adTempus User Guide

Table Of Contents

adTempus License Agreement

This license agreement ("AGREEMENT") is a legal agreement between you (either an individual or a single entity) and Arcana Development, LLC ("Arcana Development") for the software product identified above ("SOFTWARE PRODUCT"), which includes the computer software and any associated media, printed materials, and "online" or electronic documentation ("PRODUCT COMPONENTS") distributed together under the product name shown above. The SOFTWARE PRODUCT also includes any updates and supplements to the original SOFTWARE PRODUCT provided to you by Arcana Development. Any software provided along with the SOFTWARE PRODUCT that is associated with a separate end-user license agreement is licensed to you under the terms of that license agreement. By purchasing, installing, copying, downloading, executing, or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this AGREEMENT. If you do not agree to the terms of this AGREEMENT, do not install or use the SOFTWARE PRODUCT.

This AGREEMENT shall be governed by the laws of the State of Virginia and, in respect of any dispute which may arise hereunder, you consent to the jurisdiction of the federal and state courts sitting in Virginia.

Software Product License

The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

The SOFTWARE PRODUCT contains some or all of the following types of software: "Server Software" that provides the core services or functionality of the SOFTWARE PRODUCT (the computer running the Server Software shall be referred to as the "Server"); and "Client Software" that allows other computers to gain access to or utilize the services or functionality provided by the Server Software (each computer running the Client Software shall be referred to as a "Client").

1. **Grant Of License.** This AGREEMENT grants you the following rights provided that you comply with all terms and conditions of this AGREEMENT:

a. *"Licensed" Software.* If you have purchased or otherwise legally obtained a license to use the SOFTWARE PRODUCT (i.e., obtained from Arcana Development or its authorized agents a printed or electronic license certificate containing a valid license number or serial number for, and authorizing you to RUN, the SOFTWARE PRODUCT) the SOFTWARE PRODUCT is considered "Licensed Software" (and you are considered a "LICENSEE") and you are granted the following rights:

i. *"Server" Installation.* You may install, use, display, run, or otherwise interact with ("RUN") the Server Software on a single computer ("Server").

ii. *Multi-Computer Licenses*. If you have obtained a multi-computer license you may RUN additional copies of the Server Software up to the number of copies specified in your license certificate.

iii. *Site Licenses*. If you have obtained a site license you may RUN as many copies of the Server Software as you wish, on computers owned by you or under your sole control, at each physical location ("site") for which you have obtained a license. Your site license certificate specifies the site for which the license is valid; the license does not entitle you to RUN the Server Software on computers that are not at the designated site.

iv. *Unlimited Use License.* If you have obtained an unlimited use license you may RUN as many copies of the Server Software as you wish, on computers owned by you or under your sole control, regardless of location.

v. *"Client" Installation.* Regardless of the kind of license you have obtained, you may RUN the Client Software on as many computers in as many locations as you wish for the purpose of interacting with the licensed Server Software.

vi. *Backup Server Installation*. If the Server Software is installed on a computer for which you maintain a passive fail-over or backup server, you may RUN the Server Software on a temporary basis on that backup server, for backup support only, without obtaining an additional license.

b. *"Evaluation" Software.* If the SOFTWARE PRODUCT is labeled "Evaluation" or "Trial" or was provided to you by Arcana Development or its agents as evaluation software, and/or if you have not obtained a license for the SOFTWARE PRODUCT as described in **Error! Reference source not found.** above, the SOFTWARE PRODUCT is considered "Evaluation Software" (and you are considered an "EVALUATION USER")and you may RUN the SOFTWARE PRODUCT on any number of computers for up to 60 days for demonstration, test, or evaluation purposes. Your evaluation period commences at the time you first install a copy of the SOFTWARE PRODUCT; installing the SOFTWARE PRODUCT on a different computer or reinstalling the SOFTWARE PRODUCT on the same computer does not extend this evaluation period. At the end of the evaluation period you must either legally obtain a license to the SOFTWARE PRODUCT or uninstall and destroy all copies of the SOFTWARE PRODUCT in your possession.

c. *Reservation of Rights.* All rights not expressly granted are reserved by Arcana Development.

2. Description of Other Rights and Limitations.

a. Version Limitation. Your license certificate contains a specific version number. This AGREEMENT permits you to install the same or a lower version of the SOFTWARE PRODUCT as the version number specified on the license certificate. If you have obtained this SOFTWARE PRODUCT as an upgrade to a previous version for which you have a license you are permitted to RUN the SOFTWARE PRODUCT even though it may have a later version number than is specified on your license certificate, provided you meet all eligibility requirements for the upgrade as described in section **Error! Reference source not found.**.

b. *Limitations on Reverse Engineering, Decompilation, and Disassembly.* You may not reverse engineer, decompile, disassemble, or modify the SOFTWARE PRODUCT, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

c. *Separation of Components.* The SOFTWARE PRODUCT is licensed as a single product. Its component parts may not be separated for use on more than one computer except for the separation of Server Software and Client Software as provided in this AGREEMENT.

d. *Rental.* You may not rent, lease, or lend the SOFTWARE PRODUCT.

e. *Support Services.* Arcana Development may provide you with support services related to the SOFTWARE PRODUCT (“SUPPORT SERVICES”). Use of SUPPORT SERVICES is governed by the Arcana Development policies and programs described in the user manual, in “online” documentation, and/or in other materials provided by Arcana Development. Any supplemental software code provided to you as part of the SUPPORT SERVICES shall be considered part of the SOFTWARE PRODUCT and subject to the terms and conditions of this AGREEMENT. With respect to technical information you provide to Arcana Development as part of the SUPPORT SERVICES, Arcana Development may use such information for its business purposes, including for product support and development. Arcana Development will not utilize such technical information in a form that personally identifies you.

f. *Software Transfer.* The initial LICENSEE of the SOFTWARE PRODUCT may make a one-time permanent transfer of this AGREEMENT and SOFTWARE PRODUCT only directly to an end user. This transfer must include all of the SOFTWARE PRODUCT (including all component parts, the media and printed materials, any upgrades, this AGREEMENT, and, if applicable, the printed and/or electronic license certificate). Such transfer may not be by way of consignment or any other indirect transfer. The transferee of such one-time transfer must agree to comply with the terms of this AGREEMENT, including the obligation not to further transfer this AGREEMENT and SOFTWARE PRODUCT. Site licenses and unlimited use licenses may not be transferred without the consent of Arcana Development.

g. *Termination.* Without prejudice to any other rights, Arcana Development may terminate this AGREEMENT if you fail to comply with the terms and conditions of this AGREEMENT. In such event, you must destroy all copies of the SOFTWARE PRODUCT and all of its component parts.

3. **Upgrades.** If the SOFTWARE PRODUCT is labeled as an upgrade, you must be properly licensed to use a product identified by Arcana Development as being eligible for the upgrade in order to use the SOFTWARE PRODUCT. A SOFTWARE PRODUCT labeled as an upgrade replaces and/or supplements (and may disable) the product that formed the basis for your eligibility for the upgrade. You may use the resulting upgraded product only in accordance with the terms of this AGREEMENT. If the SOFTWARE PRODUCT is an upgrade of a component of a package of software

programs that you licensed as a single product, the SOFTWARE PRODUCT may be used and transferred only as part of that single product package and may not be separated for use on more than one computer.

4. **Copyright.** All title and copyrights in and to the SOFTWARE PRODUCT (including but not limited to any images, photographs, animations, video, audio, music, and text incorporated into the SOFTWARE PRODUCT), the accompanying printed materials, and any copies of the SOFTWARE PRODUCT are owned by Arcana Development or its suppliers. If this SOFTWARE PRODUCT contains documentation that is provided only in electronic form, you may print one copy of such electronic documentation. You may not copy the printed materials accompanying the SOFTWARE PRODUCT.

5. **Copies.** You may make as many copies as you wish of the installation media (if provided) or downloaded installation package, for the purpose of installing the SOFTWARE PRODUCT on computers for which you have obtained a license. After installation of the SOFTWARE PRODUCT pursuant to this AGREEMENT, you may keep the original media on which the SOFTWARE PRODUCT was provided by Arcana Development for backup or archival purposes. Except as expressly provided in this AGREEMENT, you may not otherwise make copies of the SOFTWARE PRODUCT or the printed materials accompanying the SOFTWARE PRODUCT.

6. Redistribution of Software. Copies of the SOFTWARE PRODUCT (including evaluation copies) may not be given, sold, or otherwise transferred to others without the prior permission of Arcana Development (except in the case of a license transfer as provided in section **Error! Reference source not found.** above). Software download web sites and other similar distribution services are permitted to redistribute evaluation copies provided that an evaluation copy or listing request was submitted to the service by an authorized agent of Arcana Development.

7. **U.S. Government License Rights.** This SOFTWARE PRODUCT is provided to the U.S. Government with the commercial rights and restrictions described elsewhere herein.

8. **Export Restrictions.** You may not export or re-export the SOFTWARE PRODUCT or any part thereof to any country, person or entity subject to U.S. export restrictions. You specifically agree not to export or re-export any of the SOFTWARE PRODUCT (i) to any country to which the U.S. has embargoed or restricted the export of goods or services, which currently include, but are not necessarily limited to Cuba, Iran, Iraq, Libya, North Korea, Sudan and Syria, or to any national of any such country, wherever located, who intends to transmit or transport the SOFTWARE PRODUCT back to such country; (ii) to any person or entity who you know or have reason to know will utilize the SOFTWARE PRODUCT in the design, development or production of nuclear, chemical or biological weapons; or (iii) to any person or entity who has been prohibited from participating in U.S. export transactions by any federal agency of the U.S. government. You warrant and represent that neither the BXA nor any other U.S. federal agency has suspended, revoked or denied your export privileges.

9. **Use in Hazardous Environments.** THIS SOFTWARE PRODUCT IS NOT DESIGNED, MANUFACTURED, OR INTENDED FOR USE OR RESALE IN HAZARDOUS ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, SUCH AS IN THE

OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH THE FAILURE OF THE SOFTWARE PRODUCT COULD LEAD DIRECTLY TO DEATH, PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE.

10. **License Auditing and Remote Deactivation.** The SOFTWARE PRODUCT may transmit to Arcana Development, using your network connection, data enabling Arcana Development to monitor compliance with this AGREEMENT. This data includes information that uniquely identifies the computer transmitting the data. Data collected through this mechanism will be used solely for the purpose of assessing your license compliance. The SOFTWARE PRODUCT may contain logic that allows Arcana Development to remotely disable the SOFTWARE PRODUCT if Arcana Development determines that you are in violation of this AGREEMENT. You consent to such monitoring and to Arcana Development's right to disable the SOFTWARE PRODUCT if you fail to comply with the terms of this AGREEMENT.

Disclaimer of Warranty

THIS SOFTWARE PRODUCT IS PROVIDED "AS IS." TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, ARCANA DEVELOPMENT AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NON-INFRINGEMENT, WITH REGARD TO THE SOFTWARE PRODUCT, AND THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT SERVICES. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHERS, WHICH VARY FROM STATE/JURISDICTION TO STATE/JURISDICTION.

BECAUSE OF THE VARIOUS HARDWARE AND SOFTWARE ENVIRONMENTS INTO WHICH AND USES TO WHICH THE SOFTWARE PRODUCT MAY BE PUT, NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED. GOOD DATA PROCESSING PROCEDURE DICTATES THAT ANY SOFTWARE PROGRAM BE THOROUGHLY TESTED WITH NON-CRITICAL DATA BEFORE RELYING ON IT. THE USER MUST ASSUME THE ENTIRE RISK OF USING THE SOFTWARE PRODUCT.

LIMITATION OF LIABILITY. To the maximum extent permitted by applicable law, in no event shall Arcana Development or its suppliers be liable for any special, incidental, indirect, or consequential damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or any other pecuniary loss) arising out of the use of or inability to use the SOFTWARE PRODUCT or the provision of or failure to provide Support Services, even if Arcana Development has been advised of the possibility of such damages. In any case, Arcana Development's entire liability under any provision of this AGREEMENT shall be limited to the greater of the amount actually paid by you for the SOFTWARE PRODUCT or U.S.$5.00; provided, however, if you have entered into an Arcana Development Support Services Agreement, Arcana Development's entire liability regarding Support Services shall be governed by the terms of that agreement. Because some states and jurisdictions do not allow the exclusion or limitation of liability, the above limitation may not apply to you.

Licensing

License Requirements

adTempus is licenses based on the "server" (the computer running the adTempus service). One license is required for each computer on which the service is being run. No additional licenses are required for computers running only the Console or other client tools.

To determine the license status of a server, connect to that server in the Console, then select About from the Help menu.

Evaluation Mode

If adTempus cannot find a valid license, it will run in "evaluation mode" for 30 days from the date when adTempus was first installed. During this time, adTempus is fully functional, so you can use all features. After 30 days adTempus will cease to run.

Entering License Information

The license system used by adTempus uses a file called an "electronic license certificate" to indicate that a computer is licensed to run adTempus. When you purchase a license, you will receive instructions for obtaining the necessary file, and for "installing" the license.

Note that you cannot enter a serial number or registration number directly into adTempus, or during the installation process. You must have the proper license file.

If you have lost your license information you can locate your license information using the online license management system at http://www.arcanadev.com/licensing.

Licenses can be administered locally using the "License Management" tool found in the adTempus folder on the **Start** menu.

Release Notes

For release notes and the latest information on known problems with adTempus please see www.arcanadev.com/adtempus/releasenotes.htm.

Contacting Us

Sales

For information on adTempus pricing, to purchase a license, or for sales-related questions, please visit www.arcanadev.com/shop or contact us by e-mail at sales@arcanadev.com.

Technical Support

For technical support information please visit www.arcanadev.com/adtempus/resources.asp or contact us by e-mail at support@arcanadev.com.

Product Feedback

If you have comments about a specific topic in this help document, please use the feedback link found at the bottom of the topic page. This ensures that we receive the information we need to identify the topic.

For general comments about the help document, please use the form at www.arcanadev.com/forms/docfeedback.asp?doc=adtempus.

For other comments, including feature requests, please use the form at www.arcanadev.com/forms/prodfeeback.asp?prod=6.

Obtaining Software Updates

The **Welcome** page in the adTempus Console provides an easy way to check for updates to the adTempus software.

- If the computer you are running the Console on is connected to the Internet, the Welcome page is updated from our web server each time you start the Console. If a newer version of the software is available, you will see a message on the Welcome page informing you of this.
- If your computer is not normally connected to the Internet, adTempus will not be able to check for updates when it starts. In this case you can:
  - select the "Check for Updates" option on the Welcome page while your computer is connected to the Internet, or
  - go to www.arcanadev.com/adtempus/updates.asp from a computer that is connected to the Internet.

Introduction

Overview

adTempus is a batch job scheduling and automation tool for Windows NT/2000/XP.

By "batch job" we mean programs (or collections of programs) that are meant, for the most part, to be run in an "offline" or unattended fashion, such as a program that backs up your production database each day.

Of course you can use adTempus to start Microsoft Word so you can compose a letter, but there wouldn't be much point to that.

By "scheduling and automation" we mean that adTempus executes these jobs at specific times or in response to other external events, and manages dependencies within and between jobs.

Edition Differences

adTempus is available in two editions: Scheduling Edition and the System Management Edition. See the Edition Comparison topic for information on the differences.

If you are using adTempus in evaluation mode, all features for both editions are enabled. Otherwise, System Management features will not be available unless your license is for the System Management edition.

Getting started with adTempus

The adTempus Architecture

Overview

adTempus is a batch job scheduling and automation tool for Windows NT/2000/XP.

By "batch job" we mean programs (or collections of programs) that are meant, for the most part, to be run in an "offline" or unattended fashion, such as a program that backs up your production database each day.

Of course you can use adTempus to start Microsoft Word so you can compose a letter, but there wouldn't be much point to that.

By "scheduling and automation" we mean that adTempus executes these jobs at specific times or in response to other external events, and manages dependencies within and between jobs.

Edition Differences

adTempus is available in two editions: Scheduling Edition and the System Management Edition. See the Edition Comparison topic for information on the differences.

If you are using adTempus in evaluation mode, all features for both editions are enabled. Otherwise, System Management features will not be available unless your license is for the System Management edition.

Getting started with adTempus

The adTempus Architecture

adTempus Architecture

This topic provides an overview of the adTempus architecture, introducing key concepts.

Logical Architecture

adTempus is highly modular in design. This means that there are lots of pieces (we will call them "objects"), most of which are fairly simple, that can be combined in many ways to produce complex configurations.

Before we proceed, we will set up an example that will be used to illustrate concepts as we go.

> **Example.** You need to automate the nightly update of your organization's data warehouse. This involves several different steps:
>
> 1. Download a data file using FTP.
> 2. Load the downloaded data into the data warehouse.
> 3. Run the process that reprocesses the data warehouse and rebuilds the data cubes
>
> Note that each of these three steps involves a single discrete task.

Jobs, Steps, and Tasks

Because adTempus is a job scheduler, the central object is the job. In adTempus the job contains all of the information needed for adTempus to accomplish a specific goal for you. Each job contains one or more steps; each step executes a single task.

In our example above, our goal is to complete the nightly data warehouse update; this becomes our job. Each of the steps we listed above becomes a step within the job.

Note that in many automation tools, "job" and "task" are synonymous: each of the steps above would be a separate job; the separate jobs could be linked together in "chains," "queues," or "sequences."

The task represents a single operation that adTempus is to perform (typically, a single program, script, or batch file that it is to execute). adTempus supports several different kinds of tasks that allow you to run programs, scripts, and batch files; launch documents; restart the computer; and control services. For more information, see the Tasks topic.

For each step, you can specify the rules that adTempus will use to determine whether that step succeeded; this result can then be used to alter the flow of execution for the job by using Responses.

Sometimes it makes sense to put tasks in separate jobs instead of in multiple steps within a job. More information.

More information on jobs, steps, and tasks.

Triggers

Triggers are what tell adTempus when to run a job. For example, if you want your data warehouse update to run at 11:30 each night, you would add a schedule trigger to the job to trigger it at 11:30.

More information on triggers.

Conditions

Conditions place restrictions on whether a job or step should run. For example, you can specify that a job should run only if a particular file exists.

Conditions do not cause a job to start—only triggers do this. Conditions are evaluated *after* a trigger starts a job.

More information on conditions.

Responses

Responses give you tremendous flexibility to direct the flow of execution within and between jobs.

At various points during the execution of jobs and steps, adTempus fires events. For example, an event is fired when a job starts, when a step starts, when a step finishes successfully, if a step fails, etc. To each of these events you can attach Responses, which specify Actions adTempus should execute when the event occurs. Actions allow you to execute other jobs or steps, send notification messages, capture files, and run scripts.

More information on Events, Responses, and Actions.

Getting Started

Getting Started

The following topics will help you get started using adTempus.

- If you haven't done so, you'll want to install the software.
- You should first read the overview of the adTempus architecture, will introduce you to the major "building blocks" you will be working with in adTempus.
- The adTempus Console is the administration tool for adTempus.
- If you have just installed adTempus, you will need to do some first-time setup.
- If you are upgrading from the Arcana Scheduler, you can import your Arcana Scheduler jobs.
- Finally you're ready to create a job.

adTempus Architecture

This topic provides an overview of the adTempus architecture, introducing key concepts.

Logical Architecture

adTempus is highly modular in design. This means that there are lots of pieces (we will call them "objects"), most of which are fairly simple, that can be combined in many ways to produce complex configurations.

Before we proceed, we will set up an example that will be used to illustrate concepts as we go.

> **Example.** You need to automate the nightly update of your organization's data warehouse. This involves several different steps:
>
> 1. Download a data file using FTP.
> 2. Load the downloaded data into the data warehouse.
> 3. Run the process that reprocesses the data warehouse and rebuilds the data cubes
>
> Note that each of these three steps involves a single discrete task.

Jobs, Steps, and Tasks

Because adTempus is a job scheduler, the central object is the job. In adTempus the job contains all of the information needed for adTempus to accomplish a specific goal for you. Each job contains one or more steps; each step executes a single task.

In our example above, our goal is to complete the nightly data warehouse update; this becomes our job. Each of the steps we listed above becomes a step within the job.

Note that in many automation tools, "job" and "task" are synonymous: each of the steps above would be a separate job; the separate jobs could be linked together in "chains," "queues," or "sequences."

The task represents a single operation that adTempus is to perform (typically, a single program, script, or batch file that it is to execute). adTempus supports several different kinds of tasks that allow you to run programs, scripts, and batch files; launch documents; restart the computer; and control services. For more information, see the Tasks topic.

For each step, you can specify the rules that adTempus will use to determine whether that step succeeded; this result can then be used to alter the flow of execution for the job by using Responses.

Sometimes it makes sense to put tasks in separate jobs instead of in multiple steps within a job. More information.

More information on jobs, steps, and tasks.

Triggers

Triggers are what tell adTempus when to run a job. For example, if you want your data warehouse update to run at 11:30 each night, you would add a schedule trigger to the job to trigger it at 11:30.

More information on triggers.

Conditions

Conditions place restrictions on whether a job or step should run. For example, you can specify that a job should run only if a particular file exists.

Conditions do not cause a job to start—only triggers do this. Conditions are evaluated *after* a trigger starts a job.

More information on conditions.

Responses

Responses give you tremendous flexibility to direct the flow of execution within and between jobs.

At various points during the execution of jobs and steps, adTempus fires events. For example, an event is fired when a job starts, when a step starts, when a step finishes successfully, if a step fails, etc. To each of these events you can attach Responses, which specify Actions adTempus should execute when the event occurs. Actions allow you to execute other jobs or steps, send notification messages, capture files, and run scripts.

More information on Events, Responses, and Actions.

adTempus Console

The adTempus Console is the primary user interface for adTempus, used to monitor and configure adTempus.

The Console is similar to the Microsoft Management Console and other administrative tools. The Console is divided into two main sections:

- The left section provides a "tree" view of the connected servers and the views available for each.
- Selecting a node in the tree displays the associated "view" on the right side of the Console. Each view is used to work with a different kind of object in adTempus. See the Help Contents for information on each of the views.

Most commands are issued through the pop-up menus associated with the tree nodes and the views. Right-clicking a node, view, or item in a view will display a menu of available commands.

The Console can be used to administer any number of adTempus servers. To establish a new connection, use the **Connect Server** command on the **File** menu.

adTempus uses Windows security, so Windows must be able to recognize your identity *on the server computer*. This means that you must be connecting between computers in the same domain, or between computers where you have accounts that use the same user ID and password.

Within the Console all dates and times are show using adjusted for the time zone of the server. For your convenience the clock in the adTempus status bar shows the time on the client and the time on the server.

The active view is automatically refreshed at the interval you specify in the Connection Options for the server. Inactive views are not refreshed until they are activated, except for the Failed Job list, which is always kept refreshed.

Throughout the Console, only those objects for which you have View permission are visible, and permissions may restrict your ability to modify these objects.

First-Time Setup

If you have just installed adTempus you will need to perform some initial configuration. You will need to be logged in to the computer under an Administrator account to perform the initial setup.

Security Settings

You should review the default security settings, and modify them as appropriate for your environment.

Security settings are configured using the Security Options properties.

Access to adTempus

By default only members of the Administrators group on the server are allowed to connect to adTempus (i.e., only Administrators can use adTempus). If you want other users to be able to use adTempus, make the necessary changes on the Server Security page.

Access to Jobs

By default all users are able to create jobs, and to modify and administer jobs they have created. ("All users" really means all users who are able to connect to adTempus, which, by default, is only Administrators. If you grant access to additional users, they will by default be able to create jobs.)

By default users *cannot* see or modify jobs created by other users.

To modify permissions for who is able to create jobs, and the default security settings for those jobs, use the Default Job Security page.

Note that Administrators are always a special case: Administrators have access to all objects in adTempus, regardless of who created them.

Access to Other Objects

In addition to jobs, the following objects can be secured. You should review and, if appropriate, modify the security settings for these objects:

- Notification Recipients
- Shared Schedules and Holiday Sets
- Scripts and Script Libraries

Notification

If you plan to use e-mail notification, you must configure the notification options.

Holidays

Jobs can be configured to behave differently on holidays. If you plan to use this feature you should review the default holidays and modify the list as appropriate for your organization.

Shared Schedules

Shared Schedules allow you to use a single schedule to determine the days on which many jobs will run (each job still has its own settings for the time(s) at which it will run).

You may want to set up some initial shared schedules that are available to all users, and establish policies for their use. For example, you may want to define a "Weekdays" schedule for jobs that run Monday through Friday.

Arcana Scheduler Import

The Arcana Scheduler Import tool allows you to import scheduled programs from the Arcana Scheduler. To use this tool,

- You must be running the adTempus Console from the computer on which the Arcana Scheduler service is installed.
- You must be running version 2.3 or later of the Arcana Scheduler. If you are running an earlier version you will need to upgrade to version 2.3 and then run the import.
- You must be logged in to the computer under an account that has access to the scheduled programs you want to import.

The following additional restrictions apply:

- adTempus cannot import the passwords associated with the user accounts on scheduled programs and network connections. Each time adTempus encounters a new user account during the import process it will prompt you for the password for that account.
- Imported jobs will have the permissions specified by the default job security settings. Make sure these settings are to your liking before importing the jobs.

ℹ You will be able to review the imported jobs before they are saved.

Import Wizard

Creating a Job

To create a new job, select the **Jobs** node for the appropriate server in the adTempus Console.

ℹ The New Job Wizard simplifies job creation for common scenarios. Jobs created by the Wizard can be used as-is, or can serve as a template for your own jobs.

If you are new to adTempus, we recommend that you create a job with the Wizard and then review that job, to get a feel for how the various components of adTempus work together.

To use the New Job Wizard, go to the Jobs view in the Console, then select **New Job Wizard** from the **Actions** menu.

To create a basic job without using the Wizard, in the Jobs view, select **New Job...** from the **Action** menu. The Job Properties will be displayed.

Though adTempus offers many options, a simple job requires only these four things:

1. A **Name**. Give the job a unique, descriptive name.
2. A **User Account**. Specify the user ID and password for the account the job should run under.
3. A Trigger. Generally this will be a Schedule Trigger (to execute the job at a particular date and time). On the **Trigger** page, click the **Add...** button and select "Execute according to a schedule." In the **Edit Schedule Trigger** window click **Add...** to add a new execution schedule. See the Execution Schedule topic for information on specifying the schedule.
4. A task. The Program Execution task is the most commonly used. On the **Steps** page click **Add...** and select "Execute a program, batch file, etc." See the Program Execution Task topic for information on configuring the task.

adTempus Objects

Job

Job

The Job is the central focus of adTempus. The job contains all of the information needed for adTempus to accomplish a specific goal for you. Each job contains one or more steps; each step executes a single task.

When you create or modify a job you are modifying a "template" for the job. Each time the job executes adTempus creates an instance of that job. Changes you make to a job do not have any effect on any instances that are already executing.

Information about each instance is retained in the job's history, which can be viewed from the Jobs View or the Execution History Query.

adTempus Architecture Overview

Job Properties

Steps Overview

Each step of a job executes a single task.

adTempus Architecture Overview

Triggers

Triggers

Triggers determine when a job should be run. adTempus supports the following triggers:

- The Schedule Trigger is used to execute jobs at specific dates and times.
- The Startup Trigger is used to execute jobs whenever the adTempus service is started.
- The File Trigger is used to execute jobs based on file creation, deletion, or modification.

If you are using the System Management edition of adTempus, the following additional triggers are available:

- The Computer Monitor Trigger is used to monitor remote computers and execute jobs when they fail.
- The Event Log Monitor is used to monitor the computer's Event Log and execute jobs when certain events are logged.
- The Process Trigger is used to monitor external processes and execute jobs when they start, end, or exceed memory limits.
- The WMI Trigger is used to monitor Windows Management Instrumentation (WMI) events and trigger jobs based on those events.

adTempus Architecture Overview

Schedule Trigger

A Schedule Trigger is a trigger that causes a job to be executed at specific dates and times, or at specific intervals (e.g., every 5 minutes).

When you create a Schedule Trigger, you specify one or more Schedules, which specify when the job should run. Options allow you to specify how adTempus should react to changes of the system clock. You may also specify what should happen if the job is scheduled to run on a holiday.

Schedule Trigger Properties

The Schedule Trigger Properties window is divided into three pages. Follow the links below for information on each page:

Schedules Page

Options Page

Holidays Page

Triggers Overview

Startup Trigger

A Startup Trigger is a trigger that causes a job to be executed each time the adTempus service is started.

Since the service is generally only when the computer is restarted, this trigger can be used to start programs when the computer restarts.

If adTempus is stopped and restarted for a reason other than a computer restart, your program may still be running when adTempus restarts. When you use the Startup Trigger you should also use process conditions, the Process Execution task's **Skip if already running** option, or some other safeguard to make sure that your scheduled tasks do not get re-executed when they should not.

Startup Trigger Properties

Triggers Overview

Conditions

Conditions

Conditions are used to place additional restrictions on whether a job or step should run. For example, you can specify that a job should run only if a particular file exists.

Conditions do not cause a job to start—only triggers do this. Conditions are evaluated *after* a trigger starts a job.

adTempus offers the following kinds of conditions:

- The Job Condition depends on the state of another job.
- The File Condition depends on the presence or absence of one or more files.
- The Process Condition depends on the state of an external process.
- The Script Condition allows you to use a script to write your own condition logic.

Condition Polling

adTempus Architecture Overview

File Condition

A File Condition is a condition that depends on the presence or absence of a specified file or group of files.

For example, you have a job processes data files that are transferred to your server by FTP. You want the job run each hour, but only if new files have been transferred. You attach a File Condition to the job so that it will only run if files exist.

File Condition Properties

Conditions Overview

Job Condition

A Job Condition is a condition that prevents a job or step from executing unless another job is in a specified state.

For example, you have a job that reprocesses the data cubes for your data warehouse. This job cannot run until the three jobs that extract the data for the data warehouse have run. You therefore add a Job Condition to your cube reprocess job, so that it waits on the three extract jobs.

Job Condition Properties

Conditions Overview

Process Condition

A Process Condition is a condition that depends on the state of a Windows process (program). Unlike a Job Condition the Process Condition allows you to wait on a process that is not under the control of adTempus.

For example, you have a job that requires exclusive access to a data file. This file is also used by a process that is often run by users outside of adTempus. By adding a Process Condition that targets this external process you can prevent adTempus from running the job when the data file is in use.

If you want to prevent a Program Execution Task from running if the target process is already running, the Skip this step if the process is already running option provides a one-step alternative to the Process Condition.

Process Condition Properties

Conditions Overview

Script Condition

A Script Condition is a condition that executes a script. The result of the script determines whether the condition is satisfied.

Use a Script Condition to provide your own condition checks when none of the condition types provided with adTempus meets your needs.

For example, you want a job to execute only if a specified file contains certain text. The adTempus File Condition can detect the presence of the file but does not read its contents. Using a Script Condition you can create a script (using a scripting language like VBScript) that looks for the file and then checks it for the required text.

Script Condition Properties

Conditions Overview

Script Overview

Returning Results from Scripts

Events, Responses, and Actions

Responses

Responses allow you to customize the flow of execution through an adTempus job, or to link jobs together.

While it is executing a job, adTempus fires "events" at certain key points. For example, an event is fired at the beginning of the job; another is fired at the end of the job. If the job completes successfully a "success" event is fired. If the job fails, a "failure" event is fired.

If you want adTempus to take action based on one of these events, you associate a Response with that event. The Response defines the actions that adTempus should take in response to the event. See an example.

Each response can be triggered by any number of events, and can execute any number of actions.

The events that are available vary based on whether you are creating responses for a job or a step, and based on the kind of task a step executes. For a complete list of events for a job or task, see the help topic for the Responses property page for the job or task.

Notes on using Responses

Responses are evaluated in the order that they appear in the Responses list for the job or step. Within a response, actions are executed in the order that they appear in the Actions list.

Actions are executes synchronously. That is, each action must finish before the next action starts. Each Response's actions must complete before the next Response is evaluated. And all responses must be evaluated before job execution proceeds to the next step. This is important to keep in mind when you are using Job Control actions or Script actions; see the topics for those actions for more information.

Response example

Response Properties

Actions

Events

While it is executing a job, adTempus fires "events" at certain key points. For example, an event is fired at the beginning of the job; another is fired at the end of the job. If the job completes successfully a "success" event is fired. If the job fails, a "failure" event is fired.

The events that are available vary based on whether you are creating responses for a job or a step, and based on the kind of task a step executes. For a complete list of events for a job or task, see the help topic for the Responses property page for the job or task.

If you want adTempus to take action based on one of these events, you associate a Response with that event.

Actions

Actions

Actions determine how adTempus should behave in response to events.

The following actions are available:

Action	Description
Job Control Action	Used to start, restart, terminate, hold, release, or delete a job or job step.
Notification Action	Used to send notification messages by e-

	mail or network broadcast.
File Capture Action	Used to capture and store output files produced by the job.
Script Action	Used to execute a script.

File Capture Action

The File Capture action is used to "capture" files produce by your job. Captured files are copied into the adTempus data directory and retained as part of the history for the job.

For example, if a scheduled program creates a log file of its activities each time it runs, you can capture this file so that it is retained with the job's history.

The File Capture action also gives you the option of sending captured files by e-mail.

To view captured files, view the properties for the instance once the job or step completes. The files will be found in the Captured Files list for the job; from here you can open the files or copy them to a new location.

Captured files are associated with the instance that originally captured them. When that instance is deleted, its captured files are deleted as well. See the job's General page to specify how long the job history should be retained.

File Capture Action Properties

Responses Overview

Actions

Job Control Action

The Job Control action is used to execute and manipulate jobs and job steps. It can be used to

- Run, abort, restart, hold, release, or delete a job
- Run or restart a step (in the current or another job)
- Change the status of the job or step that is executing the action

Job Control Action Properties

Responses Overview

Actions

Notification Action

The Notification action is used to send notification messages (by e-mail or network broadcast) regarding the status of a job.

Notification Action Properties

Notification Task

Responses Overview

Actions

Script Action

The Script action is used to execute a script. This gives you the flexibility to respond to events in ways not provided for by the built-in actions offered by adTempus.

Script actions also are commonly used to:

- Set environment variables for the job, when the "static" environment variable list offered by the Program Execution task is not sufficient.
- Send keystrokes to an application.

While an action is being executed, adTempus cannot perform any other "work" on the job with which the action is associated. For example:

- If more than one action has been specified in a response, each action cannot be run until the previous action has completed.
- adTempus cannot start the next step of a job until it finishes the current step. This includes running all actions associated with the step.

Therefore when you use a script action your script should be something that runs quickly, so adTempus can be about its business. If your script goes and launches some external application using the application's automation interface, and then spends 10 minutes doing something with that application, you adTempus job is going to be patiently waiting.

Using Scripts

Script Action Properties

Responses Overview

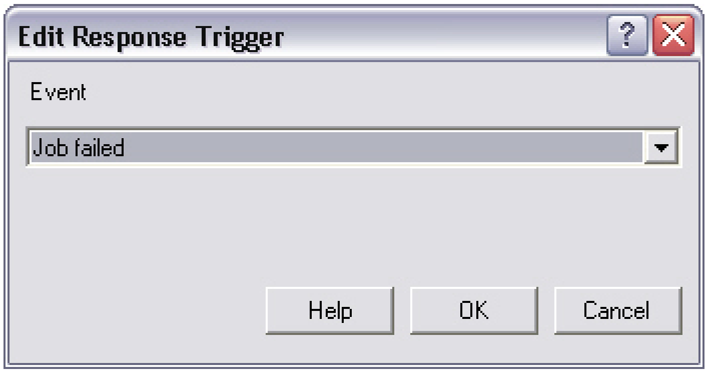
Actions

Response Example

This example illustrates the creation of a response.

