adTempus User Guide

Version 3.0

March 31, 2008

Table Of Contents

Introduction

Overview

adTempus is a batch process scheduling and automation tool for Windows platforms.

By "batch process" we mean a program (or collection of programs, scripts, etc.) that are meant, for the most part, to be run in an "offline" or unattended fashion, such as a process that reads data feeds and updates your data warehouse each day. In adTempus, such a process is represented by an object called a **job**, which stores all the information adTempus needs to run the process.

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.

The **adTempus Server** (or "engine") is the part of adTempus that is responsible for executing jobs, by running programs and scripts, downloading files, sending e-mail notification messages, etc.

The **adTempus Console** is the primary user interface and management tool for adTempus. Using the Console you manage adTempus servers and the jobs defined on those servers. The Console can run on a different computer than the Server, allowing remote administration of adTempus Servers.

See Also

Getting started with adTempus

Introduction to Key Concepts

What's New

The following sections present an overview of many of the new features in adTempus. See the adTempus Web site for a complete list of changes in this release.

Major New Features

Job Queues

Newly-introduced Job Queues allow you to set limits on the number of jobs that can be run concurrently, enabling you to better manage resource usage on a server. For example, you may have two jobs that use so much CPU time that you don't want to let them run at the same time. Previously this required setting conditions between the jobs, and could be difficult to enforce, especially if jobs were run manually. Now you can simply assign both jobs to the same queue, and limit the queue to one job at a time. If a second job is submitted for the queue, it will be held until the first job finishes.

Job Variables

The new Job Variable feature makes it easier to change parameters for job tasks and to "templatize" jobs so they can be easily duplicated or copied from server to server. Job Variables are user-defined variables that can be used in many places within a job (in the same way that environment variables can be used in earlier versions). The values can be changed easily to simplify changes to jobs.

For example, you may have a group of jobs that all act on files in a particular directory. Within each job, you call a program, passing it the directory name. Instead of "hard-coding" the path on the command line for each job or job step, you can now use a Job Variable (such as "%TargetPath%") that is defined for the group. At runtime, the "%TargetPath%" token is replaced with the actual path you have specified. If you subsequently need to change the path (or copy the jobs to another server where the path is different), you can change the path in a single place and all jobs will use the new value.

Job Variables can be inherited (from server-level variables to group-level to job-level to step-level) and overridden at any level.

Users running a job manually can also override variable values. This allows you to set up parameterized on-demand jobs, where a user can specify parameters such as selection criteria without needing to modify the job definition.

Credential Profiles

Credential Profiles allow users to schedule jobs under a Windows security account without needing to know the password for that account, provided that they have been granted permission to use the Profile. This allows administrators to set up a standard "batch" account that is used by all jobs, without having to give adTempus users the password for this account.

Improved Scripting

adTempus now supports scripts written in VB.NET and C#, and features a much-improved script editor. Now many routine administrative tasks can be easily automated through .NET programs that are written and maintained within adTempus. There's no need to write (and keep track of) a standalone program or to suffer with creaky VBScript scripts.

Improved Desktop Selection

adTempus allows jobs to run "interactively" (visible to the user), but in previous versions an interactive job was only visible from the computer's console session, and not from Remote Desktop sessions. Changes in the desktop selection options now make it possible to run a job in the console session, the session of the user whose account is used on the job, or the session of the user who is running the job manually.

Security Inheritance

The adTempus security framework has been overhauled to support security inheritance, similar to the security inheritance of folders and files in Windows. This greatly simplifies security administration compared to the old model, which required that security changes be made separately to each job.

New Tasks

Web Request Task

A new "Web Request" task has been added to adTempus (System Management Edition only) that allows your job to request a Web URL. This can be used to request a page that triggers processing on the remote server, or to download content from the server. The page or file returned by the Web server can be saved to disk and/or saved as a captured file in the job's history.

SQL Server Job Task

In previous versions of adTempus, the Job Wizard could be used to generate a script that could run and monitor SQL Server Agent jobs. In adTempus 3.0 this has been replaced by a full-fledged task that offers the same functionality but is easier to configure.

Job Execution Enhancements

More Flexibility When Running Jobs Manually

More options are now available when a user runs a job manually from the Console, offering better control over job execution. The user can now select which step(s) of the job should run, and determine whether Responses for the job should be executed.

Responses Based on Execution Time

Responses can now be triggered based on the length of time a task has been running. For example, you can now configure a Response to send a notification message if a program runs for more than one hour. Previously, the only action that could be taken based on task execution time was to terminate the task after a specified time had elapsed.

Job Management and Monitoring Enhancements

Failure Acknowledgment

Previous versions of adTempus allowed you to "acknowledge" failed job instances shown in the failed job view by changing the filter date to exclude those instances. adTempus 3.0 introduces true acknowledgment: a user can selectively and permanently "acknowledge" the failure of any instance, and doing so removes the instance from the failed jobs view of all other users as well.

History Comments

System operators or other authorized users can now add comments to a job's history. For example, if a job fails, the operator may want to record details of what caused the failure and what error resolution steps were taken. Comments for each instance are viewed and edited on the new **Comments** page of the Job Instance Details window (displayed when you view the details from an instance on the **History** page of the main job list).

Status Override

System operators or other authorized users can override the status of a completed job, forcing adTempus to show a "Succeeded" or "Failed" status for the instance. This can be used when you need to let jobs that depend on the job execute despite a failure.

Manual History Purge

In previous versions of adTempus, a job's history (execution history, log messages, and captured files) could only be purged through the automatic process controlled by the retention settings for the job. The adTempus Console now provides the option to purge the job's history complete on demand, and optionally to reset the statistics for the job as well. This new option can be found on the Statistics tab when the job is selected in the main job list.

User Interface Improvements

Many changes have been made to the adTempus Console to improve usability:

- Improved Search/Replace capability.
- New Go to Job command lets you find a job quickly
- Job Group duplication duplicates a group and all jobs in it.

Other Changes

Support for Windows Vista and Longhorn

adTempus is now fully compatible with Windows Vista and Windows Longhorn. Some visible changes to support these operating systems include:

- When configuring a job that runs under an administrator account, you can specify that the job should be run "with highest privileges," which is equivalent to running the program "As Administrator" from the Windows shell. adTempus will then run the program with full administrator privileges enabled, without a user access control (UAC) prompt.
- The adTempus Console and other client tools are marked to be run with elevated privileges when run by an Administrator. Administrators will therefore see UAC prompts when running the client tools.

XML-Based Export/Import

Previous versions of adTempus used an undocumented binary file format for exporting and importing data. adTempus now uses XML for exports and imports, making it easier for customers to modify exported data before it is imported into adTempus.

Due to this change, this version **cannot** read export files produced by earlier versions. If you want to preserve data in export files, you must import it before upgrading to adTempus 3.0, then export it again in the new format after upgrading.

Key Concepts

Introduction to Key Concepts

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

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

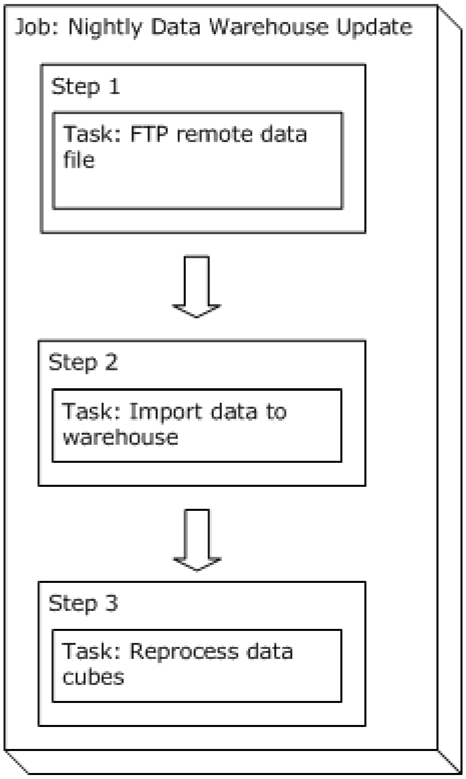
Before we proceed, we will outline an example that will be used to illustrate concepts as we go.

> **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


Job: Nightly Data Warehouse Update
Step 1
Task: FTP remote data file
Step 2
Task: Import data to warehouse
Step 3
Task: Reprocess data cubes

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 **Step**; 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 three steps we listed above becomes a step within the job.

Note that in some other automation tools, "job" and "task" are synonymous: each of the steps above would be a separate jobm and the jobs would be linked together to complete the entire process. While you could set up your processing flow this way in adTempus, it is often easier to manage fewer jobs with more steps than more jobs with fewer steps. For more information, see the Multiple Steps vs Separate Jobs topic.

The task represents a single operation that adTempus is to perform (typically, a single program, script, batch file, or other operation 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; control services; and more. For more information, see the Tasks topic.

For more information, see the Jobs and Tasks topics.

Triggers

Jobs can be started manually (a user tells adTempus to run a particular job) but are generally started automatically by adTempus at specified times, or in response to other events on the computer.

A **Trigger** is 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. Other kinds of triggers allow you to create jobs that will run in response to other events, such as when a file is uploaded to a particular directory on your server.

For more information, see the Triggers topic.

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.

For more information, see the Conditions topic.

Responses, Events, and Actions

Responses allow you to respond to state changes during the execution of a job, and to customize 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.

Using a Response, for example, you can send an e-mail notification message when a job fails or start the next job in a sequence when the job succeeds.

For more information, see the Responses topic.

Job Groups

Job Groups allow you to organize your jobs into "folders" for easier management in the Console. Job Groups are simply a visual organizational tool, and do not create any sort of relationship between the jobs in a Group.

For more information, see the Job Group topic.

Job Queues

Job Queues allow you to limit the number of jobs that can run at the same time, or prevent certain jobs from running at the same time.

For more information, see the Job Queue topic.

Job Variables

Job Variables allow you to easily use changeable values in the configuration and execution of your jobs. Job Variables are similar in concept and usage to Windows environment variable: you create a variable, giving it a name and a value. You can then insert the variable (by referencing its name) in various places within adTempus (such as the command line for a program to be executed). When adTempus runs the job, it replaces the reference to the variable with the variable's current value.

For example, supposed you have a job with 8 steps. Each of those steps runs a program that requires as a command-line parameter the name of the folder from which it should read data files. All of the steps use the same folder, which is usually "c:\clientdata\inputfiles". However, occasionally you need to do a special run of this job, reading data from a different folder. Without Job Variables, such a change would require you to edit all 8 steps to change the command-line parameters to point to the new location.

With Job Variables, you can create a variable for the job (on the Variables page of the job's properties) named "DataPath" and set the variable to "c:\clientdata\inputfiles". In each Job Step, instead of putting the path "c:\clientdata\inputfiles" in the command-line parameters, you put a reference to the job variable, using this syntax:

```
%DataPath%
```

When the job runs, adTempus will replace "%DataPath%" with the current value of the "DataPath" Job Variable.

Now when you need to do a special run using a different path you can simply change the value of the "DataPath" variable in one place (in the job's properties) and the new value will be used throughout the job. And if you run the job manually, you can override the "DataPath" variable at runtime without even modifying the job (using the Variables page of the Execute Job window).

Job Variable Inheritance

Job Variables may be defined at many levels within adTempus: Server, Job Group and Job Queue, Job, and Step. Variables set at a level are inherited by all objects at a lower level: variables set at the Server level are inherited by all Groups and Queues on that server, variables set for a Group are inherited by all Groups and Jobs within that Group, etc.

Inherited Variables can be overridden to change their values. In the previous example the "DataPath" variable was defined for a job. This variable will now appear on the "Variables" page in the properties of each step in the job. If one step needs to use a different value, you can edit the variable for that step only and give it the necessary value.

Using Job Variables

Job Variables can be used throughout adTempus in most places where you enter text that is used in the execution of a job. For example:

- The execution target and command-line parameters for a task.
- Paths for file triggers and conditions.
- Notification message subjects and bodies.

The help topic for a text entry box will note if Job Variables can be used.

To insert a Job Variable, simply type the name of the variable enclosed in percent signs, like this:

```
%Variable Name%
```

Variable names are not case sensitive, so you could also enter the variable as:

```
%variable name%
```

Variables can also be retrieved and set within scripts.

See Also

Predefined Variables

Setting and Retrieving Variables from scripts

Job Variable properties

Security

Security

adTempus uses the Windows security framework to control access to adTempus and to objects within adTempus.

When a user connects to adTempus, adTempus identifies the user and then determines the permissions for that user.

As in Windows, each secured object support various rights. Individual users or groups are either granted or denied rights. For example, most objects have a "View" right and a "Modify" right. This allows a user to be granted permission to view, but not modify, an object.

The property sheet for each secured object contains a Security page, where a standard security editor is used to modify the permissions for the object.

See the Security Configuration Guidelines topic for recommendations on how to configure security.

> This discussion applies to the management of jobs and other objects within adTempus. When a job executes, any tasks it carries out are subject to Windows security restrictions, based on the user account that the job is running under. See the Job Security topic for more information.

Security Inheritance

adTempus supports security inheritance, meaning that each object inherits permissions from its "parent." See the Security Inheritance topic for more information.

Special Permission Cases

There are several special cases to be aware of:

- Members of the server's Administrators group can view and take ownership of any object in adTempus. This is true even if the user (or the Administrators) group is not listed in the permissions for the object.
- The owner of an object always has complete control over the object. This is true even if the user is not listed in the permissions for the object.
- When you establish the default permissions for a type of object you can specify permissions for "CREATOR OWNER". This is a "placeholder" that is replaced with the identity of the creating user when the object is created.

Security Inheritance

adTempus supports security inheritance, meaning that each object can inherit permissions from its "parent." The model is similar to that of the Windows file system, where each file or folder inherits permissions from the folder above it. This inheritance makes it easy for you to manage security for large groups of users and objects.

ⓘ	Prior to adTempus version 3.0, security settings in adTempus were not inherited, meaning that each object's security had to be managed individually. If you have upgraded from a previous version, you may want to update your security settings as described in the Security Changes topic to simplify security administration.

See the Security Editor topic for information on how to control which permissions get inherited.

At the top of the adTempus security hierarchy are the security settings configured through the Server Options. These permissions are inherited by all objects within adTempus. Below this, the inheritance hierarchy follows the hierarchy outlined by the nodes in the Console Tree. For example, the **Jobs** node (which represents the "root" job group) inherits from the server settings. Each Group and Job inherits permissions from the Group above it.

See Also

Security Configuration Guidelines

Security Editor

The Security Editor is found on the Security property page for each object that can be secured in adTempus and is used to modify the permissions for the object.

To make changes you must have the "Administer Security" right for the object; otherwise you will be able to view but not change security.

The Security Editor is similar to the security properties offered by the Windows Explorer. Using the editor you may add and remove users and groups, and change the permissions for each.

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

Users and Groups

This list shows all users and groups who have permission for this object. Select an entity to view or change the permissions for that entity.

Permissions inherited from a higher-level object are indicated with "(inherited)" after the name. You cannot modify or remove inherited permissions, but you can override those permissions by adding a new entry for the same user or group and then allowing permissions that were not inherited or Denying permissions that were inherited.

Apply Permissions To

This option is only available if the object your are editing is a "security container," that is, if other objects inherit security from it. For example, a Job Group is a security container because the jobs within that group inherit its security settings.

This option determines whether the settings you are making will apply to the current object only, to the current object and its children, or only to its children.

For example, you want to give a user permission to modify all the jobs within a group, but not the group itself. To do this, you would select the "Jobs only" option before granting the "Modify" permission.

Permission

Select the permissions to Allow or Deny. The list of available permissions will depend on the object your are editing.

Inherit permissions from parent object(s)

When this option is checked, the object inherits permissions from its parent object (for example, a job inherits from the group it belongs to). Any changes made to the parent object's security settings will affect this object.

When this option is not checked, the object no longer inherits permissions, and changes to the parent do not affect the current object.

Remove explicit permissions on all child objects

This option is only available if the object your are editing is a "security container," that is, if other objects inherit security from it. For example, a Job Group is a security container because the jobs within that group inherit its security settings.

When you check this option, adTempus removes any object-specific permissions you have set for any descendents of this object, and forces them to inherit its permissions.

Owner

Indicates the owner of this object. The owner always has full control over the object. Users with the necessary permission can use the **Take Ownership** to take control of the object.

See Also

Security Overview

Installation

System Requirements and Prerequisites

System Requirements

adTempus runs on Windows platforms only. The following operating systems are supported:

- **Desktop Operating Systems:** Windows XP, Windows Vista, or later.
- **Server Operating Systems:** Windows Server 2003, Windows Server 2008 ("Longhorn"), or later.

adTempus is available in a 32-bit version and a 64-bit version (x64 only; Itanium is not supported).

Prerequisites

The following software components are required by adTempus, and will be installed if they are not present:

- Microsoft .NET Framework, version 2.0
- Microsoft XML Parser (MSXML) version 6.0 of the

Database Requirements

By default, adTempus will install, manage, and use its own instance of Microsoft SQL Server 2005 Express, and this is the recommended configuration. If you prefer to use your own existing database server, see the Database Selection topic for additional information.

Upgrading from Previous Versions

Upgrading from Previous Versions

If you are upgrading from a previous adTempus version, you can (and should) install version 3.0 without removing the previous version. The setup program will upgrade your existing installation.

Please review the information below for changes that cause problems for upgrading users. See the What's New topic for an overview of new features, and the adTempus Web site for more information about changes in this version of adTempus.

> The upgrade process will upgrade your adTempus database, making it incompatible with previous versions of adTempus. Before you install, we highly recommend that you back up your adTempus database so that you can restore it if it is necessary to revert to the previous software version.

Database Changes

If you are using the MSDE database engine for adTempus, we recommend that you upgrade to the newer SQL Server 2005 Express engine. To do this, be sure to check the **Install, configure, or reconfigure the adTempus database** option on the **Database Information** page of the Setup wizard. Setup will install SQL Server Express and migrate your database automatically.

The Microsoft Access database format is no longer supported by adTempus. If you are still using an Access database for adTempus, it will be converted to SQL Server Express.

Distributed Scheduling Changes

If you are using the Distributed Scheduling features of adTempus, note that jobs are no longer associated directly with Remote Agents. Instead, all jobs are assigned to Job Queues, and Distributed Scheduling options can be set for each Queue. During the upgrade process, adTempus will create new queues and assign jobs to them automatically based on your existing Distributed Scheduling settings. When you start adTempus for the first time after the upgrade, review the settings for each queue to ensure they are correct, and check that jobs are assigned to the correct queues.

Security Changes

The new security model supports security inheritance. See the Security Changes topic for more information.

Breaking Changes

Some "breaking" changes in adTempus 3.0 may cause problems for some users:

- adTempus does not support side-by-side installations of the server software
- adTempus 3.0 cannot interoperate with previous versions of adTempus
- The adTempus export/import format has changed
- The adTempus API has changed
- The simple Web-based monitoring tool available for adTempus 2.0 will not work with this version. A new ASP.NET interface is planned for the future.
- Some little-used features have been removed: the Application Exception trigger, SMS notification using modems, and the ability to send keystrokes to applications.
- Command-line options for the adtexec utility have changed.

The sections below provide more information on these points.

adTempus does not support side-by-side installations of the server software

You cannot have more than one version of the adTempus server software (the adTempus service) installed on a computer. If you install the server software, Setup will upgrade any existing installation.

If you install only the client software (the adTempus Console), it is possible to install it on a computer where an earlier version is installed, without removing that version. That is, you can install two different versions of the adTempus Console on the same computer.