You want adTempus to send you an e-mail message if your job fails. To do so:

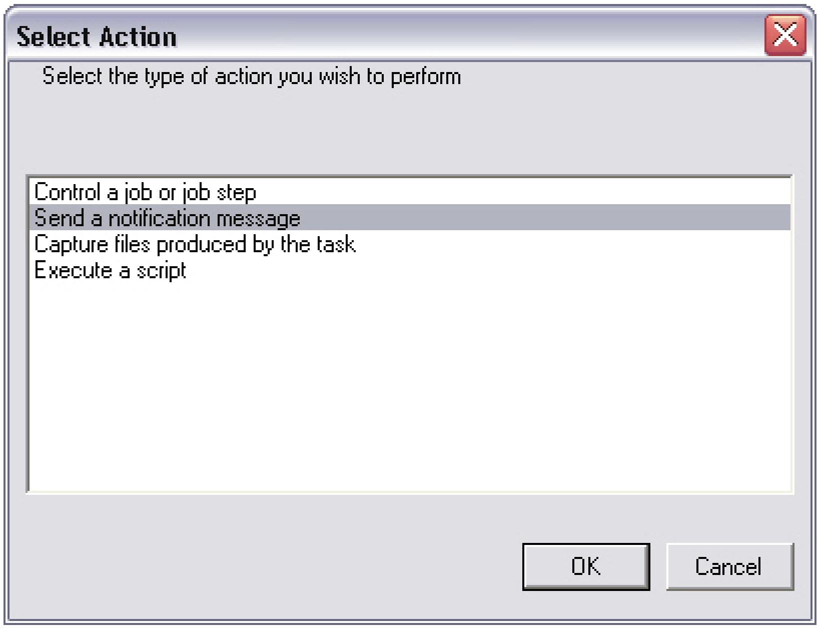
1. On the Responses page of the job's properties, click **Add** to add a new Response.
2. In the **Events** section of the Response Properties, click **New** to add a new event.
3. In the Response Trigger window, select the "Job Failed" event. *Show me*



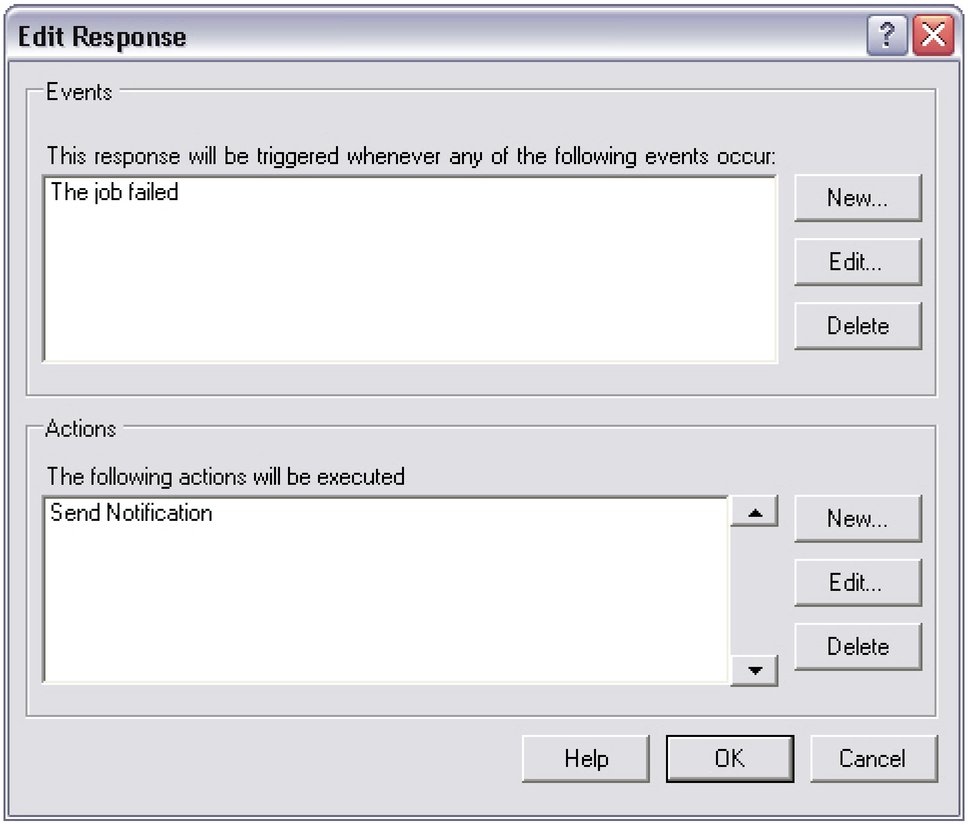
Edit Response Trigger

Event: Job failed

4. In the **Actions** section of the Response Properties, click **New** to add a new action.
5. When prompted to select an action type, select "Send a notification message." *Show me*


Select Action
Select the type of action you wish to perform
Control a job or job step
Send a notification message
Capture files produced by the task
Execute a script
OK
Cancel

6. In the Notification Action Properties window, select or create the appropriate recipient.
7. Your response is now complete. Whenever the job fails, a notification message will be sent. *Show me*


Edit Response
Events
This response will be triggered whenever any of the following events occur:
The job failed
New...
Edit...
Delete
Actions
The following actions will be executed
Send Notification
New...
Edit...
Delete
Help
OK
Cancel

Scripts

Using Scripts

The Microsoft Windows Script Host allows you to execute scripts written in various languages under a single, unified framework. Microsoft provides the VBScript and JScript scripting languages; other vendors have supplied additional scripting languages (such as PerlScript).

adTempus uses the Windows Script Host to provide extensive support for scripting. This means that adTempus supports any script in any scripting language installed on the adTempus server.

adTempus allows you to create jobs that execute scripts to accomplish external tasks (e.g., running a script that handles FTP download of data), but also allows you to use scripts to customize the behavior of adTempus. Scripts can be used:

- As conditions, to determine whether a job or job step should run.
- As actions, to perform quick tasks within adTempus, such as changing environment variables or sending keystrokes to applications.
- To determine whether a step executed successfully.
- To dynamically determine the command-line parameters for a program to be executed by adTempus.

Scripts are represented within adTempus by the Script object, which allows you to share scripts among jobs and users, and to restrict the use of scripts with security settings.

If you have subroutines or data that needs to be shared among scripts, you can place this code and data in Script Libraries, which can be referenced from any script.

Information can be shared between scripts within a job by using parameters.

Additional scripting topics:

Global variables and procedures available to adTempus scripts

Predefined script libraries

Returning a result from a script

Sending keystrokes to an application

Setting environment variables with a script

Script

A **Script** allows you to store and execute a Windows Script Host script (such as a script written in VBScript or JScript) within adTempus.

Scripts can make use of common code stored in Script Libraries, and can use globally-defined variables and procedures.

Script Properties

Using Scripts in adTempus

Script Library

Script Libraries allow you to create "libraries" of commonly used code or data that can be shared among scripts in adTempus. For example, suppose that you have several jobs where you need to pass the current date as one of the command-line parameters to the scheduled program (but the other parameters are different for each job, so you cannot use a single script for all of the jobs).

Rather than repeating the necessary formatting code in the command-line script for each task, you can create a function in a Script Library, and then reference the library from the scripts. This in turn allows you to call the functions and subroutines defined in that library.

A Script Library can contain any code or data declarations that are valid for the scripting language you specify for the library. A script can only use libraries that are in the same scripting language.

For examples, refer to the predefined libraries provided with adTempus.

Because the Windows Script Host does not support multiple namespaces, all of the code for a script and for all of the libraries it references end up in the same namespace. You therefore must take care that you do not have variables or procedures in different libraries that have the same name, if those libraries will be used by the same script.

Script Library Properties

Predefined libraries

Overview of using scripts in adTempus

adTempus Console

adTempus Console

The adTempus Console is the primary user interface for adTempus, used to monitor and configure adTempus.

The Console is similar to the Microsoft Management Console and other administrative tools. The Console is divided into two main sections:

- The left section provides a "tree" view of the connected servers and the views available for each.
- Selecting a node in the tree displays the associated "view" on the right side of the Console. Each view is used to work with a different kind of object in adTempus. See the Help Contents for information on each of the views.

Most commands are issued through the pop-up menus associated with the tree nodes and the views. Right-clicking a node, view, or item in a view will display a menu of available commands.

The Console can be used to administer any number of adTempus servers. To establish a new connection, use the **Connect Server** command on the **File** menu.

adTempus uses Windows security, so Windows must be able to recognize your identity *on the server computer*. This means that you must be connecting between computers in the same domain, or between computers where you have accounts that use the same user ID and password.

Within the Console all dates and times are show using adjusted for the time zone of the server. For your convenience the clock in the adTempus status bar shows the time on the client and the time on the server.

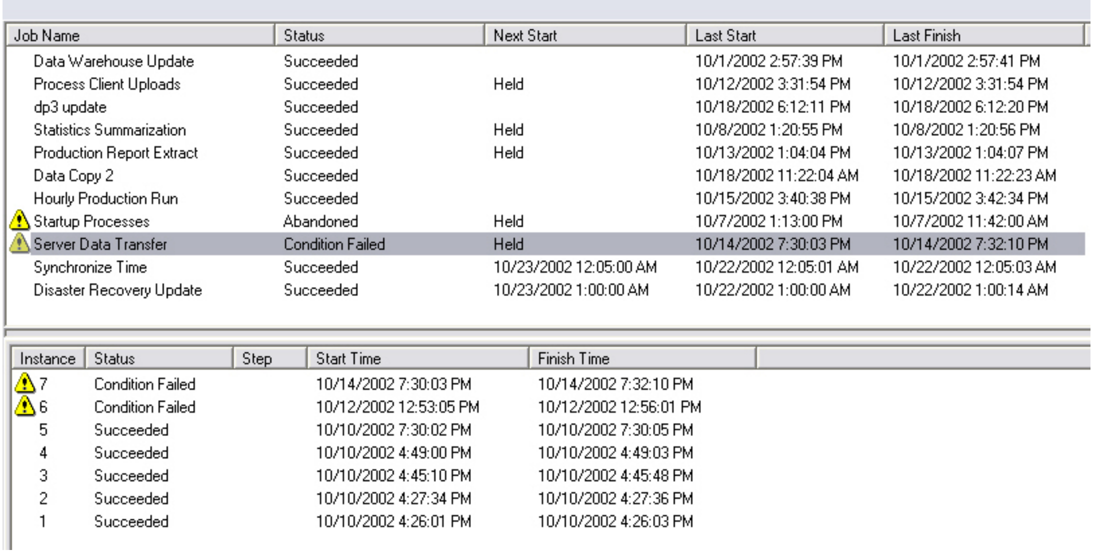
The active view is automatically refreshed at the interval you specify in the Connection Options for the server. Inactive views are not refreshed until they are activated, except for the Failed Job list, which is always kept refreshed.

Throughout the Console, only those objects for which you have View permission are visible, and permissions may restrict your ability to modify these objects.

Views

Jobs View

Show Window Sample



Job Name	Status	Next Start	Last Start	Last Finish
Data Warehouse Update	Succeeded		10/1/2002 2:57:39 PM	10/1/2002 2:57:41 PM
Process Client Uploads	Succeeded	Held	10/12/2002 3:31:54 PM	10/12/2002 3:31:54 PM
dp3 update	Succeeded		10/18/2002 6:12:11 PM	10/18/2002 6:12:20 PM
Statistics Summarization	Succeeded	Held	10/8/2002 1:20:55 PM	10/8/2002 1:20:56 PM
Production Report Extract	Succeeded	Held	10/13/2002 1:04:04 PM	10/13/2002 1:04:07 PM
Data Copy 2	Succeeded		10/18/2002 11:22:04 AM	10/18/2002 11:22:23 AM
Hourly Production Run	Succeeded		10/15/2002 3:40:38 PM	10/15/2002 3:42:34 PM
⚠ Startup Processes	Abandoned	Held	10/7/2002 1:13:00 PM	10/7/2002 11:42:00 AM
⚠ Server Data Transfer	Condition Failed	Held	10/14/2002 7:30:03 PM	10/14/2002 7:32:10 PM
Synchronize Time	Succeeded	10/23/2002 12:05:00 AM	10/22/2002 12:05:01 AM	10/22/2002 12:05:03 AM
Disaster Recovery Update	Succeeded	10/23/2002 1:00:00 AM	10/22/2002 1:00:00 AM	10/22/2002 1:00:14 AM

Instance	Status	Step	Start Time	Finish Time
⚠ 7	Condition Failed		10/14/2002 7:30:03 PM	10/14/2002 7:32:10 PM
⚠ 6	Condition Failed		10/12/2002 12:53:05 PM	10/12/2002 12:56:01 PM
5	Succeeded		10/10/2002 7:30:02 PM	10/10/2002 7:30:05 PM
4	Succeeded		10/10/2002 4:49:00 PM	10/10/2002 4:49:03 PM
3	Succeeded		10/10/2002 4:45:10 PM	10/10/2002 4:45:48 PM
2	Succeeded		10/10/2002 4:27:34 PM	10/10/2002 4:27:36 PM
1	Succeeded		10/10/2002 4:26:01 PM	10/10/2002 4:26:03 PM

The Jobs view is the primary administrative view for adTempus. The view is divided into two segments.

Job List

The top section of the view lists all adTempus jobs that you have permission to view. If the job is running the **Status** column indicates this. Otherwise, the **Status** column indicates the result of the last instance that executed.

The pop-up menu for this list allows you to create, modify, delete, hold, duplicate, and execute jobs.

To reduce clutter in the Console and simplify administration, you can organize jobs into groups, which appear as folders in the job list.

Job Details

When you select a job from the Job List, information about that job is displayed on the four tabs at the bottom of the view.

History

The **History** tab lists information about the most recent 20 executions of the job. For information about older instances, use the Execution History Query.

The pop-up menu for this list allows you to display the details for the instance and, for active instances, terminate the instance.

Agents

The **Agents** tab only appears if the adTempus server to which you are connected is a Master Server. The tab lists each agent that the job targets, and the status of the job on that agent.

The pop-up menu for this list allows you to manually start the job on a specific agent, without running it on the other agents.

Job Log

The **Job Log** tab lists any messages that have been logged for the job. Only the most recent 20 messages are listed; for information on older messages use the Message Log Query.

Note that adTempus logs very few informational messages, so do not be surprised if no messages have been logged for a job.

Statistics

The **Statistics** tab provides some basic statistics about the job as a whole (including all instances of the job). Statistics for individual instances and steps are available when you view an instance from the History list.

Job Properties

adTempus Console

Notification Recipients View

The Notification Recipients view lists all Notification Recipients, Notification Groups, and Messaging Service Providers that you have permission to view. From this view you can add, modify, and delete recipients.

You will not be permitted to delete a recipient that is being used on one or more jobs.

adTempus Console

Shared Schedules View

The Shared Schedules view lists all of the Shared Schedules that you have permission to view. In this view you can add, modify, and delete shared schedules.

You will not be permitted to delete any shared schedule that is in use on a job.

Shared Schedule Properties

adTempus Console

Holidays View

The Holidays view lists all of the Holiday Sets that you have permission to view. In this view you can add, modify, and delete holiday sets.

You will not be permitted to delete any holiday set that is in use on a job.

Holiday Set Properties

adTempus Console

Shared Scripts View

The Shared Scripts view lists all of the shared Scripts that you have permission to view. In this view you can add, modify, and delete shared scripts.

You will not be permitted to delete any script set that is in use on a job.

This view only lists scripts that were marked as shared. Scripts that were not shared can only be viewed and modified from the job, step, task, action, or condition with which they are associated.

Script Properties

adTempus Console

Script Libraries View

The Script Libraries view lists all of the Script Libraries that you have permission to view. In this view you can add, modify, and delete script libraries.

You will not be permitted to delete any script library that is referenced by a script.

Script Library Properties

adTempus Console

Execution History Query

The Execution History Query is used to query the execution history of one or more jobs. Use this query to retrieve job instances that match your specifications.

If you check the **Refresh Automatically** box adTempus will re-execute the query at the refresh interval specified for this connection. If you do not check this option, the query is only executed when you click the **Search** button.

Job Instance Properties

Failed Jobs Query

adTempus Console

Failed Jobs View

The Failed Jobs view is similar to the Execution History query except that it only retrieves instances that have failed.

If instances are found that meet the criteria, adTempus puts a failure icon next to this view in the Console Tree to draw your attention to the view. Unlike other views, this view is refreshed regularly as long as the server is connected, even if the view is not active.

When you click the **Acknowledge** button, adTempus clears the list of failed instances and sets the starting time to the current time so that it will only watch for failures you haven't already seen. adTempus will remember this starting time the next time you run the Console.

Job Instance Properties

Execution History Query

adTempus Console

Tools

Message Log Query

The adTempus Message Log is similar to the Windows Event Log and is used by adTempus to report information to users.

Most messages in the log are tied to a job or job instance. The 20 most recent messages for a job can be viewed on the Job Log tab in the Jobs view. To view additional messages, or search for messages matching specific criteria, use the Message Log query.

Be sure to check the **List non-job messages** box if you wish to see messages that are not tied to a specific jobs.

Double-click a message to see its details.

Server Options

Server Options

The Server Options window contains general settings for the adTempus server.

Default History Retention

The Default History Retention option determines how long adTempus retains the history for jobs when the job's history retention option is set to "Use global default setting."

Note that if you select the **Do not keep history** option, the history (all instance details, including messages and captured files) may be purged as soon as the job completes. This means that you will not be able to see the details of any instances of the job. Also, Job Conditions that target the job will not work correctly, as they use the history.

If you select the **Retain Indefinitely** option the history will be kept until you change the retention option to a setting that allows adTempus to purge the history.

Auditing

The Auditing options allow you to maintain an audit trail of who has created, modified, deleted, executed, and aborted each job. When the auditing options are selected, an entry is written to the message log for the job whenever a selected event occurs.

Note: When a job is deleted, the message log (including audit messages) for the job is deleted. The audit message indicating that the job has been deleted, however, is not deleted.

When a job is modified, the auditing mechanism does not record information about what changed; it only records the fact that the job has been modified.

The **Log Job Execution** and **Log Job Termination** options cause an audit message to be written whenever a user manually starts or aborts a job. Regular, scheduled executions of the job do not result in an audit message.

Audit messages are saved in the message log for the job. To review the audit messages, use the **Job Log** section of the Jobs view, or the Message Log query.

When using the Message Log query, you can query on Message Category 30 to retrieve only audit-related messages.

Retain non-job log messages for __ days

Most messages in the adTempus message log are related to a particular job; these messages are purged from the database according to the history retention settings for the job.

Some messages are not related to any job; these are purged after the number of days you specify here.

Back up adTempus database each day at

This option determines the time at which the adTempus database is compacted and backed up each day. See the adTempus Database topic for more information.

This option is only available when you are using the default Microsoft Access database for adTempus. See the database topic for more information.

Server Security Page

The options on the **Server Security** page control access to adTempus.

For information on specifying security settings see the Security Editor and Security Overview topics.

By default only members of the server's Administrators group can connect to adTempus. If you want users other than Administrators to be able to use adTempus, you must grant them Connect permission on this page.

The following permissions apply to the adTempus server:

Permission	Description
Full Control	Permission to perform all of the actions listed below.
Connect	Permission to connect to the adTempus server. Without this permission the user is not able to connect to the adTempus server, and any other permissions granted to the user therefore are irrelevant.
Changer server options	Permission to modify server options other than security settings.
Administer server security	Permission to modify server security settings (all settings in the Security Options window).

Security Overview

Security Editor

Security Options

Security Options

The Security Options window provides access to security settings for adTempus. Remember that you are editing security settings only for the adTempus server that is currently selected in the Console. If you are connected to more than one server, servers other than the one selected are not affected.

The settings are divided among the following pages:

Server Security

Default Job Security

Default Notification Recipient Security

Default Shared Schedule Security

Default Script Security

Security Overview

Default Job Security Page

Options on this page determine the default security settings for newly-created jobs.

These settings become the initial settings for newly-create jobs. Changing these settings has no effect on existing jobs.

Users must have the **Create** permission to be able to create new jobs.

The following permissions apply to jobs:

Permission	Description
Full Control	Permission to perform all actions on the job.
List/Reference	Permission to link to the job in Job Control Actions, Job Conditions, etc. Does not automatically grant permission to view the properties of the job.
View	Permission to view the properties of the job.
Modify	Permission to modify the properties of the job.
Delete	Permission to delete the job.
Execute	Permission to execute and abort the job through commands available in the console. Execute permission is also required for the user to link to the job using a Job Control Action.
Administer security	Permission to change the security settings for the job.
Change owner	Permission to take ownership of the job.
Create	Permission to create new jobs.

Security Overview

Security Editor

Default Notification Recipient Security Page

Options on this page determine the default security settings for newly-created notification recipients.

These settings become the initial settings for newly-created objects. Changing these settings has no effect on existing objects.

Users must have the **Create** permission to be able to create new notification recipients.

The following permissions apply to notification recipients:

Permission	Description
Full Control	Permission to perform all actions on the recipient.
List/User	Permission to use the recipient.
View	Permission to view the properties of the recipient.
Modify	Permission to modify the properties of the recipient.
Delete	Permission to delete the recipient.
Administer security	Permission to change the security settings for the recipient.
Change owner	Permission to take ownership of the recipient.
Create	Permission to create new notification recipients.

Security Overview

Security Editor

Default Shared Schedule Security Page

Options on this page determine the default security settings for newly-created shared schedules and holiday sets.

These settings become the initial settings for newly-created objects. Changing these settings has no effect on existing objects.

Users must have the **Create** permission to be able to create new shared schedules or holiday sets.

The following permissions apply to shared schedules and holiday sets:

Permission	Description
Full Control	Permission to perform all actions on the schedule.
List/User	Permission to use the schedule.
View	Permission to view the properties of the schedule.

Modify	Permission to modify the properties of the schedule.
Delete	Permission to delete the schedule.
Administer security	Permission to change the security settings for the schedule.
Change owner	Permission to take ownership of the schedule.
Create	Permission to create holiday sets and shared schedules

Security Overview

Security Editor

Default Script Security Page

Options on this page determine the default security settings for newly-created scripts and script libraries.

These settings become the initial settings for newly-created scripts and script libraries. Changing these settings has no effect on existing objects.

Users must have the **Create** permission to be able to create new scripts and script libraries.

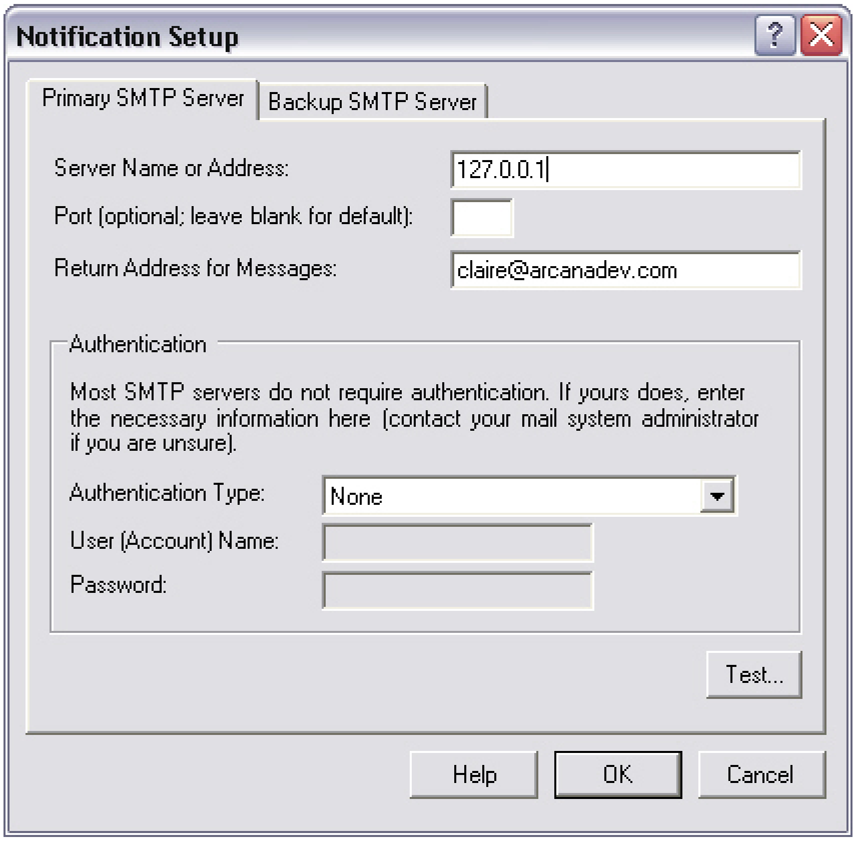
Permission	Description
Full Control	Permission to perform all actions on the script.
List/User	Permission to use the script.
View	Permission to view the properties of the script.
Modify	Permission to modify the properties of the script.
Delete	Permission to delete the script.
Administer security	Permission to change the security settings for the script.
Create	Permission to create new scripts and script libraries.

Security Overview

Security Editor

Notification Setup

Show Window Sample



Server Name or Address: 127.0.0.1

Port (optional; leave blank for default):

Return Address for Messages: claire@arcanadev.com

Authentication

Most SMTP servers do not require authentication. If yours does, enter the necessary information here (contact your mail system administrator if you are unsure).

Authentication Type: None

User (Account) Name:

Password:

The Notification Setup window contains the settings used for SMTP e-mail notification (used for Notification Actions).

To use e-mail notification you must have access to an SMTP server. (SMTP is the standard mail protocol used for Internet e-mail.) If you are unsure of what settings to use, contact your mail system administrator.

If you use an SMTP e-mail client you can use that program's settings as a guide.

You may provide settings for a primary and, optionally, a backup server. If adTempus is unable to send a message using the primary server it will attempt to use the backup server.

After specifying options, you should test your settings using the **Test** button.

Server Name or Address

Enter the name or IP address of the SMTP server. If the mail server runs on the same computer as adTempus, use `127.0.0.1` as the address.

Port

Enter the port that the SMTP server is listening on. Most servers use the default port, which is 25. Leave this box empty to use the default, or specify a value if you server uses a different port.

Return Address for Messages

Specify the return address to use on e-mail messages sent to the server. The address **must be specified**, and must be a valid address. Many mail servers will reject messages that do not have a valid return address.

Authentication

Most SMTP servers do not require authentication. If yours does, enter the necessary information here. Check with your mail system administrator if you are unsure regarding these settings.

Specifying authentication information for a mail server that does not accept it will cause messages to fail.

Test...

Click the **Test...** button to send a test message using the settings you have provided. If adTempus is unable to send the test message it will display an error message with the reason.

If the error message does not provide you with enough information to resolve the problem, you can find a transcript of the conversation between adTempus and the mail server in the file `smtp.log`, found in the "logs" subdirectory under the adTempus program directory on the server.

Data Import/Export

Data Import and Export

The adTempus Import/Export facility allows you to copy data between adTempus servers. The following kinds of objects can be transferred:

- Jobs
- Shared Schedules
- Holiday Sets
- Shared Scripts
- Script Libraries

When you export an object, any objects on which it depends are automatically exported. For example, when you export a job, any scripts, notification recipients, shared schedules, etc., that are used in the job are exported with it.

When you import data from a data file, adTempus checks each object to see if it already exists. If it does, the object already in adTempus is replaced if the object in the file has a later timestamp.

When you import data, the original security settings are not imported. Each imported object is given the default security settings. You must have permission to create each kind of object that you are importing.

Exporting data

Importing data

Importing jobs from the Arcana Scheduler

Data Export

The Data Export tool allows you to export data from adTempus. You can export data to a file to be transferred to another adTempus server. You may also export the data directly to any other server to which the Console is currently connected.

You can only export data for which you have View permission.

In the Data Export window, select an object type to view a list of objects of that type, then select the objects you wish to export. You can export objects of multiple types at the same time (e.g., you can export jobs and shared schedules).

When you export an object, any objects on which it depends are automatically exported. For example, when you export a job, any scripts, notification recipients, shared schedules, etc., that are used in the job are exported with it. Thus if you want to export a single job, that is the only object you need to select.

Export to File

Select this option to write the exported objects to a file. This is a binary file that uses a format that is specific to adTempus. The file cannot be used by any other program, and cannot be modified directly.

Export to Server

Select this option to transfer the objects directly to another server. Only servers to which the Console is currently connected will be listed.

Import/Export overview

Data Import

The Data Import tool allows you to import data that has been exported from another adTempus server.

The import tool only works on files that were created by the Export tool. Files created by other methods cannot be imported.

Import from File

Specify the file to import from. All objects in the file will be imported.

When you import data from a data file, adTempus checks each object to see if it already exists. If it does, the object already in adTempus is replaced if the object in the file has a later timestamp.

When you import data, the original security settings are not imported. Each imported object is given the default security settings. You must have permission to create each kind of object that you are importing.

Arcana Scheduler Import

The Arcana Scheduler Import tool allows you to import scheduled programs from the Arcana Scheduler. To use this tool,

- You must be running the adTempus Console from the computer on which the Arcana Scheduler service is installed.
- You must be running version 2.3 or later of the Arcana Scheduler. If you are running an earlier version you will need to upgrade to version 2.3 and then run the import.
- You must be logged in to the computer under an account that has access to the scheduled programs you want to import.

The following additional restrictions apply:

- adTempus cannot import the passwords associated with the user accounts on scheduled programs and network connections. Each time adTempus encounters a new user account during the import process it will prompt you for the password for that account.
- Imported jobs will have the permissions specified by the default job security settings. Make sure these settings are to your liking before importing the jobs.

You will be able to review the imported jobs before they are saved.

Import Wizard

Change Account Passwords

The **Change Account Passwords** window allows you to easily change the password on all jobs that use a particular user account.

For example, if you have changed you Windows password, you will need to update all jobs that use your account so that they have the correct password.

This feature will only list jobs (and accounts for jobs) that you have permission to modify.

User to change password for

This list lists the user accounts for all jobs to which you have Modify permission. When you select an account, the **Select jobs to update** list is filled with a list of all jobs that use the account.

New Password

Specify the new password to use for this account. You will be prompted to confirm the password that you enter.

Select jobs to update

This list lists the jobs that use the account you have selected. Check the jobs that you want to update with the new password.

Apply

Click **Apply** to apply the new password to the selected jobs.

Object Property Sheets

Computer Shutdown Task

Computer Shutdown Task

The Computer Shutdown task allows you to shut down or restart the computer on which adTempus is running.

When shutdown is initiated, Windows displays a warning message on the computer. This message stays on top of all other windows and remains visible until the computer shuts down.

Once shutdown is initiated it can be aborted by running the `abortshutdown` program found in the adTempus program directory.

ℹ Like all other adTempus tasks, the Restart Computer Task is executed in the security context of the user account specified for the job. Therefore that user must have the authority to shutdown/restart the computer, or the task will fail.

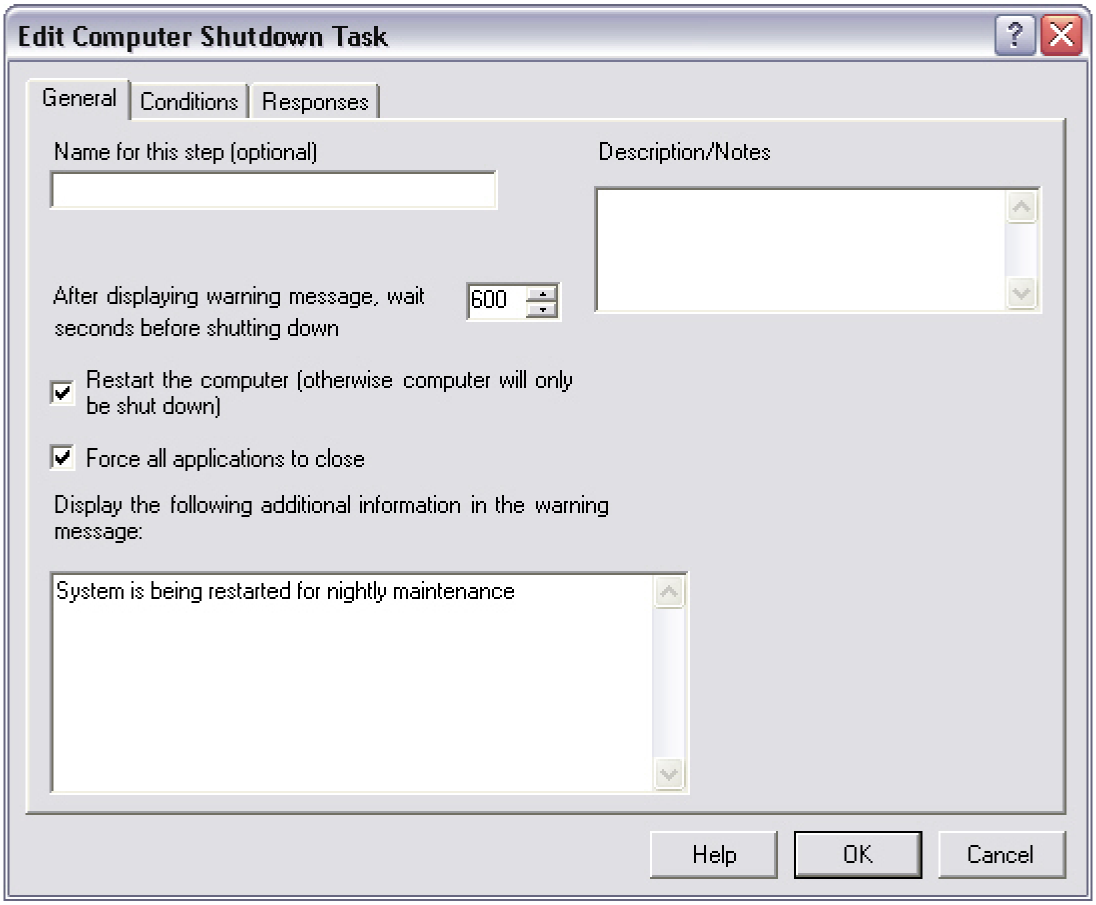
Computer Shutdown Task Properties

Tasks Overview

Steps Overview

General Page

Show Window Sample



Name for this step (optional)

Description/Notes

After displaying warning message, wait 600 seconds before shutting down

☑ Restart the computer (otherwise computer will only be shut down)

☑ Force all applications to close

Display the following additional information in the warning message:

System is being restarted for nightly maintenance

Name for this step (optional)

Optionally, specify a descriptive name for the step. Otherwise a default name is used as the description for the step.

Description / Notes

Enter any extended descriptive information or notes for this step. There is no limit on the length of the text.

After displaying warning message, wait __ seconds before shutting down

Specify the amount of time, in seconds, adTempus should wait between displaying the warning dialog and beginning the shutdown process. If the interval is set to 0, the shutdown begins immediately.

Restart the computer

Check this option if you want the computer to restart after it shuts down. If this option is not checked, the computer remains shut down.

Force all applications to close

Check this option to force all applications to close. If this option is not checked, the shutdown process may be held up by applications that do not close

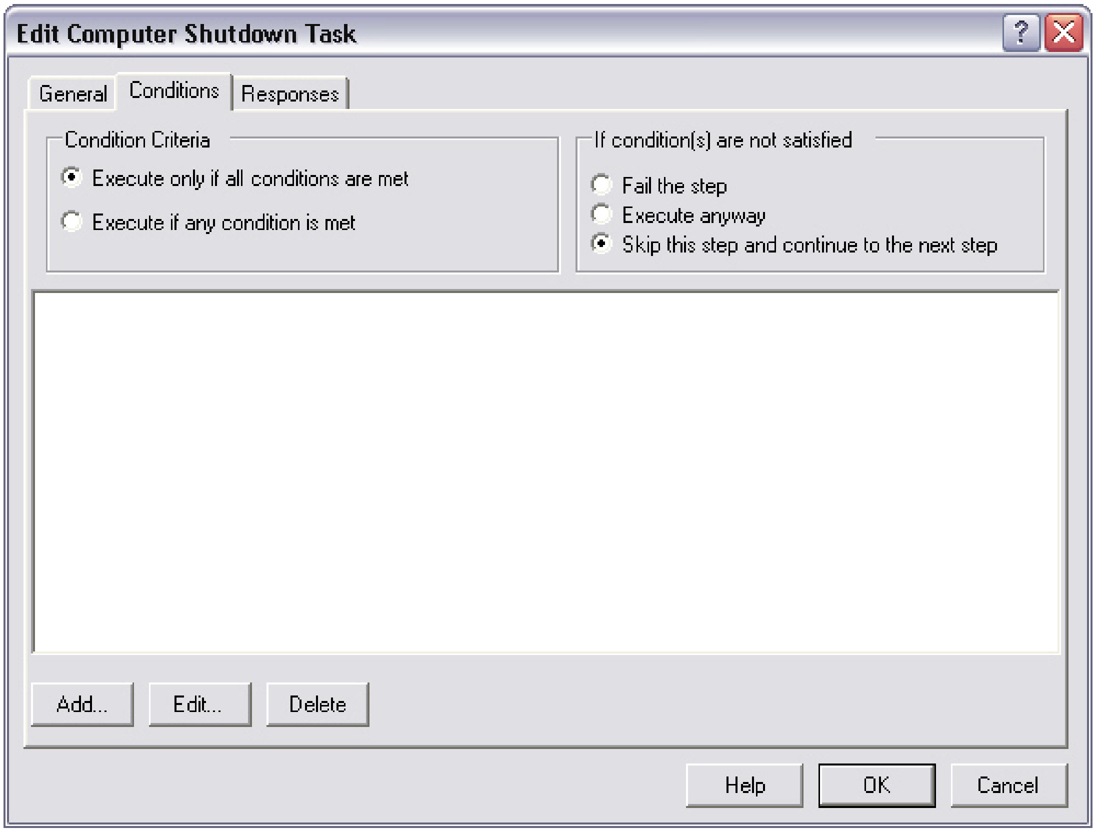
For example, an application may display a confirmation dialog box before it closes. If the application is not forced to close, shutdown will not continue until the dialog box is acknowledged by a user.

Computer Shutdown Task Properties

Computer Shutdown Task Overview

Conditions Page

Show Window Sample



Edit Computer Shutdown Task

Condition Criteria
● Execute only if all conditions are met
○ Execute if any condition is met

If condition(s) are not satisfied
○ Fail the step
○ Execute anyway
● Skip this step and continue to the next step

The **Conditions** page defines conditions that must be satisfied before the step is run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- **Execute only if all conditions are met.** The step is only executed if all of the listed conditions are met.
- **Execute if any condition is met.** The step is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- **Fail the step.** The step is not executed; the status is set to Failed.
- **Execute anyway**. The step is executed anyway.
- **Skip the step (do not report as a failure)**. The step is not run, but it is not treated as a failure. The status of the step is reported as "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the step. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

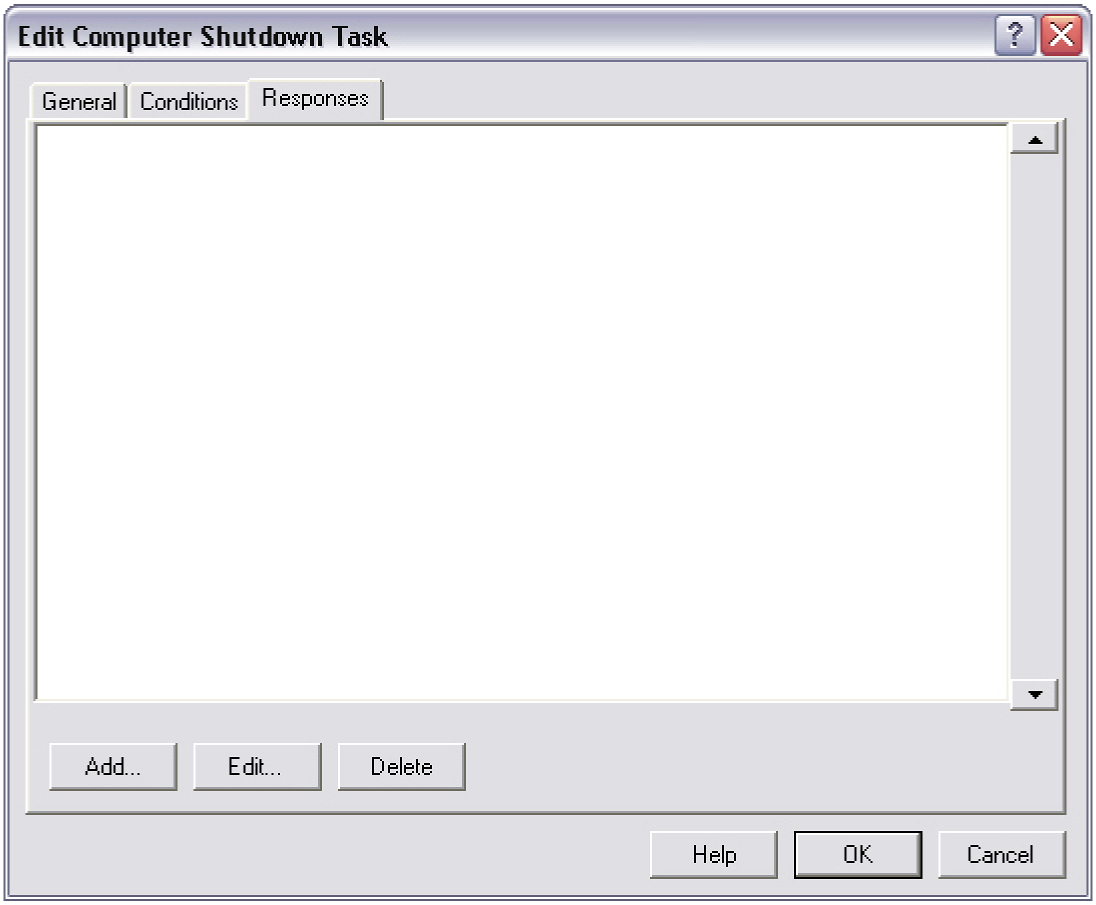
Computer Shutdown Task Properties

Computer Shutdown Task Overview

Conditions Overview

Responses Page

Show Window Sample



The **Responses** page defines the actions that adTempus should take in response to events that are fired during execution of the step.

Responses List

The **Responses** list lists the responses that have been specified for the step. You can add, edit, delete, or reorder responses. See the Responses topic for more information on the available responses.

The following events are defined for the Computer Shutdown task:

Event	Description
Beginning of step	Occurs at the beginning of the step, before adTempus has evaluated any conditions or attempted to execute the task.
Condition(s) failed	Occurs if one or more conditions is not met (occurs only once regardless of how many conditions failed).
End of step	Occurs at the end of the step, regardless of the step's outcome.
Job aborted	Occurs when the job is aborted (manually or as a

	result of a Job Control action).
Restart limit exceeded	Occurs when the step is being restarted by a Job Control action but the restart limit specified on that action has been exceeded.
Step failed	Occurs when the step's status is being set to Failed, for whatever reason.
Step is being restarted	Occurs when the step is being restarted by a Job Control action.
Step skipped	Occurs when the step is skipped because conditions were not met (if the **Skip...** option is selected on the Conditions page).
Shutdown could not be initiated	Occurs if shutdown could not be initiated (for example, if the user account used for the job does not have the necessary permission).
Shutdown initiated	Occurs once shutdown has been successfully initiated.

Computer Shutdown Task Properties

Computer Shutdown Task Overview

Responses Overview

File Capture Action

File Capture Action

The File Capture action is used to "capture" files produce by your job. Captured files are copied into the adTempus data directory and retained as part of the history for the job.

For example, if a scheduled program creates a log file of its activities each time it runs, you can capture this file so that it is retained with the job's history.

The File Capture action also gives you the option of sending captured files by e-mail.

To view captured files, view the properties for the instance once the job or step completes. The files will be found in the Captured Files list for the job; from here you can open the files or copy them to a new location.

Captured files are associated with the instance that originally captured them. When that instance is deleted, its captured files are deleted as well. See the job's General page to specify how long the job history should be retained.

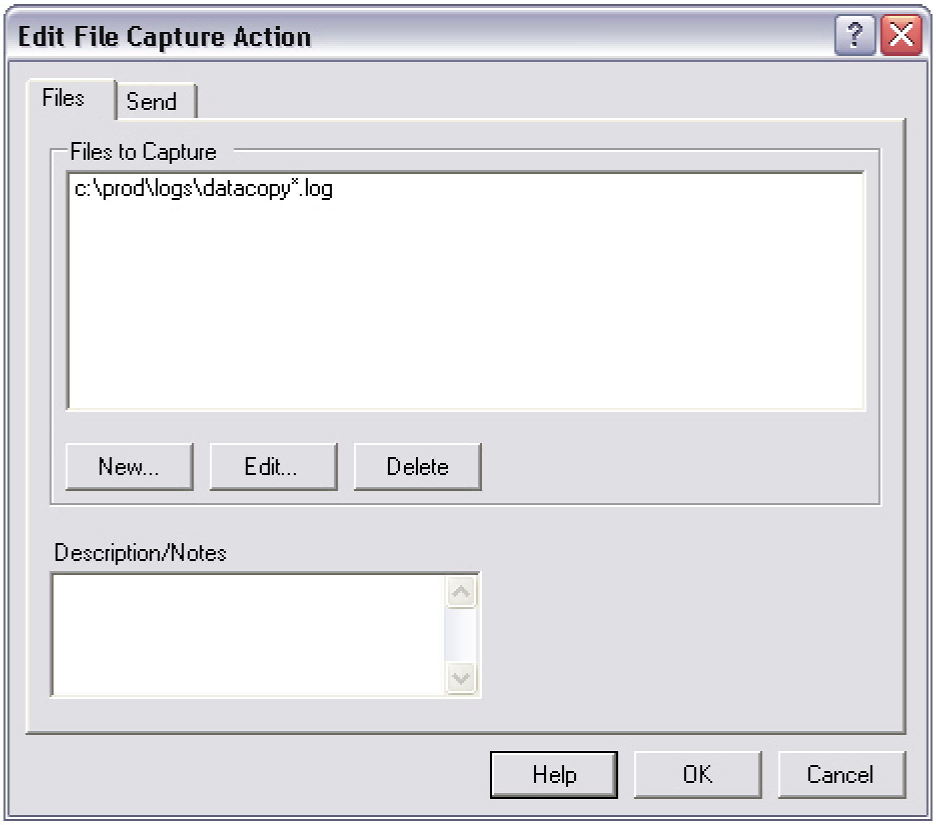
File Capture Action Properties

Responses Overview

Actions

Files Page

Show Window Sample



c:\prod\logs\datacopy*.log

The Files page lists the files that will be captured. In the Files list you can add, edit, or remove File Specifications that define the files to be captured.

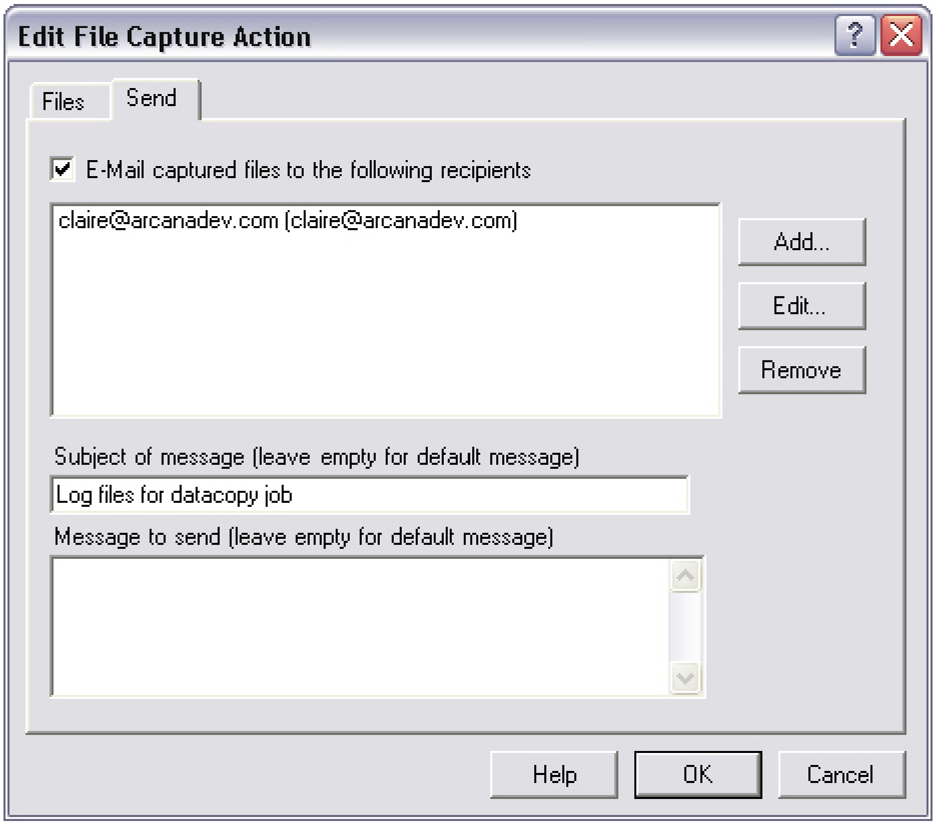
File Specification Properties

[File Capture Action Properties]

[File Capture Action Overview]

Send Page

Show Window Sample



☑ E-Mail captured files to the following recipients

claire@arcanadev.com (claire@arcanadev.com)

Subject of message (leave empty for default message)

Log files for datacopy job

Message to send (leave empty for default message)

The Send page allows you to have adTempus automatically e-mail files once they are captured.

If you want to send files without saving them in the job history, use the Notification Action instead.

E-mail captured files to the following recipients

Add, modify, or remove the notification recipients to whom the files should be sent.

Subject of message

Optionally, provide a subject for the message that will be sent. If you do not specify a subject, a default subject will be used.

Message to send

Optionally, specify the message to be sent. If you do not specify a message, a default message is used.

Notification Severity

Specify the severity (importance) of this message. Notification Recipients, Addresses, and Groups can all be configured to receive only messages that meet specified severity criteria.

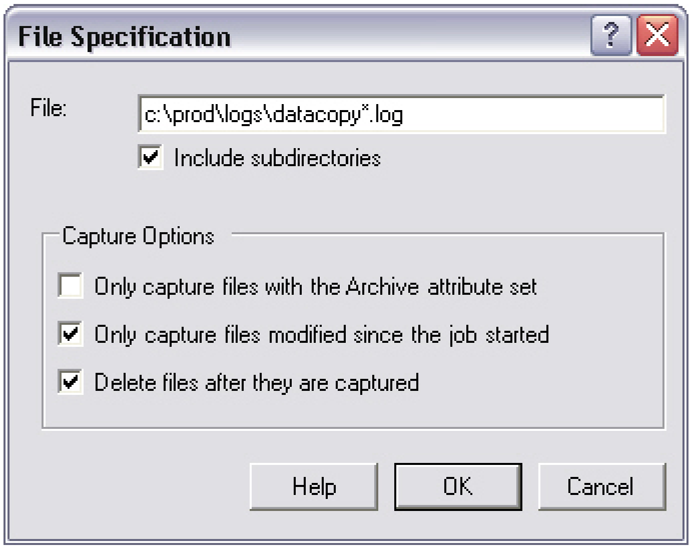
Notification Recipients

File Capture Action Properties

File Capture Action Overview

File Specification Properties

Show Window Sample



File: c:\prod\logs\datacopy*.log

☑ Include subdirectories

☐ Only capture files with the Archive attribute set

☑ Only capture files modified since the job started

☑ Delete files after they are captured

Each File Specification defines a file (or set of files) to capture.

File Specification

Specify the file(s) to be captured. You must specify the path and name, but you may use wildcards. For example:

- "c:\proddata\data1.log" looks for a specific file

- "c:\proddata\*.log" looks for any file with a ".log" extension in the "c:\proddata\" directory.

Include subdirectories

If the **File Specification** contains wildcards and the **Include subdirectories** option is checked, adTempus will check subdirectories of the directory named in the **File Specification**. For example if your **File Specification** is "c:\proddata\*.log", adTempus will look for a file with the extension ".log" in the "c:\proddata\" directory and all of its subdirectories.

Only capture files with the Archive attribute set

If this option is checked adTempus will only capture files that have the Archive attribute set (typically this indicates that the file has been modified since it was last backed up). Note that adTempus does not reset the Archive attribute after it captures the file.

Only capture files modified since the job started

If this option is checked adTempus will compare the last modification time of the file to the time that the job started. Files that were last modified before the job started will not be captured.

Delete files after they are captured

If this option is checked adTempus will delete the original file(s) after the file(s) have been copied to the adTempus data directory. Otherwise the original files are left alone.

File Capture Action Properties

File Capture Action Overview

File Condition

File Condition

A File Condition is a condition that depends on the presence or absence of a specified file or group of files.

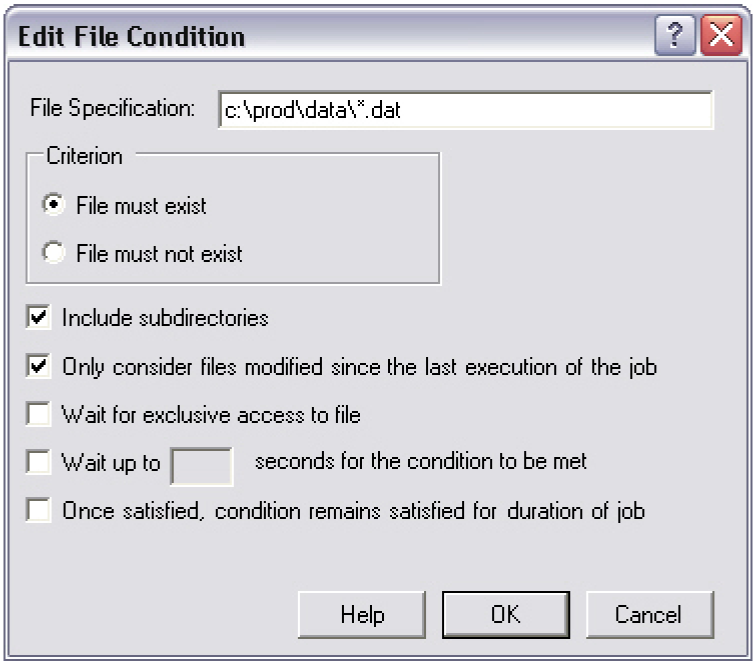
For example, you have a job processes data files that are transferred to your server by FTP. You want the job run each hour, but only if new files have been transferred. You attach a File Condition to the job so that it will only run if files exist.

File Condition Properties

Conditions Overview

File Condition Properties

Show Window Sample



File Specification: c:\prod\data\*.dat

Criterion

● File must exist

○ File must not exist

☑ Include subdirectories

☑ Only consider files modified since the last execution of the job

□ Wait for exclusive access to file

□ Wait up to   seconds for the condition to be met

□ Once satisfied, condition remains satisfied for duration of job

The options on the File Condition Properties page define the behavior of the file condition.

File Specification

Specify the files that the job is dependent on. You must specify the path and name, but you may use wildcards. For example:

- "c:\proddata\data1.log" looks for a specific file
- "c:\proddata\*.log" looks for any file with a ".log" extension in the "c:\proddata\" directory.

Criterion

Specify what the state of the file must be to satisfy the condition:

- **File must exist**. The file you specify must exist. If your **File Specification** includes wildcards, the condition is satisfied as soon as one file that matches the pattern is found.

- **File must not exist.** The file you specify must not exist. If your **File Specification** includes wildcards, the condition is satisfied only if no files that match the pattern are found.

Include subdirectories

If the **File Specification** contains wildcards and the **Include subdirectories** option is checked, adTempus will check subdirectories of the directory named in the **File Specification**. For example if your **File Specification** is "c:\proddata\*.log", adTempus will look for a file with the extension ".log" in the "c:\proddata\" directory and all of its subdirectories.

Only consider files modified since the last execution of the job

If this option is checked, adTempus will only look at files modified (or created) since the last time this job was executed.

Wait for exclusive access to the file

If this option is checked the condition is not satisfied until adTempus can get exclusive access to the file (that is, the file is not in use by any other processes).

For example, your job is dependent on a data file transferred to the server using FTP. The transfer process can take several minutes to complete. The file will appear in the directory as soon as the transfer begins, which would cause the condition to be satisfied. However, you don't want the job to run until the transfer is complete. If you check the **Wait for exclusive access to the file** option, adTempus will wait (as long as you have also checked the **Wait up to __ seconds for the condition to be met** option) for exclusive access to the file. As long as the file is still being written to, adTempus will not be able to gain exclusive access and so will keep waiting. Once the transfer is complete, adTempus will be able to gain exclusive access and will allow the job to continue.

Wait up to __ seconds for the condition to be met

Check this option and specify a wait limit if you want adTempus to wait for the condition to be met. See Condition Polling for more information.

Once satisfied, condition remains satisfied for duration of job

Check this option to indicate that adTempus should stop checking this condition once it has been satisfied. See Condition Polling for more information.

File Condition Overview

Holiday Set

Holiday Set

A holiday set is a special kind of shared schedule that can be used to specify days on which a job should *not* be run (using the settings on the Holidays page of the Schedule Trigger window).

adTempus comes preconfigured with the "Standard U.S. Holidays" set, which defines the following standard U.S. holidays:

- New Year's Day (January 1)
- Martin Luther King Day (Third Monday in January)
- Presidents' Day (Third Monday in February)
- Good Friday
- Easter
- Memorial Day (Last Monday in May)
- Independence Day (July 4)
- Labor Day (First Monday in September)
- Columbus Day (Second Monday in October)
- Veterans' Day (November 11)
- Thanksgiving (Fourth Thursday in November)
- Day After Thanksgiving
- Christmas Day (December 25)

This set can be altered to match the holiday schedule of your organization, or you may create new holiday schedules.

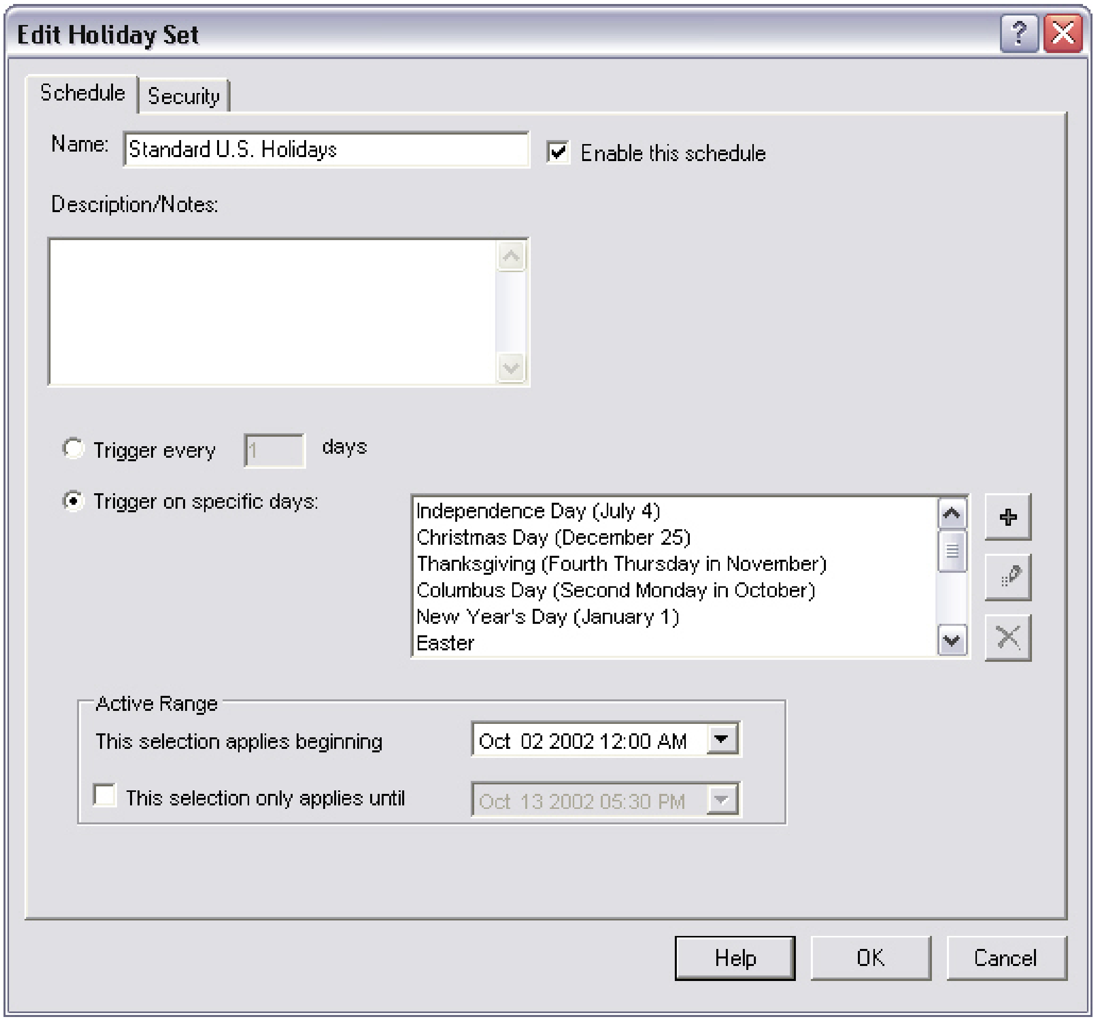
Use the settings on the Security page to specify which users are allowed to use this set of holidays.

Holiday Set Properties

Specifying holidays for jobs

Schedule Page

Show Window Sample



Edit Holiday Set

Name: Standard U.S. Holidays

☑ Enable this schedule

Description/Notes:

○ Trigger every 1 days

● Trigger on specific days:

Independence Day (July 4)
Christmas Day (December 25)
Thanksgiving (Fourth Thursday in November)
Columbus Day (Second Monday in October)
New Year's Day (January 1)
Easter

Active Range

This selection applies beginning Oct 02 2002 12:00 AM

□ This selection only applies until Oct 13 2002 05:30 PM

The **Schedule** page defines the days that constitute holidays.

Name

Provide a descriptive name for this holiday set. The name must be unique across all holiday sets and shared schedules.

Enable this schedule

Check the box to enable the holiday set or clear the box to disable it. If a holiday set is disabled, it will not appear in the list of available holiday sets when you are creating a job. Also, it will no longer cause jobs that use it to not execute on the holidays.

Date Selection

Trigger on specific days

Specify any number of specific dates or date rules to define the days that are holidays. See the Select Days topic for information on specifying days.

Active Range

Specify the earliest date and time to which this schedule applies. Optionally, specify an ending date and time.

If you specify an ending date and time, the holiday set will not be used after that date/time.

Holiday Set overview

Selecting Days

Security Page

The **Security** page defines the security settings for the shared schedule.

For information on specifying security settings see the Security Editor and Security Overview topics.

The following permissions apply to shared schedules:

Permission	Description
Full Control	Permission to perform all actions on the holiday set.
List/User	Permission to use the holiday set.
View	Permission to view the properties of the holiday set.
Modify	Permission to modify the properties of the holiday set.
Delete	Permission to delete the holiday set.
Administer security	Permission to change the security settings for the holiday set.
Change owner	Permission to take ownership of the holiday set.

Permission to create new recipients is controlled through the Default Shared Schedule Security options.

Members of the computer's Administrators group always have View and Change Owner permission, even if those permissions are not listed, or are specifically denied. An Administrator can always take ownership of a holiday set

The job's Owner always has Full Control permission for the notification recipient, even if that permission is not listed, or is specifically denied

Holiday Set overview

Specifying default holiday security

Security Overview

Security Editor

Job

Job

The Job is the central focus of adTempus. The job contains all of the information needed for adTempus to accomplish a specific goal for you. Each job contains one or more steps; each step executes a single task.

When you create or modify a job you are modifying a "template" for the job. Each time the job executes adTempus creates an instance of that job. Changes you make to a job do not have any effect on any instances that are already executing.

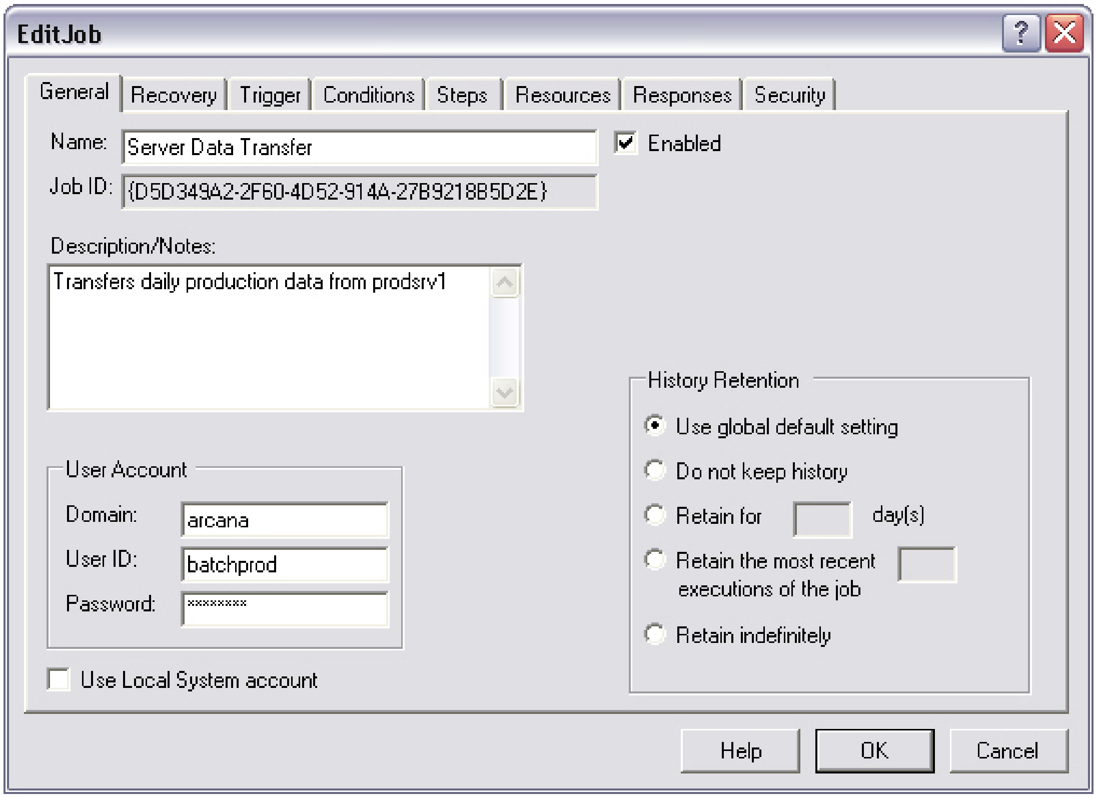
Information about each instance is retained in the job's history, which can be viewed from the Jobs View or the Execution History Query.

adTempus Architecture Overview

Job Properties

Job General Page

Show Window Sample



Name: Server Data Transfer

☑ Enabled

Job ID: {D5D349A2-2F60-4D52-914A-27B9218B5D2E}

Description/Notes: Transfers daily production data from prodsrv1

User Account

Domain: arcana

User ID: batchprod

Password: ********

□ Use Local System account

History Retention

● Use global default setting

○ Do not keep history

○ Retain for □ day(s)

○ Retain the most recent □ executions of the job

○ Retain indefinitely

Name

Provide a descriptive name for the job (255 characters maximum). The name is used throughout adTempus to identify the job, and must be unique.

If you try to use a name that is already in use on another job, you will receive an error message when you try to save the job. Note that the other job with the same name may not be visible to you (based on security settings).

The name can be changed at any time.

Enabled

Check this box to enable the job. If the box is unchecked, the job is held and will not be triggered.

Job ID

The Job ID is a 38-character system-generated unique identifier for the job. This ID remains constant even if the name of the job changes. The Job ID is shown primarily for use in diagnostic situations but can be used instead of the name in command-line utilities such as adtExec.

Description/Notes

Enter any extended descriptive information or notes for this job. There is no limit on the length of the text.

User Account Information

Specify the user ID and password for the Windows user account that the job should be run under. If the account is a domain account be sure to specify the domain in the appropriate box. If it is a local (non-domain) account, leave the domain box empty.

You can change the password for all jobs that use a particular account at once using the Global Password Change feature.

History Retention

Specify how long adTempus should retain the history for this job.

Note that if you select the **Do not keep history** option, the history (all instance details, including messages and captured files) may be purged as soon as the job completes. This means that you will not be able to see the details of any instances of the job. Also, Job Conditions that target this job will not work correctly, as they use the history.

If you select the **Retain Indefinitely** option the history will be kept until you change the retention option to a setting that allows adTempus to purge the history.

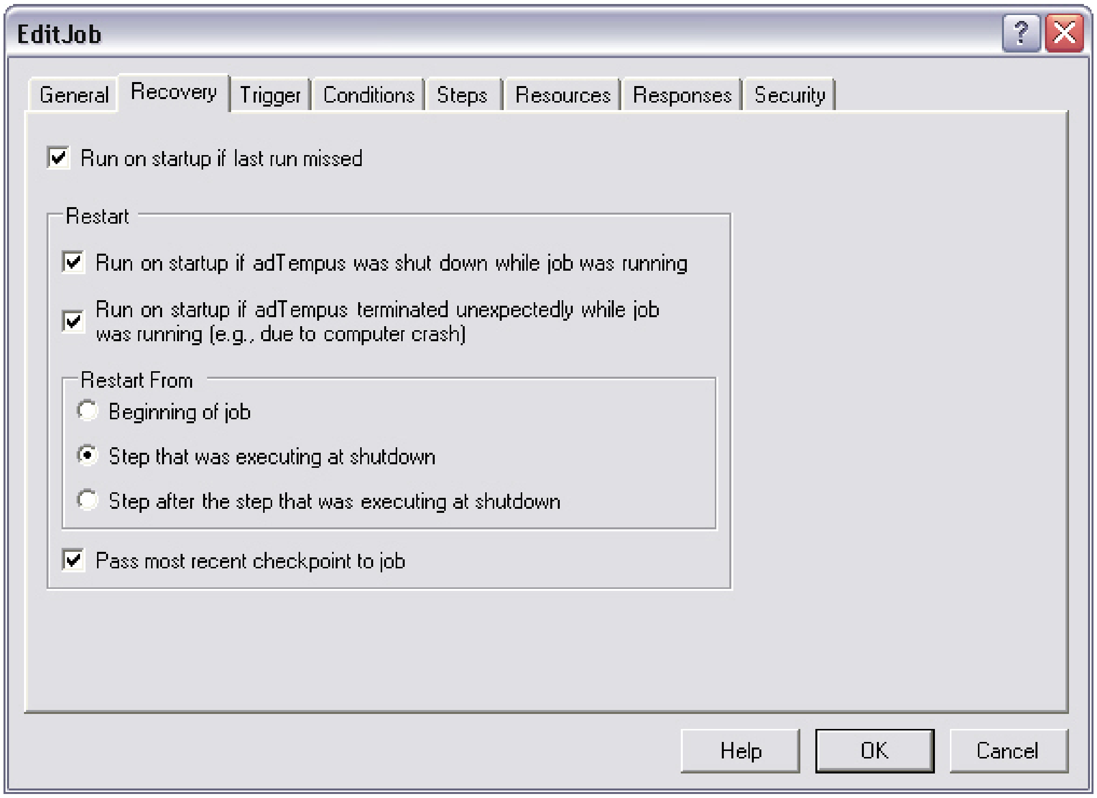
The default setting is specified in the Server Options window.