Note that you can install the adTempus 3.0 Console on a computer with version 2.x installed, but you cannot install the version 2.x Console on a computer after you have installed version 3.0. Therefore if you need to be able to run both versions of the Console on a computer, be sure to install the 2.x version first and then the 3.0 version.

adTempus 3.0 cannot interoperate with previous versions of adTempus

adTempus 3.0 uses a different communications mechanism than previous versions of adTempus. Therefore:

- If you are using Distributed Scheduling, you must upgrade the Master and all Agents at the same time. If you upgrade only one computer, it will not be able to communicate with the other computers.
- If you use inter-job dependencies (conditions or responses), you must upgrade all linked computers at the same time.
- If you do a full installation of adTempus on a computer, you will not be able to use the Console on that computer to administer computers running earlier versions of adTempus. See the previous section for information on side-by-side installations of the Console.

The adTempus export/import format has changed

adTempus 3.0 uses a new XML-based file format for data export and import. This format is not compatible with prior versions. adTempus 3.0 cannot read export files created by previous versions, and previous versions cannot read export files created by adTempus.

Therefore if you have exported data that you will need in adTempus 3.0, you need to import it into adTempus 2.x before upgrading. You can then re-export the data in the new format after the upgrade.

The adTempus API has changed

The adTempus API has changed for version 3.0. Though the architecture is the same, many changes have been made to support new features. The new API is not binary-compatible with code built for previous versions.

In previous versions of the adTempus API, collections used "one-based" indexing, consistent with Visual Basic practices. In this version the API has been modified so that all collections use "zero-based" indexing, consistent with modern C++ and .NET practices. As a result, all customer code that manipulates collections exposed by the API must be modified to use the new zero-based indexing.

The collections now expose enumerators, so customers may wish to change to enumerator access to the collections rather than index-based access.

At a minimum you will need to recompile your applications that use the API, and in many cases will have to make minor code changes. See the adTempus API page for the latest information.

Web client not available

Due to the API changes discussed above, the simple ASP-base Web monitoring tool available for adTempus 2.0 will not work with adTempus 3.0. A new ASP.NET Web interface is planned for the future.

Some little-used features have been removed

The following features have been removed from adTempus:

- The Application Exception trigger
- SMS notification using modems
- The ability to send keystrokes to applications

If you are currently using these features, contact us to discuss alternatives.

Command-line options for the adtexec utility have changed

The command-line syntax for the adtexec utility has changed. If you are using the adtexec utility to submit jobs for execution, you may need to modify your batch files, scripts, etc. See the adtexec topic for information on the new command-line syntax.

Security Changes

adTempus 3.0 introduces security inheritance to the security model. Users upgrading from previous versions of adTempus may have permissions individually assigned to jobs and other objects due to the limitations of the earlier security model. For ease of administration, users will in most cases want to remove object-level permissions in favor of inherited permissions. For example, if all your jobs have the same security settings, earlier versions of adTempus required you to make those settings on each job. With security inheritance, you can make those settings at the root job level and have all jobs inherit them.

Therefore you may need to remove the existing explicit permissions from the individual objects, so that they can inherit all of their security settings. Rather than edit each object individually, you can modify a higher level of the security hierarchy and tell adTempus to remove the non-inherited permissions on each lower level.

For example, if you want all jobs to have the same security settings, edit the settings for the root group (the "Jobs" node in the server tree). After setting the permissions you want your jobs to inherit, check the "Remove explicit permissions on all child objects" box. This will remove all permissions that have been set on individual groups and jobs, forcing them to use only the settings you have defined at the root level. If you now need to tailor the security of certain group and jobs, you can then edit them to override the inherited security.

See Also

Security Configuration Guidelines

Licensing

Licensing Overview

This topic provides an overview of licensing requirements and procedures for adTempus. See the License Agreement for the complete agreement that governs your use of the software.

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.

adTempus is not licensed based on the number of processors on the computer, or based on the processor architecture (32-bit or 64-bit).

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

Distributed Scheduling Agent Licenses

When you use the Distributed Scheduling features of adTempus, you install the adTempus server software on the Master computer and on each Agent computer. Each of these computers requires its own adTempus license. For example, if you have a Master computer and two Agent computers, you need three adTempus licenses.

Two license options are available for Agent computers:

- **A full adTempus license.** An Agent with a full adTempus license can run jobs sent to it by the Master computer. However, it can also run jobs that are managed locally.
- **A limited, agent-only license.** An Agent with an agent-only license can only run jobs sent to it by the Master computer. No jobs can be created or managed locally.

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

To view or manage licenses for an adTempus server, run the "License Management" tool found in the adTempus program group on the Windows Start menu.

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.

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 1.a 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. *"Bundled" Software.* If the SOFTWARE PRODUCT was provided to you (by Arcana Development or one of its licensees) with or as part of another software application (the "CONTAINING APPLICATION"), the SOFTWARE PRODUCT is considered “BUNDLED SOFTWARE” and you may install the SOFTWARE PRODUCT only as part of the CONTAINING APPLICATION. You are licensed to use the SOFTWARE PRODUCT only as long as you are licensed to use the CONTAINING APPLICATION.

d. *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 3.

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. BUNDLED SOFTWARE may not be transferred except as part of a transfer of the CONTAINING APPLICATION, if such a transfer is allowed under your license for that software.

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 as many copies as you wish of such electronic documentation, provided that it is for your use only. 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 2.f 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 except in compliance with 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.

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.

Installing adTempus

This topic discusses adTempus installation options and the installation process.

Before you continue please review the system requirements and prerequisites. Also be sure you are using the correct installer for your processor architecture (32- or 64-bit). If you are upgrading from a previous version of adTempus, please review the upgrade notes.

Your license information can be entered at the end of the setup process, or after setup using the License Manager tool in the adTempus program group on the Windows Start menu.

adTempus Components

adTempus is a "client/server" application, divided into two primary components:

- The adTempus Service (or "engine") does all the work of running jobs. The adTempus service must be installed on each computer on which jobs should execute. Throughout this document we refer to the Service, and the computer on which it is running, as the "server."
- The adTempus Console is the user interface for adTempus. It is used to configure and monitor adTempus. The Console may run on the same computer as the Service, or from a remote computer. A single Console can be used to monitor any number of adTempus services. Throughout this document we refer to the Console, and the computer on which it is running, as the "client."

If you are installing for the first time, you should perform a complete installation on the computer on which you want to run scheduled jobs. Optionally, you may install the client components on additional computers if you wish to administer adTempus remotely.

You must purchase a license for each computer on which you install the server components (the adTempus service). You may install the Console on any number of additional computers without purchasing additional licenses. See the licensing overview for more information.

Setup Process

When you start the adTempus setup program, it will install any prerequisites that have not already been installed, and then start the Setup Wizard.

1. Click **Next** to continue.
2. Review the installation information. For the latest information on changes in the software, click the **Release Notes** button.
3. Click **Next** to continue to the License Agreement. Review the agreement, check the option to indicate your acceptance, and click **Next** to continue.
4. Select the folder you want to install the software to, and click **Next**

Note: Some parts of the software will be installed on the system drive, even if you select a different folder for the main program components. In particular, the reporting components (installed by default; see below) will always be installed to the system drive, and require approximately 72MB of disk space.

5. Select the setup type and click **Next** to continue. The following setup types are available:
   - **Complete** installs all software components.
   - **Client Only** installs only the adTempus Console and other client utilities, which you can use to remotely administer the adTempus server installed on another computer.
   - **Custom** allows you to select which components to install. Use this option if you want to install the server components without the client components, or to not install the reporting components, which require approximately 72MB of additional disk space.
6. Select the mode that the adTempus server will operate in, then click **Next**.

   If you are not planning to use the Distributed Scheduling features of adTempus, select the **Standalone** option. Otherwise, select **Master** if this instance will manage other adTempus instances, or **Agent** if this instance will be managed by another instance. See the Distributed Scheduling Setup and Engine Mode topics for more information.

   Note: The mode can be changed after installation using the Engine Mode program.
7. Setup will display a note about database installation and configuration. Review the information and click **Next** to continue.
   - If this is a new adTempus installation, the database installation and configuration tool will run later in the setup process.
   - If you are upgrading from an earlier adTempus version, Setup by default will not change any database settings. If you want to change settings, check the box, and the installation and configuration tool will run later in the setup process.

     Note: If you are currently using the MSDE database engine installed with adTempus versions before 3.0, we recommend that you update to the newer SQL Server 2005 Express database engine included in adTempus 3.0 and later. To do this, check the box indicating that you want to run the database installation and configuration tool.
8. Click **Install** to install the software.
9. Setup will install the adTempus software, then display a message indicating that adTempus has been installed.
10. Click **Finish** to exit the installation process.
11. If adTempus cannot find a valid license for adTempus on this computer, the License Manager will start and prompt you to either activate the software using your license information, or continue running the software in Evaluation mode.
12. If this is a new adTempus installation, or if you checked the option to run the database installation and configuration tool, that tool will start now. See the Database Installation and Configuration topic for more information about this process.

    Note: If you cancel database installation and configuration, the adTempus software remains installed, but the adTempus service will fail until you have installed and/or configured the database using the Database Installation and Configuration tool.

13. Installation of adTempus is now complete. To begin using adTempus, start the adTempus Console from the adTempus group on the Windows Start menu.

    If this is a new installation of adTempus you may want to configure some adTempus settings. Refer to the Getting Started topic for more information on what to do next.

Database Installation and Configuration

Database Selection

adTempus stores most of its data in a relational database. For the most part, you do not need to know anything about the database beyond making the initial determination about which database to use.

By default, adTempus will install the Microsoft SQL Server 2005 Express database engine, and this is the recommended configuration. However, you may also choose to use your own existing database server.

Using the Default Database Engine

The default database engine, Microsoft SQL Server 2005 Express, is included as part of adTempus and does not require an additional license purchase. When you choose to use this option, adTempus Setup will install, configure, and manage SQL Server Express for you: you will not need to interact directly with the database server. A backup of your adTempus data will be created each night.

Note that even if you already have SQL Server 2005 Express installed on the computer, adTempus will install a new SQL Server instance.

SQL Server 2005 Express requires approximately 270MB of disk space for the database program files.

This is the recommended database configuration because:

- It requires no database knowledge, configuration, or management by you, the adTempus user.
- It ensures that adTempus will always be able to connect to its database, because the database will be on the same computer as adTempus.
- It isolates adTempus from problems such as database performance issues or database server shut downs that may be caused by other applications or other business needs.

Using Your Own Database Server

If you already have a Microsoft SQL Server database server, you may choose to host the adTempus database on this server instead of using the SQL Server Express instance available with adTempus. At this time, adTempus supports only Microsoft SQL Server, and requires SQL Server 2000 or later.

See the Using a Standalone SQL Server Instance topic for more information about using your own SQL Server instance.

As noted above, we recommend using the SQL Server Express instance because it provides isolation and reliability for adTempus. If you choose

	to use your own SQL Server instance, you should keep in mind that adTempus requires continuous access to the database server. If SQL Server is shut down, or the network connection between computers is lost, this will cause problems in adTempus and may lead to missed or failed jobs.

See Also

Database Installation and Configuration

Database Installation and Configuration

The Database Installation and Configuration program is run at the end of adTempus installation.

If you need to change the adTempus database configuration after installation, run the "adtdbconfig" program from the "database" folder under the adTempus program folder (e.g., "c:\program files\arcana development\adtempus\database\adtdbconfig").

When this program starts, the Database Installation and Configuration wizard will appear. If this is the initial installation of adTempus, the following choices will be presented.

- Install SQL Server Express. As discussed in the Database Selection topic, this is the recommended option.
- Use an existing SQL Server instance

See Also

Database Selection

Using a Standalone SQL Server Instance

Using a Standalone SQL Server Instance

adTempus can use an existing SQL Server to host its database. SQL Server may be running on the same computer as adTempus, or on a remote computer. SQL Server 2000 and later are supported.

Information in this topic refers to the Database Installation and Configuration program, which is the program that configures SQL Server and adTempus. This program is run either by the adTempus setup program, or manually, at a later time.

Installation Requirements

The Database Installation and Configuration program must be run with sufficient authority to create databases and users on the SQL Server. When you run the program, therefore, you must either be logged in under a Windows account that has system administrator authority for SQL Server, or have a system administrator user ID and password that the tool can use.

SQL Server must be running.

Security Setup

If SQL Server is on the same computer as adTempus, you can generally use Integrated (Windows) security, which means that adTempus can connect to the database without an explicit user ID and password. This is possible because the adTempus service runs under the SYSTEM account, which is a member of the Administrators group on the computer, which by default has system administration authority for SQL Server.

If you have changed SQL Server security settings such that members of the computer's Administrators group do not automatically have system administration authority in SQL Server, you will need to either make the necessary changes to allow the SYSTEM account full control over the adTempus database, or enable SQL Server security and have adTempus use that.

If SQL Server is on a different computer from adTempus, integrated security cannot be used because the SYSTEM account that adTempus runs under has no authority on the remote computer. For adTempus to connect to SQL Server, you must use one of the following approaches:

- Enable SQL Server security on the server and allow adTempus to use that instead. This is the default approach used by the migration tool: if you specify a remote database server, the migration tool will assume you are using explicit security, and will create a SQL Server login for adTempus to use. If SQL Server has not been configured to allow explicit security, adTempus will fail to start.
- Change the adTempus service to run under a user account instead of the SYSTEM account.

Hosting Multiple adTempus Instances

If you are installing adTempus on several servers, you can use a single SQL Server instance to host the databases for all of the adTempus instances. However, each instance of adTempus must have its own separate database on the server.

The Database Installation and Configuration program handles this automatically, by including the name of the adTempus server in the database name. For example, you have installed adTempus on three servers: ServerA, ServerB, and ServerC. You will use the SQL Server on ServerX to host the databases for all three adTempus installations. When you run the database migration tool, it will create the databases "adtempus_ServerA", "adtempus_ServerB", and "adtempus_ServerC".

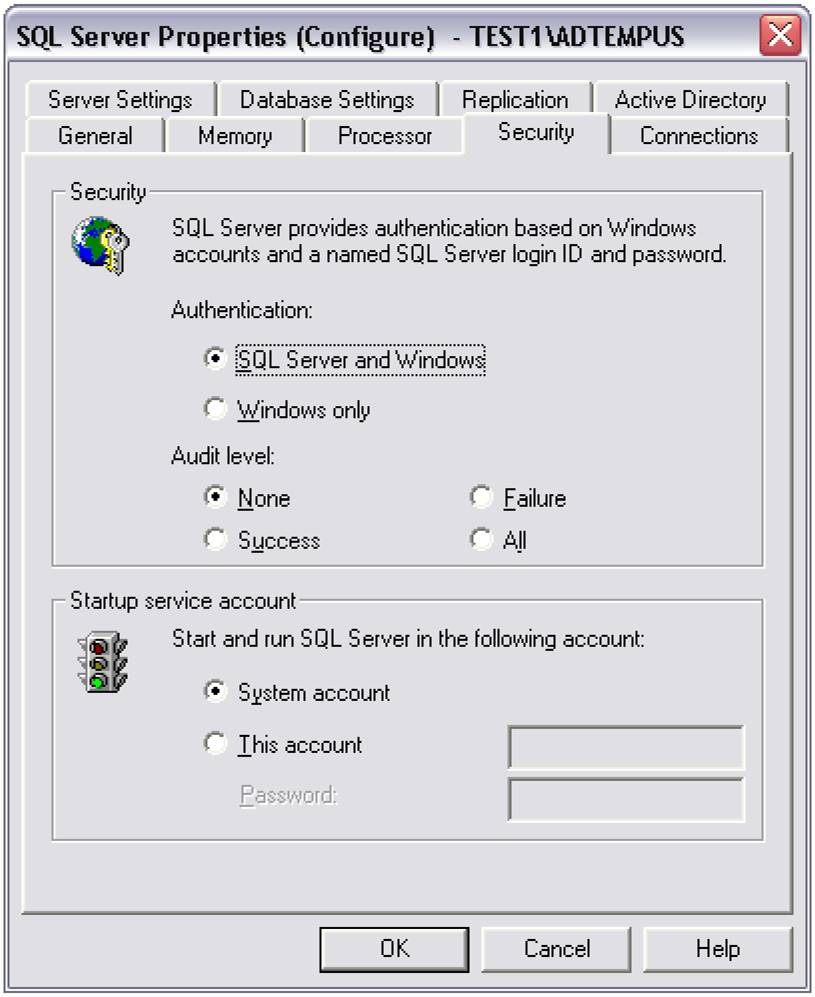
Database Backups

adTempus **does not** automatically back up the adTempus database when you are using SQL Server. You (or your SQL Server system administrator) are responsible for ensuring that the adTempus database is backed up in accordance with your backup policies.

Enabling SQL Server Security

Depending on your SQL Server configuration, you may need to enable SQL Server security on the SQL Server. In the SQL Server Enterprise Manager, display the properties for the server and go to the Security tab. Check the "SQL Server and Windows" option.

Important: Before making this change, be sure that you have provided a strong password for the sa user account.



SQL Server Properties (Configure) - TEST1\ADTEMPUS

Security

SQL Server provides authentication based on Windows accounts and a named SQL Server login ID and password.

Authentication:

● SQL Server and Windows

○ Windows only

Audit level:

● None ○ Failure

○ Success ○ All

Startup service account

Start and run SQL Server in the following account:

● System account

○ This account

Password:

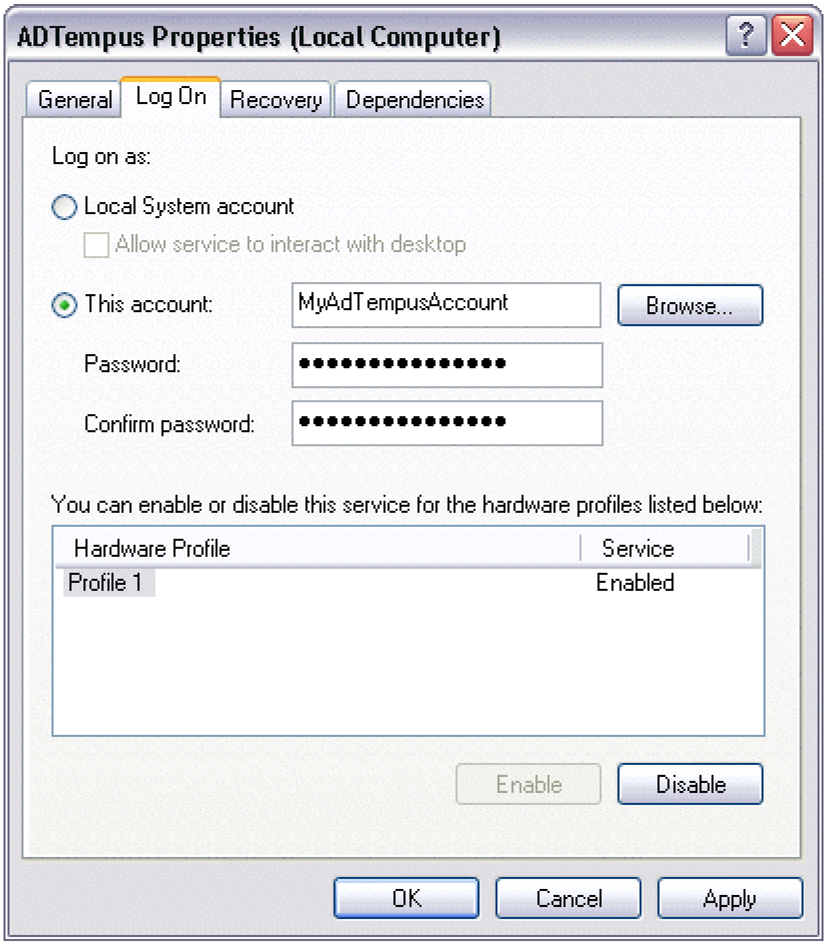
Using Integrated Security with a Remote Database Server

Integrated security for a remote server is not supported by the database migration tool: if you specify a remote database server, the migration tool will assume you are using explicit security, and will configure adTempus accordingly. If explicit security is not enabled on SQL Server, adTempus will fail to start because it will not be able to connect to the database.