Job Overview

Job Recovery Page

Options on the Recovery page determine what adTempus should do if a job is interrupted or missed.

Show Window Sample



☑ Run on startup if last run missed

Restart

☑ Run on startup if adTempus was shut down while job was running

☑ Run on startup if adTempus terminated unexpectedly while job was running (e.g., due to computer crash)

Restart From

○ Beginning of job

● Step that was executing at shutdown

○ Step after the step that was executing at shutdown

☑ Pass most recent checkpoint to job

Run on startup if last run missed

If the last scheduled execution time for the job (based on schedule triggers defined for the job) was missed because the adTempus service was not running, adTempus will run the job when the service starts.

Restart

The Restart options allow adTempus to restart a job that was interrupted.

Run on startup if adTempus was shut down while the job was running

If the adTempus service was stopped while the job was running, adTempus will restart the job when the service is started.

Stopping the adTempus service does not terminate running programs. Your program may therefore still be running when adTempus goes to restart the job. When you use this option you should use process conditions, the Process Execution task's **Skip if already running** option, or some other safeguard to make sure that your scheduled tasks do not get re-executed when they should not.

Run on startup if adTempus terminated unexpectedly while the job was running

If the adTempus stopped unexpectedly while the job was running, adTempus will restart the job when the service is started. An "unexpected termination" is any abnormal termination of the adTempus service. This could be due to a failure of the service itself or to a system failure.

adTempus is not able to determine the reason for the abnormal termination. If the termination was due to a cause other than an unexpected system shutdown, your program may still be running when adTempus goes to restart the job. When you use this option you should use process conditions, the Process Execution task's **Skip if already running** option, or some other safeguard to make sure that your scheduled tasks do not get re-executed when they should not.

Restart From

This option determines the point from which the job will be restarted:

- The beginning (first step) of the job
- The beginning of the step that was executing when the failure occurred
- The beginning of the step following the step that was executing when the failure occurred

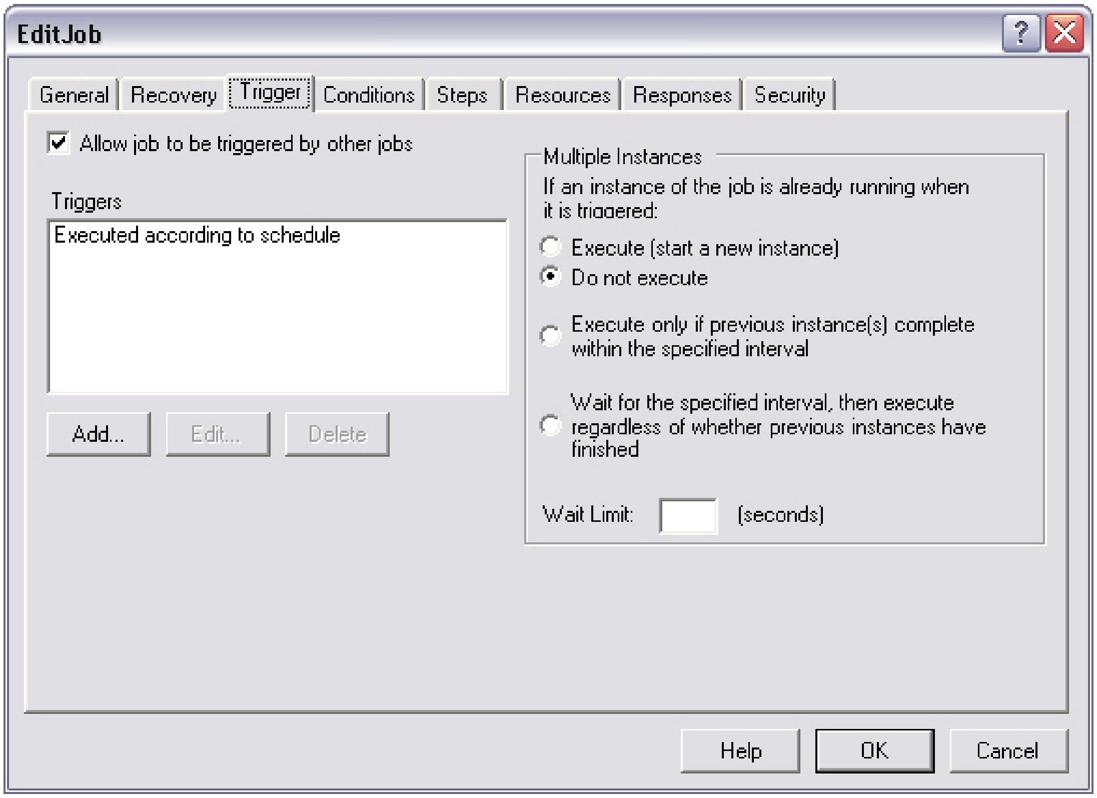
Pass most recent checkpoint to job

Regardless of what step execution begins with, adTempus can pass the task the most recent checkpoint that was set before the failure occurred. Your program or script can use this to determine where it should resume execution.

Job Overview

Job Trigger Page

Show Window Sample



☑ Allow job to be triggered by other jobs

Triggers

Executed according to schedule

Multiple Instances

If an instance of the job is already running when it is triggered:

○ Execute (start a new instance)

● Do not execute

○ Execute only if previous instance(s) complete within the specified interval

○ Wait for the specified interval, then execute regardless of whether previous instances have finished

Wait Limit: (seconds)

Options on the Trigger page determine when the job will be run.

Allow job to be triggered by other jobs

When this option is checked, the job can be executed from other jobs (by using Job Control Actions). If the option is not checked, other jobs cannot link to this job.

If you change this option after other jobs have already linked to this job, those jobs' Job Control actions will fail at execution time.

Triggers

The **Triggers** list lists the triggers that have been defined for the job. You can add, edit, or delete triggers. See the Triggers topic for information on the available trigger types.

Multiple Instances

The **Multiple Instances** options determine what adTempus should do if an instance of the job is already running when a new instance is triggered:

- **Execute (start a new instance)**. A new instance will be started. Use this option when it is acceptable for overlapping instances of a job to be running.
- **Do not execute**. The job will be skipped.

When a job is skipped due to this setting, no instance is recorded in the job's execution history, but the job's Last Status is set to "Skipped."

- **Execute only if the previous instance(s) complete within the specified interval.** adTempus will wait for the number of seconds you specify. If the previous instance(s) complete within that time, a new instance will be started. Otherwise the execution will be skipped (as in the previous option).
- **Wait for the specified interval, then execute regardless of whether previous instances have completed.** adTempus will wait for the number of seconds you specify. At the end of that time, a new instance will be started, regardless of whether previous instances have finished.

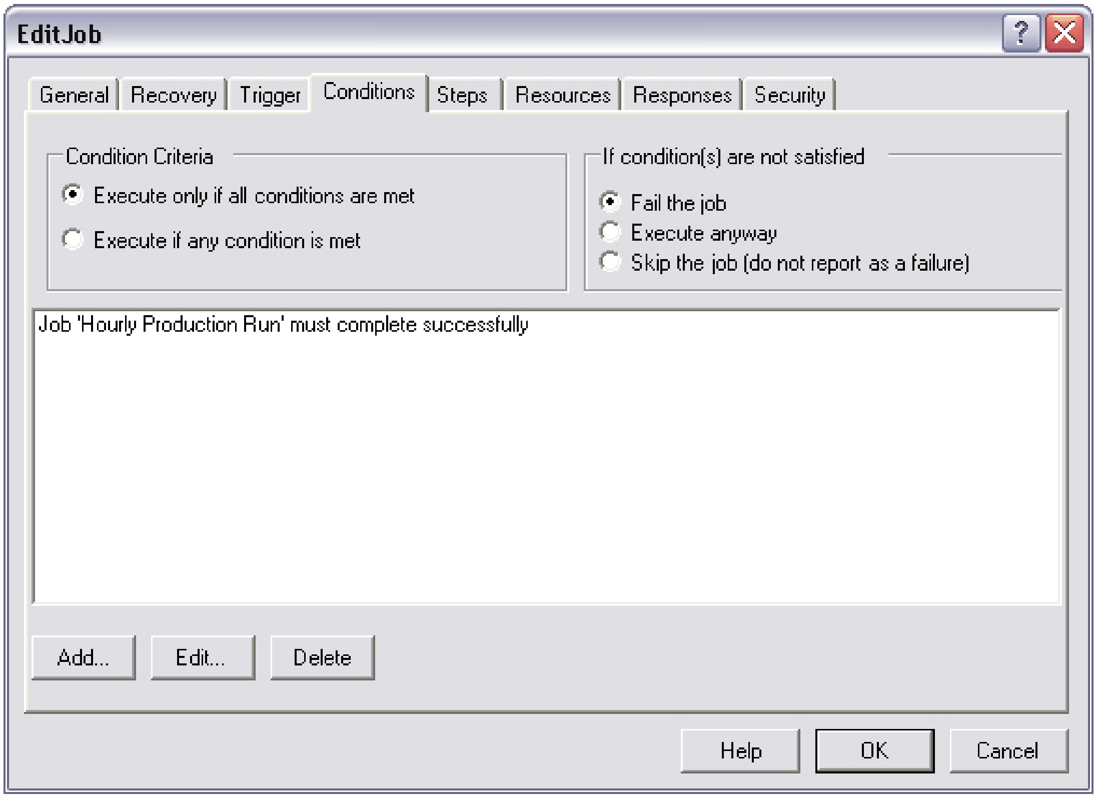
When you start a job manually using the Execute command or from another job using a Job Control action, you have the option to override these settings and force a new instance to be started.

Job Overview

Triggers Overview

Job Conditions Page

Show Window Sample



Condition Criteria

● Execute only if all conditions are met

○ Execute if any condition is met

If condition(s) are not satisfied

● Fail the job

○ Execute anyway

○ Skip the job (do not report as a failure)

Job 'Hourly Production Run' must complete successfully

The **Conditions** page defines conditions that must be satisfied before the job is run.

Do not confuse Conditions with Triggers. Conditions are evaluated after the job has been triggered; they do not *cause* the job to run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- **Execute only if all conditions are met.** The job is only executed if all of the listed conditions are met.
- **Execute if any condition is met.** The job is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- **Fail the job.** The job's steps are not executed; the instance's status is set to Failed.
- **Execute anyway**. The job is executed anyway.
- **Skip the job (do not report as a failure)**. The job is not run, but it is not treated as a failure. An instance is recorded in the job's history, with a status of "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the job. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

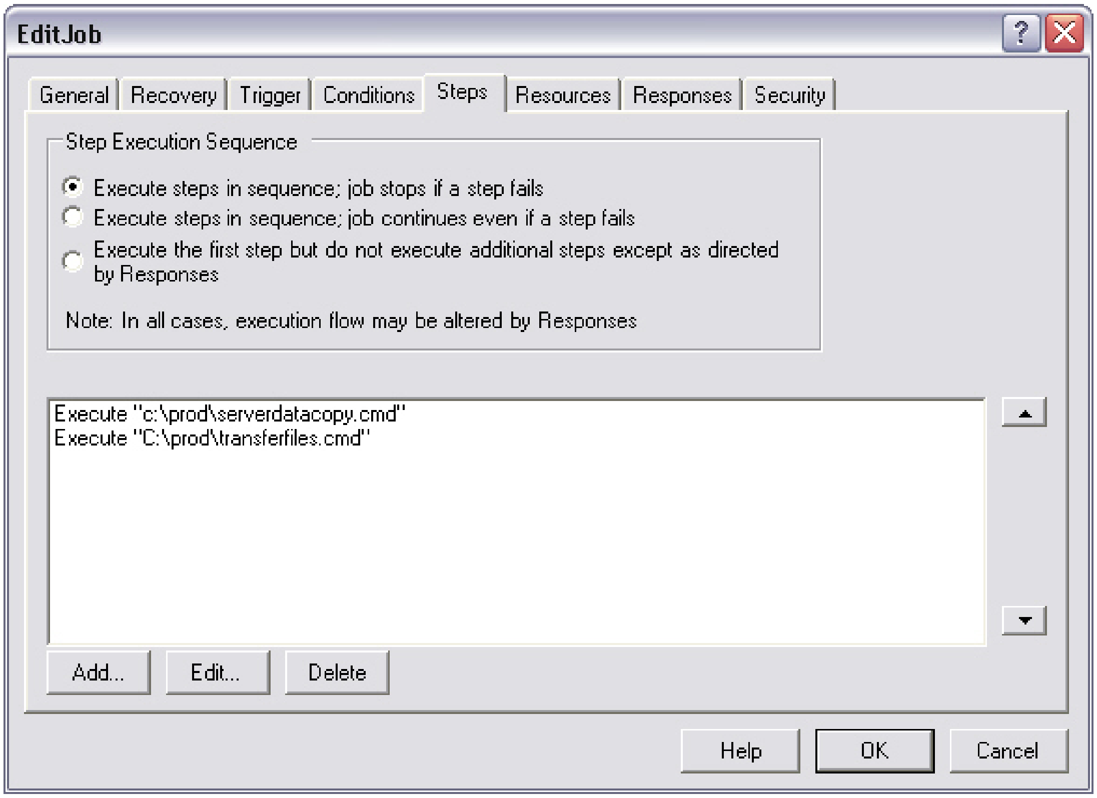
Job Overview

Conditions Overview

Job Steps Page

The **Steps** page lists the steps to be executed for the job.

Show Sample Window



● Execute steps in sequence; job stops if a step fails
○ Execute steps in sequence; job continues even if a step fails
○ Execute the first step but do not execute additional steps except as directed by Responses

Note: In all cases, execution flow may be altered by Responses

```
Execute "c:\prod\serverdatacopy.cmd"
Execute "C:\prod\transferfiles.cmd"
```

Step Execution Sequence

The **Step Execution Sequence** determines how steps should be executed in jobs with more than one step:

- **Execute steps in sequence; job stops if a step fails.** adTempus executes each job in sequence as long as all steps succeed. If a step fails, the job stops, and the job's status is "Failed."

- **Execute steps in sequence; job continues even if a step fails**. adTempus executes each job in sequence regardless of the outcome of individual steps. After all steps have been executed, the job's status is based on the status of the last step executed.
- **Execute the first step, but do not execute additional steps except as directed by Responses.** Use this option if you want to control the sequence of steps. When this option is selected, adTempus only runs the first step automatically. You must then use Job Control actions to direct the execution flow. The job's status is based on the status of the last step executed.

Regardless of which option you specify here, Job Control actions executed by Responses within the job can alter the flow of the job.

Steps

The **Steps** list lists the steps that have been defined for the job. You can add, edit, delete, or reorder steps. Each step executes a single task; see the Tasks topic for information on the available task types.

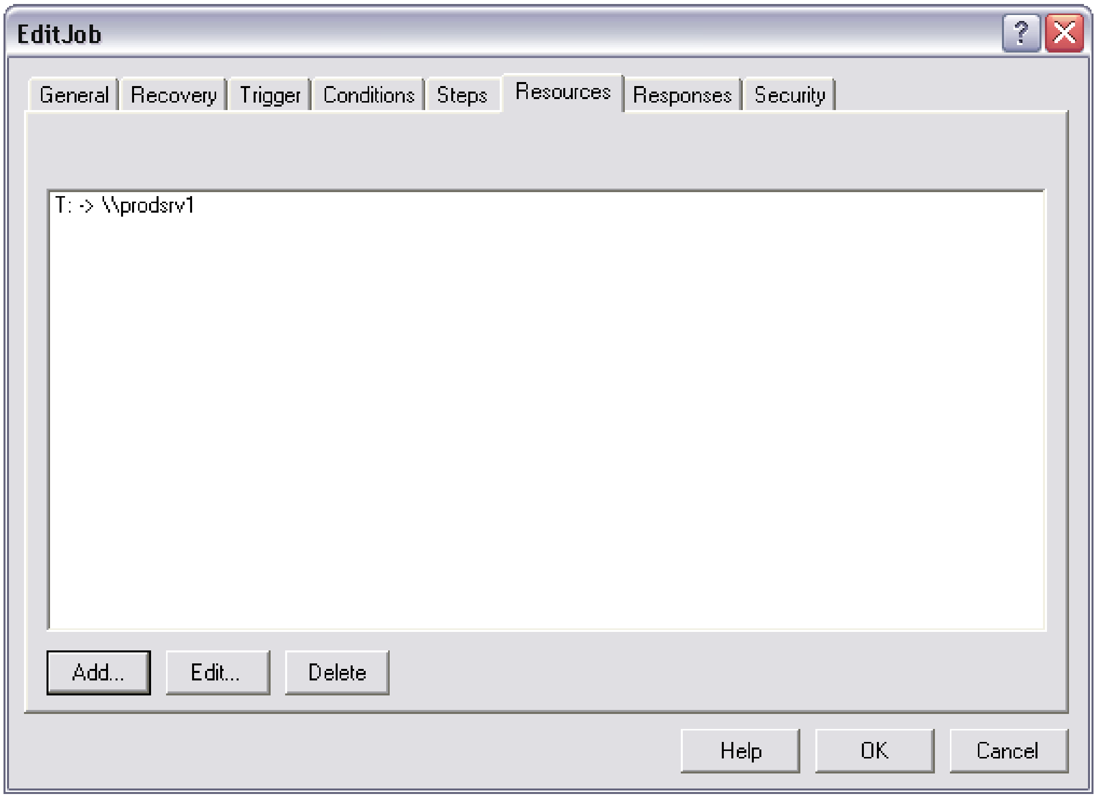
Tasks Overview

Job Overview

Job Resources Page

The **Resources** page defines resources that the job needs.

Show Window Sample



T: -> \\prodsrv1

Resources List

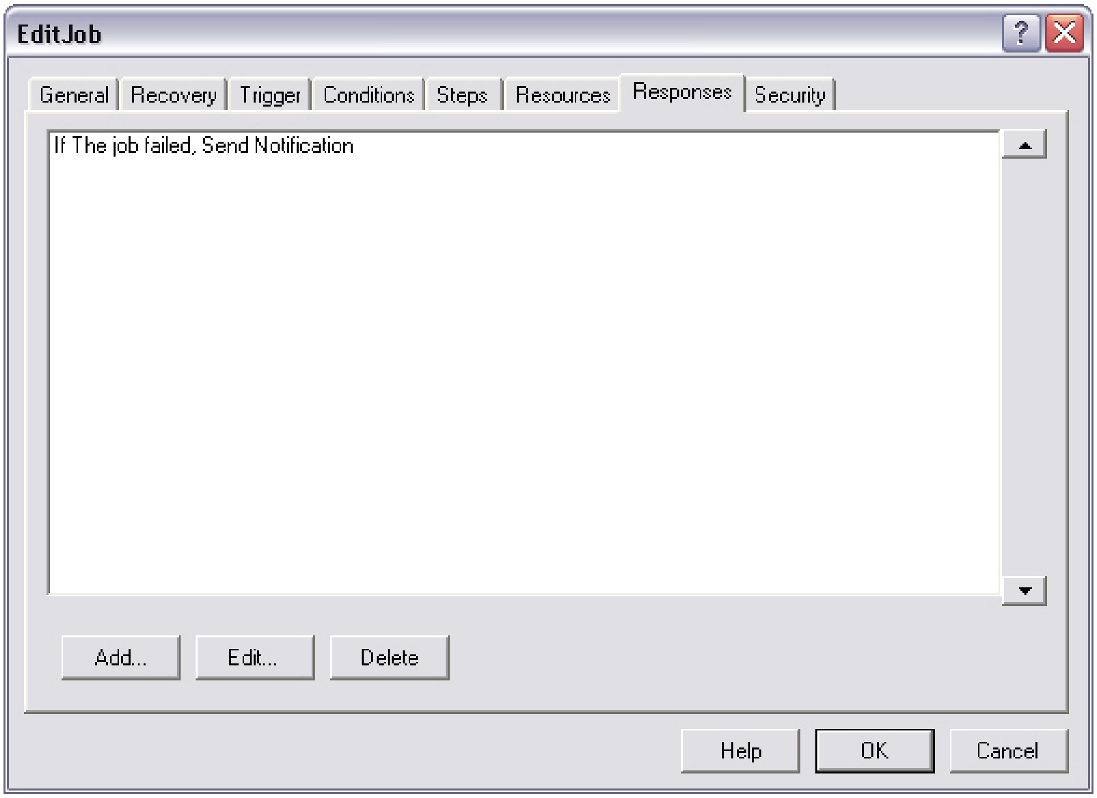
The **Resources** list lists the resources that have been specified for the job. You can add, edit, or delete resources. See the Resources topic for information on the available resource types.

Job Overview

Resources Overview

Job Responses Page

Show Window Sample



If The job failed, Send Notification

The **Responses** page defines responses that should be executed for the job.

Responses List

The **Responses** list lists the responses that have been specified for the job. You can add, edit, delete, or reorder responses. See the Responses topic for more information.

The following events are defined for jobs:

Event	Description
Job started	Occurs at the beginning of the job, before adTempus has evaluated any conditions or attempted to execute any steps.
Condition(s) failed	Occurs if one or more conditions is not met (occurs only once regardless of how many conditions failed).
End of job	Occurs at the end of the job, regardless of the job's outcome.
Job aborted	Occurs when the job is aborted (manually or as a result of a Job Control action).
Restart limit exceeded	Occurs when the job is being restarted by a Job Control action but the restart limit specified on that action has been exceeded.

Job failed	Occurs when the job is ending with a Failed status, for whatever reason.
Job is being restarted	Occurs when the job is being restarted by a Job Control action.
Job skipped	Occurs when the job is skipped, either because conditions were not met (if the **Skip...** option is selected on the Conditions page) or because another instance is already running.
Job succeeded	Occurs when the job finishes with a successful result.

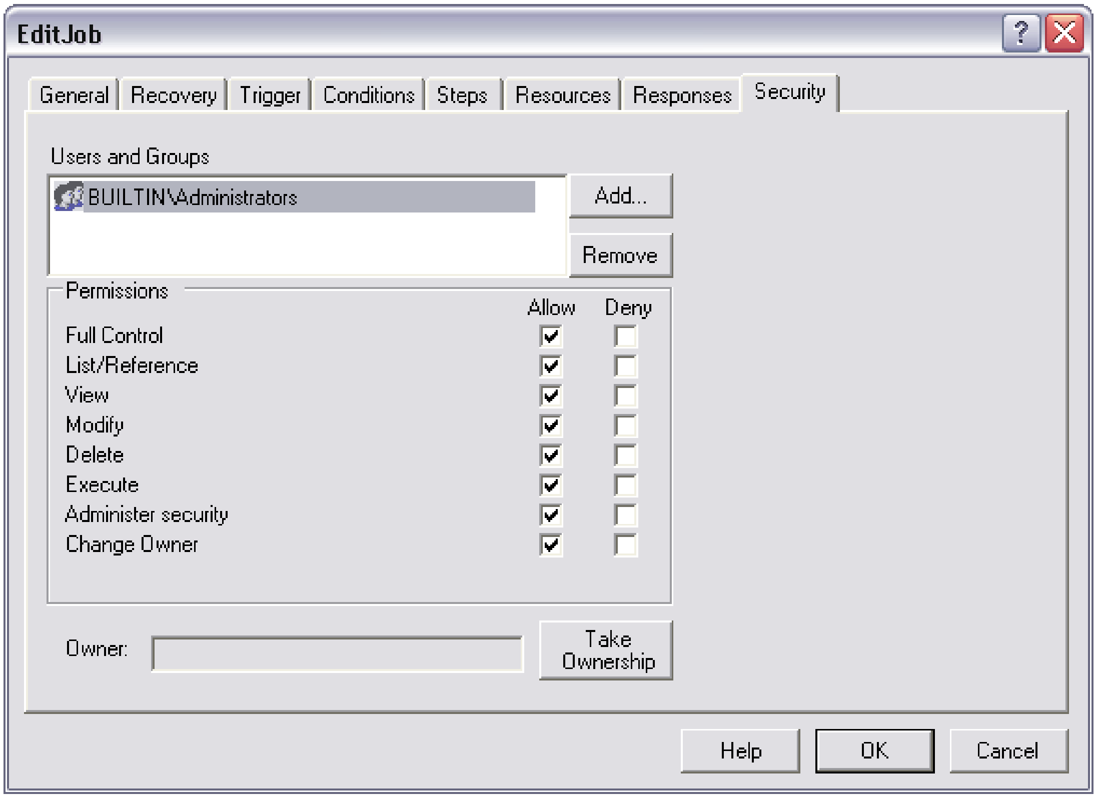
Job Overview

Responses Overview

Job Security Page

The **Security** page defines the security settings for the job.

Show Window Sample



Users and Groups

BUILTIN\Administrators

Permissions

	Allow	Deny
Full Control	☑	□
List/Reference	☑	□
View	☑	□
Modify	☑	□
Delete	☑	□
Execute	☑	□
Administer security	☑	□
Change Owner	☑	□

Owner:

For information on specifying security settings see the Security Editor and Security Overview topics.

The following permissions apply to jobs:

Permission	Description
Full Control	Permission to perform all actions on the job.
List/Reference	Permission to link to the job in Job Control Actions, Job Conditions, etc. Does not automatically grant permission to view the properties of the job.
View	Permission to view the properties of the job.
Modify	Permission to modify the properties of the job.
Delete	Permission to delete the job.
Execute	Permission to execute and abort the job through commands available in the console. Execute permission is also required for the user to link to the job using a Job Control Action.
Administer security	Permission to change the security settings for the job.
Change owner	Permission to take ownership of the job.

Permission to create new jobs is controlled through the Default Job Security options.

Members of the computer's Administrators group always have View and Change Owner permission, even if those permissions are not listed, or are specifically denied. An Administrator can always take ownership of a job.

The job's Owner always has Full Control permission for the job, even if that permission is not listed, or is specifically denied

Job Overview

Specifying default job security

Security Overview

Security Editor

Job Condition

Job Condition

A Job Condition is a condition that prevents a job or step from executing unless another job is in a specified state.

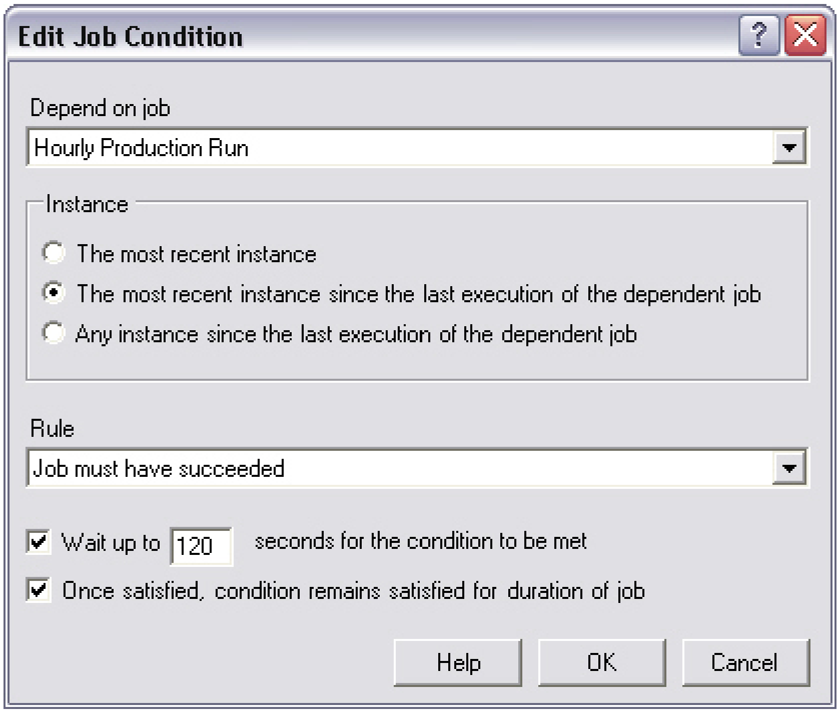
For example, you have a job that reprocesses the data cubes for your data warehouse. This job cannot run until the three jobs that extract the data for the data warehouse have run. You therefore add a Job Condition to your cube reprocess job, so that it waits on the three extract jobs.

Job Condition Properties

Conditions Overview

Job Condition Properties

Show Window Sample



Edit Job Condition

Depend on job: Hourly Production Run

Instance

○ The most recent instance

● The most recent instance since the last execution of the dependent job

○ Any instance since the last execution of the dependent job

Rule: Job must have succeeded

☑ Wait up to 120 seconds for the condition to be met

☑ Once satisfied, condition remains satisfied for duration of job

The options on the Job Condition Properties page define the behavior of the job condition.

Depend on Job

Select the job to depend on. Only jobs that you have "List/Reference" permission for are listed. The job may be on the same computer as the job you are editing, or on a different computer.

Instance

Specify which instance of the target job adTempus should look at to see if the condition is me:

- **The most recent instance.** adTempus will look at the most recent instance of the target job, regardless of when that instance executed.
- **The most recent instance since the last execution of the dependent job.** adTempus will look at the most recent instance that executed since the last execution of the dependent job.
- **Any instance since the last execution of the dependent job.** adTempus will look at all instances that have completed since the last execution of the dependent job. If any of these instances matches the Rule (see below), the condition will be satisfied.

*Example illustrating the effect of the three **Instance** options*

You are making job "Data Copy" dependent on job "Data Extract," with the Rule (see below) set to "Job must have succeeded" (that is, "Data Copy" cannot run until "Data Extract" has run successfully).

At 12:00 PM instance 312 of job "Data Copy" finishes.

At 12:15 PM instance 403 of job "Data Extract" finishes successfully.

At 12:30 PM instance 404 of job "Data Extract" fails.

At 12:45 PM instance 313 of job "Data Copy" starts.

- If the **Instance** is set to "The most recent instance," the condition is not met, because the most recent instance (404) failed.

- If the **Instance** is set to "The most recent instance since the last execution of the dependent job," the condition is not met, because the most recent instance (404) failed.
- If the **Instance** is set to "Any instance since the last execution of the dependent job," the condition is satisfied, because instance 403 succeeded.

At 1:00 PM instance 405 of job "Data Extract" finishes successfully.

At 1:15 PM instance 314 of job "Data Copy" starts. Regardless of which instance you have selected, the condition is met.

At 1:30 PM instance 315 of job "Data Copy" starts (note that "Data Extract" has not run again in the interim).

- If the **Instance** is set to "The most recent instance," the condition is met, because the most recent instance (405) succeeded. However, "Data Copy" has not been run since the last "Data Extract," so you probably would not want the "Data Copy" to run again yet.

- If the **Instance** is set to "The most recent instance since the last execution of the dependent job," the condition is not met, because "Data Extract" has not been run since the last time "Data Copy" was run.
- If the **Instance** is set to "Any instance since the last execution of the dependent job," the condition is not met, because "Data Extract" has not been run since the last time "Data Copy" was run.

Rule

Specify the rule that the target instance must satisfy:

- **Job must be running.** The target instance of the job must currently be running.
- **Job must not be running.** The target instance of the job must not currently be running.
- **Job must have succeeded.** The target instance of the job must have completed with a successful result.
- **Job must have failed.** The target instance of the job must have completed with a failure result.
- **Job must have completed.** The target instance of the job must have run, but the result is not important.

Wait up to __ seconds for the condition to be met

Check this option and specify a wait limit if you want adTempus to wait for the condition to be met. See Condition Polling for more information.

Once satisfied, condition remains satisfied for duration of job

Check this option to indicate that adTempus should stop checking this condition once it has been satisfied. See Condition Polling for more information.

Job Condition Overview

Job Control Action

Job Control Action

The Job Control action is used to execute and manipulate jobs and job steps. It can be used to

- Run, abort, restart, hold, release, or delete a job
- Run or restart a step (in the current or another job)
- Change the status of the job or step that is executing the action

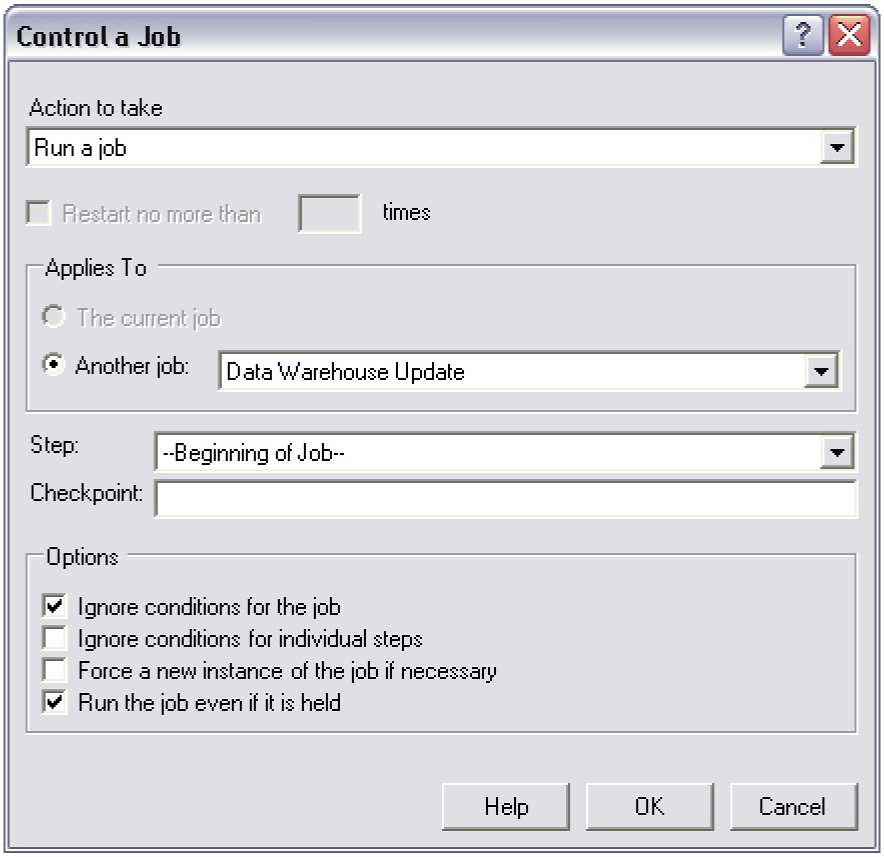
Job Control Action Properties

Responses Overview

Actions

Job Control Action Properties

Show Window Sample



Control a Job

Action to take: Run a job

☐ Restart no more than ___ times

Applies To

○ The current job

● Another job: Data Warehouse Update

Step: --Beginning of Job--

Checkpoint:

Options

☑ Ignore conditions for the job

☐ Ignore conditions for individual steps

☐ Force a new instance of the job if necessary

☑ Run the job even if it is held

The Job Control Action can be used to perform a number of job-related tasks. The following options are available.