After you have completed the rest of the migration process (ignoring the failure of the adTempus service to start), you must make the configuration changes described below.

Stop the adTempus service, if it is running, before proceeding.

Change the account under which the adTempus service runs



Log on as:

○ Local System account

□ Allow service to interact with desktop

● This account: MyAdTempusAccount

Password: •••••••••••••••

Confirm password: •••••••••••••••

You can enable or disable this service for the hardware profiles listed below:

Hardware Profile	Service
Profile 1	Enabled

In the Windows **Services** tool, display the properties for adTempus. On the **Log On** page, change the service to run under a user account.

The account you select must meet the following requirements:

- The account must have the necessary permissions for the adTempus database in SQL Server (must have full permission to read and write data).
- The account must be a member of the Administrators group on the computer where adTempus is running.
- The account must have the following rights:
  - Act as part of the operating system
  - Create a token object
  - Replace a process level token

To grant the rights, edit the "User Rights Assignment" in the Local Security Policy. If your computer is a member of a domain, or is a domain controller, you may also need to grant these rights in the domain or domain controller security policies.

Change the adTempus database configuration

On the computer where adTempus is installed, run the Registry Editor and go to the key **HKEY_LOCAL_MACHINE\Software\Arcana Development\adTempus\Database**. Then make the following changes:

- **Delete** the "UserID" and "Password" values under the key.
- Edit the "ConnectionString" value, which will look similar to this:

```
provider=SQLOLEDB;
server=myserver;Database=adtempus_test1
```

Add the following to the end of the value:

```
; Integrated Security=SSPI
```

The value will now appear similar to the following:

```
provider=SQLOLEDB;
server=myserver;Database=adtempus_test1; Integrated
Security=SSPI
```

Now start the adTempus service.

Obtaining Software Updates

If your computer is connected to the Internet, the adTempus Update Center will automatically keep you informed of product updates.

The Console will check for updates once a day, and display any new update messages when you start the Console. To check for new updates or review previously-seen updates at any time, select **Check for Updates** from the **Help** menu in the adTempus Console or select the **Check for Updates** shortcut in the adTempus group on the Windows Start menu.

If your computer is not connected to the Internet, you can go to www.arcanadev.com/adtempus from a computer that is connected to the Internet.

Uninstalling adTempus

adTempus can be uninstalled by using the **Add or Remove Programs** tool in the Windows Control Panel.

ℹ	You do not need to uninstall adTempus before you install a newer version: the setup program for the newer version will upgrade your existing installation.

License Deactivation

If you have activated the adTempus software on this computer using your license, the uninstall process will ask you if you want to "deactivate" the software and return the license to your license account. If you do not plan to reinstall the software on this computer, you should answer "Yes," to make the license available for use on another computer. If you will be reinstalling the software on this computer, you should not return your license.

Database Removal

Uninstalling adTempus will not remove the adTempus database engine (SQL Server 2005 Express), if installed. If you will be reinstalling adTempus on this computer, you should leave the database engine installed. If you will not be reinstalling adTempus, you should remove the database engine.

To remove SQL Server Express, select "Microsoft SQL Server 2005" in the Windows **Add or Remove Programs** tool and click **Remove**. In the Microsoft SQL Server 2005 Uninstall wizard, check the "Remove SQL Server 2005 instance components" box and select the "ADTEMPUS: Database Engine" instance, then click **Next** and continue with the removal process.

⚠	Be sure you select only the "ADTEMPUS: Database Engine" instance. If any other instances are listed, they were installed by other products and should not be removed.

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 Introduction to Key Concepts, 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.

First-Time Setup

If you have installed a new adTempus instance, there are a few initial settings that you will probably need to configure. You will need to be logged in to the computer under an account that is a member of the Administrators group to perform the initial setup (by default, only Administrators can connect to adTempus).

To begin, run the adTempus Console from the adTempus group on the Windows Start menu. The Console should automatically connect to the local adTempus server.

Security Settings

The default security settings for adTempus allow only members of the computer's Administrators group to connect to and administer adTempus. If only Administrators will be using adTempus, you do not need to make any changes to security settings.

If you want non-Administrators to be able to use adTempus, you must grant them the necessary permissions. See the Security Configuration Guidelines topic for more information.

Notification

If you plan to use e-mail or SMS notification (e.g., for sending messages about failed jobs), 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.

Security Configuration Guidelines

Security Configuration Guidelines

The adTempus security framework is integrated with the Windows security framework: security in adTempus is managed by granting permissions to Windows users and groups. Further, the security inheritance model in adTempus uses the same concepts as the security inheritance in the Windows file system. Therefore many principals for effective Windows security management can be applied to adTempus as well.

This following topics cover basic security setup:

Permission Management Guidelines

Initial Security Setup

Permission Management Guidelines

Security is easier to manage when it is kept simple. The best way to do this is to work with groups (of jobs and other objects, and users) rather than individuals (individual users, jobs, etc.).

User Groups

adTempus does not allow you to define groups of users, but it does recognize user groups defined in Windows. Whenever possible, grant permissions within adTempus to groups of Windows users, and then grant permissions to individual users indirectly, by assigning them to groups.

For example, you must specify which users have permission to connect to adTempus (using the Security Settings Page of the Server Options). Instead of adding individual users here, you should instead create an "adTempus Users" user group in Windows, and grant permission to that group. Then add the appropriate users to this group.

Within adTempus you may then need to further define which users have access to which jobs (and other objects). Again, use Windows user groups to simplify this. For example, at a basic level you may have one set of users who are allowed to view and monitor jobs but not to create or modify them, and a second set of users who are allowed to create and modify jobs. In Windows, create two more groups, such as "adTempus Read-Only Users" and "adTempus Update Users" and assign the appropriate people to the correct group. Make the two new groups members of the "adTempus Users" group you already created and now, through Windows security inheritance, the people you assign to "adTempus Read-Only users" also inherit the permission to connect to adTempus, without you needing to assign them that permission explicitly within adTempus.

Object Groups

Just as you should grant permissions to groups of users rather than individual users whenever possible, so you should also assign security access to groups of objects rather than individual objects. Practically speaking, you will most often be tailoring permissions to jobs, so following this principal you should try to set permissions based on Job Groups rather than the individual jobs. If a set of jobs have a related function (e.g., they are all related to one application on your server) it makes sense to put them in the same Job Group. By the same token, they probably have substantially the same security requirements as well. So rather than modifying the security settings for each one of these jobs, you should set the permissions for the Group that contains them instead. Then as new jobs are added to the group, they automatically inherit the permissions from the group. If a few jobs within the group have specialized security requirements, you can customize them either by modifying those jobs individually to override the inherited permissions, or by grouping them into a lower-level group and customizing the permissions for that group.

Initial Permission Setup

Members of the computer's Administrators group always have permission to connect to adTempus, and full control over all objects within adTempus. If adTempus is only going to be used by Administrators, you do not need to modify any security settings. However, if non-Administrators will be using the software, you will need to grant permissions as discussed below.

General Access to adTempus

Before a user can do anything within adTempus, that user must have permission to connect to adTempus, and this permission is assigned independently of all other permissions. For example, you can grant user "Bob" full control over all jobs within adTempus, but if you haven't granted Bob "Connect" permission, he will not be able to connect to adTempus using the Console. Similarly, if you give Bob "Connect" permission but he does not have permissions for any objects within adTempus, he will be able to connect with the Console but will not be able to do anything once he gets there.

As discussed above, we recommend that you create a Windows security group, assign the appropriate users to that group, and then grant permissions to that group within adTempus.

To grant Connect permissions

1. Go to the Server Options window (**Tools > Server Options > General Options**) and select the Security Settings page.
2. Click the **Add** button and add the appropriate Windows security group (or user).
3. Select the group or user you added in Step 2.

4. Check the "Allow" box for the "Connect to Server" permission.
5. Click OK to save the changes (or continue granting additional permissions as discussed below).

General Object Permissions

The change made in the previous section will allow the designated user(s) to connect to adTempus, but they still will not be able to do anything as they do not have permissions for any objects within adTempus.

A typical basic security model allows all adTempus users to view all objects and to create new objects, and full control over objects that they have created. To implement this model:

1. Go to the Server Options window (**Tools > Server Options > General Options**) and select the Security Settings page.
2. Select the user group you added in the previous section.
3. Make sure the **Apply permissions to** option is set to "Server Settings, all adTempus objects."
4. Check the "Allow" box for the following permissions:
    - Connect to the server
    - List/Use
    - Associate
    - View
    - Create
5. Click the **Add** button to add another user.
6. In the **Add User or Group** window, type the name "creator owner" and click **OK**.
7. Make sure the **Apply permissions to** option is set to "Objects only".
8. Check the "Allow" box for the "Full Control" permission.
9. Click **OK** to save the settings.

Object Permission Refinements

Once you have set a general security policy as outlined above, you can easily refine the security settings for different kinds of objects.

For example, you may want to prevent users from creating new Job Queues. To do this:

1. Right-click the **Queues** node and select the **Queue Security** command.
2. In the Security Properties window, you will see an inherited entry for the user or group you added in the previous section. This inherited entry cannot be modified.
3. Click the **Add** button and add the same user or group.
4. You will now see a new, non-inherited entry for that user or group. Select it and make sure the **Apply Permissions To** option is set to "Queue Only."
5. Check the "Deny" box for the "Create New Queues" permission.
6. Click **OK** to save your changes.

You have now overridden the permissions that the Queues node inherited from the server, denying users the right to create new Queues. If some non-Administrative users need to be able to create Queues, you can further modify the permission list to grant them that permission.

Create a Basic Job

This example shows how to create a simple job to run the "Notepad" program from adTempus.

To create the job:

1. Select the Job Group that you want to create the job in. Right-click the group's name and select **New job** from the pop-up menu. (If you don't have any Groups defined, or want to create the job in the root group, right-click the **Jobs** node.)
2. The Job Properties window will open. Though adTempus offers many options, a simple job requires only a few things.
3. Enter a **Name**. Give the job a unique, descriptive name, such as "Run Notepad."
4. Enter the **User Account**. Type the user ID for the Windows user account the job should run under. If the account is a domain account, include the domain, for example "mydomain\myuserid". When you leave the User Account box, you will be prompted to enter the password for this account. For this example, use your own user ID.
5. Click to the Steps page.
6. Click **Add...** and select the "Execute a program, batch file, etc." task type. This creates a Program Execution Task, which runs a program or batch file and is the most commonly-used kind of task.
7. The Program Execution Task properties window will open.
8. Next to the **Target** box, click the "**...**" button to browse for the file to execute. Browse to the Windows directory (usually "c:\windows") and select "notepad.exe".
9. Click **OK** to save the task properties and return to the job properties.
10. Click **OK** to save the job.
11. The new job will now appear in the folder.

Because this job does not have any Triggers assigned to it, adTempus will run it only if you request it. To run the job:

1. Right-click the job's name and select the **Run** command. The Execute Job window will be shown.
2. Click **OK** to start the job, then click **OK** to acknowledge the informational message.
3. The job's status in the Console will be updated to show that it is running, and the Notepad program will open on your desktop.

ℹ This example assumes you are running the adTempus Console on the same computer where the adTempus server is installed. If you are not, or if you are connected to that computer through a Remote Desktop or Terminal Services connection, you may not be able to see the Notepad program running on your desktop. In this case you will need to terminate the job by right-clicking it and selecting the **Terminate** command.

4. Close Notepad. The job's status will be updated to show that the job has finished.

Next Steps

Most jobs will have a Trigger so that adTempus will run them automatically. The most common trigger is the Schedule Trigger, which runs the job at specified days and times. We'll add a simple Schedule Trigger to the job you just created.

1. Double-click the job to display its properties, then click to the Triggers page.
2. Click the **Add...** button and select "Execute according to a schedule." The Execution Schedule Properties window will open.
3. Keep the default settings on the **Date Selection** page. These settings will cause the job to run every day.
4. Click to the **Time Selection** page.
5. Select the **Trigger at these times** option.
6. Below the list, type in a time when the job should run. For this example, enter a time that is two or three minutes in the future.
7. Click the "**+**" to add the time to the list.
8. Click **OK** to save the schedule properties. The Schedule Trigger properties window now appears, where you could set additional trigger options.
9. Click **OK** to return to the job.
10. Click **OK** again to save the job and return to the Console.
11. After a moment you will see a **Next Start** time for your job.
12. When the scheduled time arrives, the job will start.

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.

Security

The adTempus security framework determines who is able to manage a job within adTempus, but these settings do not affect what the job can do at run-time. Instead, for each job you must specify the Windows user account that the job will use. When adTempus runs the job it logs in the specified user and runs the job in that user's security context.

This means that any programs, scripts, or other tasks run by the job will have the same security limitations as if that user were running the program directly.

See Also

Introduction to Key Concepts

Job Properties

Tasks

Notification Task

The Notification Task allows you to send notification messages, and optionally capture and attach files. The Notification Task provides the same functionality as the Notification Action and File Capture Action. Those actions can only be run as Responses for a job or a step, but the Notification Task, because it is a task, can be the primary function that a job or step performs.

Generally, you would use a Notification or File Capture Action when you want to send a notification message in response to something that occurs during job execution (i.e., the notification is of secondary importance). You would use Notification Task when the notification is of primary importance.

For example, if you wanted to monitor a computer and send a notification message when it failed, you would create a job that uses a Computer Monitor Trigger and a Notification Task.

See Also

Notification Task Properties

Notification Action

Tasks Overview

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.

See Also

Process Termination Task Properties

Tasks Overview

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.

Though the Program Execution Task can be used to run scripts as well, the Script Execution Task is tailored to script execution and is often a better choice.

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.

See Also

Program Execution Task Properties

Script Execution Task

Tasks Overview

Steps Overview

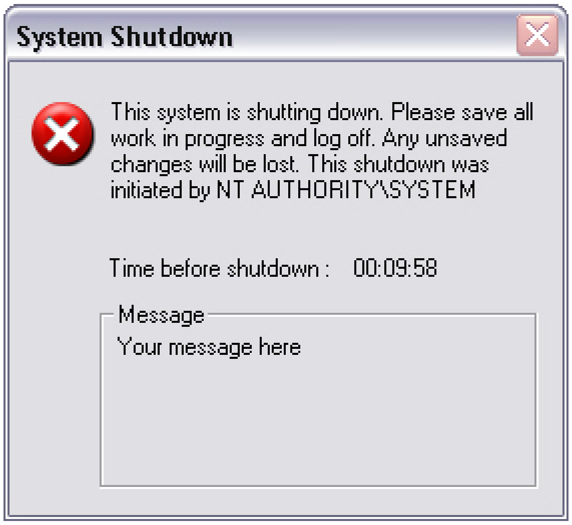
Network Access for Scheduled Tasks

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.



This system is shutting down. Please save all work in progress and log off. Any unsaved changes will be lost. This shutdown was initiated by NT AUTHORITY\SYSTEM

Time before shutdown : 00:09:58

Message

Your message here

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.

See Also

Computer Shutdown Task Properties

Tasks Overview

Script Execution Task

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

Although external script files can also 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.

See Also

Script Execution Task Properties

Scripts Overview

Tasks Overview

Steps Overview

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.

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.

See Also

Service Control Task Properties

Tasks Overview

Steps Overview

SQL Server Job Task

The SQL Server Job task allows you to execute a SQL Server Agent job (a job defined within SQL Server) from adTempus.

This can be useful if you need to incorporate a SQL Server Agent job into a processing flow that is otherwise managed by adTempus. For example, you may want adTempus to execute a sequence like this:

1. Run a program that loads data into a SQL Server database.
2. Start a SQL Server job that processes the new data, and wait for that job to complete.
3. Run a program that produces reports based on the new data.

The SQL Server Job Task allows you to do this without the need to write your own program or database script to start and monitor the SQL Server job.

Responses can be executed based on the results of the SQL Server job.

See Also

SQL Server Job Task Properties

Web Request Task

The Web Request task allows adTempus to request a page or file from a Web server.

This task can be used to simply "ping" a URL on the server (for example, if the Web application has offline processing that is triggered by requesting a special URL on the server) or to download a page or file from the Web server.

Pages and files retrieved by the task can be saved in the job's history and/or written to a specified location on the adTempus server.

The Web Request Task is used to request a URL as part of job processing. If you want to trigger a job based on the response (or non-response) of a Web server, use the Computer Monitor Trigger instead.

See Also

Web Request Task Properties

Computer Monitor Trigger

Triggers

Computer Monitor Trigger

The Computer Monitor Trigger allows adTempus to monitor remote computers and trigger a job when the remote computer fails.

The Computer Monitor can use a simple network ping to verify that the computer is alive and reachable, or it can make an HTTP request to the server and evaluate the result. This feature can be used, for example, to verify that your Web application is running properly.

If you need to request and save a Web page or file as part of your job's processing after the job is triggered by some other mechanism, use the Web Request Task instead.

See Also

Computer Monitor Properties

Using adTempus to Monitor Your Web Application

Web Request Task

Triggers Overview

Event Log Monitor Trigger

The Event Log Monitor Trigger allows adTempus to monitor the Event Log on the host computer and trigger jobs when selected events are logged.

For example, you could have a job that is run whenever a failed logon attempt is reported by the operating system, or a job that is run whenever SQL Server reports a replication error.

Each time adTempus is started, it reviews any events that were logged while it was stopped, and triggers based on those events as appropriate. It then begins watching for new events and triggering based on them.

See Also

Event Log Trigger Properties

Triggers Overview

File Trigger

A File Trigger is a trigger that causes a job to be executed whenever specified files are created, modified, or deleted.

When you create a File Trigger, you specify the files or directories that adTempus should watch for, and under which conditions the job should be triggered.

The File Trigger works by periodically (approximately every 10 seconds) scanning the directories being watched by the trigger and comparing the current state to the previous state.

Multiple Files and Job Instances

Each time the File Trigger scans the directories, it produces a list of all the files that meet the trigger criteria. Because adTempus makes this evaluation at intervals of several seconds, it is possible for more than one file to meet the trigger criteria. For example, suppose you have a trigger configured to fire any time files are added to a particular directory. In between evaluations of the directory, five new files are copied into the directory. On the next evaluation, adTempus will find those five files, and the trigger's **Start a separate instance for each file** setting determines how it will handle the scenario.

Separate Instance for Each File

If the **Start a separate instance for each file** option is checked, adTempus will start a separate instance of the job for each of the matching files it has found, and each instance will have the FileName job variable set to the name of the file for which it is being executed.

Typically you would use this option if your program is being called to process files individually. For example, you need to run a data processing utility on each file that is uploaded into a directory. When you configure the Program Execution Task to run your program, you can include the token "%FileName%" in the command-line parameters, and adTempus will replace "%FileName%" with the name of the triggering file.

Single Instance for All Files

If the **Start a separate instance for each file** option is *not* checked, adTempus will start a single instance of the job, even if more than one file has been found. The FileName job variable will contain the names of all the files found on the current evaluation pass.

Typically you would use this option if your program is being called to process an entire directory. For example, your program already has logic to scan an entire folder and process all the files in it; you only need adTempus to start it when there are files there to process.

In this scenario, you need to make sure you have configured the job so that it will not be triggered too often. For example, set the job's Multiple Instances rule to prevent duplicate instances of the job from being started at the same time. Otherwise, adTempus may trigger your program again while it is still processing files from the first trigger.

Changes Made While adTempus is Not Running

adTempus saves the current state of the directories it is monitoring each time it makes an evaluation pass. This state information is stored in the adTempus database, and so is not lost if the adTempus service is shut down. Therefore when adTempus starts, it compares the current file state to the state information it saved before it shut down, and triggers jobs based on the differences. That is, adTempus will respond to files that were added, changed, or deleted while adTempus was not running.

Potential Problems with Network Drives

If your file trigger is watching files on a network drive, you may encounter problems if the network connection is broken. Because of the way adTempus scans for files, if the network connection is broken, any files on the network drive will not be visible to adTempus, and it will respond as if they had been deleted, triggering the job if your trigger is set to fire on deletions. Similarly, when the connection is restored and the files suddenly reappear, adTempus will see them as new files (because they were not there on its previous evaluation pass). adTempus will therefore trigger the job if the trigger is set to fire when new files are created.

See Also

File Trigger Properties

Triggers Overview

File Condition

Process Trigger

The Process Trigger allows you to trigger a job whenever external processes start, end, or exceed a specified memory limit.

For example, you may have a server process that occasionally fails. Using a Process Trigger, you can have adTempus monitor this process and run a job (to restart the process, send a notification message, etc.) when it fails.

See Also

Process Trigger Properties

Triggers 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.

If a Schedule Trigger has more than one schedule, the job will be executed whenever any of the schedules is satisfied.

Multiple Schedules

A job can have more than one Schedule Trigger, and each Schedule Trigger can have any number of Schedules. The job will execute whenever any of the schedules on any of the Schedule Triggers is satisfied. However, if any Schedules or Schedule Triggers produce overlapping times, adTempus will still only run the job once. For example, even if two separate Schedules specify that the job should run at 12:30 PM on Monday, the job will only run once at that time.

For example, you can have one schedule that defines the run times for weekdays, and a second schedule for weekends.

You might want to have separate Schedule Triggers (instead of multiple Schedules on a single Schedule Trigger) if you want to use different time zone or holiday options for different schedules: all Schedules for a Schedule Trigger share the same options and holidays, but each separate Schedule Trigger for a job has its own options and holidays.

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

See Also

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.

See Also

Startup Trigger Properties

Triggers Overview

WMI Trigger

The WMI Trigger allows adTempus to receive Windows Management Instrumentation (WMI) events and trigger jobs based on them.

See Also

WMI Trigger Properties

Microsoft Platform SDK documentation for WMI

Sample WMI Queries

Triggers Overview

Conditions

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.

⚠	A File Condition will not cause a job to start executing: it will only cause a job to pause execution to wait for files. To start a job based on files, use a File Trigger instead.

See Also

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.

⚠	A Job Condition will not cause a job to start executing: it will only cause a job to pause execution to wait for other jobs. Some other mechanism (a Trigger or a Response on another job) must start the job. If you want one job to cause another to start when it finishes, you probably want to use a Job Control Action instead of a Job Condition. See the Comparison of Triggers and Conditions topic for more information about the differences.

See Also

Job Condition Properties

Comparison of Triggers and Conditions

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.

See Also

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.

See Also

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 sequentially, one at a time. 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.

See Also

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.

See Also

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

See Also

Job Control Action Properties

Responses Overview

Actions

Notification Action

The Notification action is used to send notification messages (by e-mail, network broadcast, SMS, etc.) regarding the status of a job. To send a notification message, add a Response to a Job or Job Step.

You can also send notification messages without using Responses by using a Notification Task as a step in your job.

See Also

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.

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, your adTempus job is going to be patiently waiting.

See Also

Script Action Properties

Responses Overview

Actions

Using Scripts

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.
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."
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.

Scripts

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.

See Also

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 Shared 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.

Script Library Code and Namespaces

The various scripting languages all handle namespaces slightly differently, and this may affect the way you are able to define and call code in your script Library.

WSH Scripting Languages

The Windows Scripting Host (WSH) languages (VBScript, JScript, and other WSH-compatible languages) do not support multiple namespaces. As a result, 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.

VB.NET

All script and library code used by adTempus is placed in the global namespace. Code for each Script Library must be contained by a public Module (by default the Script Library Editor will set the module name to match the library name, but this is not required).

Therefore you may have variables and procedures in different libraries that have the same name, but you may need to qualify a the variable or procedure name using the library name when you use it in a script.

For example, you have created a Script Library named "MyLibrary", and the module defined in it is also named "MyLibrary." Within that module you have a public function named "MyFunction."

When you reference this library from a script, you can call your function using either "MyLibrary.MyFunction" or "MyFunction".

However, if you create a second library that also contains a function named "MyFunction", and both libraries are referenced by your script, "MyFunction" by itself is ambiguous, and you will need to call "MyLibrary.MyFunction".

C#

All scripts and libraries by are created to use the "UserScript" namespace. You can change the namespace if you wish, and make references across namespaces as necessary. The "adTempus" object used for interaction with adTempus is always in the "UserScript" namespace.

See Also

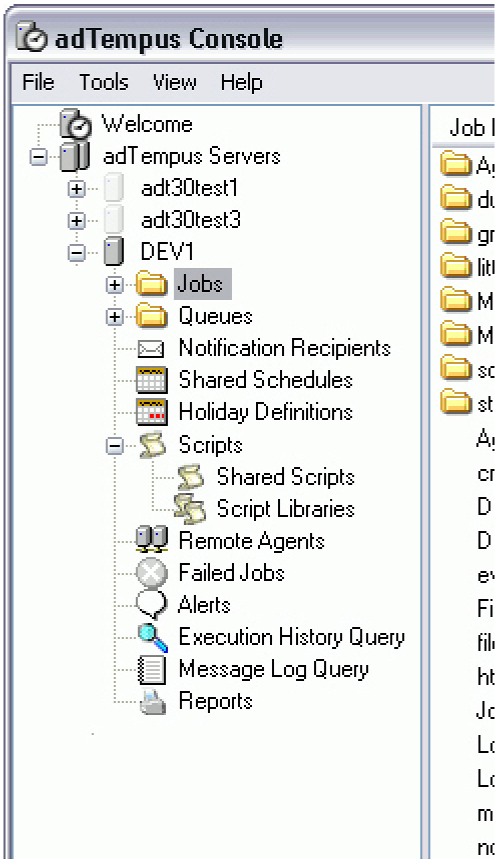
Script Library Properties

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 (or "folder") 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.

Within the Console all dates and times are by default 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. If you prefer, you can have dates and times shown in your local time zone instead using the settings available in the Console Options window.

The active view is automatically refreshed at the interval you specify in the Console Options. You can refresh the Console at any time by pressing **F5** or selecting **Refresh** from the **View** menu.

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

The icons for some of the nodes in the Console Tree change based on the status of the jobs or other items they contain. For example, Job Groups that contain jobs that have recently failed will display an error symbol over the folder.

View Reference

Welcome Page

The Welcome page displays news, announcements, and other information about adTempus. The contents of this page are updated from the Web each time the Console starts.

Server Folders

The **adTempus Servers** folder contains a separate node for each adTempus server the Console is connected to.

How To...

Connect to a new server

Choose **Connect Server...** from the **File** menu

--or--

Right-click the **adTempus Servers** folder in the Console Tree and select **Add Server...**.

Jobs Folder

The **Jobs** folder is the primary administrative view for adTempus. If you have defined Job Groups, they will appear as folders beneath the Jobs folder.

Click the Jobs folder or one of the folders beneath it displays the Jobs View, which is divided into two panes:

- The Job List (top pane)
- Job Details (bottom pane)

Job List

The top pane 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. See the Job Status Descriptions topic for a list of possible statuses.

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.

When you select a group from the tree at the left of the Console, only the jobs and groups in that group are displayed in the Job List. You can move a group or job to a different group by dragging it to the destination group's folder in the tree.

Items in the Job List may display special icons (▶⊗⚠📁📁📁) to draw your attention to their status.

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 instances (executions) of the job. For information about older instances, or to search for instances based on various criteria, 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.

Items in the History List may display special icons to draw your attention to their status.

If an instance completed with a failure or warning status, the instance will be marked with an error (⊗) or warning (⚠) icon until the instance is acknowledged by an authorized user.

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 or to search for messages based on various criteria, use the Message Log Query.

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.

The **Clear job history** link opens the Clear Job History window, which allows you to purge the history for the job and optionally reset the statistics. History information is also purged automatically based on the job's history retention settings.

How To...

Create a new Job

1. In the Console Tree, select the Group you want to create the job in.
2. Right-click the group's name and select **New Job...** from the pop-up menu.

   --or--

   Right-click In the Job List pane, and select **New Job...** from the pop-up menu.

Move a Job or Group to a different Group

1. If necessary, expand the Console Tree so that the target group is visible in the Console Tree.
2. In the Console Tree, select the group that contains the jobs or groups you want to move.
3. In the Job List Pane, select the jobs or groups you want to move.
4. Drag the selected jobs or groups to the target group.

Create a new Group

1. In the Console Tree, select the Group you want to create the group in.
2. Right-click the group's name and select **New Group...** from the pop-up menu.

   --or--

   Right-click In the Job List pane, and select **New Group...** from the pop-up menu.

View or modify the properties of a Group

1. In the Console Tree, select the group whose properties you want to view or modify.
2. Right-click the group's name and select **Edit...** or **View...** from the pop-up menu.

Delete a Group

1. In the Console Tree, select the group in the Job List Pane or in the Console Tree.
2. Right-click the group's name and select **Delete** from the pop-up menu.
3. If the group (or any of the groups below the group) contains jobs, you will be asked whether you want to delete the jobs as well, or just to delete group and move the jobs to the Root group.

Hold or release all jobs in a Group

1. In the Console Tree or the Job List, select the group you want to hold or release.
2. Right-click the group and select **Hold all jobs in group** or **Release all jobs in group**.

Note: This procedure holds or releases the individual jobs in the group at the time you perform the action, not the group itself (groups cannot be held). It affects only jobs that you have permission to hold/release.

See Also

Job Properties

Job Group Properties

Queues Folder

The **Queues** folder contains all Queues that have been defined for the server.

Selecting a Queue displays all jobs that are assigned to that queue. The view is identical to the main Jobs View,except that it does not show any job groups (all jobs assigned to the queue are listed, regardless of what group they are in).

You can move a job to a different queue by dragging it to the destination queue in the tree.

How To...

Create a new Queue

Right click the **Queues** folder in the Console Tree and select the **New Queue...** option.

View or modify a Queue's properties

Right-click the queue's name in the Console Tree and select **Edit...** or **View...** from the pop-up menu.

Delete a Queue

1. Right-click the queue's name in the Console Tree and select **Delete** from the pop-up menu.
2. If the queue contains jobs, you will be asked whether to delete the jobs as well, or to delete only the queue and move the jobs to the Default queue.

Note: You cannot delete a queue if it contains jobs that you do not have permission to modify.

Create a new Job

1. In the Console Tree, select the queue you want to create the job in.
2. Right-click the queue's name and select **New Job...** from the pop-up menu.

   --or--

   Right-click In the Job List pane, and select **New Job...** from the pop-up menu.

Note: This will assign the new job to the Root job group. You can move the job to a new group after you finish editing it by dragging the job from the Root group to the desired group.

To create a new job directly in the desired group,

1. In the Console Tree, select the Group you want to create the job in.
2. Right-click the group's name and select **New Job...** from the pop-up menu.

   --or--

   Right-click In the Job List pane, and select **New Job...** from the pop-up menu.
3. You can select the Queue for the job in the job properties.

Hold or release a Queue

Holding a queue prevents jobs assigned to the queue from running. Releasing a queue allows jobs to begin running again.

To hold or release a queue,

1. Select the queue in the Console Tree.
2. Right-click the queue's name and select **Hold** or **Release** from the pop-up menu.

Holding or releasing a queue affects the queue itself rather than individual jobs. The jobs themselves may also be individually held: a job will not run if either the job or the queue is held.

Move a Job to a new Queue

1. In the Console Tree, select the queue that contains the job.
2. Drag the job to the desired queue.

   --*or*--

   Edit the job and select the new queue in the job's properties.

See Also

Jobs View

Job Queues

Notification Recipients Folder

The Notification Recipients view lists all Notification Recipients and Notification Groups that you have permission to view. From these views you can add, modify, and delete recipients and groups.

How To...

Create a new Notification Recipient or Notification Group

1. Right-click the **Notification Recipients** folder in the Console Tree

   --*or*--

Select the **Notification Recipients** folder in the Console Tree and then right-click in the Notification Recipients list.
2. Select **New** from the pop-up menu, then select **Individual** or **Group**.

Delete a notification Recipient or Notification Group

1. Right-click the name of the Recipient or Group in the **Notification Recipients** folder and select **Delete** from the pop-up menu.
2. If the Recipient or Group is being used in adTempus, you will be asked to confirm that you want to remove all references to the Recipient or Group.

See Also

Notification Setup

Notification Recipient Security

Shared Schedules

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.

See Also

Shared Schedule Overview

Shared Schedule Properties

Using Shared Schedules

Shared Schedule Security

Holiday Definitions

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.

See Also

Holiday Set Overview

Holiday Set Properties

Using Holidays

Scripts

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.

Shared Scripts are used when you want to be able to use the same script in many places in adTempus, without having to maintain multiple copies of the same code. A Shared Script can be referenced anywhere in adTempus that a script can be used.

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

See Also

Script Properties

Script Security

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.

See Also

Script Library Overview

Script Security

Remote Agents

This folder only appears if the selected server is configured as a Distributed Scheduling Master computer.

The Remote Agents View lists all Remote Agents that have been defined, and shows the current status for each.

See Also

Remote Agent Properties

Distributed Scheduling Overview

Remote Agent Security

Failed Jobs

The Failed Jobs view is similar to the Execution History query except that it only retrieves instances that have failed (or completed with a warning status) and that have not yet been acknowledged.

You may choose whether to include instances that completed with warnings in this view. When this option is checked, the list will include instances that were abandoned, aborted, or killed.

If instances are found that meet the criteria, adTempus puts a failure (⊗) or warning (⚠) icon next to this view in the Console Tree to draw your attention to the view. This view is refreshed regularly as long as the server is connected, even if the view is not active. If no failed instances are present, the view's icon will appear dimmed (⊗).

When you click the **Acknowledge All** button, adTempus marks all instances as "Acknowledged," and they no longer appear in the Failed Jobs View for you or for any other user. To acknowledge individual instances, select them in the list, then right-click and choose the **Acknowledge Failure** command.

See Also

Job Instance Properties

Execution History Query

Alerts

Alerts are messages from the adTempus server about adTempus operation. Alerts are generally issued when adTempus needs to inform you of a problem that is not related to a specific job (for example, if there is a problem with the adTempus database).

If unacknowledged alerts are listed, adTempus puts an informational (ℹ), warning (⚠), or error (⊗) icon next to this view the Console Tree to draw your attention to it. If no alerts are displayed, the view's icon will be an empty balloon (🗨).

You may choose whether to include informational alerts in this view. If this option is not checked, only warning and error messages will be displayed.

When you click the **Acknowledge All** button, adTempus marks all Alerts in the view as "Acknowledged," and they no longer appear in the Alerts view for you or for any other user. To acknowledge individual Alerts, select them in the list, then right-click and choose the **Acknowledge** command.

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 (executions) that match your specifications.

If you check the **Refresh Automatically** box adTempus will refresh the query automatically each time it refreshes data from the server. If you do not check this option, the query is only executed when you click the **Search** button.

The Job History Report provides the same information in a printable format, and also provides counts of the number of instances in each status.

See Also

Job Instance Properties

Failed Jobs View

Job History Report

Reports

Reports

adTempus includes a report execution and viewing facility, allowing you to produce reports on job settings, history, statistics, etc.

Included Reports

adTempus includes the following reports:

- The Job Inventory Report lists job settings and dependencies.
- The Execution Schedule Report lists the scheduled execution times for a job.
- The Job Accounting Report provides statistics on executions and CPU time usage.

Create Your Own Reports

adTempus uses Crystal Reports for reporting. If you have access to Crystal Reports, you can create your own reports, or adapt ours to suit your needs.

More information for report developers

Request a Report

If you need a report and can't create your own, contact us; we can probably create it for you.

Job History Report

The **Job History Report** provides data similar to that shown in the Execution History Query.

The report lists all instances of selected jobs that ran within a specified date range, optionally filtered by status.

> This report is built from the job's execution history. The history retention settings for the job therefore must be set to retain the history long enough for you to report on it. For example, if you plan to run the Job History Report at the end of each month, you must make sure that the job's history will be saved for at least one month.
>
> If no history records exist for part of the date range you have specified for your report, the report will be incomplete.

Information on the report can be grouped as follows:

- **No grouping.** The report lists all instances in chronological order.
- **Status**. The report groups the instances by their status. Within each status, instances are listed in chronological order. The report displays the total number of instances for each status.
- **Status and Job.** The report groups the instances by their status. Within each status, instances are grouped by job, then listed in chronological order. The report displays the total number of instances for each status and for each job within a status.
- **Job**. The report groups the instances by job. Within each job, instances are listed in chronological order. The report displays the total number of instances for each job.
- **Job and Status.** The report groups the instances by job. Within each job, instances are grouped by status, then listed in chronological order. The report displays the total number of instances for each job and for each status within a job.

Additional Information

This report uses a direct database connection, which may require additional configuration. See the Direct Database Connection topic for more information.

See Also

Execution History Query

Job Accounting Report

Job Inventory Report

The Job Inventory Report produces an inventory listing of the selected jobs, showing basic settings and dependencies for each job.

Job Execution Schedule Report

The Job Execution Schedule Report produces a listing of all the scheduled execution times for the selected jobs. Every scheduled execution for the specified date range is listed.

The Start Date/Time and End Date/Time should be specified using the time zone of the server.

Note that this report may take a long time to produce, and may be quite large, if you have a large number of reports and/or have specified a wide range of dates.

Job Accounting Report

The Job Accounting Report provides information on job run times and CPU time usage for selected jobs over a selected range of dates.

The main report shows summary information for each job. By double-clicking a job you can drill down to a list of instances for that job, with statistics for each instance. Finally, you can double-click an instance to drill down to the step-level details for that instance.

> This report is built from the job's execution history. The history retention settings for the job therefore must be set to retain the history long enough for you to report on it. For example, if you plan to run the Job Accounting Report at the end of each month, you must make sure that the job's history will be saved for at least one month.
>
> If no history records exist for part of the date range you have specified for your report, the report will be incomplete.

Additional Information

This report uses a direct database connection, which may require additional configuration. See the Direct Database Connection topic for more information.

See Also

Job History Report

Creating Your Own Reports

The report viewer built in to adTempus can read any report produced by Crystal Reports.

The framework for incorporating custom reports into the adTempus Console has changed for adTempus 3.0, and documentation is not yet available. Please check the adTempus support site for the latest information.

Window Reference

Clear Job History Window

Clear Job History Window

The Clear Job History window (available through the **Clear job history** link on the **Statistics** page in the Job List) allows you to clear the history (list of previous instances and log messages) for a job immediately, instead of allowing the history to be cleared automatically based on the job's retention settings (specified in the job's properties).

When you click **OK**, the history records will be purged immediately and permanently. If you check the **Also reset statistics for the job** box, adTempus will reset the execution statistics (total executions, total execution time, average execution time, total processor time, and average processor time) to zero. If you do not check this option, the statistics will not be affected.

After the history is cleared, adTempus records an audit message in the job's log, indicating the identity of the user who cleared the history.

Computer Monitor Trigger

Computer Monitor Trigger

The Computer Monitor Trigger allows adTempus to monitor remote computers and trigger a job when the remote computer fails.

The Computer Monitor can use a simple network ping to verify that the computer is alive and reachable, or it can make an HTTP request to the server and evaluate the result. This feature can be used, for example, to verify that your Web application is running properly.