Action to take

Click to view a list of available actions

Action	Description
Hold a job	Places the target job on hold. Can be used to hold the job that is executing the action, or another action.
Release a job	Releases the target job. Can be used to hold the job that is executing the action, or another action.

Restart the job from the beginning	Restarts the target job from the beginning. Applies only to the job that is executing the action. This results in a new instance of the job being started. Subsequent steps in the original instance are not executed.
Run a job	Runs a job. The job can be started from any step. The new job is executed asynchronously. That is, the Job Control action returns control immediately to the caller, without waiting for the target job to complete.
Terminate a job	Terminates (aborts) a running job.
Restart the step	Restarts the target step. Applies only to the step that is executing the action.
Run another step in the current job	Transfers control to another step in the current job. Once control is transferred, execution continues from the new step. Execution doe
Delete a job	Deletes the target job. Can delete the job that is executing the action, or another job. Note that if you delete the job that is executing the action, any executing instances (including the one responsible for the deletion) will continue executing until they complete.
Stop executing steps in this job	Instructs adTempus to stop automatically executing steps in the job. After this action is executed, you are responsible for executing subsequent steps as appropriate, using further Job Control actions.
Set the job's status to failed	Overrides normal success determination rules and sets the job's status to Failed.
Set the job's status to succeeded	Overrides normal success determination rules and sets the job's status to Succeeded.
Set the step's status to failed	Overrides normal success determination rules and sets the step's status to Failed.
Set the step's status to succeeded	Overrides normal success determination rules and sets the step's status to Failed.

Restart no more than __ times

This option is available only when the action is "Restart the job from the beginning" or "Restart the step."

When this option is selected, adTempus will limit the number of times that the job or step is restarted. Once the limit is exceeded, adTempus will not restart the job or step, and the "Restart limit exceeded" event will be fired.

Applies To

Determines whether the action applies to the current job (the job that is executing the action) , another job on the same server, or a job on another computer. To target another job, you must have "Execute" authority for that job.

Step

This option is available only when the action is "Run a job" or "Run another step in the current job."

If the action is "Run a job" you may optionally specify the step at which execution should begin. If you do not select a step, the job is executed from the beginning.

If the action is "Run another step in the current job" you must specify the step to execute.

Checkpoint

This option is available only when the action is "Run a job" or "Run another step in the current job." You may optionally specify a checkpoint to be passed to the job or step.

Ignore conditions for the job

If this option is checked, adTempus will ignore any conditions that are defined for the target job (this forces the job to execute even if the conditions are not met). Conditions at the step level are not ignored (see next option).

Ignore conditions for individual steps

If this option is checked, adTempus will ignore any conditions that are defined for the target step, or for the steps of the target job (this forces the step(s) to execute even if the conditions are not met).

Force a new instance of the job if necessary

This option overrides the Multiple Instances option for the target job, forcing adTempus to start a new instance of the job even if another instance is already running.

Run the job even if it is held

If this option is checked, adTempus runs the target job regardless of whether it is held. If the option is not checked, the target job will not be run if it is held.

Job Control Action Overview

Network Resource

Network Resource

The Network Resource allows you to specify a network connection that must be established in order for a job to run. Using the network resource you may optionally map network drive letters for the job.

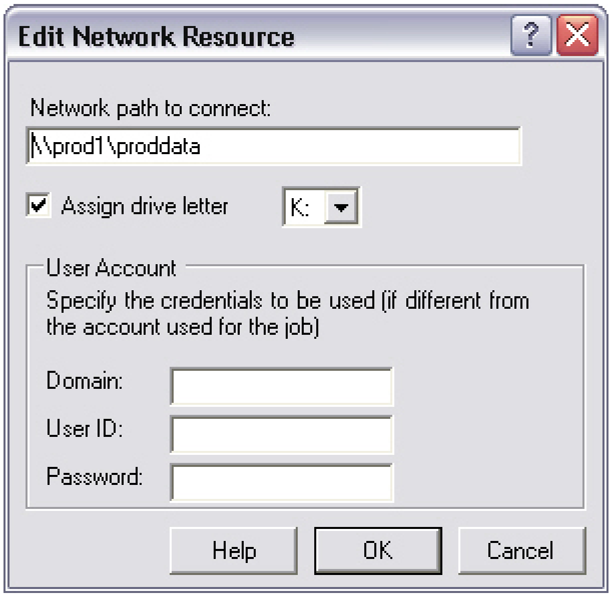
For more information on network access for jobs, see the Network Access topic.

Network Resource Properties

Resources Overview

Network Resource Properties

Show Window Sample



Network path to connect: \\prod1\proddata

☑ Assign drive letter K:

User Account

Specify the credentials to be used (if different from the account used for the job)

Domain:

User ID:

Password:

The Network Resource properties define a network connection that needs to be made for your job.

See the Network Access topic for important information on network resources.

Generally there is no need to explicitly connect network resources unless you need to assign a drive letter or specify user credentials other than those used by the job. If neither of these is the case, your tasks can refer to the resources by their UNC paths without explicitly making a connection.

Network path to connect

Specify the network path to connect to, using UNC notation. For example, specify `\\prod1\data` to connect to the `data` share on the `prod1` server.

Assign drive letter

Check this option to assign a drive letter to the path you have selected. The drive letter will be mapped to the network connection for the duration of the job, allowing your tasks to refer to the resources using drive letters.

Drive letters are assigned at the computer—not user—level. This means that the drive letter you select must be available at the time the job runs. If it is not, adTempus will not be able to assign the drive letter, and the job will fail.

User Account

Specify the credentials adTempus should use when making the network connection. If the account that the job is running under has access to the resource, there is no need to specify credential here: specify credentials only if you need to user a User ID different from the one the job is using.

Notification Action

Notification Action

The Notification action is used to send notification messages (by e-mail or network broadcast) regarding the status of a job.

Notification Action Properties

Notification Task

Responses Overview

Actions

Notification Action Properties

The Notification Action properties are divided among the following pages:

Notification

Attachments

Notification Recipient

Notification Address

The **Notification Address** window defines the properties for an e-mail or other notification address for a Notification Recipient. Each recipient may have any number of addresses.

For example, a person may have an e-mail address that goes to her mailbox on the company mail system, and another address for sending messages to her text pager.

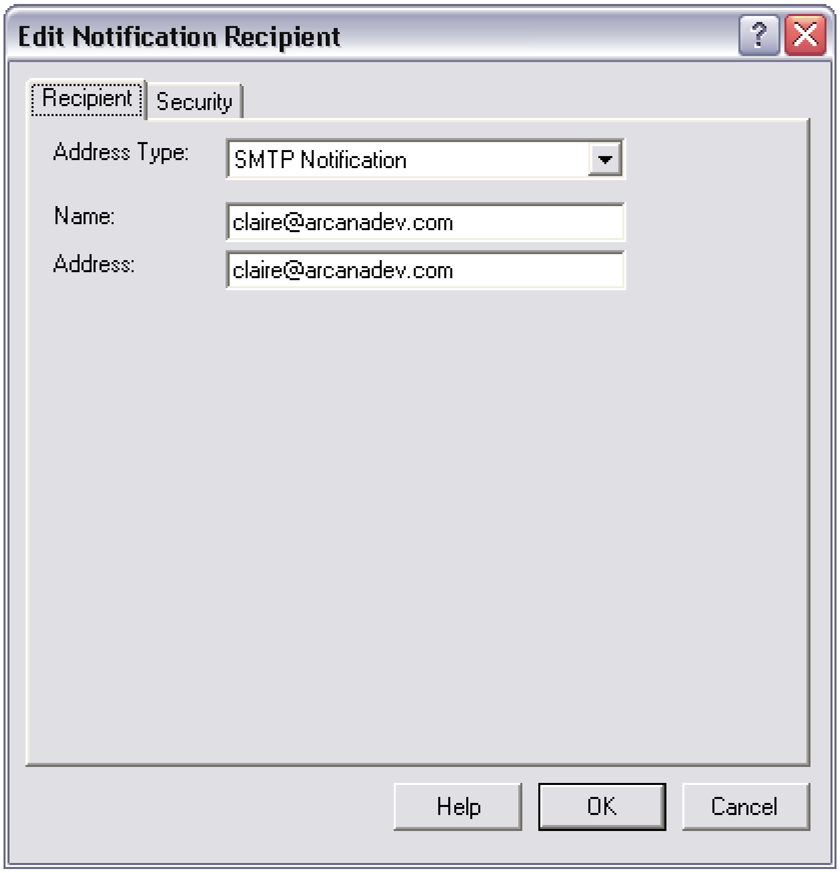
For each address, you can use the Severity and Schedule settings to determine the conditions under which messages will be sent to the address.

Notification Address Properties

Notification Recipient

Address Page

Show Window Sample



Address Type: SMTP Notification

Name: claire@arcanadev.com

Address: claire@arcanadev.com

The Address page defines the general properties for the address.

Address Type and Recipient

Select the type of recipient, and specify the recipient. The following recipient types are supported:

Address Type	Description	Recipient Format
SMTP Notification	The recipient is notified using SMTP e-mail messages. Before you use SMTP recipients you must configure SMTP notification.	Specify a valid SMTP e-mail address, e.g., "claire@arcanadev.com".
Network Broadcast Notification	The recipient is notified using Windows Network Broadcast messages	**Note:** If you specify the name of a user, and that user is logged in to more than one computer, notification will only be sent to the first computer the user logged in to. This is a Windows limitation.
Pager or SMS Notification	The message is sent to a pager or SMS-compatible messaging	Specify the ID of the device to send the message to. Generally this is the device's telephone

using Modem	device. The message is sent by dialing in to the service provider using a modem attached to the computer.	number. Depending on your service provider, you may need either the 10-digit or 7-digit telephone number. See the SMS messaging topic for more information.
Pager or SMS Notification using SNPP	The message is sent to a pager or SMS-compatible messaging device. The message is sent by connecting to the service provider over the Internet, using the SNPP Protocol.	Specify the ID of the device to send the message to. Generally this is the device's telephone number. Depending on your service provider, you may need either the 10-digit or 7-digit telephone number. See the SMS messaging topic for more information.
Pager or SMS Notification using SMPP	The message is sent to a pager or SMS-compatible messaging device. The message is sent by connecting to the service provider over the Internet, using the SMPP Protocol.	Specify the ID of the device to send the message to. Generally this is the device's telephone number. Depending on your service provider, you may need either the 10-digit or 7-digit telephone number. See the SMS messaging topic for more information.
Jabber Instant Messenger	The message is sent using the Jabber instant messaging protocol.	Specify the Jabber ID of the recipient, e.g., "arcanadevelopment@jabber.org".
Microsoft MSN Messenger	The message is sent using a Microsoft Windows (MSN) messaging server.	Specify the Windows Messenger ID of the recipient, e.g., "claire@arcanadev.com".

Service Provider

If you have selected an SMS or Instant Messaging address type, you must select the service provider that will be used to deliver the message. If no service provider has been defined, you will need to create one.

If a single service provider has been defined that matches the type of address you are editing, that provider will be selected automatically. Otherwise you will be presented with a choice of the available providers.

To change to a different provider, click the "**...**" button.

Note: When you are sending messages to SMS devices you will generally need to select the service provider that provides service for the device. For example, if you are creating notification addresses for a SkyTel pager and a Metrocall pager, each will have a separate service provider, because SkyTel will not deliver messages to Metrocall devices, and vice-versa. See the SMS messaging topic for more information.

Limit messages to

Check this option to limit the length of messages sent to this address. For example, if your text pager has a 300-character limit, you can specify that limit here.

When this option is checked, adTempus will not include any attachments on messages sent to this address.

If you are sending notification to a pager or other SMS device, the maximum message length accepted by the carrier is defined in the Service Provider Properties. The Service Provider Properties also include an option to have adTempus automatically split longer messages into multiple messages so that the entire message can be delivered to you.

In some cases, though, you may not want the full message sent. For example, if you pay for each page or character sent, you may not want a 10-part message delivered to you in its entirety. You can therefore limit the total number of characters that will be sent, which will also limit the number of pages.

The various options work together as follows:

Auto-split option set for service provider?	Message limit specified for address?	Result
Yes	Yes	The message text is first truncated to the limit specified for the address. It is then split into as many messages as necessary to keep each message within the limit specified for the provider.
Yes	No	The message text is split into as many messages as necessary to keep each message within the limit specified for the provider.
No	Yes	The message text is first truncated to the limit specified for the address. If the limit specified for the provider is less, the message is further truncated to stay within that limit.
No	No	The message text is truncated to the limit specified for the service provider.

Severity

Use this option to specify the severity conditions that a notification message must meet in order to be sent to this address.

Configuring SMTP notification

SMS Messaging

Sending notification for jobs

Notification Recipient Security Page

The **Security** page defines the security settings for the notification recipient.

For information on specifying security settings see the Security Editor and Security Overview topics.

The following permissions apply to notification recipients:

Permission	Description
Full Control	Permission to perform all actions on the recipient.
List/Use	Permission to use the recipient.
View	Permission to view the properties of the recipient.
Modify	Permission to modify the properties of the recipient.
Delete	Permission to delete the recipient.
Administer security	Permission to change the security settings for the recipient.
Change owner	Permission to take ownership of the recipient.

Permission to create new recipients is controlled through the Default Notification Recipient Security options.

Members of the computer's Administrators group always have View and Change Owner permission, even if those permissions are not listed, or are specifically denied. An Administrator can always take ownership of a notification recipient.

The job's Owner always has Full Control permission for the notification recipient, even if that permission is not listed, or is specifically denied

Specifying default notification recipient security

Security Overview

Security Editor

Process Condition

Process Condition

A Process Condition is a condition that depends on the state of a Windows process (program). Unlike a Job Condition the Process Condition allows you to wait on a process that is not under the control of adTempus.

For example, you have a job that requires exclusive access to a data file. This file is also used by a process that is often run by users outside of adTempus. By adding a Process Condition that targets this external process you can prevent adTempus from running the job when the data file is in use.

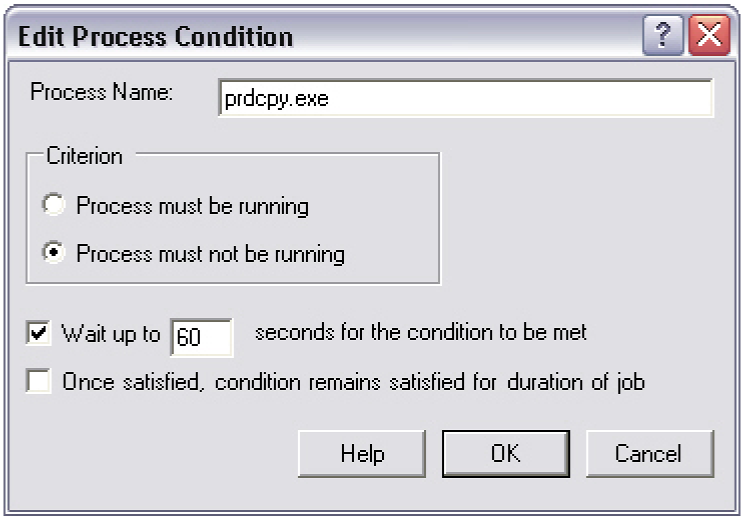
If you want to prevent a Program Execution Task from running if the target process is already running, the Skip this step if the process is already running option provides a one-step alternative to the Process Condition.

Process Condition Properties

Conditions Overview

Process Condition Properties

Show Window Sample



Edit Process Condition

Process Name: prdcpy.exe

Criterion

○ Process must be running

● Process must not be running

☑ Wait up to 60 seconds for the condition to be met

□ Once satisfied, condition remains satisfied for duration of job

The options on the Process Condition Properties page define the behavior of the process condition.

Process Name

Specify exact name of the executable process to depend on. The process name must be specified exactly as it appears in the Processes list in the Windows Task Manager.

Note that this must be an executable (.exe). The process condition cannot target batch files (.bat, .cmd), scripts (.vbs, .js), etc. The process condition also cannot currently target 16-bit processes (processes that appear under NTVDM.EXE in the Task Manager).

Criterion

Specify what the state of the process must be to satisfy the condition:

- **Process must be running**. The process you specify must be running.
- **Process must not be running.** The process you specify must not be running.

Wait up to __ seconds for the condition to be met

Check this option and specify a wait limit if you want adTempus to wait for the condition to be met. See Condition Polling for more information.

Once satisfied, condition remains satisfied for duration of job

Check this option to indicate that adTempus should stop checking this condition once it has been satisfied. See Condition Polling for more information.

Process Condition Overview

Process Termination Task

Process Termination Task

The Process Termination task allows you to have adTempus terminate a process that is executing outside of adTempus.

Be sure you understand how adTempus terminates a process before using this task. Terminating a process is not always "clean," and this can lead to problems (such as data corruption) with your application.

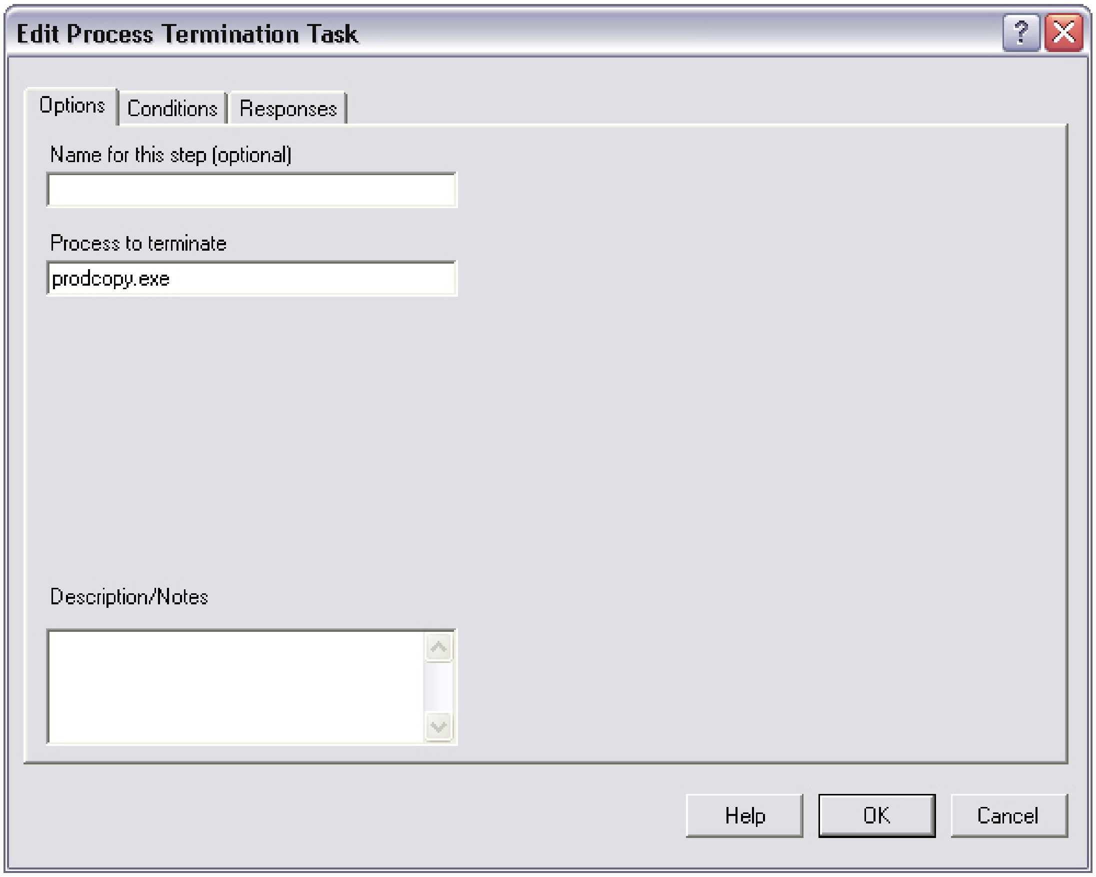
We suggest that you use a "cleaner" method for terminating applications if possible. See the How to Tell an Application to Close topic for suggestions.

Process Termination Task Properties

Tasks Overview

Options Page

Show Window Sample



Name for this step (optional)

Process to terminate
prodcopy.exe

Description/Notes

Name for this step (optional)

Optionally, specify a descriptive name for the step. Otherwise the name of the target process is used as the description for the step.

Description / Notes

Enter any extended descriptive information or notes for this step. There is no limit on the length of the text.

Process to terminate

Specify exact name of the executable process to terminate. The process name must be specified exactly as it appears in the Processes list in the Windows Task Manager.

Note that this must be an executable (.exe). The task cannot target batch files (.bat, .cmd), scripts (.vbs, .js), etc. The task also cannot currently target 16-bit processes (processes that appear under NTVDM.EXE in the Task Manager).

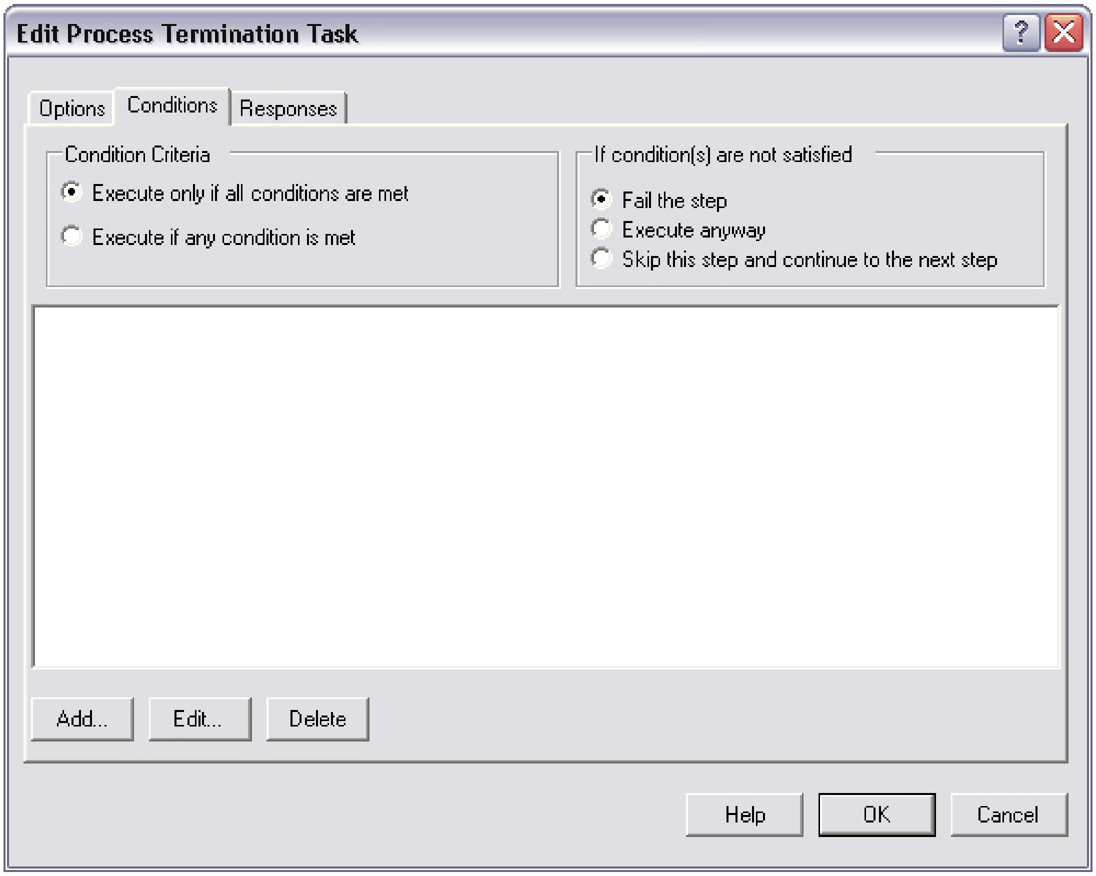
If more than one instance of the process is running, all instances will be terminated.

Process Termination Task Properties

Process Termination Task Overview

Conditions Page

Show Window Sample



Condition Criteria

● Execute only if all conditions are met

○ Execute if any condition is met

If condition(s) are not satisfied

● Fail the step

○ Execute anyway

○ Skip this step and continue to the next step

The **Conditions** page defines conditions that must be satisfied before the step is run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- **Execute only if all conditions are met.** The step is only executed if all of the listed conditions are met.
- **Execute if any condition is met.** The step is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- **Fail the step.** The step is not executed; the status is set to Failed.
- **Execute anyway**. The step is executed anyway.
- **Skip the step (do not report as a failure)**. The step is not run, but it is not treated as a failure. The status of the step is reported as "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the step. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

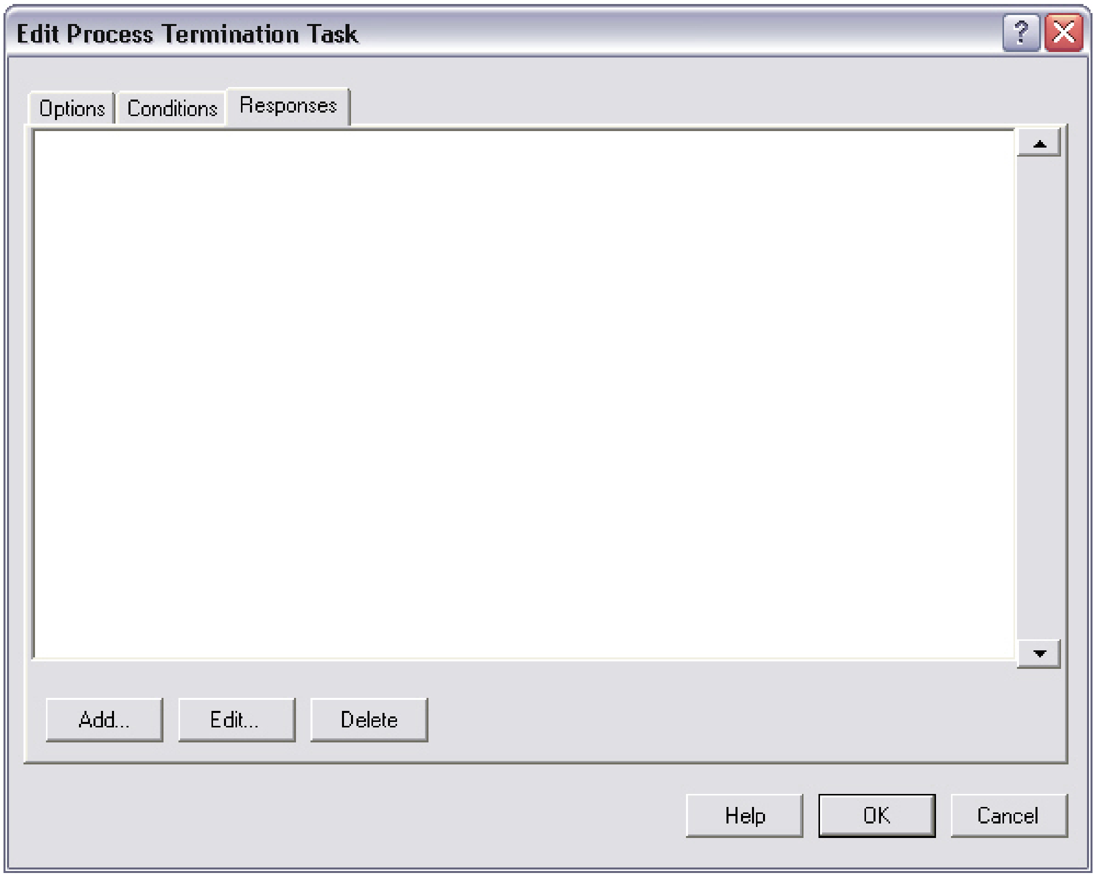
Process Termination Task Properties

Process Termination Task Overview

Conditions Overview

Responses Page

Show Window Sample



The **Responses** page defines the actions that adTempus should take in response to events that are fired during execution of the step.

Responses List

The **Responses** list lists the responses that have been specified for the step. You can add, edit, delete, or reorder responses. See the Responses topic for more information on the available responses.

The following events are defined for the Process Termination task:

Event	Description
Beginning of step	Occurs at the beginning of the step, before adTempus has evaluated any conditions or attempted to execute the task.
Condition(s) failed	Occurs if one or more conditions is not met (occurs only once regardless of how many conditions failed).
End of step	Occurs at the end of the step, regardless of the step's outcome.
Restart limit exceeded	Occurs when the step is being restarted by a Job Control action but the restart limit specified on that

	action has been exceeded.
Step failed	Occurs when the step's status is being set to Failed, for whatever reason.
Step is being restarted	Occurs when the step is being restarted by a Job Control action.
Step skipped	Occurs when the step is skipped because conditions were not met (if the **Skip...** option is selected on the Conditions page).
Task returns a successful result	Occurs when adTempus is able to terminate all executing instances of the target process (or if no instances were running).
Task returns a failure result	Occurs when adTempus is unable to terminate one or more instances of the target process (for example, because the user account used for the job did not have the necessary permissions).

Process Termination Task Properties

Process Termination Task Overview

Responses Overview

Program Execution Task

Program Execution Task

The Program Execution Task is likely the main task that you will use in adTempus. The Program Execution task allows you to run programs and batch files and launch documents.

The program is executed under the user account specified for the job, and therefore will only have access to the files and other resources that the user has access to. This includes the target program itself: the user must have the necessary permissions to launch the program.

Note that scheduled programs do not have access to network drive letters unless those drive letters have been mapped either by adTempus (on the Resources page for the job) or by the target itself (e.g., using a "net use" command in a batch file). See the Network Access topic for more information.

Program Execution Task Properties

Script Execution Task

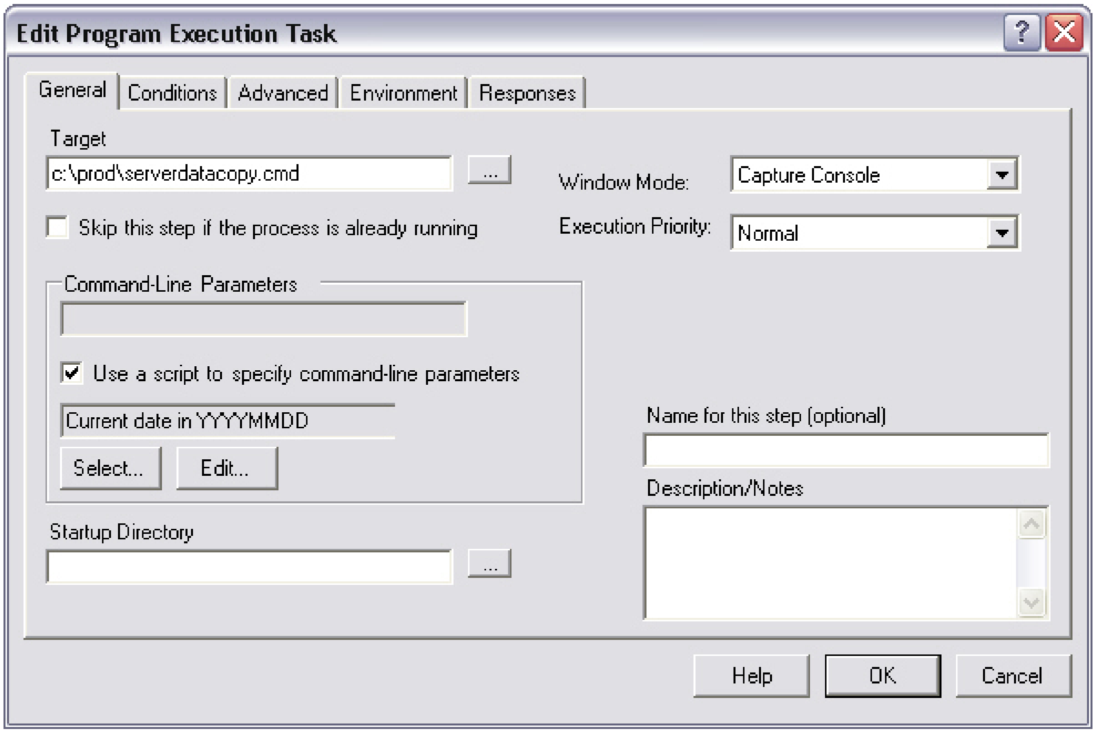
Tasks Overview

Steps Overview

Network Access for Scheduled Tasks

General Page

Show Window Sample



Edit Program Execution Task

Target: c:\prod\serverdatacopy.cmd

□ Skip this step if the process is already running

Window Mode: Capture Console

Execution Priority: Normal

Command-Line Parameters:

☑ Use a script to specify command-line parameters

Current date in YYYYMMDD

Startup Directory:

Name for this step (optional):

Description/Notes:

The **General** page defines the basic properties of the program to be run.

Target

Specify the program, batch file, script file, document file, etc., to be executed. You may specify here essentially anything that is valid when using the **Run** command from the Windows **Start** menu. Note that you must include quotes around the target if it contains spaces.

For example:

`"c:\program files\serverdatacopy\prod1.exe"`	Runs the "prod1.exe" program.
`c:\data\mydocument.doc`	Launches Microsoft Word and loads "mydocument.doc".
`c:\cmd\copydata.cmd`	Runs the "copydata.cmd" batch file.

If you use the **...** button to browse for the target, be aware that you are browsing the file system of the client. If the server is a different computer this is not going to produce the result you want. Be sure that the target you specify is valid on the server.

The Program Execution Task can be used to run Windows scripts (either by specifying the script file as the target, or specifying `wscript.exe` or `cscript.exe` as the target and passing the appropriate command-line parameters.

However, this approach does not allow you to return a result code from the script. If you use the Script Execution task instead your script can return a result code.

Skip this step if the process is already running

When this option is checked, it is like having a Process Condition on the target process. If the target process is already running (regardless of whether it is under the control of adTempus), the step will be skipped.

This option is only valid when you have specified an executable file (.exe) as the **Target**. The option will not work with other kinds of targets; nor will it work with 16-bit processes (processes that appear under NTVDM.EXE in the Task Manager).

Command-Line Parameters

Specify any command-line parameters that should be passed to the program.

There are two ways to insert dynamic data into the command line:

- use a script (see next section) to build the command line
- Use substitution tokens within the command line.

Use a script to specify command-line parameters

Check this option to use a script to set the command-line parameters. For example, you can use a script to pass the current date to the target application. See the Script Overview topic for information on working with scripts.

Before it returns your script must set the global **Result** variable to the string that you wish to have passed on the command line.

Startup Directory

Specify the startup ("working") directory to be associated with the program.

Window Mode

The Window Mode option determines how the program will appear:

- **Hidden**. The program is run on a hidden desktop and will not be visible to any user who is logged in to the computer. This option should be used for programs that do not prompt for user input.
- **Capture Console.** This option can be used with console-mode programs only. When this option is selected the program is run on a hidden desktop (and so is not visible to any user who is logged in to the computer). The output that the program writes to the console is captured in a file, which is stored in the execution history for the job. To review the output, view the properties for the instance once the step completes. The console output will be found in the Captured Files list for the job, with the name "console output.txt".
- **Normal.** The program is run on the interactive desktop, where it will be visible to any user who is logged in to the computer. This is equivalent to the behavior you would see if you launched the program using the **Run** command on the Windows **Start** menu.
- **Minimized.** The program is run on the interactive desktop but is instructed to start with its main window minimized.
- **Maximized.** The program is run on the interactive desktop but is instructed to start with its main window maximized.