If you need to request and save a Web page or file as part of your job's processing after the job is triggered by some other mechanism, use the Web Request Task instead.

See Also

Computer Monitor Properties

Using adTempus to Monitor Your Web Application

Web Request Task

Triggers Overview

Computer Monitor Trigger

The Computer Monitor Trigger allows adTempus to monitor remote computers and trigger a job when the remote computer fails.

The Computer Monitor can use a simple network ping to verify that the computer is alive and reachable, or it can make an HTTP request to the server and evaluate the result. This feature can be used, for example, to verify that your Web application is running properly.

If you need to request and save a Web page or file as part of your job's processing after the job is triggered by some other mechanism, use the Web Request Task instead.

See Also

Computer Monitor Properties

Using adTempus to Monitor Your Web Application

Web Request Task

Triggers Overview

Using adTempus to Monitor Your Web Application

A simple ping can tell you whether a remote computer is running and reachable, but this often does not provide you with enough information about the computer.

Just because the computer is "alive", for example, doesn't mean that the Web server software is running properly.

adTempus provides additional monitoring flexibility through its ability to make an HTTP request to a remote computer, and then evaluate the response that the computer returns.

A Simple Test

In a simple situation, you may only want to verify that Internet Information Server is running and serving pages properly. To do this, you would create a simple ASP page on the Web server, called, for example, "testapp.asp". It might look like this:

```
<%@ Language=VBScript%>
```

```
Application Status: OK
```

You would then configure the Computer Monitor to request this page, and to trigger if the response did not match "Application Status: OK".

Note that we did not format the output from this ASP page as a properly-formed HTML page. Since it's not intended for a browser, there's no need to do this. In fact, the HTML tags would make the error message harder to read if you include it in notification messages.

Doing More

The ASP page or other scripted page you have adTempus request can do as much processing as you need and then return an appropriate response. For example, it could verify that your web server is able to connect to your web server, check that disk space is sufficient, etc.

Using the Results

The computer monitor sets two Job Variables that you can use when assessing the state of the target computer.

The **ConnectionStatusCode** parameter indicates whether adTempus actually received a response from the computer.

- If the **ConnectionStatusCode** is 0, adTempus was able to retrieve the requested page, and the **ResponseText** parameter contains the complete text returned by the server.
- If the **ConnectionStatusCode** is not 0, this indicates a network or server error that prevented adTempus from receiving a reply. For example, the target computer may be unreachable due to a network problem. In this case, the **ResponseText** parameter contains an error message describing the problem.

Using the **ResponseText** Script Parameter, you can execute certain steps of your job depending on the nature of the error returned, and include the error message from the server in notification messages that get sent by adTempus.

See Also

Sample script for customizing the response to a failure

Properties

General Page

Optional Descriptive Name

Optionally, provide a descriptive name for this trigger. Otherwise, a description will be created automatically based on the target computer name.

Enable this Trigger

If this box is checked, the trigger is enabled and will cause the job to run. If this box is not checked, adTempus will ignore this trigger.

Identity

Specify the credentials that will be used for the connection to the target computer.

- If you are using a simple ping test, or requesting a page from a Web site that does not require authentication, check the **The connection can be made without authentication...** option.
- If you are requesting a page from a Web site that requires authentication,
    - Check the **The connection can be made without authentication...** option if the Web server accepts Windows authentication and the user whose account is being used to run the job has permission to request the target Web page.
    - Otherwise, you need to specify a user ID and password for adTempus to use when connecting to the target server. Enter the user ID to select an existing Credential Profile or to create a new one.

Description / Notes

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

See Also

Computer Monitor Overview

Target Page

Optional Descriptive Name

Optionally, provide a descriptive name for this trigger. Otherwise, a description will be created automatically based on the target computer name.

Enable this Trigger

If this box is checked, the trigger is enabled and will cause the job to run. If this box is not checked, adTempus will ignore this trigger.

Target

If the **Test Method** is "Simple Ping", specify the name or IP address of the computer to ping.

If the **Test Method** is "HTTP Request", specify the URL that adTempus should request. For example:

```
http://10.10.10.201/testapp.asp
```

The **Target** may contain Job Variables, which will be expanded to resolve the target address.

Test Method

Select the type of test you want to perform:

- **Simple Ping.** adTempus will ping the computer at the interval you specify. If a response is received, adTempus considers the computer to be alive. If no response is received, adTempus considers the computer to be dead, and triggers the job. Note that adTempus cannot distinguish between a failed computer and a failed network connection. Use the "HTTP Request" method for a better evaluation tool.
- **HTTP Request.** adTempus requests a specific page from a Web server, and then evaluates the response it receives to decide whether to trigger the job. See Using adTempus to Monitor Your Web Application for more information.

Test Interval

These options determine how often adTempus will test the server:

- Test every X minutes
- Wait for a reply for Y seconds
- Trigger after Z minutes

The options work together as follows:

- Every X minutes, adTempus will send a ping or HTTP request to the target computer
- adTempus will wait up to Y seconds to receive a response from the target.
    - If no response is received, it considers the target to be dead.
    - If a response is received, it checks the other criteria to see if the target should be considered alive.
- If the target is dead, adTempus will continue checking every X minutes until Z minutes have elapsed since the first failure. If no valid response has been received after Z minutes, the job will be triggered.

If you want the job to be triggered the first time a failure is detected, set Z to be <= X.

Trigger at test interval until successful response received

If this option is checked, after adTempus first triggers the job it will continue to trigger the job at the test interval until the target is found to be alive.

If this option is not checked, adTempus will trigger the job only once per failure (that is, it will not trigger the job again unless the target is restored and then fails again).

Trigger again when successful response received

If this option is checked, adTempus will trigger the job again when a successful response is received from the target.

This option is useful if you are using the trigger to send a notification message to someone when the target fails. If this option is checked, a message would be sent when the target fails, and another sent when the target is restored.

The trigger sets a Job Variable named "Action" that indicates whether the job is being triggered because the target failed or because it was restored. You can test this variable using a Script Condition to determine which steps of your job to run depending on the action.

For example, Step 1 could use a Script Condition so that it only executes if the Action variable is "ConnectionFailed". Step 1 therefore would be run when the target computer fails.

Step 2 could use a Script Condition so that it only executes if the Action variable is "ConnectionRestored". Step 2 therefore would be run when the target computer was restored.

Response Evaluation

These options apply only when you have selected the "HTTP Request" test method, and determine how adTempus should evaluate the reply it receives from the target computer.

- **Trigger if response matches.** If the response matches the specified text, the job will be triggered.
- **Trigger if response does not match.** If the response does not match the specified text, the job will be triggered.
- **Use a script to evaluate the response.** adTempus will run the script you specify. Information about the response will be passed to the script using the Job Variables collection. Your script then evaluates the response and sets the global **Result** variable to **True** if adTempus should trigger the job, or **False** if it should not. Script example.

When you specify text for adTempus to match, adTempus looks for that text anywhere within the response it receives. For example, if you specify that the job should trigger if the response does not match "Application Status: OK", adTempus will look for the string "Application Status: OK" anywhere within the response, and trigger if it does not find it.

Use regular expressions

If this option is checked, the match text you enter is treated as a regular expression. If a match for that regular expression is found anywhere within the event message, the condition is satisfied.

Minimum Time Between Triggers

Specify the minimum time (in seconds) that must elapse after this trigger fires before adTempus will fire it again.

Description / Notes

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

See Also

Job Variables set by the Computer Monitor

Computer Monitor Overview

Using adTempus to Monitor Your Web Application

Sample setup for customizing the response to a failure

Job Variables

The Computer Monitor Trigger sets the following Job Variables, which can be used by a selection script or by other scripts or notification messages in the job.

Parameter Name	Description
ResponseText	The body of the HTML page returned by the remote server.
ConnectionStatusCode	A code indicating the result of the connection attempt.
Action	A string indicating whether the trigger is firing because the connection has been broken or because it has been restored: • "ConnectionFailed": No connection could be made to the server, or it returned an

	invalid response. • "ConnectionRestored": The connection has been restored, or the server is now returning a good response.
FailureCount	If Action="ConnectionFailed", FailureCount Indicates the number of consecutive unsuccessful responses that have been received since the last successful response. If Action="ConnectionRestored", FailureCount Indicates the number of consecutive unsuccessful responses that were received before the connection was restored. You can use this parameter to escalate your responses. For example, you could configure your job to send a warning message to certain users after the connection has been down for 5 minutes, and send a more urgent message if the connection is still down after 20 minutes.
ComputerMonitorTarget	The Target that the trigger is trying to connect to.

See Also

Using Job Variables

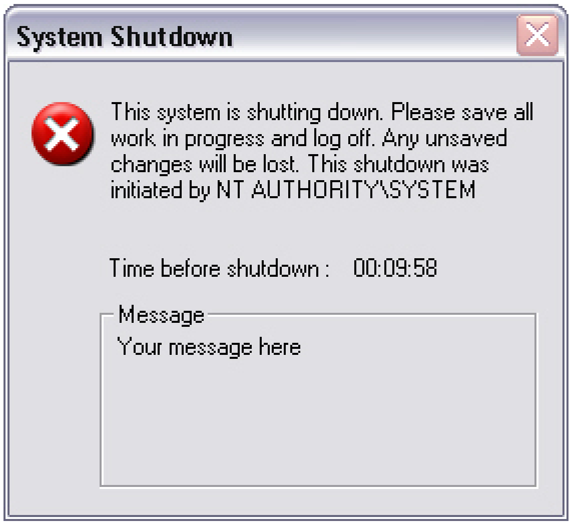
Computer Shutdown Task

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.



System Shutdown

This system is shutting down. Please save all work in progress and log off. Any unsaved changes will be lost. This shutdown was initiated by NT AUTHORITY\SYSTEM

Time before shutdown : 00:09:58

Message

Your message here

ⓘ	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.

See Also

[Computer Shutdown Task Properties]

[Tasks Overview]

General Page

The General page contains the basic settings for the [Computer Shutdown Task].

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.

See Also

[Computer Shutdown Task Overview]

Conditions Page

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.

See Also

Computer Shutdown Task Properties

Computer Shutdown Task Overview

Conditions Overview

Responses Page

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.

See Also

Computer Shutdown Task Properties

Computer Shutdown Task Overview

Responses Overview

Console Options Window

The Console Options window (**Tools > Console Options**) allows you to set your preferences for the Console. These settings are saved in your user profile, and changes do not affect other users.

Console Auto-Refresh Interval

The Console Auto-Refresh Interval specifies the interval (in seconds) at which the Console will retrieve the latest data from the adTempus server. When you are connected to many servers, or to a server with a large number of jobs, you may need to increase the interval to improve the performance of the Console.

With the auto-refresh interval set to a longer interval, you can always refresh the Console manually by pressing **F5** in any Console view.

Reset all hidden messages

The Reset all hidden messages option turns back on all messages you have previously hidden by checking the "Do not show this message again" box when the message was displayed.

Credential Profile

Credential Profiles

Credential Profiles simplify the use and management of user credentials in adTempus. Each Credential Profile stores the user ID and password for a single user account. In most cases this will be a Windows user account, but you may also need to store other kinds of credentials, such as the credentials for connecting to a protected Web server.

Instead of storing user IDs and passwords with each object that uses them (for example, in each job that runs under a user account), adTempus stores the information in one place (the Credential Profile) and then stores a reference to the Credential Profile in each object that uses it.

Features and Benefits

Credential Profiles offer several benefits.

Simplified Password Management

If the password for an account is changed, the password only has to be changed once in adTempus, in the Credential Profile. The new password will immediately be used by all jobs that use the profile.

Delegated Account Use for Improved Security

Users can be granted permission to use a Credential Profile without knowing the password for the account.

For example, it is often necessary that jobs be run under user accounts with Administrative privileges, and non-administrative users may need to configure and manage these jobs. Instead of telling these users the password for an administrative account to use on jobs, an authorized Administrator (who knows the account's password) can create a Credential Profile, and grant other users permission to use the profile. These users will now be able to schedule jobs using this account without having access to the account's password.

Simplified Data Entry

Once a user has been granted permission to use a Credential Profile, that user does not have to type in (or even know) the password for the profile. When entering credentials for a new job or other object, the user only has to type the user ID.

Creating and Using Credential Profiles

Generally, Credential Profiles will be created automatically, as needed. In places where credentials need to be specified (such as the **User Account** box in a job's properties), the user simply types in the user ID for the account. adTempus will then prompt the user for the password and create a new Credential Profile if necessary, as described in the Entering User Credentials topic.

You can also create profiles through the Credential Profiles window.

Managing User Profiles

Authorized users can view and manage Credential Profiles through the Credential Profiles window.

If the password changes for an account, use the Credential Profiles to edit the associated Credential Profile and change the password in adTempus.

Entering User Credentials

Type in the user ID for the account you want to use. For Windows domain accounts, enter the account name in the following form:

```
domain\username
```

After you type the user name, adTempus will check to see if there is already a Credential Profile for that user ID.

- If no Credential Profile exists, the Credential Profile Properties window will be displayed, allowing you to specify the password and other properties for the Credential Profile. You become the owner of this Credential Profile, and can use the security settings to grant permission for other users to use the Profile without needing to know the password.
- If a Credential Profile exists for the user ID, adTempus checks to see if you already have permission to use it.
    - If you already have permission to use the Credential Profile, you are not prompted to enter the password.

- If you have not already been granted permission to use the Credential Profile, adTempus will prompt you for the password. If the password you enter matches the password stored for the profile, you will be granted permission to use the profile, and in the future will not have to enter the password.

See Also

Credential Profile Overview

Credential Profile Properties

Properties

Credentials Page

The **Credentials** page defines the basic properties of the Credential Profile. A Credential Profile generally represents a Windows user account, but may also represent other kinds of account.

Domain

For a Windows user account, enter the domain name, if appropriate. If the account is not a domain account, leave this box empty.

User ID

Enter the user ID for the account.

Password

Enter the password for the account.

Comments

For a Windows user account, you may enter comments about the account. For other account types, the Comments box will show a description of the account type, which cannot be changed.

See Also

Credential Profiles Overview

Security Page

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

Credential Profiles inherit security settings from the defaults specified in the Credential Profiles window.

Automatic Permission Granting

When a user tries to use an existing Credential Profile (e.g., when configuring a job) but does not have "Use Credentials" permission for that profile, she will be prompted to enter the password associated with the profile. If she enters the correct password, she will automatically be granted "Use Credentials" permission for the profile.

Once this permission is assigned, the user can use the profile in the future without being prompted for the password, even if the password is subsequently changed.

Available Permissions

The following permissions apply to Credential Profiles:

Permission	Description
Full Control	Permission to perform all actions on the profile.
Use Credentials	Permission to use the credentials on a job or other object. If a user has Use Credentials permission, she can use the profile without knowing the password for the account.
Modify	Permission to modify the properties of the profile.
Delete	Permission to delete the profile.
Administer security	Permission to change the security settings for the profile.
Change owner	Permission to take ownership of the profile.

See Also

Security Overview

Credential Profiles Overview

Duplicate Job Group Window

This window allows you to duplicate a group and all the jobs and groups it contains. To reach this window, right-click a group in the Console Tree or Job List and select the **Duplicate Group** option from the pop-up menu.

When you duplicate a group, adTempus creates a copy of the group and of all the groups and jobs contained within the group. The new group is created as a sibling of the source group (both have the same parent group).

You will be able to review the results of the copy operation before you save the new groups and jobs.

Name for New Group

Specify the name for the new group. The name must be unique within the parent group.

Hold all jobs in new group

If this box is checked, all of the newly-created jobs will be held (disabled). If this box is not checked, each job will have the same held status as the original from which it is copied.

Prefix new job names with

Check this box if you want to add a prefix to the name of each duplicated job. If you do not specify a prefix, each job will have the same name as the job from which it is copied. Since the new jobs will be in a different group than the originals, the names will still be considered unique.

Copy

Click the **Copy** button to begin the copy process. When you copy a group, all the groups and jobs it contains are copied as well, creating a copy of the group's hierarchy.

Once the copy is complete, click **OK** to save the new groups and jobs, or **Cancel** to discard them.

Event Log Monitor Trigger

Event Log Monitor Trigger

The Event Log Monitor Trigger allows adTempus to monitor the Event Log on the host computer and trigger jobs when selected events are logged.

For example, you could have a job that is run whenever a failed logon attempt is reported by the operating system, or a job that is run whenever SQL Server reports a replication error.

Each time adTempus is started, it reviews any events that were logged while it was stopped, and triggers based on those events as appropriate. It then begins watching for new events and triggering based on them.

See Also

Event Log Trigger Properties

Triggers Overview

Event Log Trigger Properties

This page defines the conditions for triggering a job. adTempus will trigger your job each time a new event is reported that meets all of the criteria you specify.

Optional Descriptive Name

Optionally, provide a descriptive name for this trigger. Otherwise, a description will be created automatically based on trigger settings.

Enable this Trigger

If this box is checked, the trigger is enabled and will cause the job to run. If this box is not checked, adTempus will ignore this trigger.

Log

Select the log to monitor. A trigger can only monitor a single log.

Sources

Select the event sources (applications) whose events you want to check. You may select any number of sources; if you do not select any source, all sources will be considered.

For example, you might want to trigger based on Error messages logged by a particular application in the Application log. Or you might want to trigger any time an Error message is logged in the System log.

Category

This option is available only if you have selected a single event source. Select the category of messages you want adTempus to examine.

Types

Select the types (severities) to consider. If none are selected, all types are considered.

Include Event IDs

If you want to trigger only when specific events are logged, enter the IDs of those events here. You can specify as many events as you wish, separated by commas.

Event IDs are specific to each application (Event Source). To determine which events IDs you want to capture, consult the documentation for the application in question, or use the Windows Event Log Viewer to examine events the application has logged. The Event Log Viewer will show you the Event IDs for those events.

If you have selected multiple Event Sources, you should not specify Event IDs unless you are sure you know what you are doing (i.e., all of the sources you have selected use the same Event IDs to mean the same thing).

Exclude Event IDs

If you want to ignore certain events, enter the IDs of those events here. You can specify as many events as you wish, separated by commas.

For example, you might want to trigger any time an event of type Error is logged by Source "My Application", except if the event ID is 117.

Minimum Time Between Triggers

Specify the minimum time (in seconds) that must elapse after this trigger fires before adTempus will fire it again. Once adTempus triggers the job, it will ignore all new events until the specified time has passed.

Select events whose message matches

Check this option to select events based on the text of the event message. If the message contains the text you specify, it will be selected. The message does not have to match the specified text exactly, it may contain the specified text anywhere within it.

Use regular expressions

If this option is checked, the match text you enter is treated as a regular expression. If a match for that regular expression is found anywhere within the event message, the condition is satisfied.

Use a script to select events

adTempus will run the script you specify. Information about the event will be passed to the script using the JobVariables collection. Your script then evaluates the event and returns **True** if adTempus should trigger the job, or **False** if it should not.

adTempus calls the script only for events that have satisfied all of the other criteria for the trigger.

Description / Notes

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

See Also

Job Variables set by this trigger

Job Variables

The Event Log Monitor Trigger sets the following job variables, which can be used by a selection script or by other scripts or notification messages in the job.

Parameter Name	Description
EventLog	The name of the log in which the event was recorded (e.g., "Application").
EventSource	The name of the event source. This is the name of the source as it appears in the Registry, and may not be the same as the name displayed in the user interface.
EventCategory	The numeric event category.
EventType	The numeric event type (severity).
EventID	The numeric event ID.
EventMessage	The full text of the message.
EventTimestamp	A VARIANT containing the timestamp for the message.

Execute Job Window

Execute Job

The Run command is used to submit a job to run immediately. This feature can be used to test the setup of a job, or to run an "on-demand" job.

The **Execute Job** window allows you to specify how the job will be run. It is divided into the following pages:

General Options

Steps

Variables

See Also

Running jobs using the adtExec utility

General Options Page

The **General Options** page contains basic execution options for the job

Ignore conditions for the job

If this option is checked, adTempus will ignore any conditions that are defined for the 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 steps of the 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 job, forcing adTempus to start a new instance of the job even if another instance is already running.

Force the job to run even if the queue is held

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

adTempus runs the job regardless of whether the job itself is held.

Override queue limits and force the job to run immediately

If this option is checked, adTempus runs the job immediately, even if the Queue to which it belongs is currently at its quota for the maximum number of concurrent jobs. That is, this option will force the Queue to exceed its limit if necessary.

Run only on Master

If this option is checked, adTempus runs the job only on the Master computer, even if the job is associated with Agents.

To run the job only on a specific agent, select the agent from the Agents list and use the **Run** command on the pop-up menu.

Force job to run on Master

If this option is checked, adTempus runs the job on the Master, even if the Queue's Run on Master option is not checked.

Save the selected options as the default options

When this box is checked, adTempus remembers your choices for all of the above settings (settings in the "Options" group) and makes them the default the next time you run a job manually.

Make the job visible on my desktop

This option is available only if you are running the Console on the same computer where the adTempus service is running. When this box is checked, it overrides the job's User Interaction settings and causes adTempus to run the job in your current Windows logon session, if possible. This ensures that you will be able to see any windows shown by programs that the job runs.

Responses

Choose which Responses you want to execute (this option applies to Responses for the job and for steps within the job):

- **Execute Responses:** All Responses are executed as normal.
- **Do not execute any Responses:** No Responses will be executed.
- **Do not execute any Job Control Responses:** All Responses will be executed *except* Job Control Responses (Responses that run, terminate, hold, or release a job or job step). Use this option if you do not want adTempus to execute any other jobs that are chained to the current job.

Checkpoint

Specify the checkpoint to pass to the job.

Steps Page

The options on the **Steps** page of the Execute Job window determine which steps adTempus will run.

Run the job from the beginning

The job will run as normal, with all steps executed.

Run the job from the specified step

Execution will begin with the selected step and continue to the end of the job. If you check **Run only the selected step**, the job will end after the selected step is run.

Run only the selected steps

Only the steps you select will be executed. Steps will be executed in their original order (as listed here), but you may skip steps.

Variables Page

The **Variables** page allows you to set or override Job Variables that have been defined for this job. Values you provide here are used only for this execution of the job.

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.

See Also

File Capture Action Properties

Responses Overview

Actions

Files Page

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.

See Also

File Capture Action Overview

File Specification Properties

File Capture Action Properties

Send Page

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.

Include captured console output

If this option is checked, the e-mail message will also include any captured console output for the step or job invoking the action.

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.

See Also

Notification Recipients

File Capture Action Overview

File Specification Properties

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.

The **File Specification** may contain Job Variables, which will be expanded to resolve the file name.

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.

See Also

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.

> A File Condition will not cause a job to start executing: it will only cause a job to pause execution to wait for files. To start a job based on files, use a File Trigger instead.

See Also

File Condition Properties

Conditions Overview

File Condition Properties

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

> A File Condition will not cause a job to start executing: it will only cause a job to pause execution to wait for files. To start a job based on files, use a File Trigger instead.

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.

The **File Specification** may contain Job Variables, which will be expanded to resolve the file name.

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.

Once satisfied, do not re-evaluate for this instance

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

Condition Wait

This option determines whether adTempus will wait for the condition to be met:

- **Do not wait for condition to be met.** If the condition is not satisfied the first time adTempus checks it for a given instance, adTempus treats the condition as "Failed."
- **Wait until the condition is met.** If the condition is not satisfied, adTempus will wait until it is satisfied. The job or step will show a status of "Waiting for Condition."
- **Wait up to __ minutes for the condition to be met.** adTempus will wait up to the specified time limit for the condition to be satisfied. If the condition is satisfied within that time limit, job execution will continue immediately (as depending on other conditions). If the condition is not satisfied at the end of the time limit, adTempus treats the condition as "Failed."

See the Condition Polling topic for more information.

See Also

File Condition Overview

File Trigger

File Trigger

A File Trigger is a trigger that causes a job to be executed whenever specified files are created, modified, or deleted.

When you create a File Trigger, you specify the files or directories that adTempus should watch for, and under which conditions the job should be triggered.

The File Trigger works by periodically (approximately every 10 seconds) scanning the directories being watched by the trigger and comparing the current state to the previous state.

Multiple Files and Job Instances

Each time the File Trigger scans the directories, it produces a list of all the files that meet the trigger criteria. Because adTempus makes this evaluation at intervals of several seconds, it is possible for more than one file to meet the trigger criteria. For example, suppose you have a trigger configured to fire any time files are added to a particular directory. In between evaluations of the directory, five new files are copied into the directory. On the next evaluation, adTempus will find those five files, and the trigger's **Start a separate instance for each file** setting determines how it will handle the scenario.

Separate Instance for Each File

If the **Start a separate instance for each file** option is checked, adTempus will start a separate instance of the job for each of the matching files it has found, and each instance will have the FileName job variable set to the name of the file for which it is being executed.

Typically you would use this option if your program is being called to process files individually. For example, you need to run a data processing utility on each file that is uploaded into a directory. When you configure the Program Execution Task to run your program, you can include the token "%FileName%" in the command-line parameters, and adTempus will replace "%FileName%" with the name of the triggering file.

Single Instance for All Files

If the **Start a separate instance for each file** option is *not* checked, adTempus will start a single instance of the job, even if more than one file has been found. The FileName job variable will contain the names of all the files found on the current evaluation pass.

Typically you would use this option if your program is being called to process an entire directory. For example, your program already has logic to scan an entire folder and process all the files in it; you only need adTempus to start it when there are files there to process.

In this scenario, you need to make sure you have configured the job so that it will not be triggered too often. For example, set the job's Multiple Instances rule to prevent duplicate instances of the job from being started at the same time. Otherwise, adTempus may trigger your program again while it is still processing files from the first trigger.

Changes Made While adTempus is Not Running

adTempus saves the current state of the directories it is monitoring each time it makes an evaluation pass. This state information is stored in the adTempus database, and so is not lost if the adTempus service is shut down. Therefore when adTempus starts, it compares the current file state to the state information it saved before it shut down, and triggers jobs based on the differences. That is, adTempus will respond to files that were added, changed, or deleted while adTempus was not running.

Potential Problems with Network Drives

If your file trigger is watching files on a network drive, you may encounter problems if the network connection is broken. Because of the way adTempus scans for files, if the network connection is broken, any files on the network drive will not be visible to adTempus, and it will respond as if they had been deleted, triggering the job if your trigger is set to fire on deletions. Similarly, when the connection is restored and the files suddenly reappear, adTempus will see them as new files (because they were not there on its previous evaluation pass). adTempus will therefore trigger the job if the trigger is set to fire when new files are created.

See Also

File Trigger Properties

Triggers Overview

File Condition

File Trigger Properties

The File Trigger Properties window defines the circumstances under which the job will be triggered.

Optional Descriptive Name

Optionally, provide a descriptive name for this trigger. Otherwise, a description will be created automatically based on the file names you specify.

Enable this Trigger

If this box is checked, the trigger is enabled and will cause the job to run. If this box is not checked, adTempus will ignore this trigger.

Trigger Conditions

Select the conditions under which the job will be executed. You must select at least one option.

Files

Specify the files or directories to watch. You may specify a single file (e.g., "c:\data\jobtrigger.txt") or use wildcards (e.g., "c:\datafiles\newfiles\*.dat").

You may add as many file specifications as you wish. The job will be triggered when a file is found that matches any of the file specifications you have provided and matches all the other conditions you have specified.

Start a separate instance for each file

Check this option to have adTempus start a separate instance of the job for each matching file that is found. If this option is not checked, adTempus will start only one instance of the job regardless of how many matching files are found in a single evaluation "pass". See the File Trigger Overview topic for more information.

Minimum Time Between Triggers

Specify the minimum time (in seconds) that must elapse after this trigger fires before adTempus will fire it again.

adTempus evaluates the files approximately every 10 seconds; specify a longer time if appropriate.

> You should use this option and/or the options available on the job to prevent duplicate instances to prevent your job from being triggered continually.
>
> For example, suppose you are monitoring a log file for changes. The first time adTempus detects a change to the file, it will trigger your job. If the application that is writing to the log file continues to write to it, adTempus will continue to trigger your job each time it detects a change (it will check for changes roughly every 10 seconds).
>
> See the File Trigger Overview topic for more information.

Wait for exclusive access to the file

If this option is checked, adTempus will not trigger the job until it can get exclusive access to the file (i.e., no other application is reading or writing the file).

Capture the file

If this option is checked, adTempus will capture the file that satisfied the trigger and save it in the job history.

Selection Script

You may specify a script that is to be run to determine whether a file should cause the job to trigger. When you specify a script, adTempus evaluates files as follows:

1. It finds all files that match your file specifications.
2. It discards any files that do not meet other selection criteria you have specified.
3. For each remaining file, it calls the script you specify. Information about the file to be tested is passed to your script in the JobVariables collection.
4. Your script evaluates the file in whatever way is appropriate. If the job should be triggered based on the file, your script sets the global **Result** variable to **True**, and adTempus triggers the job when your script returns. Otherwise, your script sets the **Result** variable to **False**, and adTempus continues to the next file.

Description / Notes

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

See Also

File Trigger Overview

File Specification Properties

Each File Specification defines a file (or set of files) that will trigger the job.

File Specification

Specify the file(s). 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.

The **File Specification** may contain Job Variables, which will be expanded to resolve the file name.

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.

See Also

File Trigger Properties

File Trigger Overview

Job Variables

The File Trigger sets the following Job Variables, which can be used by a selection script or by other scripts or notification messages in the job.

Variable Name	Description

FileName	The full name (including path) of the file.
FileSizeHigh	The high-order bytes of the file size.
FileSizeLow	The low-order bytes of the file size.
FileCreationTime	The date/time at which the file was created.
FileLastAccessTime	The date/time at which the file was last accessed.
FileLastWriteTime	The date/time at which the file was last modified.
FileAttributes	An integer indicating the file's attributes.

See Also

Using Job Variables

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).

To view or manage holiday sets, select the Holiday Definitions folder in the adTempus Console.

adTempus comes preconfigured with the "Standard U.S. Holidays" set, which includes **standard U.S. holidays**.

The Standard U.S. Holidays set defines the following 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.

See Also

Holiday Set Properties

Specifying holidays for jobs

Schedule Page

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.

See Also

Holiday Set overview

Selecting Days

Security Page

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

The following permissions apply to Holiday Sets:

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 Holiday Sets is controlled through the Shared Schedule and Holiday Security window.

See Also

Holiday Set overview

Specifying default holiday security

Jabber Service Provider

Jabber Service Provider

The Jabber Service Provider allows you to send notification messages using the Jabber instant messaging protocol.

For more information on Jabber, Jabber clients, and Jabber servers, see www.jabber.org.

Once you have configured a provider, you can create Notification Addresses that use it.

See Also

Jabber Provider Properties

Notification Service Providers Overview

Provider Page

The **Provider** page defines the messaging server that adTempus should connect to and the account it should use when connecting.

ℹ	Unlike the Microsoft Messenger, Jabber supports multiple logins using the same account. You can therefore have adTempus use the same account you normally use, or use the same account for all adTempus installations.

Provider Name

Provide a descriptive name for this provider. The default "Jabber" will generally be sufficient, unless you plan to be connecting to more than one Jabber server.

User ID and Password

Specify the user ID and password used to log in to the Jabber server. This is the part of the Jabber ID before the @ sign. For example, if your Jabber ID is "arcanadevelopment@jabber.org", the User ID is "arcanadevelopment".

Jabber Server

Many public and private Jabber servers are available. If you are not using the public server at jabber.org, you will need to change the settings here to point to the correct server.

Security Page

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

The following permissions apply to notification service providers:

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

Permission to create new providers is controlled through the Notification Security options.

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.

Security

The adTempus security framework determines who is able to manage a job within adTempus, but these settings do not affect what the job can do at run-time. Instead, for each job you must specify the Windows user account that the job will use. When adTempus runs the job it logs in the specified user and runs the job in that user's security context.

This means that any programs, scripts, or other tasks run by the job will have the same security limitations as if that user were running the program directly.

See Also

Introduction to Key Concepts

Job Properties

Checkpoints

adTempus provides the capability of restarting jobs that were interrupted due to a system shutdown or failure. In such cases, though, you may not want to start executing your task from the beginning in the event of a restart.

For example, if you have a batch file that executes several different steps, you may want to skip the steps that completed before the failure.

Checkpoints provide a means of doing this. As your program, batch file, or script executes, it can periodically pass a checkpoint value to adTempus (this can be any string that will be meaningful to your program). If adTempus subsequently restarts the job after a failure, it will pass back the latest checkpoint it received; your task can use that to resume execution at the correct place.

The method for getting and setting the checkpoint depends on the kind of task you are executing:

- Batch files should use the adtChkpt utility program.
- Scripts (as long as they are run by an adTempus Script Execution task) can use the global Checkpoint variable.
- Applications can retrieve the checkpoint by using Windows API functions to retrieve the "ADJobCheckpoint" environment variable. To set the checkpoint the application must use the automation interface to the adTempus engine.

Properties

Job General Page

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 within the group that the job belongs to.

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.

Queue

Select the Queue to which this job is assigned.

Priority

Specify the priority of this job within the queue. In a queue that limits the number of jobs that can run simultaneously, jobs with higher priorities are executed before jobs with lower priorities. New jobs have a default priority of 100.

User Account

Specify the Windows user account that the job should be run under. See Entering User Credentials for more information on how to enter the credentials.

All tasks performed by this job will be run under the user account you specify. For example, when the job executes a program, the program will operate under the same security context (and restrictions) as if the specified user had run the job herself.

Run with highest privileges

Note: This option is available only if the adTempus server is running Windows Vista, Windows Server 2008, or later.

If the specified **User Account** is an administrative account, check the **Run with highest privileges** option to have administrative privileges enabled when the program runs. This is equivalent to using the "Run as Admin" command in Windows. If this option is not checked, administrative privileges will not be enabled, and the program may not behave as you expect.

If the **User Account** is not an administrative account, checking this option has no effect on program execution.

User Interaction

Select the kinds of Windows logon session that this job is allowed to use. The session that a job runs in determines whether it will be visible to interactive users at the computer, and to which users.

Each job runs in one of the following sessions:

- **Console.** The console session is the keyboard and monitor that are physically connected to the computer, or the first Terminal Services/Remote Desktop session to attach to the server. A program running in the console session is visible only to a user logged in to the console session. If you are connected to the computer through a Terminal Services/Remote Desktop session that is not the console session, you will not be able to see the program's user interface.
- **User's Session.** If the user whose account is used to run the job is logged on interactively at the computer, adTempus will run the job in that logon session. The user will be able to see the user interface for any programs run by the job. If the user has more than one interactive logon session, the program will run in the logon session that was created first.
- **Hidden.** The job will be run in a hidden session. If the programs run by the job display a user interface, no user will be able to see them.

Explanation of User Interaction options

The **User Interaction** selection list specifies which session types the job is allowed to use, and in what order of preference.

The following options are available in the **User Interaction** selection list:

Setting	Meaning
Console, User's Session, Hidden	- If there is a console session for the computer, the job will run in that session. - Otherwise, if there is an interactive session for the job's user, the job will run in that session. - Otherwise, the job will run in a hidden session.
Console, User's Session	- If there is a console session for the computer, the job will run in that session. - Otherwise, if there is an interactive session for the job's user, the job will run in that session. - Otherwise, the job will fail.
Console, Hidden	- If there is a console session for the computer, the job will run in that session. - Otherwise, the job will run in a hidden session.
Console	- If there is a console session for the computer, the job will run in that session. - Otherwise, the job will fail.
User's Session, Console, Hidden	- If there is an interactive session for the job's user, the job will run in that session. - Otherwise, if there is a console session for the computer, the job will run in that session. - Otherwise, the job will run in a hidden session.
User's Session, Console	- If there is an interactive session for the job's user, the job will run in that session. - Otherwise, if there is a console session for the computer, the job will run in that session. - Otherwise, the job will fail.
User's Session, Hidden	- If there is an interactive session for the job's user, the job will run in that session. - Otherwise, the job will run in a hidden session.
User's Session	- If there is an interactive session for the job's user, the job will run in that session. - Otherwise, the job will fail.

Hidden	The job will run in a hidden session.

Guidelines for selecting the interaction level

A program that is properly designed for "batch" execution should not rely on direct interaction with users, and therefore should not need to display a user interface. For such programs, use the "Hidden" session whenever possible. This provides more security, and eliminates the annoyance of program windows popping up while a user is logged in to the computer.

If you need to be able to see or interact with programs run in the job, choose one of the interactive options (Console or User's Session) depending on your needs.

For an interactive program, also consider what adTempus should do if no interactive session is available (i.e., no user logged in).

- If it is imperative that a user be able to see/interact with the programs, choose a **User Interaction** option that does *not* include the "Hidden" option. This will cause the job to fail if no interactive session is available.
- If it is more important that the job be run at its scheduled time, be sure to include a **User Interaction** option that *does* include the "Hidden" option. Since a hidden session is always available, this will ensure that your job is run even if no interactive session is available.

Regardless of what interaction level is selected, if you run the job manually while logged in to the computer, you can choose to have the job executed in your current logon session. This is useful if you need to be able to see the user interfaces of the job's programs during job debugging.

The option you specify here may override the "Window Mode" option specified for a Program Execution task.

Description / Notes

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

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.

See Also

Job Overview

Variables Page

The **Variables** page lists all Job Variables for the job. A job inherits variables from the group and queue that it belongs to, and from the server-level variables. You can add, modify, or delete new variables, or modify inherited variables. Inherited variables cannot be deleted.

The variables defined here are inherited by the job's steps, triggers, conditions, etc.

See Also

Job Variables Overview

Job Recovery Page

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

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.

ℹ	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.

ℹ	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.

Trigger Page

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.

> ℹ 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.

> ℹ 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.

See Also

Triggers Overview

Conditions Page

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

> ℹ 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.

See Also

Conditions Overview

Steps Page

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

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.

ℹ	Regardless of which option you specify here, Job Control actions executed by Responses within the job can alter the flow of the job. For example, a Job Control Action may cause the job to end early, or may jump to a step out of the specified sequence.

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.

See Also

Tasks Overview

Job Resources Page

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

Resource Failure

Select the action that adTempus should take if one or more resources cannot be loaded.

- **Fail the job**. The Job will terminate immediately with a Failed status.
- **Run the job anyway**. adTempus will write a warning message to the job log but will execute the job anyway. Your tasks may fail to execute correctly if they depend on network resources that are not available. This option should be used only if your tasks can operate properly without the requested resources.

Resources List

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

See Also

Resources Overview

Job Responses Page

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
Beginning of Job	Occurs at the beginning of the job, before adTempus has evaluated any conditions or attempted to execute any steps.
End of job	Occurs at the end of the job, regardless of the job's outcome.
Conditions failed	Occurs if one or more conditions is not met (occurs only once regardless of how many conditions failed).
Job aborted	Occurs when the job is aborted (manually or as a result of a Job Control action).
Job failed	Occurs when the job is ending with a Failed status, for whatever reason.
Job restarted	Occurs when the job is being restarted by a Job Control action.
Job skipped because another instance is running	Occurs when another instance of the same job is already running and the Multiple Instances rule for the job specifies that the job should be skipped in this situation.
Job succeeded	Occurs when the job finishes with a successful result.
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.