If a program is run in one of the "interactive" modes (Normal, Minimized, or Maximized) it will be visible to any user logged in to the computer, regardless of whether the program is running under that user's account. The user will be able to interact with the program and could therefore "hijack" the program and use it to carry out actions the user is not normally authorized to perform.

Also, "interactive" programs are terminated when the interactive user logs off, regardless of whose account the programs are running under.

Therefore if access to the computer is not restricted, you should run programs in one of the "non-interactive" modes (Hidden or Capture Console) instead.

Execution Priority

Specify the priority that the operating system should assign to the process.

Name for this step (optional)

Optionally, specify a descriptive name for the step. Otherwise the name of the target is used as the description for the step.

Description / Notes

Enter any extended descriptive information or notes for this step. There is no limit on the length of the text.

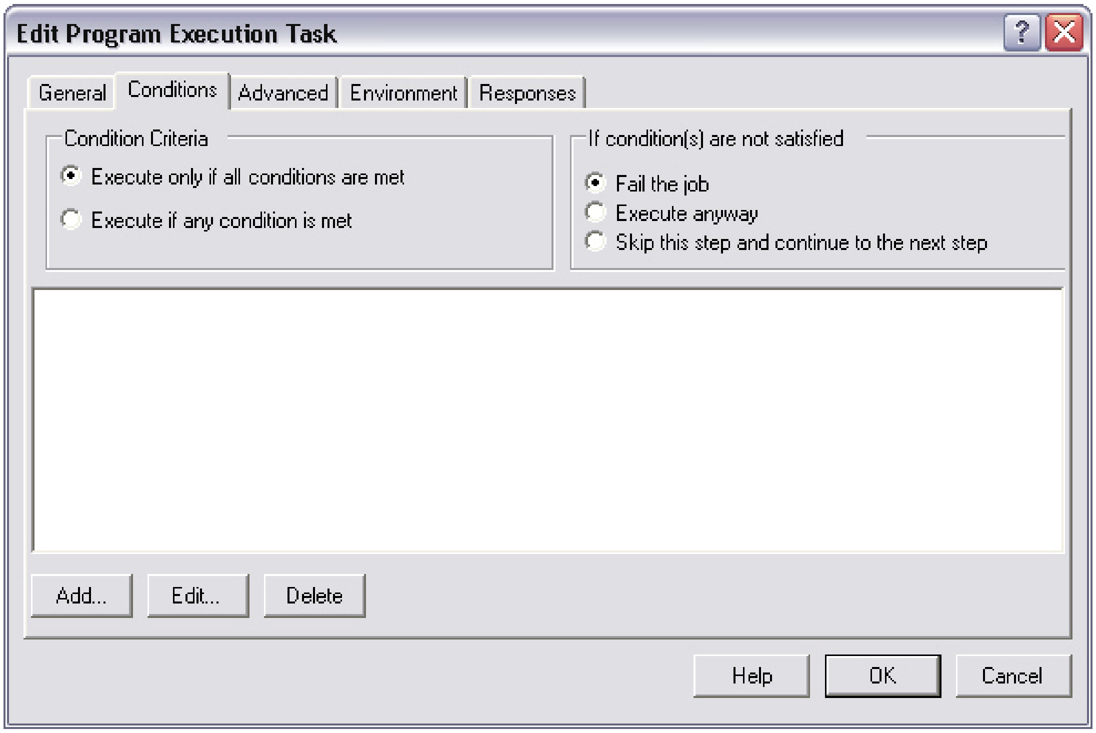
Script Execution Task

Program Execution Task Properties

Program Execution Task Overview

Conditions Page

Show Window Sample



Condition Criteria

● Execute only if all conditions are met

○ Execute if any condition is met

If condition(s) are not satisfied

● Fail the job

○ Execute anyway

○ Skip this step and continue to the next step

The **Conditions** page defines conditions that must be satisfied before the step is run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- **Execute only if all conditions are met.** The step is only executed if all of the listed conditions are met.
- **Execute if any condition is met.** The step is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- **Fail the step.** The step is not executed; the status is set to Failed.
- **Execute anyway**. The step is executed anyway.
- **Skip the step (do not report as a failure)**. The step is not run, but it is not treated as a failure. The status of the step is reported as "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the step. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

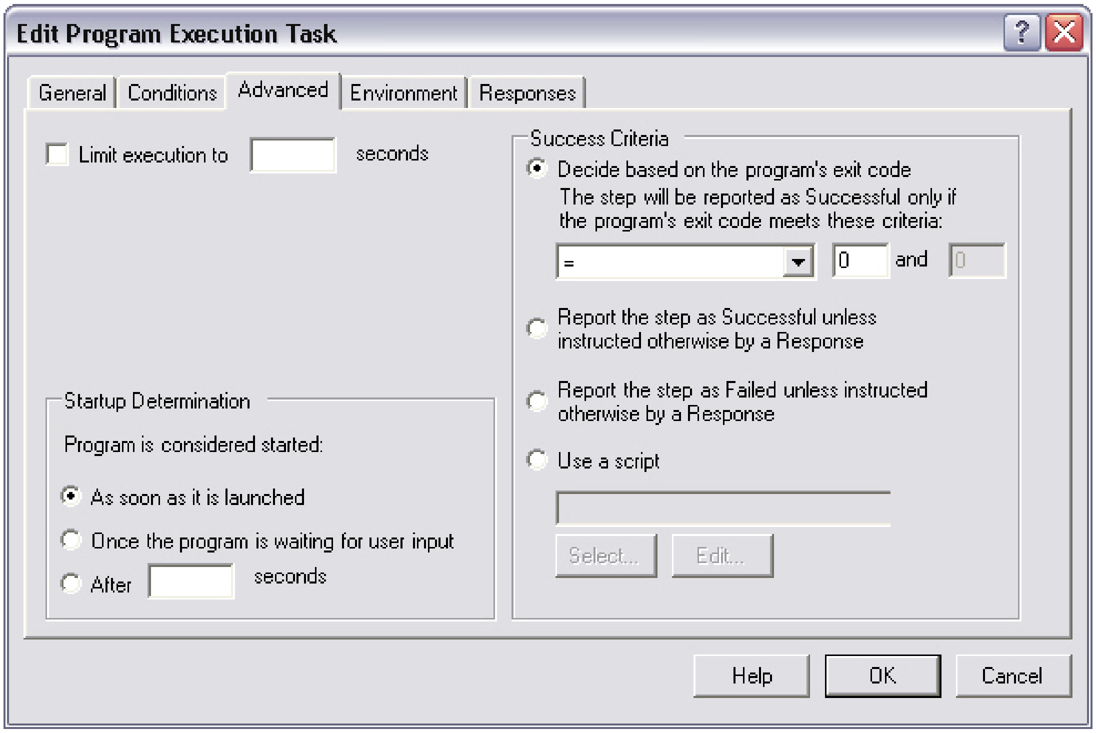
Program Execution Task Properties

Program Execution Task Overview

Conditions Overview

Advanced Page

Show Window Sample



Edit Program Execution Task

General | Conditions | Advanced | Environment | Responses

☐ Limit execution to ____ seconds

Success Criteria

● Decide based on the program's exit code

The step will be reported as Successful only if the program's exit code meets these criteria:

= ▾ 0 and 0

○ Report the step as Successful unless instructed otherwise by a Response

○ Report the step as Failed unless instructed otherwise by a Response

○ Use a script

Select... Edit...

Startup Determination

Program is considered started:

● As soon as it is launched

○ Once the program is waiting for user input

○ After ____ seconds

Help OK Cancel

The **Advanced** page defines additional options for the task

Limit execution to __ seconds

When this option is checked, adTempus will terminate the process if it is still running after the specified number of seconds.

This option should generally be used only as a last resort to terminate an application that is not behaving properly. adTempus cannot always terminate the process cleanly, and this can lead to problems (such as data corruption) with your application. See How adTempus Terminates a Process for more information.

If the task being executed is a batch file rather than an executable process, terminating it terminates only the batch file. This does not terminate any program(s) started by the batch file.

Before the process is terminated adTempus fires the "Before process is killed" event. You can attach a Response to this event and use it to try to close the application gracefully. For example, you could use a script action to send the keystrokes necessary to close whatever document is open.

If you are looking for a way to close an application at a certain time, see the How to Tell an Application to Close topic for suggestions.

Startup Determination

The **Startup Determination** option can be used to allow an application to finish initializing before job execution continues. This option is useful when a subsequent action or step needs to interact with the program. For example, if you are launching a program and then in a subsequent step sending keystrokes to the program, you must wait until the program has initialized and begun accepting keystrokes. Three options are available:

- **As soon as it is launched**. adTempus does not wait at all. As soon as it has launched the program, it moves on.
- **Once the program is waiting for user input.** adTempus waits until the program has completed initialization and its Windows message loop is waiting for messages.
- **After __ seconds.** Specify a delay, in seconds.

Success Criteria

The **Success Criteria** determine whether the step is reported as Successful or Failed. Four options are available:

- **Decide based on the program's exit code**. adTempus will look at the exit code returned by the process and evaluate it based on the rule you specify. If the exit code meets the criteria, the step will be reported as Successful; otherwise it will be reported as Failed.

  Many—but not all—programs return an exit code to indicate their status. By convention, an exit code of 0 indicates success; an exit code greater than 0 indicates failure.

  Note that the meaning of a particular exit code is dependent on the program being run. adTempus does not know what if anything an exit code means—it can only detect and respond to the exit code.

  For more information see the Exit Codes topic.

- **Report the step as successful unless instructed otherwise by a Response.** adTempus will ignore the exit code and assume that the task was successful. You can change the status of the step using a Job Control Action.
- **Report the step as Failed unless instructed otherwise by a Response.** adTempus will ignore the exit code and assume that the task failed. You can change the status of the step using a Job Control Action.
- **Use a script.** adTempus will ignore the exit code and will execute the script that you specify. The result of this script will determine whether the step succeeded.

  Before it returns your script must set the global **Result** variable to either **True** (step succeeded) or **False** (step failed). Any other value will cause the step to be reported as Failed.

This feature can be used, for example, when you are running a program that does not produce a meaningful exit code, but produces a file if it succeeds. You could use a script to check for the existence of the file and set the step's status accordingly.

You can also test for any number of specific exit codes and take action based on them using Responses for the step. For example, you may want to report the step as failed if the exit code is greater than 0, but send e-mail notification to an administrator if the exit code is greater than 128.

Program Execution Task Properties

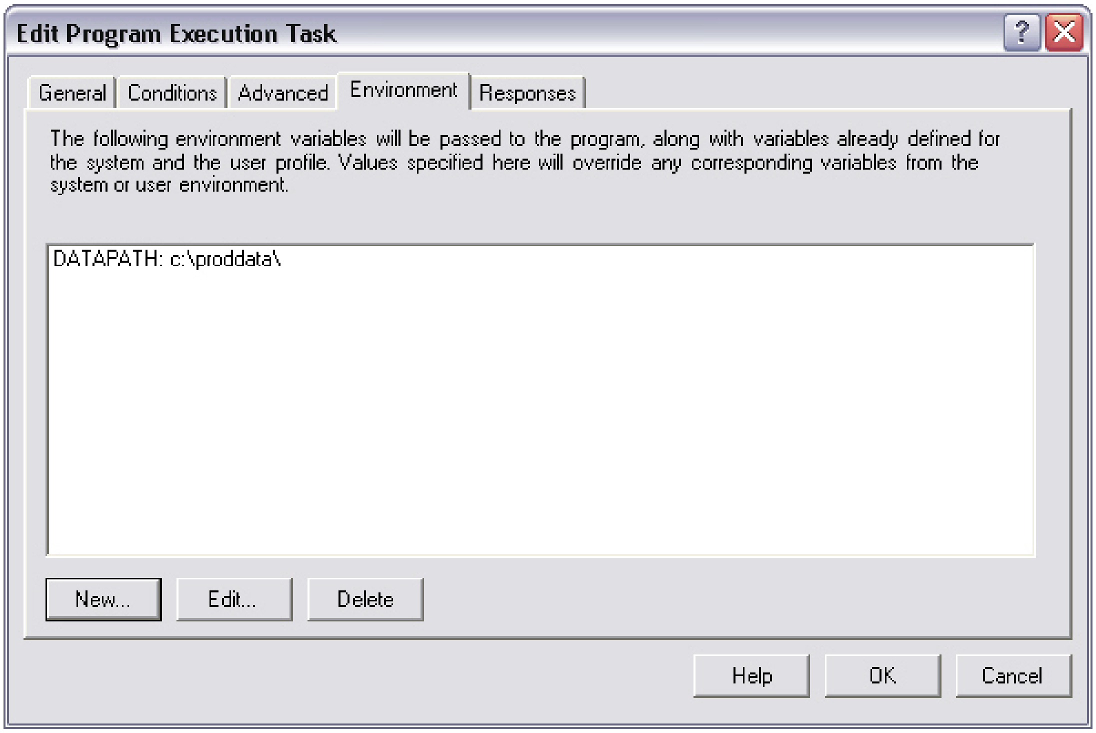
Program Execution Task Overview

Script Overview

Returning Results from Scripts

Environment Page

Show Window Sample



Edit Program Execution Task

The following environment variables will be passed to the program, along with variables already defined for the system and the user profile. Values specified here will override any corresponding variables from the system or user environment.

DATAPATH: c:\proddata\

The **Environment** page allows you to define additional environment variables to be placed in the environment passed to the program.

adTempus automatically passes the environment variables that have been defined for the System and for the user whose account the job is running under; variables you specify on the **Environment** page are in addition to those. If you specify an environment variable that already exists in the System or User environment, the value you specify here will replace the previous value.

Environment variables you set here apply only to the program being run by this task; they do not affect other steps in the job.

Note that you can also use a script to set environment variables. Using this approach you can:

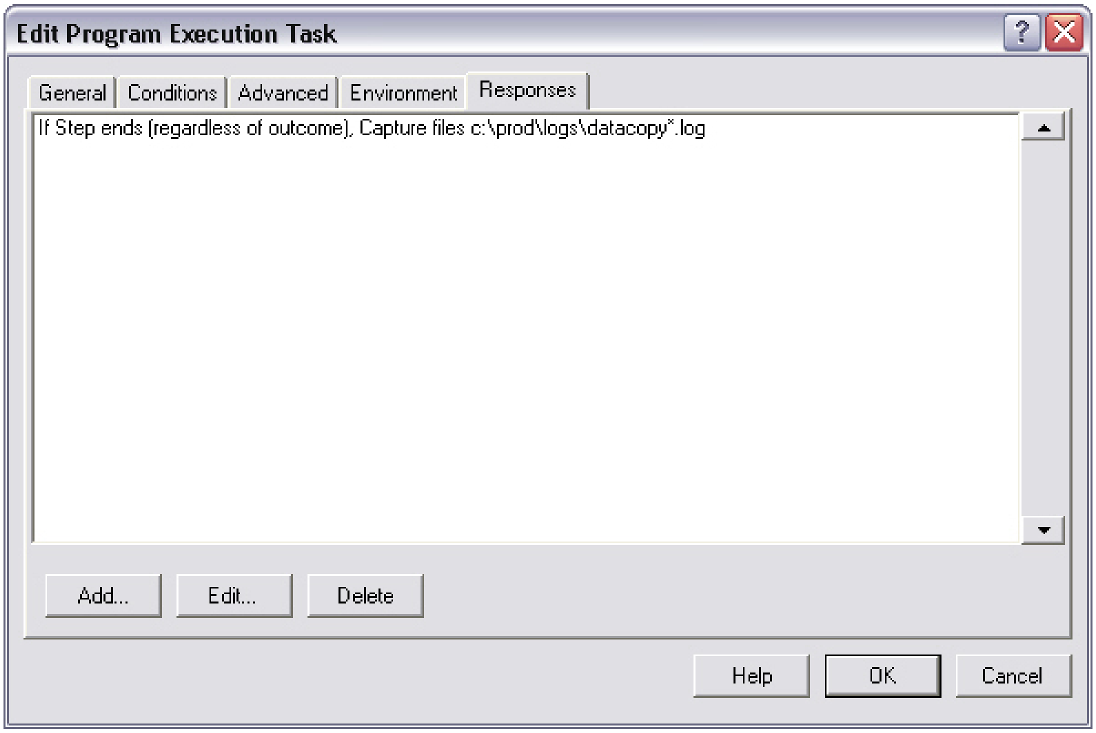
- Set environment values that affect more than one step
- Set environment variables using dynamic values.

Program Execution Task Properties

Program Execution Task Overview

Responses Page

Show Window Sample



If Step ends (regardless of outcome), Capture files c:\prod\logs\datacopy*.log

The **Responses** page defines the actions that adTempus should take in response to events that are fired during execution of the step.

Responses List

The **Responses** list lists the responses that have been specified for the step. You can add, edit, delete, or reorder responses. See the Responses topic for more information on the available responses.

The following events are defined for the Program Execution task:

Event	Description
Beginning of step	Occurs at the beginning of the step, before adTempus has evaluated any conditions or attempted to execute the task.
Condition(s) failed	Occurs if one or more conditions is not met (occurs only once regardless of how many conditions failed).
End of step	Occurs at the end of the step, regardless of the step's outcome.
Execution of task finished	Occurs when the target program finishes executing, regardless of the result.
Restart limit exceeded	Occurs when the step is being restarted by a Job Control action but the restart limit specified on that action has been exceeded.
Step failed	Occurs when the step's status is being set to Failed,

	for whatever reason.
Step is being restarted	Occurs when the step is being restarted by a Job Control action.
Step skipped	Occurs when the step is skipped, either because conditions were not met (if the **Skip...** option is selected on the Conditions page) or because the target process is already running (and the **Skip this step...** option is selected on the General page).
Task completed with an exit code that meets these criteria	Occurs when the target program finishes and returns a result that matches the criteria you specify.
Task could not be executed	Occurs if the target program could not be executed (for example, if the program does not exist, or the user account used for the job does not have the necessary permission).
Task executed but returned failure result	Occurs when the target program finishes and, based on the **Success Criteria** on the Advanced page, is determined to have failed.
Task returned successful result	Occurs when the target program finishes and, based on the **Success Criteria** on the Advanced page, is determined to have succeeded.
Task started	Occurs once the target program has been successfully started. Does not occur until after adTempus has waited as directed in the **Startup Determination** options on the Advanced page.
Task will be terminated because it exceeded its allowed execution time	Occurs once the time limit specified on the Advanced page has elapsed, but before adTempus attempts to terminate the process. Respond to this event if you want to use your own script or other approach to end the process.
Task was terminated because it exceeded its allowed execution time	Occurs once the time limit specified on the Advanced page has elapsed, after adTempus has terminated the process.

Program Execution Task Properties

Program Execution Task Overview

Responses Overview

Responses

Responses

Responses allow you to customize the flow of execution through an adTempus job, or to link jobs together.

While it is executing a job, adTempus fires "events" at certain key points. For example, an event is fired at the beginning of the job; another is fired at the end of the job. If the job completes successfully a "success" event is fired. If the job fails, a "failure" event is fired.

If you want adTempus to take action based on one of these events, you associate a Response with that event. The Response defines the actions that adTempus should take in response to the event. See an example.

Each response can be triggered by any number of events, and can execute any number of actions.

The events that are available vary based on whether you are creating responses for a job or a step, and based on the kind of task a step executes. For a complete list of events for a job or task, see the help topic for the Responses property page for the job or task.

Notes on using Responses

Responses are evaluated in the order that they appear in the Responses list for the job or step. Within a response, actions are executed in the order that they appear in the Actions list.

Actions are executes synchronously. That is, each action must finish before the next action starts. Each Response's actions must complete before the next Response is evaluated. And all responses must be evaluated before job execution proceeds to the next step. This is important to keep in mind when you are using Job Control actions or Script actions; see the topics for those actions for more information.

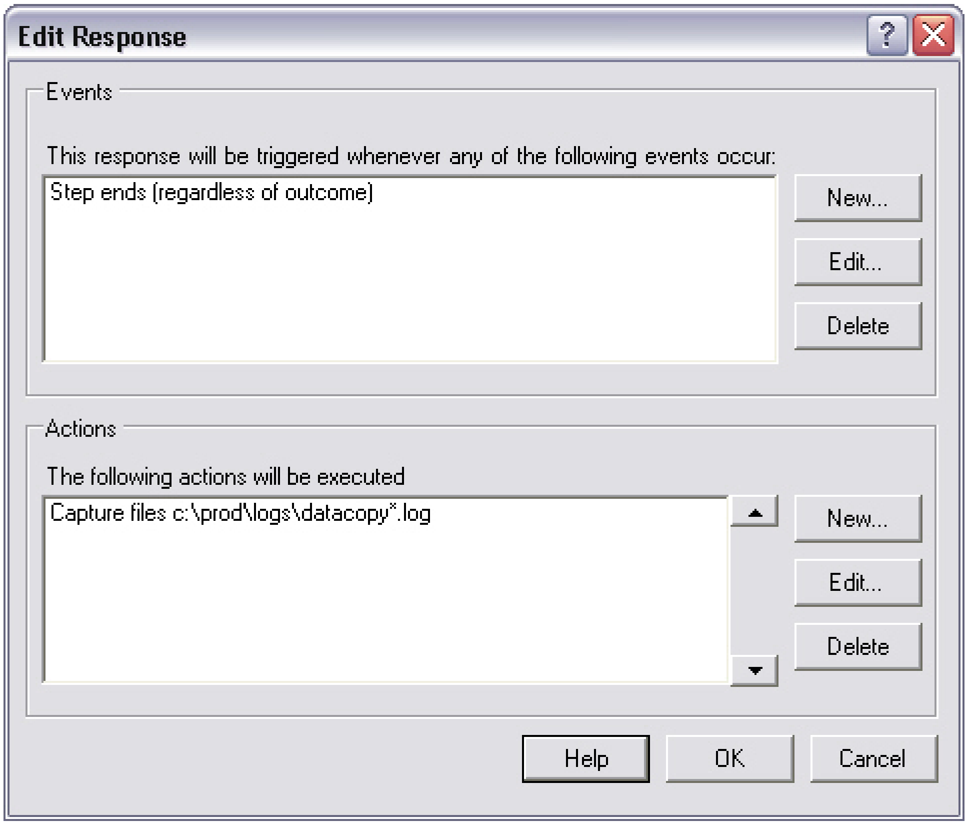
Response example

Response Properties

Actions

Response Properties

Show Window Sample


Edit Response
Events
This response will be triggered whenever any of the following events occur:
Step ends (regardless of outcome)
New...
Edit...
Delete
Actions
The following actions will be executed
Capture files c:\prod\logs\datacopy*.log
New...
Edit...
Delete
Help
OK
Cancel

The Response ties events to actions. A Response is triggered when any of the events in its Events list occurs; when a Response is triggered all of its Actions are executed in the order that they are listed.

Events

Add, modify, or remove events that trigger this response.

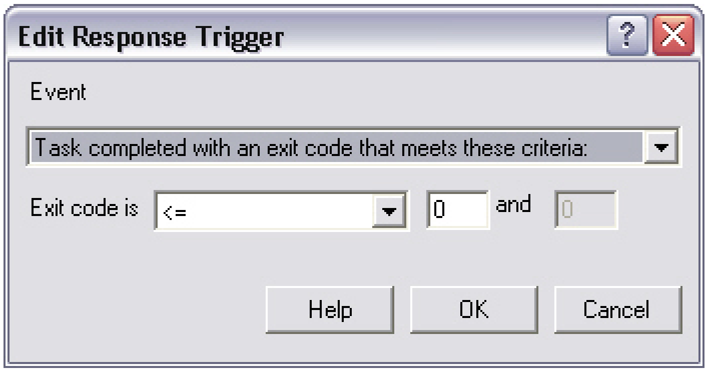
Actions

Add, modify, remove, or reorder actions that will be executed when this response is triggered.

Response Overview

Response Trigger

Show Window Sample



Task completed with an exit code that meets these criteria:

Exit code is <= 0 and 0

A Response Trigger associates events with actions to define a response. In the Response Trigger window, select the event that you want to respond to.

The events that are available will depend on the job or kind of task you are creating the response for. For a list and description of the events that are valid for a job or task, see the help topic for the Responses page of the job or task type.

Responses Overview

Schedule Trigger

Schedule Trigger

A Schedule Trigger is a trigger that causes a job to be executed at specific dates and times, or at specific intervals (e.g., every 5 minutes).

When you create a Schedule Trigger, you specify one or more Schedules, which specify when the job should run. Options allow you to specify how adTempus should react to changes of the system clock. You may also specify what should happen if the job is scheduled to run on a holiday.

Schedule Trigger Properties

The Schedule Trigger Properties window is divided into three pages. Follow the links below for information on each page:

Schedules Page

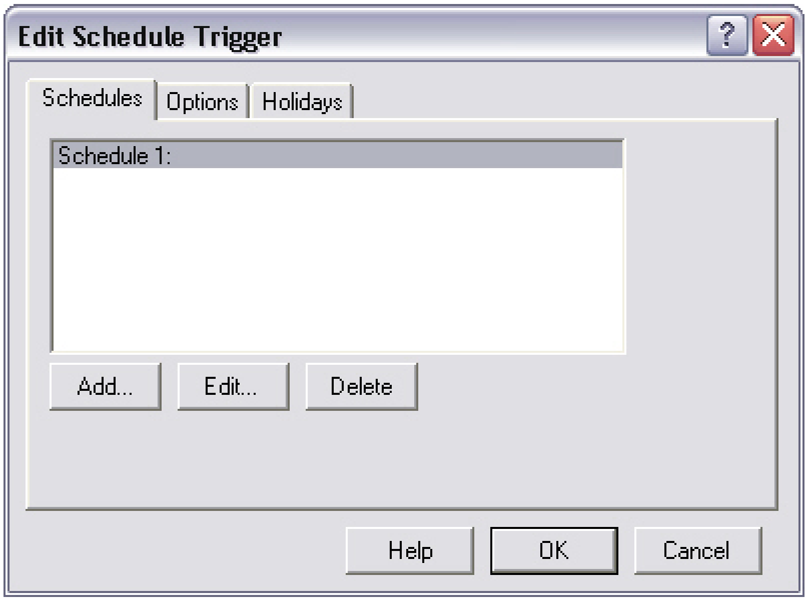
Options Page

Holidays Page

Triggers Overview

Schedules Page

Show Window Sample



On the **Schedules** page you specify the Schedules that determine when the job should be run. You may add, edit, or remove schedules.

If you have specified more than one schedule, each schedule will trigger the job independently. For example, if one schedule includes all Mondays and another includes all Tuesdays, the job will be triggered on Mondays and Tuesdays.

If the schedules have overlapping times, this will *not* cause the job to be triggered twice at the same time. For example, if Schedule 1 and Schedule 2 both call for the job to be run at 12:30 PM on Monday, the job will only be run once at 12:30.

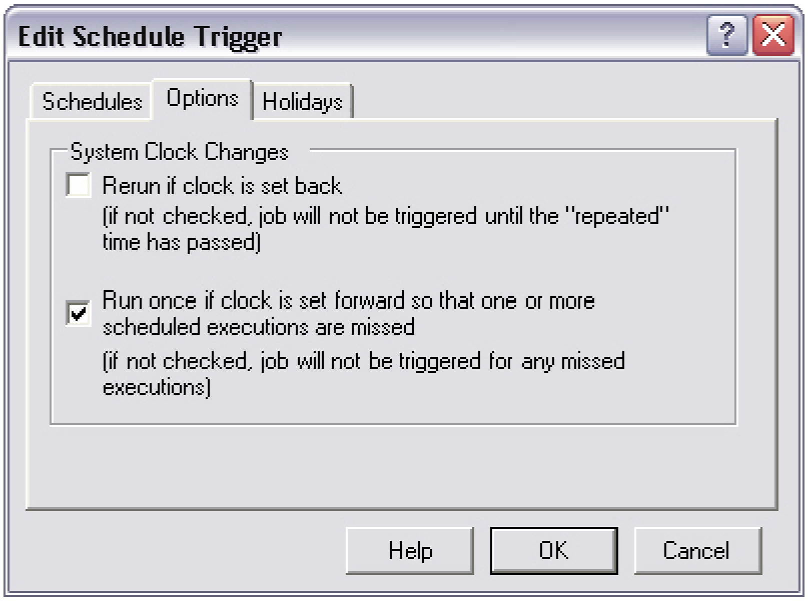
Schedule Trigger Overview

Schedules

Triggers Overview

Options Page

Show Window Sample



Edit Schedule Trigger

System Clock Changes

□ Rerun if clock is set back
(if not checked, job will not be triggered until the "repeated" time has passed)

☑ Run once if clock is set forward so that one or more scheduled executions are missed
(if not checked, job will not be triggered for any missed executions)

On the **Options** page you specify how adTempus should treat this job if a change of the system clock affects a scheduled run time.

Rerun if clock is set back

If this option is checked, the job will continue to run at its regular schedule even if the clock is set back so that executions are repeated. If the option is not checked, any "repeated" times will be ignored.

For example, your job is scheduled to run at 1:30 every morning. At 1:32 AM on 21 August 2002, just after the job has run, the system clock is set back to 1:29 AM when the clock is synchronized with a master clock. If the **Rerun...** option is checked, the job will be run when the clock reaches 1:30 again. If the option is not checked, the job will not be run again until 1:30 the next day.

Guidance: If your job is only meant to run once a day, you will want to leave this option unchecked (the default setting). If, however, your job is meant to run at regular intervals (e.g., to transfer data every 5 minutes), you will probably want to check the option so that the job continues to execute regularly even when the clock is set back by an hour for daylight saving time.

Run once if clock is set forward so that one or more scheduled executions are missed

If this option is checked, adTempus will run the job once if a time change causes a regular execution to be missed. If the option is not checked, executions may be missed as a result of the time change.

For example, your job is scheduled to run at 2:30 every morning. At 2:00 one morning the system clock is set forward to 3:00 AM due to daylight saving time. If the **Run once...** option is checked, adTempus will run the job when the time change is detected to make up for the missed time. If the option is not checked, the execution will be missed.

adTempus will only execute the job once, regardless of the number of executions that are missed. For example, even if your job runs every minute and the clock has been set forward by a full hour, adTempus will run the job once, not 60 times.

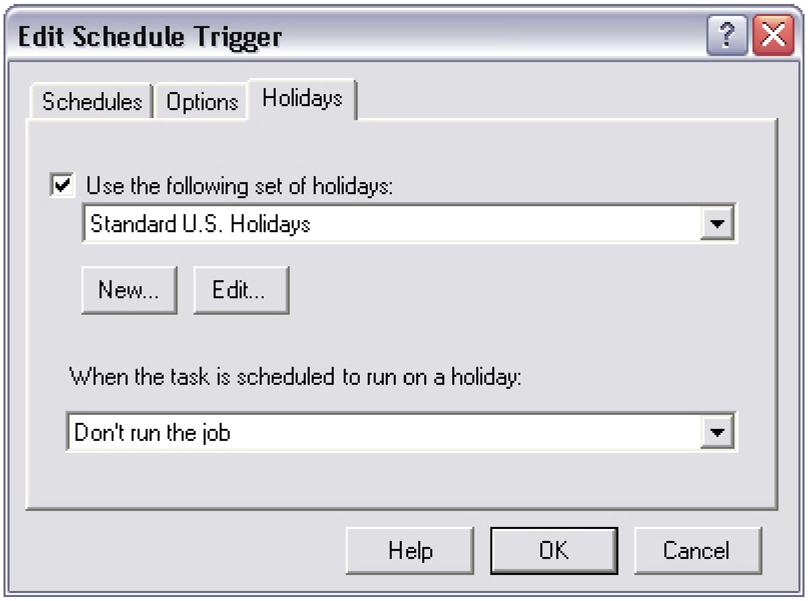
Guidance: If your job only runs once a day (or less frequently) you will want to check this option (the default setting) to ensure that the job gets run. If, however, the job runs frequently (e.g., once an hour) you may want to uncheck this option, as missing one execution is not likely to be significant.

Schedule Trigger Overview

Triggers Overview

Holidays Page

Show Window Sample


Edit Schedule Trigger

☑ Use the following set of holidays: Standard U.S. Holidays

When the task is scheduled to run on a holiday: Don't run the job

On the **Holidays** page you specify how adTempus should treat this job if it is scheduled to run on a holiday.

Use the following set of holidays

If you want the job to behave differently on holidays, check this option and select the list of holidays that applies. You may optionally create or edit a holiday set if none of the existing sets meets your needs.

If you do not want the job to behave differently on holidays, uncheck this option.

When the job is scheduled to run on a holiday

Specify how adTempus should behave if a scheduled execution for the job falls on a holiday. The following options are available:

- **Don't run the job.** The job is not run on any day that is defined as a holiday.
- **Run the job on the nearest business day.** The job is run on the nearest day that is not a weekend or holiday. This could be either before or after the originally scheduled date.
- **Run the job on the previous business day.** The job is run on the nearest prior day that is not a weekend or holiday.
- **Run the job on the next business day.** The job is run on the nearest subsequent day that is not a weekend or holiday.
- **Run the job on the nearest non-holiday day.** The job is run on the nearest day that is not a holiday. This could be either before or after the originally scheduled date and could be a weekend.
- **Run the job on the previous business day.** The job is run on the nearest prior day that is not a holiday (but may be a weekend).
- **Run the job on the next business day.** The job is run on the nearest subsequent day that is not a holiday (but may be a weekend)

Schedule Trigger Overview

Defining Holiday Sets

Schedule

Schedule

A Schedule defines the dates and times at which a job will run.