See Also

Responses Overview

Job Security Page

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

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. Includes the ability to hold and release the job.
Delete	Permission to delete the job.
Execute/Terminate	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.
Hold/Release	Permission to hold and release the job, but not to modify any of its other properties.
Administer security	Permission to change the security settings for the job.
Change owner	Permission to take ownership of the job.

To create a new job, a user must have permissions for both the group and queue that the job is being created in:

- For the group, the "Create jobs and groups in this group" permission.
- For the queue, the "Associate (assign jobs to this queue)" permission.

See Also

Security Overview

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.

A Job Condition will not cause a job to start executing: it will only cause a job to pause execution to wait for other jobs. Some other mechanism (a Trigger or a Response on another job) must start the job. If you want one job to cause another to start when it finishes, you probably want to use a Job Control Action instead of a Job Condition. See the Comparison of Triggers and Conditions topic for more information about the differences.

See Also

Job Condition Properties

Comparison of Triggers and Conditions

Conditions Overview

Job Condition Properties

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.

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.

Instance

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

- **The most recent instance since the previous execution of job.** adTempus will look at the most recent instance of the target job that executed since the last execution of the dependent job.
- **The most recent instance.** adTempus will look at the most recent instance of the target job, regardless of when that instance executed.
- **Any instance since the previous execution of 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, the condition will be satisfied.

It is helpful to think of a "cycle" of jobs. If job "Data Copy" is dependent on job "Data Extract," a normal cycle would be an execution of "Data Extract" followed by an execution of "Data Copy." Once "Data Copy" has run, it generally should not run again until "Data Extract" has run again.

The default Instance option "The most recent instance since the previous execution of job" will produce this kind of cycle, and is the recommended setting. The other two options can be used for specialized scenarios. See this example for an illustration of the effects of the three settings.

Condition Wait

This option determines whether adTempus will wait for the condition to be met:

- **Do not wait for condition to be met.** If the condition is not satisfied the first time adTempus checks it for a given instance, adTempus treats the condition as "Failed."
- **Wait until the condition is met.** If the condition is not satisfied, adTempus will wait until it is satisfied. The job or step will show a status of "Waiting for Condition."
- **Wait up to __ minutes for the condition to be met.** adTempus will wait up to the specified time limit for the condition to be satisfied. If the condition is satisfied within that time limit, job execution will continue immediately (as depending on other conditions). If the condition is not satisfied at the end of the time limit, adTempus treats the condition as "Failed."

See the Condition Polling topic for more information.

Once satisfied, do not re-evaluate for this instance

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

See Also

Job Condition Overview

Job Condition Instance Example

This example illustrates the behavior of the Job Condition based on the three available **Instance** options.

Suppose you are making job "Data Copy" dependent on job "Data Extract," with the Rule 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 previous execution of job," the condition is not met, because the most recent instance (404) failed.
- If the **Instance** is set to "Any instance since the previous execution of 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 previous execution of 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 previous execution of job," the condition is not met, because "Data Extract" has not been run since the last time "Data Copy" was run.

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

See Also

Job Control Action Properties

Responses Overview

Actions

Action Page

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

Action to take

Select the action that will be carried out:

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.

Wait __ seconds before restarting

This option is only available when the **Action to take** is "Restart the job from the beginning" or "Restart the step".

Use this option to have adTempus wait a specified number of seconds before it restarts the job or step.

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.

Set the status of the calling job/step to "Resubmitted"

This option is only available when the **Action to take** is "Restart the job from the beginning" or "Restart the step".

When this option is checked, the current job instance (if the action is "Restart the job") and step have their status set to "Resubmitted." If this option is not checked, they retain their current status.

For example, suppose that you are setting up this action to restart the job if it fails. Instance 1 of the job runs and fails, so the action is invoked and the job restarts, creating instance 2. If the **Set the status...** option is checked, the status of instance 1 is changed from "Failed" to "Resubmitted," making it clear that the restart has occurred, and preventing instance 1 from appearing in the "Failed Jobs" view in the Console. If the **Set the status...** option is not checked, instance 1 retains its original "Failed" status.

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.

Run only the selected step

If this option is checked, adTempus will run only the step you have selected. Otherwise, adTempus runs the job starting with the selected step and continuing with the remaining steps in the job.

Responses

Choose which Responses you want to execute (this option applies to Responses for the job and for steps within the job):

- **Execute Responses:** All Responses are executed as normal.
- **Do not execute any Responses:** No Responses will be executed.
- **Do not execute any Job Control Responses:** All Responses will be executed *except* Job Control Responses (Responses that run, terminate, hold, or release a job or job step). Use this option if you do not want adTempus to execute any other jobs that are chained to the current job.

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.

Run only on same Agent

This option is only available if the current server is a Distributed Scheduling Master computer. If this option is checked, the target job will be run only on the same computer as the calling job. If this option is not checked, the job will execute according to the Distributed Scheduling settings for its Queue, and may not execute on the same computer as the calling job.

Run only on Master

This option is only available if the current server is a Distributed Scheduling Master computer. If this option is checked, the target job will be run only on the Master computer; any Agents associated with the job's Queue will be ignored. If the Queue is not configured to run jobs on the Master, the job will not run unless the **Force job to run on Master** option is also checked.

Force job to run on Master

This option is only available if the current server is a Distributed Scheduling Master computer. If this option is checked, the target job will run on the Master computer even if the Queue is not configured to run jobs on the Master.

See Also

Variables Page

Job Control Action Overview

Variables Page

The **Variables** page allows you to set or override Job Variables that have been defined for the target job. Values you provide here are used only when the job is executed by this Job Control Action.

For example, you may have a job that is triggered to run independently but can also be started by another job through a Job Control Action. If the job needs to behave differently depending on how it is started, you can set a Variable in the Job Control Action so that the job knows it is being started by another job.

See Also

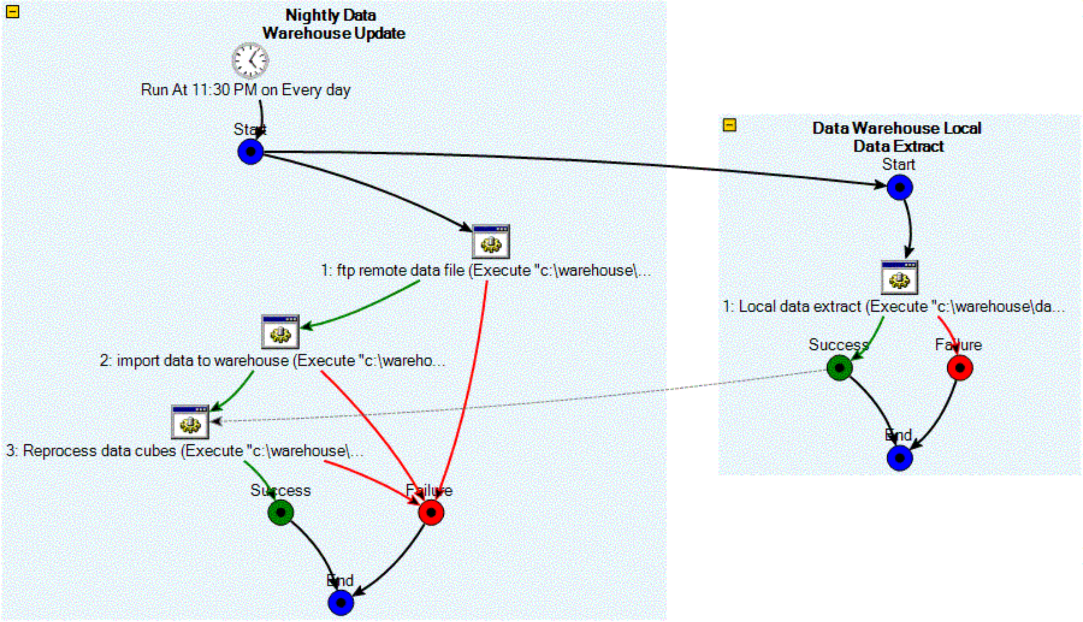
Job Control Action Overview

Job Flow Diagram

The **Job Flow Diagram** tool allows you to create job flow diagrams, which are graphical representations of your jobs that can be used to review and document your job setup.

To diagram a job chain, right-click a job in the Job List and select the **Diagram job/chain...** command. adTempus will diagram the job and the complete chain of all jobs linked to it (and jobs linked to those jobs, and so on).

Diagrams are created automatically based on your job setup. For example, this diagram shows the "Nightly Data Warehouse Update" job and the "Data Warehouse Local Data Extract" job from the job setup example.


Nightly Data Warehouse Update
Run At 11:30 PM on Every day
Start
1: ftp remote data file (Execute "c:\warehouse\...
2: import data to warehouse (Execute "c:\wareho...
3: Reprocess data cubes (Execute "c:\warehouse\...
Success
Failure
End
Data Warehouse Local Data Extract
Start
1: Local data extract (Execute "c:\warehouse\da...
Success
Failure
End

The diagramming tool does not allow you to build or edit your job graphically. You can display and edit the properties for jobs, steps, actions, and links by double-clicking them in the diagram.

Navigating the Diagram

Each job is represented by a shaded box. Initially the diagram will show the complete internal details of all of the selected jobs (and the jobs linked to them). Each trigger, condition, step, and action for the job is displayed. Links between jobs are also shown.

To hide or show the details for an individual job, use the collapse (⊟) and expand (⊞) buttons found on each job. You can also expand (⁺₊) and collapse (⁻₋) all jobs at once.

By default, jobs, triggers, steps, and actions are labeled, but links are not. To see a description of a link, click it, and information will be displayed in the Properties box to the right of the diagram. You can turn on display of labels if you prefer, using labels toggle button (💬).

Link Types

Links are color-coded to allow you to get a sense of the job flow without needing labels on the links:

Solid black link →	Represents a course that is always taken
Green link →	Represents a course that is taken on success
Red link →	Represents a course that is taken on failure
Yellow link →	Represents a course that is taken in "warning" conditions
Grey dashed link ⇢	Represents a condition that must be met (does not indicate a direct action). In the sample diagram above, the end node of "Data Warehouse Local Data Extract" has a condition link to step 3 of "Nightly Data Warehouse Update". This indicates that step 3 of "Nightly Data Warehouse Update" cannot run until "Data Warehouse Local Data Extract" completes, but does not mean that "Data Warehouse Local Data Extract" directly runs step 3 of "Nightly Data Warehouse Update".

Editing Objects

You can edit an object (job, step, link, action, condition, or trigger) by double-clicking it on the diagram. If you make any changes, the diagram will be redrawn automatically.

Printing the Diagram

The diagram can be printed using the Print (🖨) and Print Preview (🔍) buttons on the Toolbar.

Saving the Diagram

Diagrams can be saved to a file using most common graphics image formats.

Job Group

Job Group

Job Groups allow you to group related jobs into "folders" in the adTempus Console. This allows you to better organize your workspace when you have a large number of jobs.

Each group may contain jobs and/or "child" groups. Groups may be nested to as many levels as necessary.

Groups and group hierarchies are global (not personal) settings. That is, when you put a job in a group, the job appears in that group for all adTempus users.

When you delete a group, adTempus will ask you whether you also want to delete all of the jobs within the group. If you choose not to delete the jobs, they will be moved to the next group up.

All jobs in a group inherit the security settings and job variables defined for the group.

Aside from security and job variable inheritance, Job Groups are strictly an organizational tool. Placing jobs in the same Group does not create any link or dependency between those jobs, or in any way affect the way that the jobs execute. If you need to be able to limit the number of related jobs that are executing at the same time, you can use Job Queues to do this.

See Also

Job Group Properties

Job Queues

Group Page

The group page defines general settings for the group.

Name

Provide a descriptive name for the group (255 characters maximum). The name can be changed at any time. The name must be unique within the parent group.

Description

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

See Also

Job Group Overview

Variables Page

The **Variables** page defines Job Variables that are inherited by all Jobs in this group. You may add, modify, or delete Job Variables, or override the values of Variables that the group has inherited.

A Group inherits Variables from its parent group. The top-level (root) group inherits the server-level variables.

See Also

Job Group Overview

Security Page

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

Permissions set for the Group are inherited by jobs within the Group.

The following permissions apply to Job Groups:

Permission	Description
Full Control	Permission to perform all actions on the group.
List/Reference	Permission to list this group.
View	Permission to view the properties of the group.
Create jobs and groups in this group	Permission to create new jobs or groups in this group, or move existing jobs to this group.
Modify	Permission to modify the properties of the group.
Delete	Permission to delete the group.
Administer security	Permission to change the security settings for the group.
Change owner	Permission to take ownership of the group.

Job Instance

Job Instance

Each time a job is executed, this is called an "instance" of the job. Information about each instance is saved in adTempus, based on the job's history retention settings.

You can review the history for a job by selecting the job in the Console's Job List View or using the Execution History Query.

Instance Acknowledgment

When a job instance ends with an error or warning status, it is flagged for review and appears in the Failed Jobs view in the Console. An authorized user can "acknowledge" the failure or warning, clearing the flag status of the instance and removing it from the view.

See Also

Job Instance Details

Execution History Query

Job Page

The **Job** page lists information about the instance.

Instance

The instance ID for this instance. Instances are numbered sequentially.

Status

The status of the instance.

Execution Start

The date/time at which execution started (dates and times are shown using the time zone of the server).

Execution Finish

The date/time at which execution finished (dates and times are shown using the time zone of the server).

Elapsed Time

The total elapsed time for the job (the difference between the Start and Finish times).

Processor Time

The amount of CPU time used by the job. Note that processor times listed for jobs and steps only include program execution tasks and script tasks. Other tasks do not update the processor times.

See Also

Job Instance Overview

Log Page

The **Log** page lists messages logged by adTempus related to this job. Messages for the job and for individual steps are listed. To view the details of a message, double-click the appropriate row in the list.

Steps Page

The **Steps** page lists information for each step of the job. For more information on a step, double-click the appropriate row to display the Step Details.

> Only steps that were executed (or that adTempus attempted to execute) are listed. If the step was never reached (e.g., because the job was aborted or failed before reaching the step, or because a job control action within the job caused the step to be skipped over), the step will not be listed here.

See Also

Step Details

Job Instance Overview

Captured Files Page

The **Captured Files** page lists any files captured by the job (using File Capture actions or the Capture Console window mode).

Captured console output has the name "console output.txt".

To view or save a captured file, right-click the appropriate row for a menu of options.

- Select **Open** to open the file using the default application for the file type.
- Select **Save As...** to save the file to the location of your choice. This saves a copy of the file that you can do with as you wish; the captured file is still saved on the server as part of the job history.

Comments Page

The **Comments** page allows authorized users to record comments about this job instance. For example, if the job failed, the system operator can record notes about what error resolution procedures were followed.

Job Queue

Job Queue Properties

The properties of a Job Queue are divided among the following pages:

Queue

Distributed Scheduling

Variables

Security

See Also

Job Queue Overview

Properties

Queue Page

The **Queue Page** defines the basic settings for the Job Queue.

Name

Enter a name for this Queue. The name is limited to 255 characters and must be unique.

Enabled

Check this box to enable execution of jobs in the Queue. If this box is not checked, jobs in the Queue will not be executed.

Resubmit pending jobs when adTempus restarts

If jobs were queued (awaiting an available execution slot in the queue) when the adTempus service was shut down, adTempus will resubmit those jobs on startup if this option is checked. Otherwise, they will not be resubmitted.

For example, suppose this Queue has an execution limit of 1 (only one job can run at a time). When the adTempus service is shut down, there is one job executing and three more from this queue that have been triggered and are waiting to be executed. If this option is checked, when the service is restarted adTempus will resubmit those three pending jobs to the queue, in the same order they were in at shutdown.

Note that any jobs that were executing at shutdown are not affected by this option, only jobs waiting to be executed.. The restart options for each job determine whether that job will be restarted.

Limit the number of jobs from this queue that can run concurrently

Use this option to limit the number of jobs from this queue that can run at the same time. See the Job Queue Overview topic for more information.

If Distributed Scheduling is in use, this limit applies individually to each computer. For example, if the limit is set to "1" but the Queue is configured to run jobs on two different Agents, each Agent has a limit of one concurrent job, meaning that there may be two jobs (one on each Agent) running in this Queue at any time. If the Queue is configured for Load Balancing, you can additionally limit the number of concurrent jobs against all computers. See the Distributed Scheduling properties for more information.

Description / Notes

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

See Also

Job Queue Overview

Distributed Scheduling Page

When you are using the Distributed Scheduling features of adTempus, the **Distributed Scheduling** settings for the Queue determine which computers the jobs assigned to the Queue will be executed on. If the server you are managing is not configured as a Distributed Scheduling Master computer, this page of the Queue's properties will not be available.

> In adTempus version 2.x, Distributed Scheduling options were set individually for each job. Beginning with adTempus 3.0, these options are set at the Queue level to simplify job management.

Agent Mode

Select the mode adTempus should use when running the jobs in this Queue.

Limit the number of jobs from this queue that can run concurrently across all target computers

This option is only available if the **Agent Mode** is set to "Load Balance."

Use this option to limit the maximum number of jobs that can run concurrently in this Queue across all computers assigned to this Queue. For example, if the limit is set to "1," only one job at a time will run in this Queue, regardless of how many Agents are assigned to the Queue, or what limit has been set on the Queue page.

Run on Master

Check this option to run the job on the Master computer, in addition to any specified agents.

- If you select the **Slave** or **Mirror** modes, the job will be run on the Master and on any specified agents. If no agents are specified, the job will run only on the Master.
- If you specify the **Load Balance** mode, adTempus will consider the Master and all specified agents when determining which computer to run the job on.

Run on Remote Agents

Specify the Agents on which the job should run.

When you add or edit an agent, adTempus displays the Remote Execution Options window, which allows you to customize the way the job will be run on this agent.

Evaluate Conditions On

This option is only available if the **Agent Mode** is set to "Slave."

Use this option to specify where job-level conditions for the job should be evaluated (step-level conditions are always evaluated on the computer where the job is actually running).

- **Master:** When the job is triggered, the Master computer evaluates the job-level conditions for the job. If the conditions are met, the execution command is sent to the Agent computers and the job is run. If the conditions are not met, the job is not run.
- **Slave:** When the job is triggered, the Master computer sends the execution commands to the Agents. Each Agent then evaluates the conditions for the job and determines whether the run the job. Thus it is possible that the job will execute on some Agents but not on others.

See Also

Distributed Scheduling Overview

Job Queue Overview

Variables Page

The **Variables** page defines Job Variables that are inherited by all Jobs in this Queue. You may add, modify, or delete Job Variables, or override the values of Variables that the Queue has inherited.

A Queue inherits the server-level variables.

Security Page

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

Permissions set for the Queue are inherited by jobs within the Queue.

The following permissions apply to Job Queues:

Permission	Description
Full Control	Permission to perform all actions on the Queue.
List/Reference	Permission to list this Queue.
View	Permission to view the properties of the Queue.
Associate (assign jobs to this queue)	Permission to assign jobs to this Queue.
Modify	Permission to modify the properties of the Queue.
Delete	Permission to delete the Queue.
Administer security	Permission to change the security settings for the Queue.
Change owner	Permission to take ownership of the Queue.

Permission to create new Queues is defined through the Default Queue Security options.

Remote Execution Options

These options determine how jobs from a Queue will be run on the selected Agent. Changing these settings only affects the Agent's behavior for the Queue being edited: if the Agent is also used by other Queues, it may use different settings for those Queues.

To change these settings, select an Agent on the Distributed Scheduling page of the Job Queue's properties, and click the **Edit...** button.

Enable execution on this agent

Check this box to run jobs from this Queue on this Agent. If the option is not checked, the Agent remains associated with the Queue, but jobs from the Queue will not be run on the Agent.