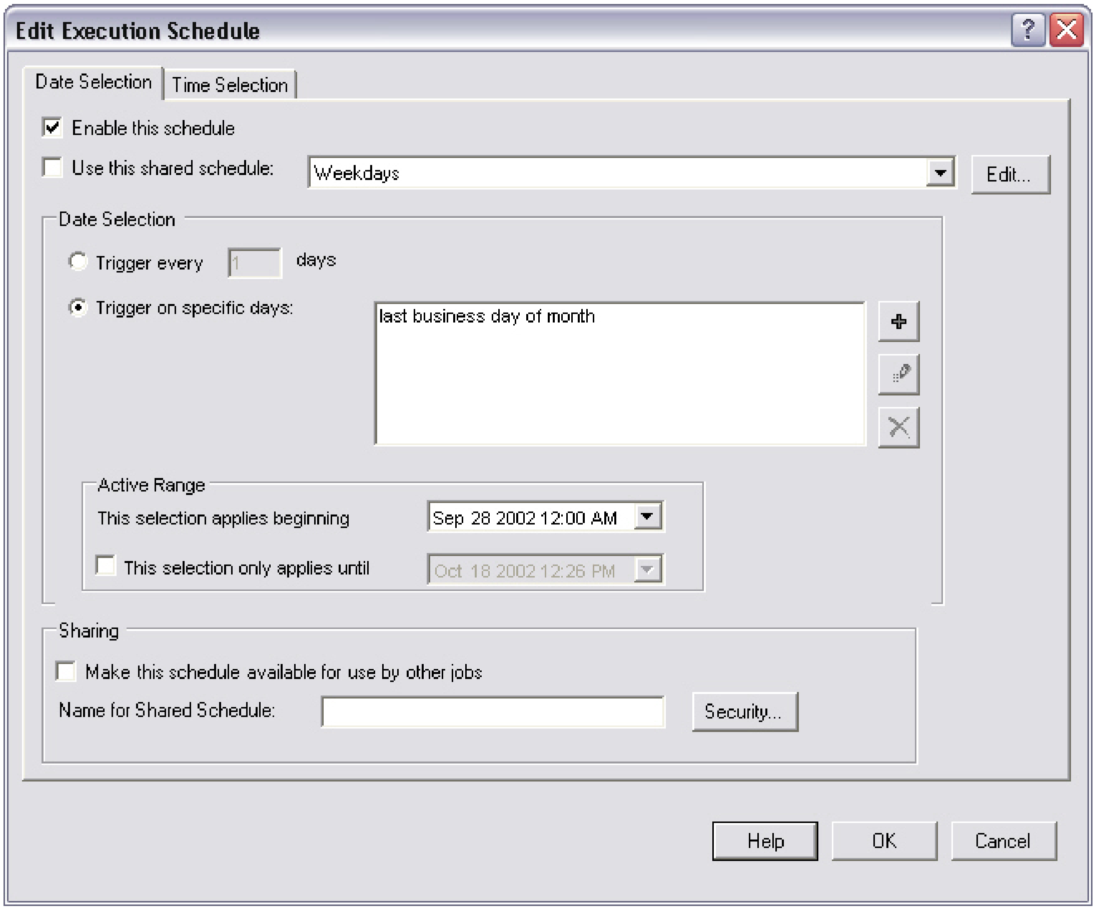
Schedules are attached to Jobs using Schedule Triggers. A job may have any number of schedules.

Schedule Properties

Schedule Trigger

Date Selection Page

Show Window Sample



☑ Enable this schedule

□ Use this shared schedule: Weekdays

Date Selection

○ Trigger every 1 days

● Trigger on specific days: last business day of month

Active Range

This selection applies beginning Sep 28 2002 12:00 AM

□ This selection only applies until Oct 18 2002 12:26 PM

Sharing

□ Make this schedule available for use by other jobs

Name for Shared Schedule:

The **Date Selection** page defines the days on which the job will run. You may either use an existing Shared Schedule or specify dates for the job.

Enable this schedule

Check this option to make the schedule "active." If the Schedule is not enabled, it will be ignored by adTempus (jobs will not be triggered on the specified days).

Use this shared schedule

Check this option to use an existing Shared Schedule to determine the days on which the job will run. Select the shared schedule that you wish to use.

You still must specify the times the job will run, using the Time Selection page.

Date Selection

The date selection options are not available if you have chosen to use a shared schedule. There are two options for specifying the days that the job will run:

- **Trigger every __ days.** adTempus will trigger the job at the fixed interval you specify. The job will be triggered on the starting date you specify in the **Active Range** and every *n* days afterward (where *n* is the interval you have specified).

- **Trigger on specific days.** When you select this option you have complete flexibility to define as many specific dates or rules as you wish. Use the list to add, edit, or remove Day Selections that specify the days on which the job should run.

Active Range

Specify the earliest date and time to which this schedule applies. Optionally, specify an ending date and time.

If you specify an ending date and time, the schedule will not be used (will not trigger jobs) after that date/time.

Sharing

The Sharing options are not available if you have chosen to use a shared schedule.

Once you have created a schedule for your job, you can share it to make it available for use in other jobs.

Make this schedule available for use by other jobs

Check this option to share the schedule.

Once you have checked this option and saved your changes by clicking **OK**, your schedule becomes a Shared Schedule. When you return to this window you will find that the **Use this shared schedule** option is now checked and your schedule is selected in the shared schedule list. To modify the schedule you must edit it as a shared schedule.

Once a schedule has been shared, it cannot be "unshared" without deleting it.

Name for shared schedule

Provide a descriptive name for the schedule. The name is required, and must be unique across all existing shared schedules.

Security

Click this button to display the security properties for the schedule. This allows you to specify who will be able to use the shared schedule.

Day Selection Properties

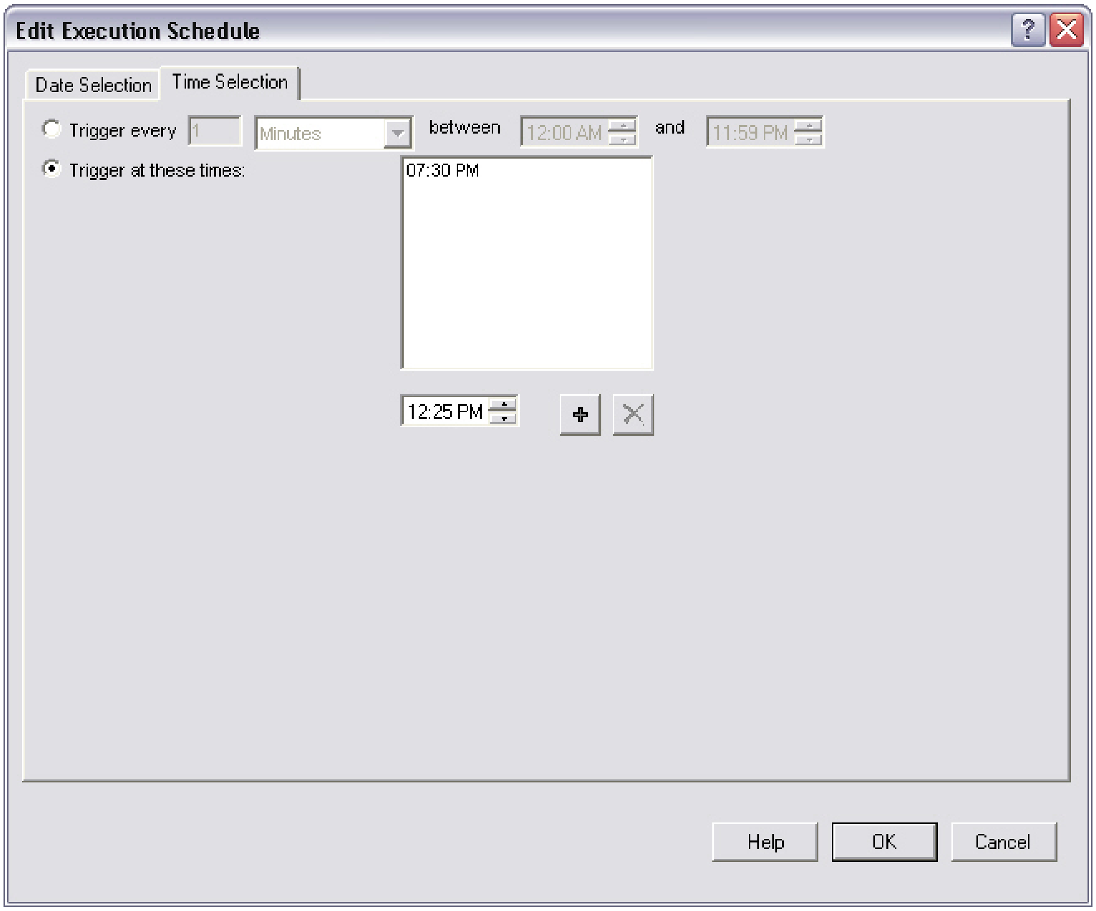
Shared Schedules

Schedule Properties

Schedule Overview

Time Selection Page

Show Window Sample



Edit Execution Schedule

○ Trigger every 1 Minutes between 12:00 AM and 11:59 PM

● Trigger at these times: 07:30 PM

12:25 PM

The **Time Selection** page defines the times at which the job will run.

Trigger every...

Select this option to execute the job at a fixed interval (e.g., every 5 minutes or every 2 hours). The first execution will occur at the starting time you specify.

Trigger at these times

Select this option to specify exact times at which the job should execute. You can list any number of times.

Schedule Properties

Schedule Overview

Script

Script

A **Script** allows you to store and execute a Windows Script Host script (such as a script written in VBScript or JScript) within adTempus.

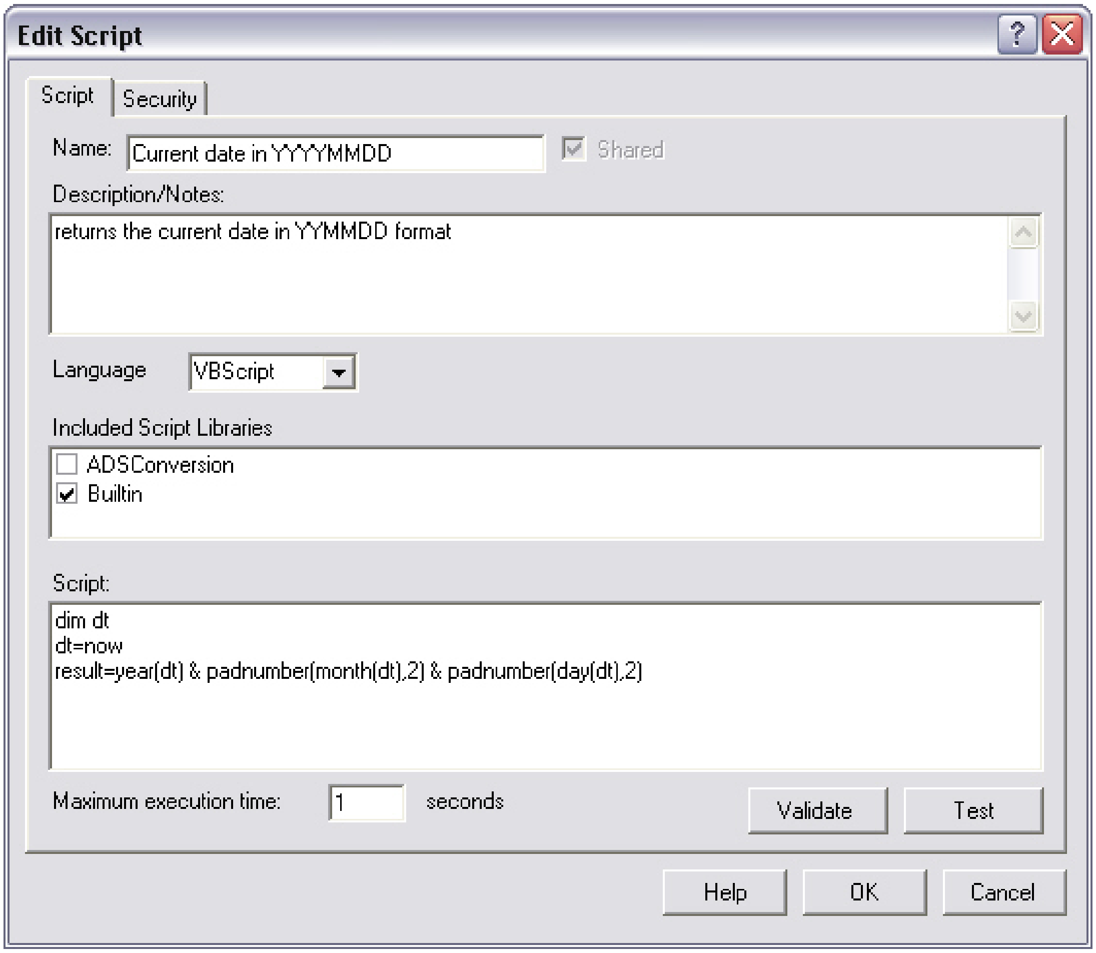
Scripts can make use of common code stored in Script Libraries, and can use globally-defined variables and procedures.

Script Properties

Using Scripts in adTempus

Script Page

Show Window Sample



Name: Current date in YYYYMMDD ☑ Shared

Description/Notes: returns the current date in YYMMDD format

Language: VBScript

Included Script Libraries:
☐ ADSConversion
☑ Builtin

Script:

```
dim dt
dt=now
result=year(dt) & padnumber(month(dt),2) & padnumber(day(dt),2)
```

Maximum execution time: 1 seconds

The Script page defines the script to be executed.

Name

Provide a descriptive name for this script. If the script is **Shared**, a name is required. Otherwise the name is optional. The name must be unique across all scripts.

Shared

Check this option to allow the script to be shared among jobs, responses, etc. If the script is not **Shared** it will be available only for the object with which it is associated.

When a script is shared the Security page will be visible, allowing you to specify permissions for the script.

Once you have checked the **Shared** option and clicked **OK**, the script cannot be "unshared." If you no longer want it to be shared you must delete it and recreate it.

Description / Notes

Enter any extended descriptive information or notes for this script. There is no limit on the length of the text.

Language

Select or enter the language that the script uses. The language must be a scripting language recognized by the Window Scripting Host on the server. Note that the list only lists VBScript and JScript as predefined values, even if other scripting languages are installed. You must type in the name of the script language if you are using a language other than one of these.

Included Script Libraries

If your script uses procedures from any Script Libraries, select them here to make them available to the script.

Script

Provide the body of the script.

Maximum Execution Time

Specify the maximum time (in seconds) that the script should be allowed to run. (This feature prevents against "runaway" scripts.)

If you do not want to set a limit, specify a value of 0. This will cause the scripting host to monitor execution of the script and abort it if the script host detects a problem.

Validate

Click the **Validate** button to validate your script. adTempus will check to ensure that the code appears valid. The script is automatically validated when you click the **OK** button.

Test

Click **Test** to validate and execute your script. If the script returns a value, that value will be displayed.

Validation and testing of the script occur on the computer that the Console is running on. If you are connected to the adTempus server from a remote computer and your script refers to external objects, files, etc., you should test the script from a Console running on the computer that hosts the server to be sure that it will work correctly on the server.

Script Properties

Script Overview

Security Page

The **Security** page defines the security settings for the script. **Note:** This page is not available unless the script is shared.

For information on specifying security settings see the Security Editor and Security Overview topics.

The following permissions apply to scripts:

Permission	Description
Full Control	Permission to perform all actions on the script.
List/Use	Permission to use the script.
View	Permission to view the properties of the script.
Modify	Permission to modify the properties of the script.
Delete	Permission to delete the script.
Administer security	Permission to change the security settings for the script.
Change owner	Permission to take ownership of the script.

Permission to create new recipients is controlled through the Default Script Security options.

ⓘ Members of the computer's Administrators group always have View and Change Owner permission, even if those permissions are not listed, or are specifically denied. An Administrator can always take ownership of a script.

The job's Owner always has Full Control permission for the notification recipient, even if that permission is not listed, or is specifically denied

Specifying default script security

Script Properties

Security Editor

Script Action

Script Action

The Script action is used to execute a script. This gives you the flexibility to respond to events in ways not provided for by the built-in actions offered by adTempus.

Script actions also are commonly used to:

- Set environment variables for the job, when the "static" environment variable list offered by the Program Execution task is not sufficient.
- Send keystrokes to an application.

While an action is being executed, adTempus cannot perform any other "work" on the job with which the action is associated. For example:

- If more than one action has been specified in a response, each action cannot be run until the previous action has completed.
- adTempus cannot start the next step of a job until it finishes the current step. This includes running all actions associated with the step.

Therefore when you use a script action your script should be something that runs quickly, so adTempus can be about its business. If your script goes and launches some external application using the application's automation interface, and then spends 10 minutes doing something with that application, you adTempus job is going to be patiently waiting.

Using Scripts

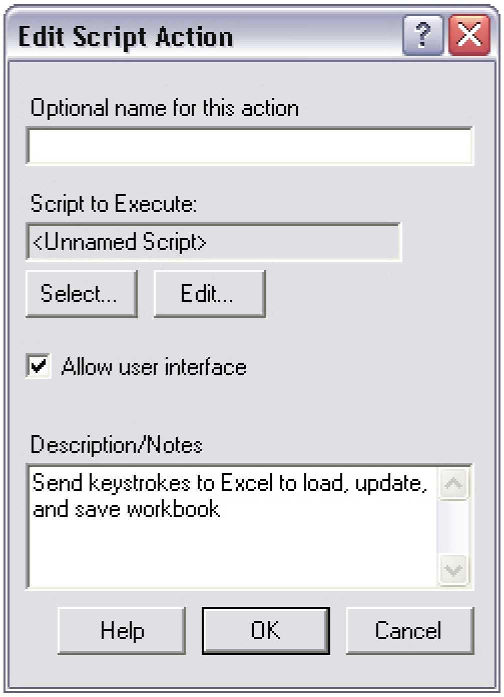
Script Action Properties

Responses Overview

Actions

Script Action Properties

Show Window Sample



<Unnamed Script>

☑ Allow user interface

Description/Notes

Send keystrokes to Excel to load, update, and save workbook

The Script Action properties define the script that should be run.

Optional name for this action

Optionally, specify a descriptive name for the action. Otherwise the name of the target script is used as the description for the action.

Description / Notes

Enter any extended descriptive information or notes for this action. There is no limit on the length of the text.

Script to Execute

Select or create the script you want to execute.

Allow user interface

Check this option if your script needs to show a user interface. If the option is not checked, the scripting host will not permit a user interface to be displayed.

Script Action Overview

Using Scripts

Script Condition

Script Condition

A Script Condition is a condition that executes a script. The result of the script determines whether the condition is satisfied.

Use a Script Condition to provide your own condition checks when none of the condition types provided with adTempus meets your needs.

For example, you want a job to execute only if a specified file contains certain text. The adTempus File Condition can detect the presence of the file but does not read its contents. Using a Script Condition you can create a script (using a scripting language like VBScript) that looks for the file and then checks it for the required text.

Script Condition Properties

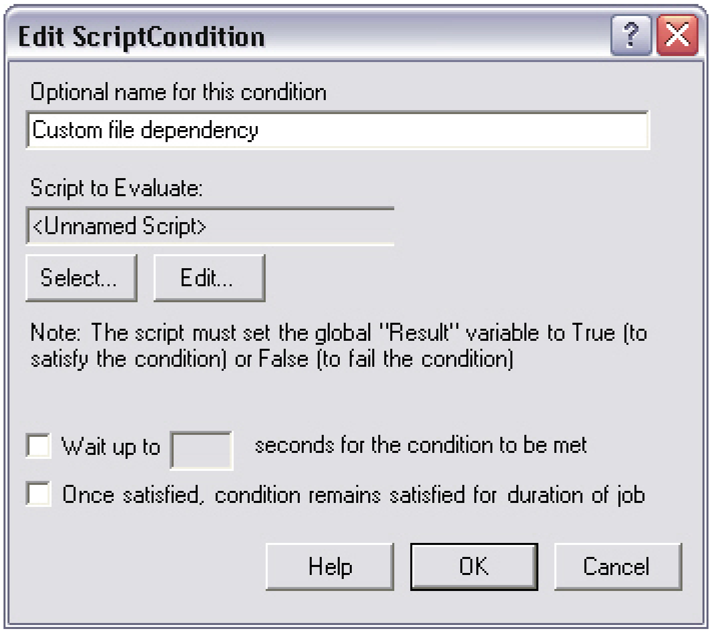
Conditions Overview

Script Overview

Returning Results from Scripts

Script Condition Properties

Show Window Sample



Edit ScriptCondition

Optional name for this condition: Custom file dependency

Script to Evaluate: <Unnamed Script>

Note: The script must set the global "Result" variable to True (to satisfy the condition) or False (to fail the condition)

□ Wait up to [ ] seconds for the condition to be met

□ Once satisfied, condition remains satisfied for duration of job

The options on the Script Condition Properties page define the behavior of the script condition.

Optional name for this condition

You may optionally provide a descriptive name for this condition to make it easier to identify in the Conditions list. Otherwise the name of the script is used.

Script to evaluate

Select or create the script that adTempus should evaluate. See the Script Overview topic for information on working with scripts.

Before it returns your script must set the global **Result** variable to either **True** (condition is satisfied) or **False** (condition is not satisfied). Any other value will cause the condition to fail.

Wait up to __ seconds for the condition to be met

Check this option and specify a wait limit if you want adTempus to wait for the condition to be met. See Condition Polling for more information.

Once satisfied, condition remains satisfied for duration of job

Check this option to indicate that adTempus should stop checking this condition once it has been satisfied. See Condition Polling for more information.

Script Condition Overview

Script Overview

Returning Results from Scripts

Script Execution Task

Script Execution Task

The Script Execution task allows you to run Windows Script Host scripts. The script may be a script that is stored in adTempus or it may be an external script file.

Although external script files can just as easily be executed using a Program Execution task, executing them using a Script Execution task allows your script to return a result code to adTempus, used to indicate the success or failure of the script.

Script Execution Task Properties

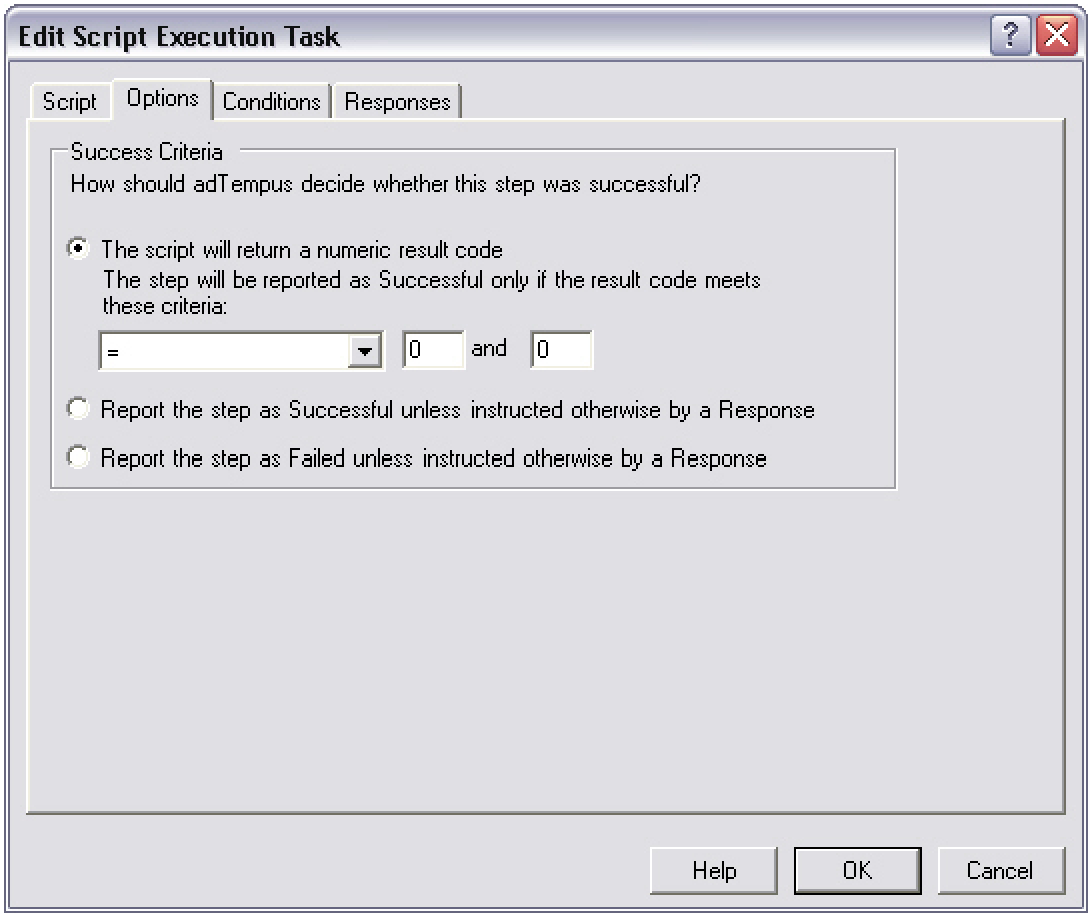
Scripts Overview

Tasks Overview

Steps Overview

Options Page

Show Window Sample



Success Criteria

How should adTempus decide whether this step was successful?

● The script will return a numeric result code

The step will be reported as Successful only if the result code meets these criteria:

= | 0 | and | 0

○ Report the step as Successful unless instructed otherwise by a Response

○ Report the step as Failed unless instructed otherwise by a Response

The **Options** page defines additional options for the task.

Success Criteria

The **Success Criteria** determine whether the step is reported as Successful or Failed. Three options are available:

- **The script will return a numeric result code**. Before it returns your script must set the global Result variable to either True (step succeeded) or False (step failed). Any other value will cause the step to be reported as Failed.

- **Report the step as successful unless instructed otherwise by a Response.** adTempus will ignore the exit code and assume that the task was successful. You can change the status of the step using a Job Control Action.
- **Report the step as Failed unless instructed otherwise by a Response.** adTempus will ignore the exit code and assume that the task failed. You can change the status of the step using a Job Control Action.

You can also test for any number of specific exit codes or string results and take action based on them using Responses for the step. For example, you may want to report the step as failed if the exit code is greater than 0, but send e-mail notification to an administrator if the exit code is greater than 128.

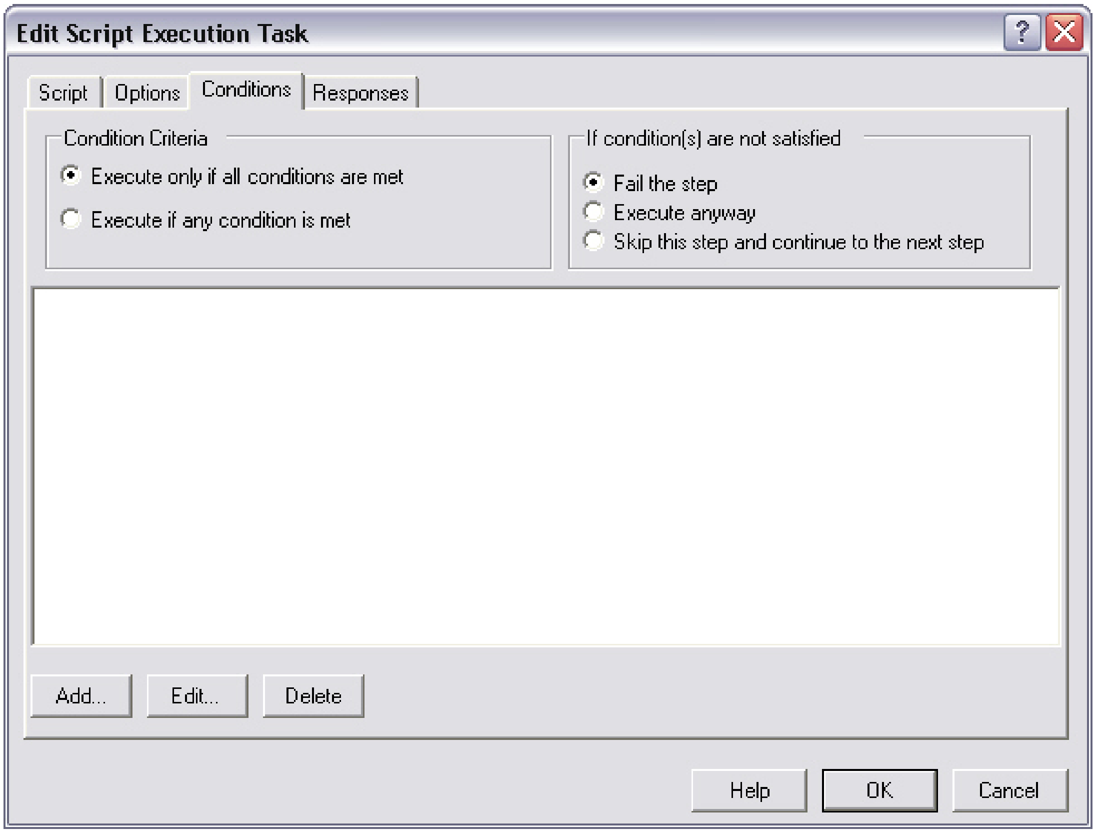
Script Execution Task Properties

Script Execution Task Overview

Returning Results from Scripts

Conditions Page

Show Window Sample



Edit Script Execution Task

Condition Criteria

● Execute only if all conditions are met

○ Execute if any condition is met

If condition(s) are not satisfied

● Fail the step

○ Execute anyway

○ Skip this step and continue to the next step

The **Conditions** page defines conditions that must be satisfied before the step is run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- **Execute only if all conditions are met.** The step is only executed if all of the listed conditions are met.
- **Execute if any condition is met.** The step is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- **Fail the step.** The step is not executed; the status is set to Failed.
- **Execute anyway**. The step is executed anyway.
- **Skip the step (do not report as a failure)**. The step is not run, but it is not treated as a failure. The status of the step is reported as "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the step. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

Script Execution Task Properties

Script Execution Task Overview

Conditions Overview

Script Library

Script Library

Script Libraries allow you to create "libraries" of commonly used code or data that can be shared among scripts in adTempus. For example, suppose that you have several jobs where you need to pass the current date as one of the command-line parameters to the scheduled program (but the other parameters are different for each job, so you cannot use a single script for all of the jobs).

Rather than repeating the necessary formatting code in the command-line script for each task, you can create a function in a Script Library, and then reference the library from the scripts. This in turn allows you to call the functions and subroutines defined in that library.

A Script Library can contain any code or data declarations that are valid for the scripting language you specify for the library. A script can only use libraries that are in the same scripting language.

For examples, refer to the predefined libraries provided with adTempus.

Because the Windows Script Host does not support multiple namespaces, all of the code for a script and for all of the libraries it references end up in the same namespace. You therefore must take care that you do not have variables or procedures in different libraries that have the same name, if those libraries will be used by the same script.

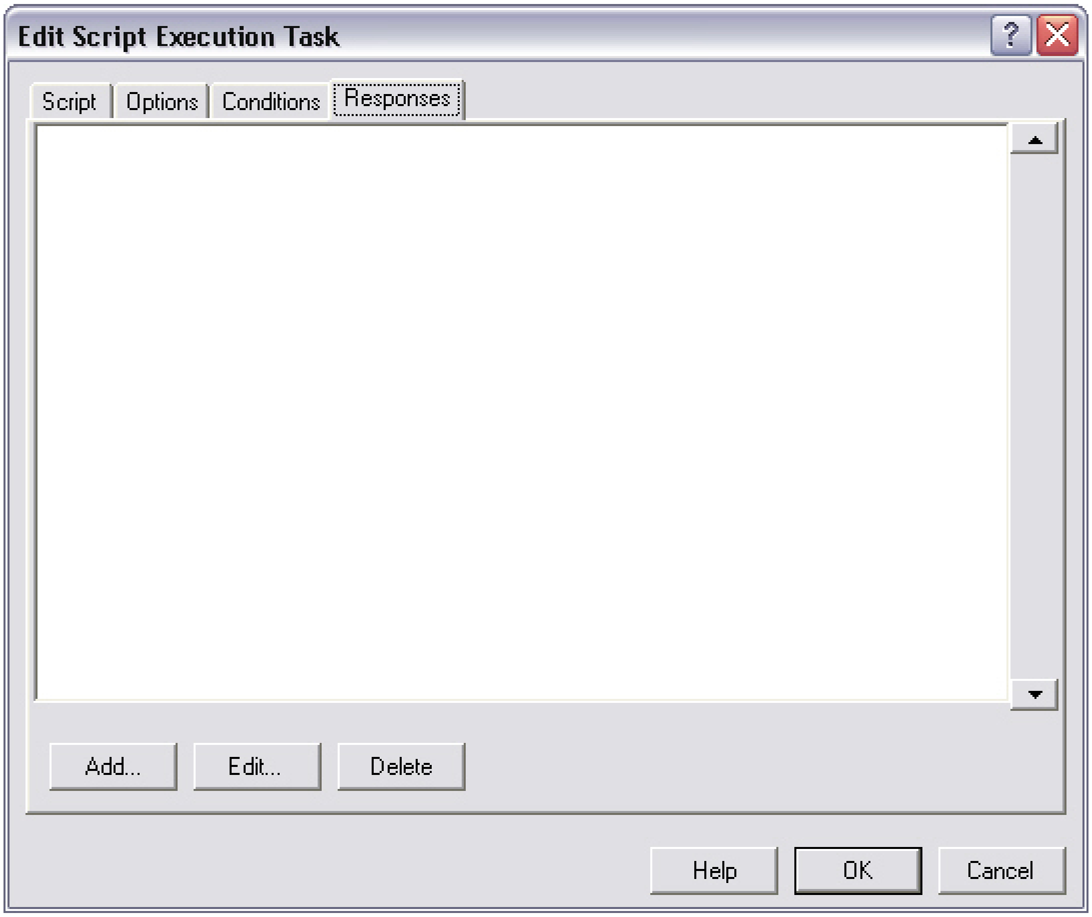
Script Library Properties

Predefined libraries

Overview of using scripts in adTempus

Responses Page

Show Window Sample



The **Responses** page defines the actions that adTempus should take in response to events that are fired during execution of the step.

Responses List

The **Responses** list lists the responses that have been specified for the step. You can add, edit, delete, or reorder responses. See the Responses topic for more information on the available responses.

The following events are defined for the Script Execution task:

Event	Description
Beginning of step	Occurs at the beginning of the step, before adTempus has evaluated any conditions or attempted to execute the task.
Condition(s) failed	Occurs if one or more conditions is not met (occurs

	only once regardless of how many conditions failed).
End of step	Occurs at the end of the step, regardless of the step's outcome.
Execution of task finished	Occurs when the target script finishes executing, regardless of the result.
Restart limit exceeded	Occurs when the step is being restarted by a Job Control action but the restart limit specified on that action has been exceeded.
Step failed	Occurs when the step's status is being set to Failed, for whatever reason.
Step is being restarted	Occurs when the step is being restarted by a Job Control action.
Step skipped	Occurs when the step is skipped because conditions were not met (if the **Skip...** option is selected on the Conditions page).
Task completed with an exit code that meets these criteria	Occurs when the target script finishes and returns a result that matches the criteria you specify.
Task could not be executed	Occurs if the target script could not be executed (for example, if the script does not exist, or the user account used for the job does not have the necessary permission).
Task executed but returned failure result	Occurs when the target script finishes and, based on the **Success Criteria** on the Options page, is determined to have failed.
Task returned successful result	Occurs when the target script finishes and, based on the **Success Criteria** on the Options page, is determined to have succeeded.
Task started	Occurs once the target script has been successfully started.
Task will be terminated because it exceeded its allowed execution time	Occurs once the time limit specified on the Script page has elapsed, but before adTempus attempts to terminate the process. Respond to this event if you want to use your own script or other approach to end the script.
Task was terminated because it exceeded its allowed execution time	Occurs once the time limit specified on the Script page has elapsed, after adTempus has terminated the process.
Script returns a result that meets these criteria	Occurs when the script returns a result that matches (exactly) the string you specify.

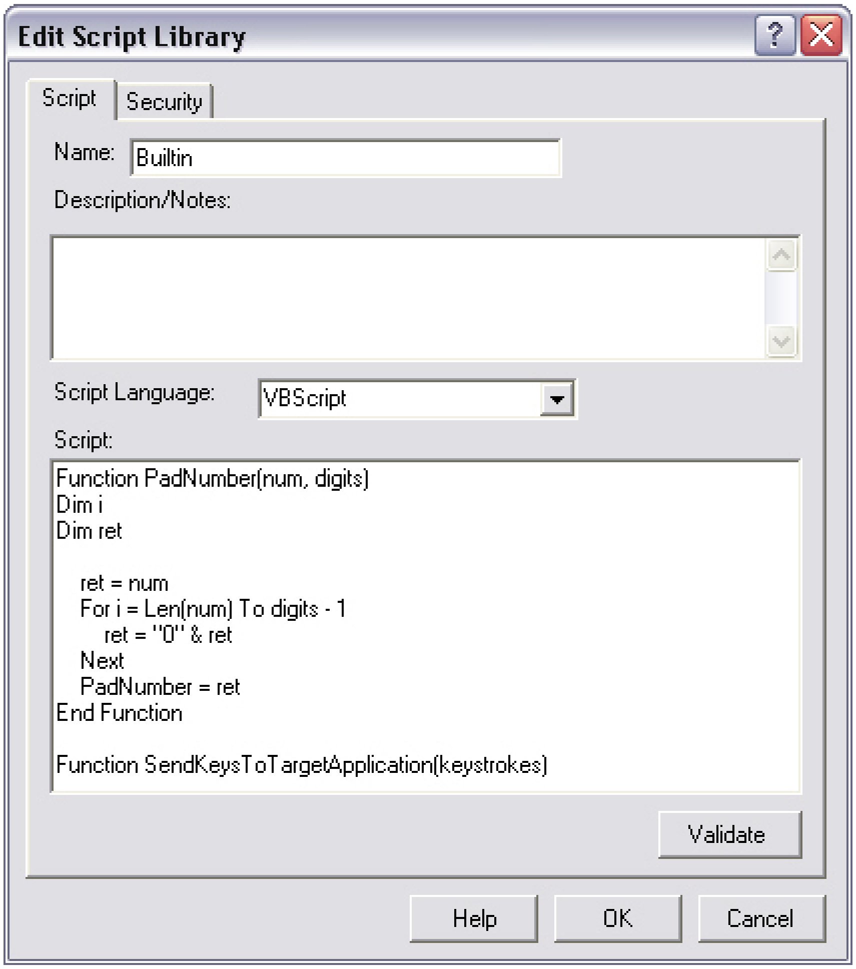
Script Execution Task Properties

Script Execution Task Overview

Responses Overview

Script Page

Show Window Sample



Name: Builtin

Script Language: VBScript

```
Function PadNumber(num, digits)
Dim i
Dim ret

    ret = num
    For i = Len(num) To digits - 1
        ret = "0" & ret
    Next
    PadNumber = ret
End Function

Function SendKeysToTargetApplication(keystrokes)
```

The Script page defines the script code for the library.

Name

Provide a descriptive name for this library. The name must be unique across all scripts, and must contain only letters, numbers, and the underscore ("_") character. No other punctuation or spaces are allowed in the name.

Description / Notes

Enter any extended descriptive information or notes for this library. There is no limit on the length of the text.

Language

Select or enter the language that the script uses. The language must be a scripting language recognized by the Window Scripting Host on the computer running the adTempus service.

Script

Provide any functions, subroutines, constants, etc., that are part of this library. The code you specify here will be available to any scripts that use this library.

Validate

Click the **Validate** button to validate your script library. adTempus will check to ensure that the code appears valid. The script library is automatically validated when you click the **OK** button.

Validation of the script library occurs on the computer that the Console is running on. If you are connected to the adTempus server from a remote computer and your script refers to external objects, files, etc., you should test the script from a Console running on the computer that hosts the server to be sure that it will work correctly on the server.

Script Library Properties

Script Library Overview

Script Overview

Security Page

The **Security** page defines the security settings for the script library.

For information on specifying security settings see the Security Editor and Security Overview topics.

The following permissions apply to script libraries:

Permission	Description
Full Control	Permission to perform all actions on the script library.
List/Use	Permission to use the script library.
View	Permission to view the properties of the script library.

Modify	Permission to modify the properties of the script library.
Delete	Permission to delete the script library.
Administer security	Permission to change the security settings for the script library.
Change owner	Permission to take ownership of the script library.

Permission to create new recipients is controlled through the Default Script Security options.

Members of the computer's Administrators group always have View and Change Owner permission, even if those permissions are not listed, or are specifically denied. An Administrator can always take ownership of a script library.

The job's Owner always has Full Control permission for the notification recipient, even if that permission is not listed, or is specifically denied

Specifying default script security

Security Overview

Security Editor

Script Library Properties

Script Library Overview

Service Control Task

Service Control Task

The Service Control task allows adTempus to start, stop, and monitor system services.

When you configure adTempus to monitor a service, adTempus can detect when the service is stopped (e.g., due to an application failure) and restart the service automatically.

Like all other adTempus tasks, the Service Control Task is executed in the security context of the user account specified for the job. Therefore that user must have the authority to control the service you select, or the task will fail.

Using adTempus to automatically restart a failed service

To automatically detect and restart a failed service, configure the task as follows:

1. On the Options page select the service you wish to monitor and set the **Control Type** to "Start the service".
2. Check the **Monitor the service** option.
3. On the Responses page create a new response.
4. Add the "Service failed or was stopped outside of adTempus" event to the response.
5. Add a "Control a job or job step" (Job Control) action to the response. Set the Job Control action's **Action to take** to "Restart the step".
6. Optionally (but recommended) specify a retry limit for the action so that adTempus does not end up continuously restarting a service that fails repeatedly.

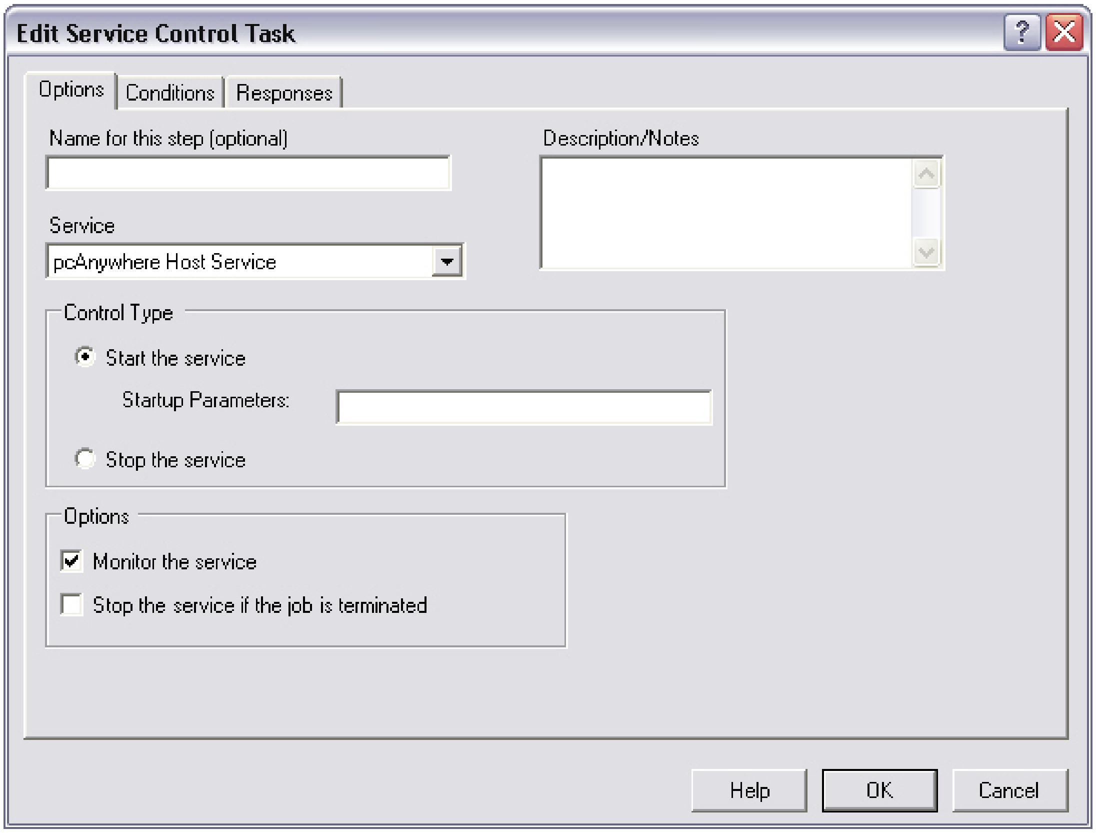
Service Control Task Properties

Tasks Overview

Steps Overview

Options Page

Show Window Sample



Name for this step (optional)

Description/Notes

Service: pcAnywhere Host Service

Control Type

● Start the service

Startup Parameters:

○ Stop the service

Options

☑ Monitor the service

□ Stop the service if the job is terminated

The **Options** page defines the information adTempus needs to control the service.

Name for this step (optional)

Optionally, specify a descriptive name for the step. Otherwise the name of the target service is used as the description for the step.

Description / Notes

Enter any extended descriptive information or notes for this step. There is no limit on the length of the text.

Service

Select the service you wish to control. The list of services is taken from the server.

Control Type

Select the action you want adTempus to perform:

- **Start the service.** adTempus will start the specified service. If the service is already started, adTempus will log an informational message but the step will succeed.

Optionally, specify any **Startup Parameters** that should be passed to the service.

- **Stop the service.** adTempus will stop the specified service. If the service is not running, adTempus will log an informational message but the step will succeed.

Options

When **Start the service** is selected as the control type, the following additional options are available:

Monitor the service

When this option is selected, adTempus will monitor the service. If adTempus detects that the service has stopped, it will fire the "Service Failed" event. By creating a Response to this event you can have adTempus restart the service (by restarting the step) or take other actions as appropriate.

When this option is selected, the job step does not end until the service has stopped (and no responses cause it to restart). Therefore, execution of the job will not continue past this step until the service stops. If you use this option you generally will want this step to be the only—or last—step of the job.

adTempus can only detect that the target service is no longer running. It cannot determine *why* the service is no longer running. Therefore any of the following circumstances could trigger the task's failure response:

- The service application aborts ("crashes") in some way.
- A user, or some other application, stops the service (e.g., using the Service Control Manager).
- The operating system stops the service because the computer is shutting down. You should therefore make sure that adTempus is the first service to be stopped when the computer is being shut down or restarted.

Stop the service if the job is terminated

When this option is selected, adTempus will stop the service if the job is terminated (either through a manual abort or as a result of a Job Control action). The service is **not** stopped if the adTempus service is stopped.

This option is only available if the **Monitor the service** option is selected.

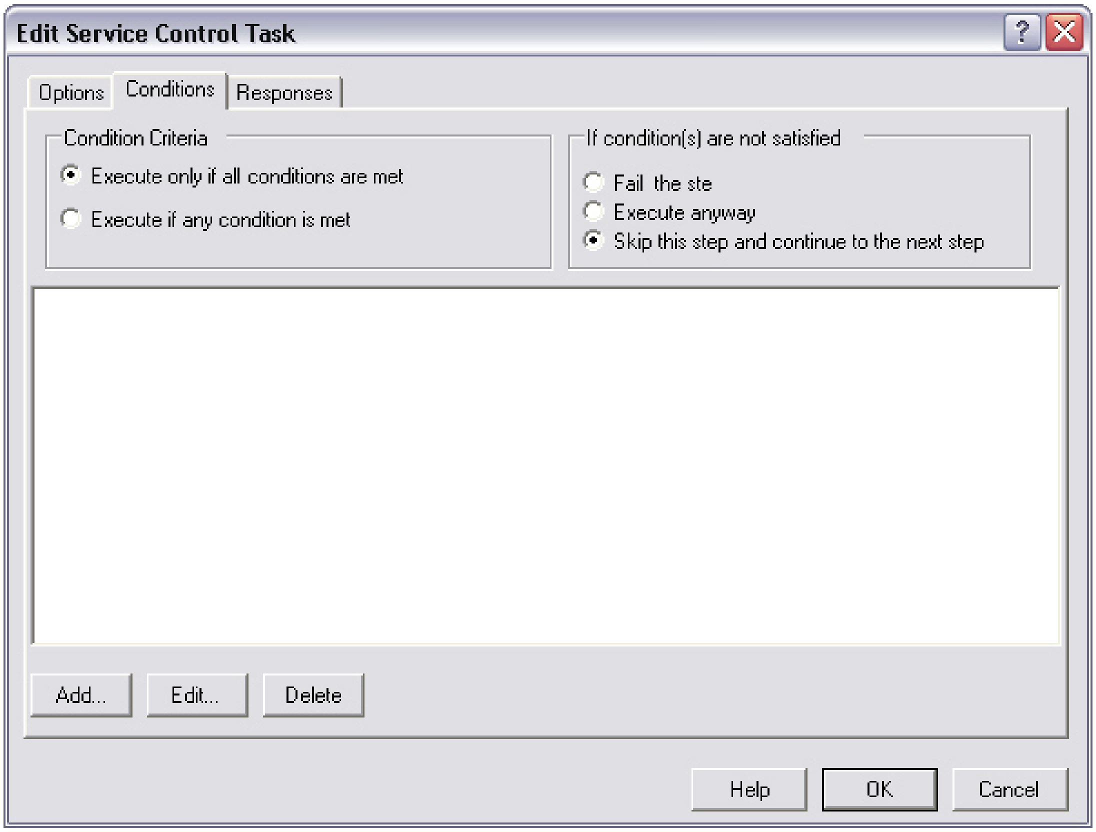
[Using adTempus to automatically restart a failed service]()

[Service Control Task Properties]()

[Service Control Task Overview]()

Conditions Page

Show Window Sample



Edit Service Control Task
Options
Conditions
Responses
Condition Criteria
● Execute only if all conditions are met
○ Execute if any condition is met
If condition(s) are not satisfied
○ Fail the ste
○ Execute anyway
● Skip this step and continue to the next step
Add...
Edit...
Delete
Help
OK
Cancel

The **Conditions** page defines conditions that must be satisfied before the step is run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- **Execute only if all conditions are met.** The step is only executed if all of the listed conditions are met.
- **Execute if any condition is met.** The step is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- **Fail the step.** The step is not executed; the status is set to Failed.
- **Execute anyway**. The step is executed anyway.
- **Skip the step (do not report as a failure)**. The step is not run, but it is not treated as a failure. The status of the step is reported as "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the step. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

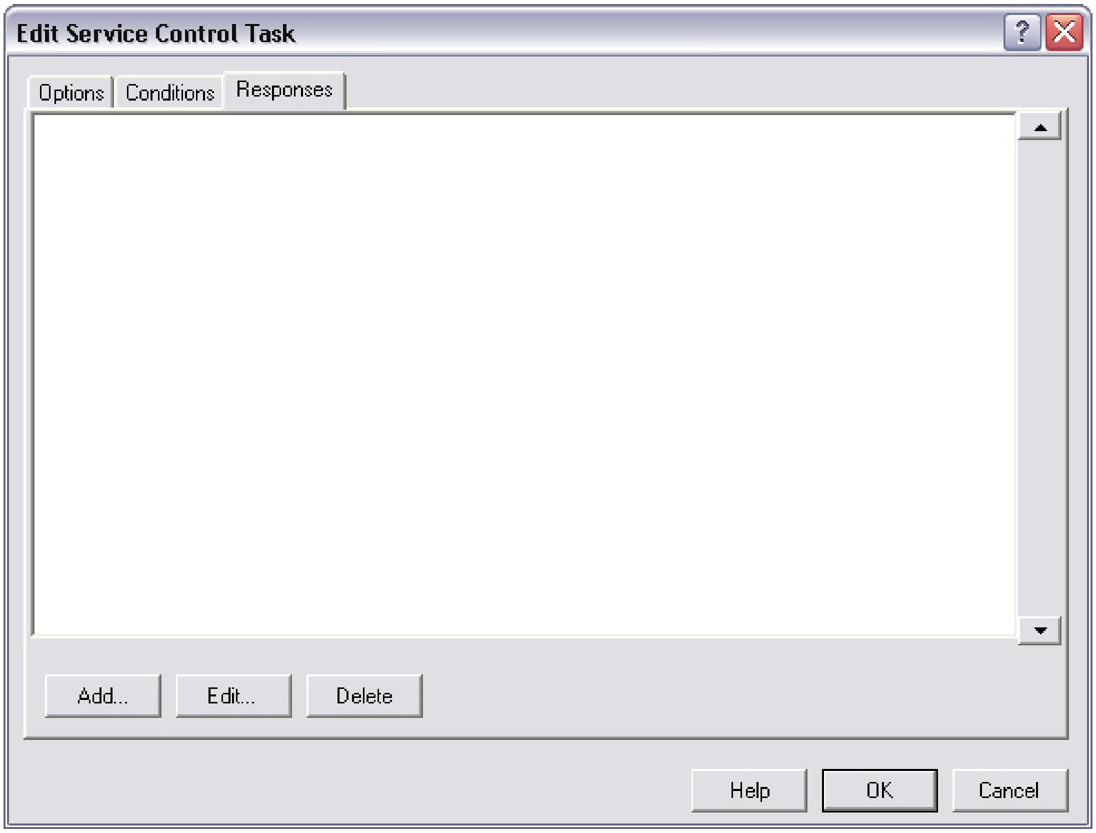
Service Control Task Properties

Service Control Task Overview

Conditions Overview

Responses Page

Show Window Sample



The **Responses** page defines the actions that adTempus should take in response to events that are fired during execution of the step.

Responses List

The **Responses** list lists the responses that have been specified for the step. You can add, edit, delete, or reorder responses. See the Responses topic for more information on the available responses.

The following events are defined for the Service Control task:

Event	Description
Beginning of step	Occurs at the beginning of the step, before adTempus has evaluated any conditions or attempted to execute the task.
Condition(s) failed	Occurs if one or more conditions is not met (occurs only once regardless of how many conditions failed).
End of step	Occurs at the end of the step, regardless of the step's outcome.
Restart limit exceeded	Occurs when the step is being restarted by a Job Control action but the restart limit specified on that action has been exceeded.

Step failed	Occurs when the step's status is being set to Failed, for whatever reason.
Step is being restarted	Occurs when the step is being restarted by a Job Control action.
Step skipped	Occurs when the step is skipped because conditions were not met (if the **Skip...** option is selected on the Conditions page).
Task returns a failure result	Occurs if adTempus is unable to start or stop the target service.
Task returns a successful result	Occurs if adTempus succeeds in starting or stopping the service. **Note:** This event occurs before adTempus begins monitoring the service, so you can use it to respond to the service start without waiting for the monitoring to complete.
Service failed or was stopped outside of adTempus	Occurs if adTempus detects that the service is no longer running. Valid only if the Monitor the service option has been selected.

Using adTempus to automatically restart a failed service

Service Control Task Properties

Service Control Task Overview

Responses Overview

Shared Schedule

Shared Schedule

A shared schedule is a schedule that can be used by more than one job. For example, you might have a predefined "weekdays" schedule for jobs that executed Monday through Friday. When you create a new job that needs to run on weekdays, you can link it to this shared schedule, rather than having to define a new schedule for the job.

Note that the shared schedule only defines the days on which jobs should run; each job has its own settings to determine the time(s) at which it should run.

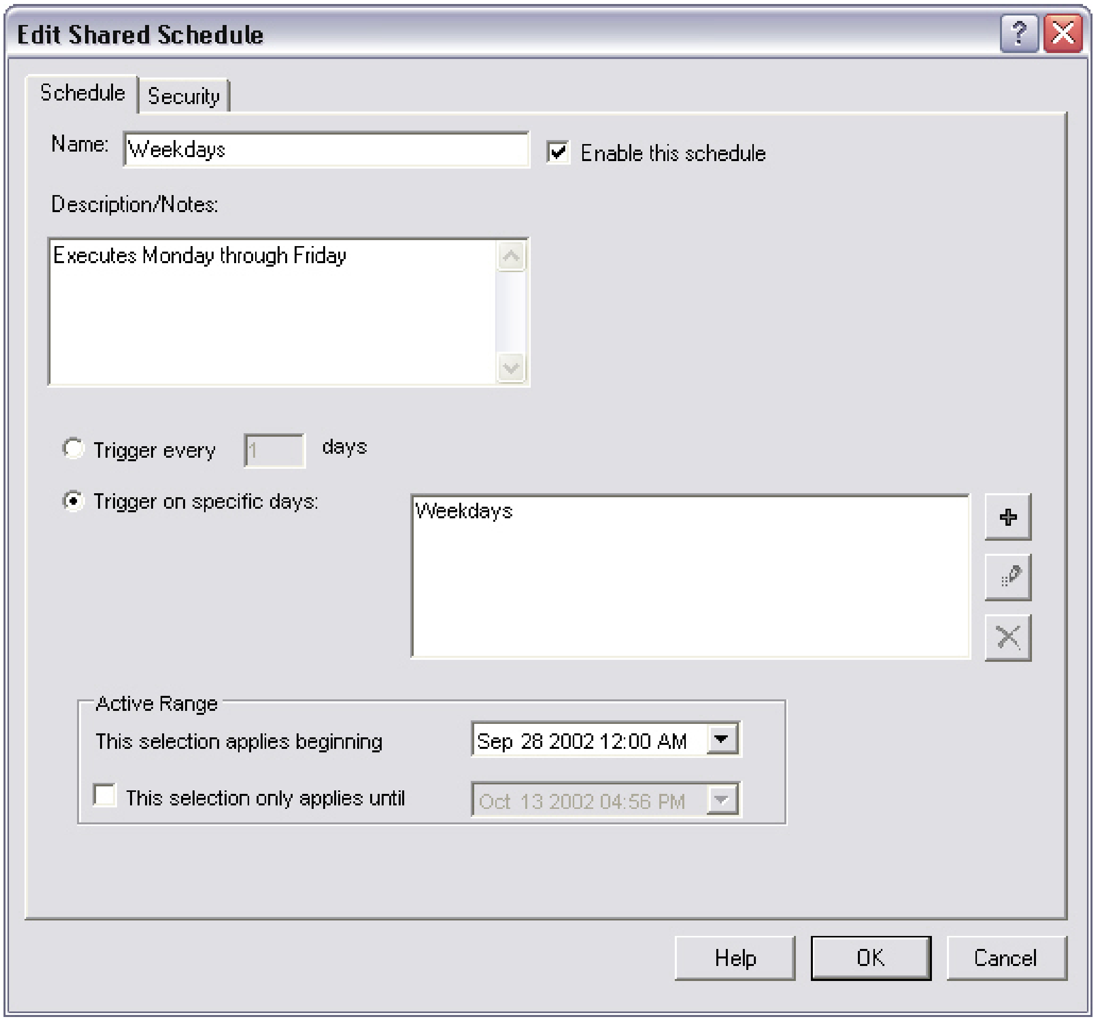
Shared schedules are used on the Date Selection page of a Schedule.

Use the settings on the Security page to specify which users are allowed to use this schedule.

Shared Schedule Properties

Schedule Page

Show Window Sample



Edit Shared Schedule

Name: Weekdays ☑ Enable this schedule

Description/Notes:

Executes Monday through Friday

○ Trigger every 1 days

● Trigger on specific days: Weekdays

Active Range

This selection applies beginning Sep 28 2002 12:00 AM

□ This selection only applies until Oct 13 2002 04:56 PM

The **Schedule** page defines the days on which jobs that use this shared schedule should execute.

Name

Provide a descriptive name for this schedule. The name must be unique across all shared schedules and holiday sets.

Enable this schedule

Check the box to enable the schedule or clear the box to disable it. If a shared schedule is disabled, it will not appear in the list of available shared schedules when you are creating a job.

If the schedule is disabled, jobs that use it will not be triggered by the schedule.

Date Selection

Two options are available to specify the days on which this item will be triggered:

Trigger every ___ days

Specify the interval (in days) at which the item should be triggered. The interval is calculated from the starting date specified in the **Active Range** section. For example, if you specify an interval of 3 and the **Active Range** has a starting date of October 1, 2002, the item will be triggered on October 1, 4, 7, 10, etc.

Trigger on specific days

Specify any number of specific dates or date rules to define the days on which the item will be triggered. See the Select Days topic for information on specifying days.

Active Range

Specify the earliest date and time to which this schedule applies. Optionally, specify an ending date and time.

If you specify an ending date and time, the schedule will not be used (will not trigger jobs) after that date/time.

Shared Schedule overview

Selecting Days

Security Page

The **Security** page defines the security settings for the shared schedule.

For information on specifying security settings see the Security Editor and Security Overview topics.

The following permissions apply to shared schedules:

Permission	Description
Full Control	Permission to perform all actions on the schedule.
List/User	Permission to use the schedule.
View	Permission to view the properties of the schedule.
Modify	Permission to modify the properties of the schedule.
Delete	Permission to delete the schedule.
Administer security	Permission to change the security settings for the schedule.
Change owner	Permission to take ownership of the schedule.

Permission to create new recipients is controlled through the Default Shared Schedule Security options.

Members of the computer's Administrators group always have View and Change Owner permission, even if those permissions are not listed, or are specifically denied. An Administrator can always take ownership of a shared schedule.

The job's Owner always has Full Control permission for the notification recipient, even if that permission is not listed, or is specifically denied

Shared Schedule overview

Specifying default shared schedule security

Security Overview

Security Editor

Startup Trigger

Startup Trigger

A Startup Trigger is a trigger that causes a job to be executed each time the adTempus service is started.

Since the service is generally only when the computer is restarted, this trigger can be used to start programs when the computer restarts.

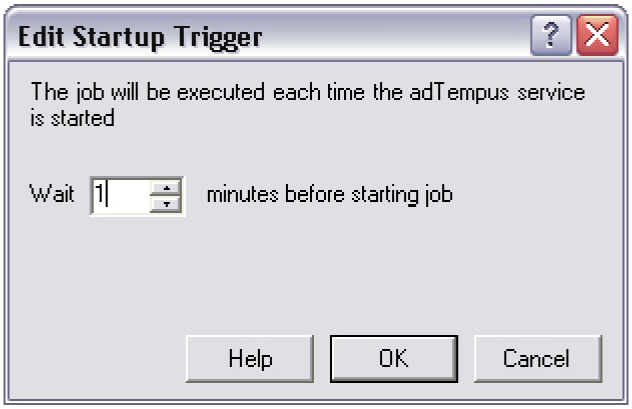
If adTempus is stopped and restarted for a reason other than a computer restart, your program may still be running when adTempus restarts. When you use the Startup Trigger you should also use process conditions, the Process Execution task's **Skip if already running** option, or some other safeguard to make sure that your scheduled tasks do not get re-executed when they should not.

Startup Trigger Properties

Triggers Overview

Startup Trigger Properties

Show Window Sample


Edit Startup Trigger

The job will be executed each time the adTempus service is started

Wait 1 minutes before starting job

The Startup Trigger has only one option:

Wait ___ minutes before starting job

Specify how long (in minutes) adTempus should wait before starting the job. For example, if the program run by the job depends on other Windows system services (such as a database server), you may want to have adTempus wait a few minutes before starting the job, to ensure that the required services have been started.

You can also use Process Conditions to make sure that programs on which the job depends have been started.

Startup Trigger Overview

The adTempus Service

The adTempus service is the "engine" that does the work of executing adTempus jobs. The adTempus service must be installed on each computer on which jobs are to be executed.

Because it runs as a system service the adTempus service does not have a user interface; all configuration is done using the adTempus Console.

The adTempus service is configured when adTempus is installed. Please note that for adTempus to execute jobs properly, you **must not** change this configuration:

- The service **must** be configured to run under the Local System account. This does not pose a security risk because each job must be run under a user account, which maintains system security.
- The **Allow service to interact with desktop** option must be checked.

Utility Programs

adtChkpt Utility

The adtChkpt utility can be used to set or retrieve the checkpoint for an executing job. This utility is meant to be used when you have a batch file that executes several different processes and you want to let adTempus know where things stand in case the job needs to be restarted from this point later.

This program relies on information put in the environment by adTempus and therefore will only work within a batch file run by adTempus.

To set the checkpoint, use the syntax

```
adtchkpt /s "checkpoint name"
```

Where "checkpoint name" is whatever string you want to pass as the checkpoint.

To query the checkpoint, use the syntax

```
adtchkpt /q "candidate1" "candidate2" "candidate3" [...]
```

Where each "candidate" value is a possible checkpoint for the job.

adTempus sets the exit code to the index of the candidate that matches the checkpoint, or to 0 if no match is found. For example, if the checkpoint were "candidate2", the example above would return an exit code of 2. You can test the exit code in your batch file and use it to jump to the correct place in the batch file when a job is being resumed.

For example, your batch file might look like this:

```
@echo off
rem check to see if the job is being started with a checkpoint.
rem if so, jump to the proper step
"c:\program files\arcana development\adtempus\adtchkpt" /s "step1"
"step2" "step3" "finished"
if errorlevel 4 goto the_end
if errorlevel 3 goto step3
if errorlevel 2 goto step2

:step1
rem tell adTempus we're starting step 1
"c:\program files\arcana development\adtempus\adtchkpt" /s "step1"
rem run the preprocessing program...
c:\prod\preprocess.exe

:step2
rem tell adTempus we're starting step 2
"c:\program files\arcana development\adtempus\adtchkpt" /s "step2"
rem run the main process
```

```
c:\prod\theprocess.exe

:step3
rem tell adTempus we're starting step 3
"c:\program files\arcana development\adtempus\adtchkpt" /s "step3"
rem do cleanup
c:\prod\cleanup.exe

"c:\program files\arcana development\adtempus\adtchkpt" /s
"finished"
:the_end
```

adtExec Utility

The adtExec utility can be used to execute adTempus jobs from the command line. You can use this utility, for example, to start a job from a batch file, or to create a desktop shortcut that starts a job.

adtExec uses the same authentication as the adTempus Console: the user executing the utility must have Execute permission for the job.

adtExec has the following syntax:

```
adtexec job [options]
```

All parameters except the job are optional.

Parameter	Meaning
job	Specify the name (enclose in quotes if it contains spaces or other punctuation) or Job ID (available from the job properties window) of the job you want to execute.
-f	Forces a new instance of the job if another instance is already running. This is the default behavior.
-f-	Does not force a new instance of the job if another instance is already running.
-j	Ignores conditions for the job. This is the default behavior.
-j-	Does not ignore conditions for the job.
-s	Ignores conditions for steps.
-s-	Does not ignore conditions for steps. This is the default behavior.
-m	Forces the job to run on the Master computer.
-a	Runs the job on all enabled agents associated with the job. If not specified, job runs on the Master only.
-t*stepNumber*	Runs the job starting with the specified step number. (1 is the first step)

-c"*checkpoint*"	Passes the *checkpoint* you specify to the job.
-u*userID*	Uses the specified Windows *userID* for authentication, rather than using the identity of the user running the program.
-p*password*	Specifies the *password* to use for the account specified with the -u parameter.
-r*ComputerName*[:*port*]	Connects to adTempus on the specified remote computer, rather than on the local computer.
-w	Waits for the job to run. adtExec will submit the job, then wait for it to complete and report its status. If the job completes successfully, adtExec returns an exit code of 0. If the job does not complete successfully, adtExec returns an exit code greater than 0. If this parameter is not included, adtExec returns immediately after submitting the job and provides no further information about the status of the job.

Technical Information

The adTempus Database

adTempus uses a relational database to store its data. adTempus is installed with a Microsoft Access database, which will be adequate for most users. However, you may wish to switch to a SQL Server database.

Microsoft Access Database

The adTempus database is found as `adtempus.mdb`, in the "data" subdirectory under the adTempus program directory.

The Microsoft Access database engine is adequate for most users in terms of performance and database capacity. Users with high levels of scheduling activity may see problems in database performance or excessive database size.

Database size problems

The Microsoft Access database engine is inefficient when it comes to reclaiming unused space in the database file. This problem is exacerbated in adTempus because adTempus writes a large amount of temporary data to the database when it executes a job, which is then deleted after the job completes. Because the disk space required for this data is not effectively reclaimed by Microsoft Access, the database file continues to grow, well in excess of the amount of data currently in the database. (For example, a database with only 5 MB of "real" data can easily grow to well over 200 MB in size after a week of heavy use.)

Space can only be reclaimed by "compacting" the database, which can only be done when the database is not in use. adTempus automatically backs up and compacts the database as described in the **Database Backups** section below. If your database continues to grow, it may be because adTempus is unable to perform the daily backup and compact operation; see the next section for information on resolving or working around this.

Database Backups

Many backup applications will not be able to back up the adTempus database, because the database file is continually in use by adTempus.

Each time adTempus starts, it makes a backup copy of the database (saved in the "data\backup" subdirectory under the Arcana Scheduler program directory). This file can be backed up by the backup software, because it will not be in use at the time of the backup. This process also compacts the database, releasing unused space.

Because it is expected that the adTempus service may be running for months at a time, adTempus also attempts to perform the backup and compact operation daily (by default around 1:30 AM; see the Server Options to change this time).

adTempus can only perform this operation at a time when no database activity is occurring (no jobs are running, being updated, etc.) adTempus monitors database activity and attempts to squeeze in this maintenance activity. If it is unsuccessful it will write a warning message to both the adTempus and Windows event logs. If adTempus is occasionally unable to perform this maintenance that is not a problem. If, however, you see that adTempus is consistently (or generally) unable to perform this maintenance, you should try one of the following solutions:

- If there are jobs scheduled to run at the time that the backup runs, change the backup time to a more "quiet" time.
- Or you may wish to switch to a SQL Server database (see below).

SQL Server Database

adTempus also supports Microsoft SQL Server databases. If SQL Server is already installed on the computer that is hosting adTempus you can use that instance of SQL Server.

If you do not have SQL Server, you can use the SQL Server Desktop Engine, which is available on the adTempus CD or by download from our web site. The SQL Server Desktop Engine is a "scaled down" version of SQL Server. It is available to you at no charge as part of your license for adTempus.

We recommend that you use a SQL Server database if:

- You have a high scheduling load (>100-200 jobs being executed per hour) or many jobs that are run at the same time.
- You are experiencing a problem with the size of the Microsoft Access database.
- You are experiencing performance problems (such as excessive memory usage by adtempussrv.exe or sluggish job execution).

For more information on converting to SQL Server, see www.arcanadev.com/adtempus/sqlserver.htm.

Data Security

All adTempus data is stored in the adTempus database, except captured files, which are stored in the "data\capturedfiles" subdirectory under the adTempus program directory.

The adTempus database and the captured files should be secured against unauthorized access using Windows file system security. For adTempus to operate, the SYSTEM account must have full control over the "data" subdirectory (and its contents and subdirectories). adTempus users do *not* need access to the data files.

User IDs and passwords within the adTempus database are stored in an encrypted format to protect against users who might gain access to the database.

All data is encrypted before being transmitted between the adTempus Console and Service, so it is protected when transmitted over a network connection.

Port Information

adTempus uses the TCP protocol for communication between the client and the server. The server listens on port 3760; the client uses a dynamically-assigned port.

If you are using remote administration and your server is behind a firewall, you will need to make sure that the firewall is configured to allow incoming connections on port 3760.

Changing the Port

If for some reason you need adTempus to use a different port, you can do so as follows:

1. Using the Registry Editor, locate the key `HKEY_LOCAL_MACHINE\Software\Arcana Development\adTempus`.
2. If this key does not have a subkey named `Options`, create one.
3. Under the Options key, create a new **string** (REG_SZ) value called "`Server:Port`". Set this value to the port number you want to use. Important: the registry value must be a string value, not a DWORD value.
4. You must restart the adTempus service for the change to take effect.

For clients to connect to the server, you must specify the correct port number:

1. From the **File** menu in the Console, select **Connect Server...**
2. In the Connection Properties window, select **Connect to a remote computer**. Note: You must select this option even if you are connecting to the local computer, because you need to be able to override the address for the server.
3. Specify the name or address of the server, followed by a colon and the port number. For example:
    - `localhost:60102` connects to the server on the local computer if you have configured it to listen on port 60102.
    - `server1:63109` connects to "server1" if you have configured it to listen on port 63109.

Index