Queue execution commands

This option determines what happens if adTempus cannot deliver an execution command to the Agent (because the Agent is not running or the network connection is broken).

- If this option is checked, adTempus will queue the command. Once the connection is re-established, the command will be delivered to the Agent and the job will be run.
- If this option is not checked, the command will be discarded, and the job will not be run on the Agent.

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 resources for a job are managed from the Resources page of the job's properties.

See Also

Network Access

Network Resource Properties

Resources Overview

Network Resource Properties

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 letter mappings may fail for a variety of reasons. Therefore you should avoid using drive letters if possible. See the Network Access topic for more information.

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 Properties

The Notification Action properties are divided among the following pages:

Notification

Attachments

Notification Page

Define the basic properties for a Notification Action.

Recipients

Add, edit, or remove the notification recipients who will receive the notification message.

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.

The subject may contain Job Variables, which will be expanded when the message is sent. For more information, see Specifying Notification Subjects, Messages, and Severities at Run-Time.

Message to send

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

The default message varies depending on the event that triggered the action. Generally the message contains the job name and instance ID, step number, and a description of the event that triggered the action.

If you specify a message, it will be sent instead of the default message. If you wish to include the default message as part of your message, include the token %DEFAULTMESSAGE% in your message. The token will be replaced with the default message.

The message may contain Job Variables, which will be expanded when the message is sent. For more information, see Specifying Notification Subjects, Messages, and Severities at Run-Time.

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.

See Also

Notification Action Overview

Notification Recipients

Attachments Page

The Attachments page allows you to specify files that will be attached to the notification message. In the Files list you can add, edit, or remove File Specifications that define the files to be attached.

Note the attached files are not captured and saved as part of the job's history. If you want to save the files, use a File Capture action instead.

If you check the **Include captured console output** option, any console output that has been captured for the current step (if this action is being called by a step) or for the entire job (if this action is being called by the job) will be attached to the message.

Specifying Notification Subjects, Messages, and Severities at Run-Time

When you use a Notification Action, File Capture Action, or Notification Task, you specify the subject and message text to be used in the notification messages sent by adTempus.

In many cases it is useful to be able to customize the content of these messages at run-time. For example, if you have a notification message that gets sent when a job starts because of a File Trigger, you might want the name of the file to be included in the message.

This can be accomplished using Job Variable tokens.

The predefined %DEFAULTMESSAGE% token can be used in Notification Actions. It expands to the default message for the event that triggered the action; the message describes the event that occurred (such as "Job XXX Failed").

Notification Severity

You can also override the notification severity using a script (run by a Script Condition, Script Execution Task, etc.). Simply set the **MessageSeverity** variable to the severity you want to use (1 through 9). When you make this change, it applies only to the next notification action taken by the job (i.e., the next Notification Action, Notification Task, or File Capture Action executed by the job).

If you want to override the severity for all subsequent notifications in the entire job, set the **MessageSeverityAll** variable instead.

For an example of this, see the Computer Monitor example.

Notification Address

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.

See Also

Notification Address Properties

Notification Recipient

Address Page

The Address page defines the general properties for the address.

Address Type and Address

Select the type of address, and specify the address. The following address 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 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	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

	the SMPP Protocol.	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".

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	Message limit specified for	Result

for service provider?	address?	
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.

See Also

Configuring SMTP notification

SMS Messaging

Sending notification for jobs

Schedule Page

The **Schedule** page allows you to define a schedule that determines when this address will be used. For example, you may want messages sent to your pager only during certain hours.

Override schedule when severity is...

Check this option to override the schedule for messages that meet certain criteria. For example, you may want to receive notification messages on your pager only between 9 AM and 5 PM, except if the message has a severity of 9.

Schedule

Specify as many criteria as necessary. When a notification message is sent to the recipient, adTempus compares the current time to the criteria, and uses an address only if at least one "include" criterion is met and no "exclude" criteria are met.

Notification Group

Notification Group

A Notification Group is a group of Notification Recipients or other groups. When a notification message is sent to a group, it is sent to all members of the group (subject to the conditions defined for each member).

By using the schedule option available for each member, you can use a Notification Group to reflect an on-call duty schedule. For example, you may have certain people who handle server problems during the business day and others who are on-call for after-hours problems.

Jobs that you set up don't need to know about the details: they can be configured simply to notify a group. Within the group, you can specify which members should receive messages at which times.

Messages are sent to Notification Groups using Notification Actions, File Capture Actions, and Notification Tasks.

Notification Groups are managed from the Notification Recipients folder in the adTempus Console.

See Also

Notification Group Properties

Notification Overview

Notification Action

File Capture Action

Notification Task

Group Page

The **Group** page defines the Job Group and its members.

Name

Enter a unique descriptive name for this group.

Enable this group

If this option is not checked, the group can still be used on jobs, but it will not receive notification messages until it is enabled.

Severity

Use this option to specify the severity conditions that a notification message must meet in order to be sent to this group. Any message that does not meet the condition will not be sent to members of this group.

You can specify severity conditions for an individual member in the properties for that member.

Members

Add, change, or remove group members.

Security Page

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

The following permissions apply to notification groups:

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

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

Notification Group Member

Notification Group Member Properties

The Notification Group Member properties define settings for a member of a Notification Group. The options are divided among the following pages:

Member Page

Schedule Page

See Also

Notification Group

Member Page

The **Member** page contains basic settings for the member.

Settings here only affect messages sent to the member through this group. For example, disabling the recipient here only prevents the recipient from receiving messages through this group.

Recipient

Select the recipient (an individual or another group) who is the member.

Enable this member

If this option is not checked, the member does not receive notification messages sent to the group.

Severity

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

Schedule Page

The **Schedule** page allows you to define a schedule that determines when this member will be notified.

Override schedule when severity is...

Check this option to override the schedule for messages that meet certain criteria.

Schedule

Specify as many criteria as necessary. When a notification message is sent to the group, adTempus compares the current time to the criteria, and notifies a member only if at least one "include" criterion is met and no "exclude" criteria are met.

Notification Recipient

Notification Recipient

A Notification Recipient, or individual, represents a single person or entity who receives notification messages. Each recipient may have any number of addresses, which may be used in different circumstances. For example, an individual may have an e-mail address that receives all notification messages sent to that person, and a pager e-mail address that only receives urgent notification messages.

Each Notification Recipient can be assigned to any number of Notification Groups, which are used simplify notification administration.

Messages are sent to Notification Recipients using Notification Actions, File Capture Actions, and Notification Tasks.

Notification Recipients are managed through the Notification Recipients folder in the adTempus Console.

See Also

Notification Recipient Properties

Notification Overview

Notification Groups

Recipient Page

The Recipient page defines the properties for the Notification Recipient.

Name

Provide a unique name for this recipient.

Enable this recipient

If this option is not checked, the recipient can still be used on jobs, but will not receive notification messages until it is enabled.

Severity

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

Addresses

Add, modify, or remove addresses for this recipient. The recipient must have at least one address, or it will never receive notification.

When a notification message is sent to a Recipient, the message is sent to each of the recipient's addresses (depending on the selection criteria defined for each address).

See Also

Notification Recipient Overview

Security Page

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

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.

See Also

Notification Recipient Overview

Notification Setup Window

Notification Setup Window

The Notification Setup Window (**Tools > Server Options > Notification Setup**) contains the server settings used for sending notification messages (e.g., for Notification Actions).

The settings are divided among the following pages:

- Primary SMTP Server (for e-mail notification settings)
- Backup SMTP Server (for e-mail notification settings)
- Notification Service Providers (for other notification services, such as SMS and instant messaging)

SMTP Servers

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.

By default messages will use "adTempus" as the name of the sender. If you want to use a different name, specify the return address in this form:

    Desired Name <returnaddress@example.com>

Replace *`Desired Name`* with the name you want to appear in messages. Be sure to include the angle brackets (<>) around the address.

You can specify different return addresses for different jobs by setting the **NotificationFromName** and **NotificationFromAddress** Job Variables for a job, group, queue, step, etc.

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.

SMTP Servers

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.

By default messages will use "adTempus" as the name of the sender. If you want to use a different name, specify the return address in this form:

> *Desired Name <returnaddress@example.com>*

Replace *Desired Name* with the name you want to appear in messages. Be sure to include the angle brackets (<>) around the address.

You can specify different return addresses for different jobs by setting the **NotificationFromName** and **NotificationFromAddress** Job Variables for a job, group, queue, step, etc.

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.

Notification Service Providers

Notification Service Providers

adTempus supports the delivery of notification messages using several different technologies, such as SMS and instant messaging.

In order to use these technologies, you must first define a Notification Service Provider in adTempus. A Notification Service Provider simply provides the information adTempus needs to connect to the carrier or server and deliver messages.

For example, if you want to be able to send SMS messages to your cell phone, adTempus needs to know how to connect to your cell phone company to deliver the message.

adTempus supports the following messaging services in addition to SMTP e-mail:

- SMS notification through a network gateway
- Instant messaging using the Jabber protocol

SMS Network Service Provider

The SMS Network Service Provider supports notification using the Short Message Service (SMS) protocol. Messages are delivered to the carrier over a network connection.

To deliver SMS messages, adTempus must connect to an SMS server. Two kinds of servers are available:

- Your company may operate or subscribe to an SMS gateway service. The gateway accepts SMS messages and automatically routes them to the proper carrier for delivery to the device. That is, messages for both your SkyTel pager and you Cingular cell phone can go to the same gateway.
- The carrier may provide public access to its SMS server, allowing adTempus to connect directly to it. When you use this option, messages for each device must be sent by adTempus directly to the SMS server for the company that provides service for the device. That is, messages intended for your SkyTel pager must be delivered to the SkyTel SMS server; messages intended for your Cingular cell phone must be delivered to the Cingular SMS server.

Finding the Proper Settings

To configure the service provider, you will need to know the network address that adTempus should connect to, and any user information needed to connect to the server.

If you are using a gateway service, check with that service for more information.

If you want to connect directly to the carrier, you will need to check with the carrier for the connection information. adTempus has a list of carriers whose public servers we know about; you can check to see if your carrier is on that list by using the Import Settings... button.

See Also

SMS Network Service Provider Properties

SMS Service Messaging

Notification Service Providers Overview

Jabber Service Provider

The Jabber Service Provider allows you to send notification messages using the Jabber instant messaging protocol.

For more information on Jabber, Jabber clients, and Jabber servers, see www.jabber.org.

Once you have configured a provider, you can create Notification Addresses that use it.

See Also

Jabber Provider Properties

Notification Service Providers Overview

Notification Task

Notification Task

The Notification Task allows you to send notification messages, and optionally capture and attach files. The Notification Task provides the same functionality as the Notification Action and File Capture Action. Those actions can only be run as Responses for a job or a step, but the Notification Task, because it is a task, can be the primary function that a job or step performs.

Generally, you would use a Notification or File Capture Action when you want to send a notification message in response to something that occurs during job execution (i.e., the notification is of secondary importance). You would use Notification Task when the notification is of primary importance.

For example, if you wanted to monitor a computer and send a notification message when it failed, you would create a job that uses a Computer Monitor Trigger and a Notification Task.

See Also

Notification Task Properties

Notification Action

Tasks Overview

Notification Message Page

Name for this step (optional)

Optionally, specify a descriptive name for the step. Otherwise a description will be automatically generated.

Description / Notes

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

Enable this Step

When this option is checked, the step is enabled and will be executed. If this option is not checked, the step will be skipped when the job is run.

Subject of message

Provide a subject for the message that will be sent.

The subject may contain Job Variables, which will be expanded when the message is sent. For more information, see Specifying Notification Subjects, Messages, and Severities at Run-Time.

Message to send

Specify the message to be sent.

The message may contain Job Variables, which will be expanded when the message is sent. For more information, see Specifying Notification Subjects, Messages, and Severities at Run-Time.

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.

Recipients Page

Recipients

Add, edit, or remove the notification recipients who will receive the notification message.

Attachments Page

The Attachments page allows you to specify files that will be attached to the notification message. In the Files list you can add, edit, or remove File Specifications that define the files to be attached.

Save these files in the job history

If this option is checked, the files you specify will also be captured and saved as part of the job's history. If the option is not checked, the files are sent, but are not saved with the job history.

Include captured console output

If this option is checked, any captured console output files from previous steps will be attached to the message.

Conditions Page

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.

Responses Page

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 Notification 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 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).

See Also

Responses Overview

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.

ℹ	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.

See Also

Process Condition Properties

Conditions Overview

Process Condition Properties

The options on the Process Condition Properties window 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.

Condition Wait

This option determines whether adTempus will wait for the condition to be met:

- **Do not wait for condition to be met.** If the condition is not satisfied the first time adTempus checks it for a given instance, adTempus treats the condition as "Failed."
- **Wait until the condition is met.** If the condition is not satisfied, adTempus will wait until it is satisfied. The job or step will show a status of "Waiting for Condition."
- **Wait up to __ minutes for the condition to be met.** adTempus will wait up to the specified time limit for the condition to be satisfied. If the condition is satisfied within that time limit, job execution will continue immediately (as

depending on other conditions). If the condition is not satisfied at the end of the time limit, adTempus treats the condition as "Failed."

See the Condition Polling topic for more information.

Once satisfied, do not re-evaluate for this instance

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

See Also

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.

See Also

Process Termination Task Properties

Tasks Overview

Options Page

The **Options** page of the Process Termination Task contains options for terminating the process.

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.

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 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.

Description / Notes

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

See Also

Process Termination Task Overview

Conditions Page

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.

Responses Page

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).

See Also

Process Termination Task Properties

Process Termination Task Overview

Responses Overview

Process Trigger

Process Trigger

The Process Trigger allows you to trigger a job whenever external processes start, end, or exceed a specified memory limit.

For example, you may have a server process that occasionally fails. Using a Process Trigger, you can have adTempus monitor this process and run a job (to restart the process, send a notification message, etc.) when it fails.

See Also

Process Trigger Properties

Triggers Overview

Process Trigger Properties

This page defines the conditions for triggering a job.

Optional Descriptive Name

Optionally, provide a descriptive name for this trigger. Otherwise, a description will be created automatically based on trigger settings.

Enable this Trigger

If this box is checked, the trigger is enabled and will cause the job to run. If this box is not checked, adTempus will ignore this trigger.

Trigger Conditions

Select the conditions under which the job will be triggered. You must select at least one option.

Processes

Specify the process names that adTempus should monitor. You must specify the name only, not the path, using the form that appears in the Processes list in the Task Manager.

If you specify more than one process, adTempus will trigger the job whenever any of the processes you list satisfies the other criteria.

If you do not specify any processes, adTempus will trigger for every process on the system, which is probably not what you want.

Minimum Time Between Triggers

Specify the minimum time (in seconds) that must elapse after this trigger fires before adTempus will fire it again. Once adTempus triggers the job, it will ignore all process changes until the specified time has passed.

Use a script to select processes

adTempus will run the script you specify. Information about the process will be passed to the script using the JobVariables collection. Your script then evaluates the information and determines whether the trigger should fire.

adTempus calls the script only for events that have satisfied all of the other criteria for the trigger.

Description / Notes

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

Job Variables

The Process Trigger sets the following Job Variables, which can be used by a selection script or by other scripts or notification messages in the job.

Parameter Name	Description
WMIEvent	An ISWbemObject interface to an __InstanceModificationEvent returned by a WMI query. **Note:** The event parameter is only available to the selection script called for the trigger. It is not available to scripts run later in the job, as most script parameters are. If you need information from the event later in your job, you must use the selection script to extract the information and save it in other parameters.

See Also

Using Job Variables

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.

Though the Program Execution Task can be used to run scripts as well, the Script Execution Task is tailored to script execution and is often a better choice.

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.

See Also

Program Execution Task Properties

Script Execution Task

Tasks Overview

Steps Overview

Network Access for Scheduled Tasks

General Page

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. You do not need to include quotation marks around the target: they will be added for you if necessary.

Only specify the program or document path and name here. Any parameters or options that need to be passed to the program must be entered in the command-line parameters box below.

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.

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 Job Variables 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.

Startup Directory

Specify the startup ("working") directory to be associated with the program. For most programs, it is not necessary to specify this directory.

Window Mode

The Window Mode option indicates how you want the program to appear on the desktop.

The following two options make the program invisible to any users logged in to the computer, regardless of the User Interaction settings for the job.

- **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".

The following three options may make the program visible to users logged in to the computer, depending on the User Interaction settings for the job. If the job's User Interaction settings cause the job to be run in "Hidden" mode, the program will run in Hidden mode even if you have selected one of these options. If the job runs in the "Console Session" or the "User's Session," the program will appear on the user's desktop in the selected mode:

- **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.

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.

See Also

Script Execution Task

Program Execution Task Properties

Program Execution Task Overview

Advanced Page

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.

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. 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.

  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.

See Also

Program Execution Task Overview

Conditions Page

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.

See Also

Program Execution Task Properties

Program Execution Task Overview

Conditions Overview

Variables Page

The **Variables** page allows you to define Job Variables for this step. You can add new Variables, or override the values of Variables inherited from the Job. Any Variables you define or override here affect only this step of the Job.

To set Variables that apply to the entire job, use the Variables page in the Job properties.

Any variables that have the Include in program's environment variables option checked will be set as Windows environment variables when the program is run. Variables that do not have this option checked are available for use in adTempus but will not be added to the program's environment.

The **Variables** page and functionality replaces the **Environment** page found in adTempus version 2.x. If you have upgraded from version 2.x, any environment variables that you had defined for this step have been converted to Job Variables.

See Also

Program Execution Task Properties

Program Execution Task Overview

Job Variables Overview

Responses Page

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.

See Also

Program Execution Task Properties

Program Execution Task Overview

Responses Overview

Remote Agent

Remote Agent

A **Remote Agent** is a remote computer on which jobs managed by the Master computer can be run. See the Distributed Scheduling Overview for more information on Remote Agents.

Before you add a Remote Agent to the Master computer, you must install the adTempus software on the remote computer, and configure adTempus on that computer to run as an Agent. You can then create the Remote Agent definition on the Master, through the Remote Agents folder in the adTempus Console.

Once you have defined Remote Agents, they can be associated with a Job Queue by editing the queue's Distributed Scheduling properties. Jobs assigned to the Queue will then run on the Remote Agent.

See Also

Distributed Scheduling Overview

Remote Agent Properties

Agent Page

The Agent Page defines the basic settings for the Remote Agent.

Agent Address

Specify the network name or IP address of the remote computer. The adTempus service must be installed on this computer, and adTempus must be configured to run as an Agent.

Optional Name

Optionally, provide a descriptive name for the Agent to be used instead of the address in the user interface.

Enable this agent

Check this box to enable the Agent.

If the Agent is not enabled, no jobs will be run on the agent.

See Also

Remote Agent Properties

Variables Page

The **Variables** page allows you to define Job Variables for this Agent. You can add new Variables, or override the values of Variables inherited from the server level.

Variables you define here will be available to all jobs assigned to this Agent. When a job runs on more than one Agent, it receives the appropriate variables on each Agent.

Job Variables can be used to accommodate differences in the file structure or environment between different Agent computers. For example, suppose you have a job that is being mirrored to three different Remote Agents. This job runs program "myprogram.exe". On two of the Agents, this program is installed in folder "c:\myprogram," but on the third it is installed in folder "c:\program files\myprogram". Your job needs to be able to find the program in the correct folder on each of the three computers.

To accomplish this, you would create a Job Variable named "MyProgramPath" for each of the three Remote Agents, and set this variable to the correct path for that Agent. In your job, you would set the target of your Program Execution step to "%MyProgramPath%\myprogram.exe". When the job is executed, adTempus will replace "%MyProgramPath%" with the correct path on each Agent.

Security Page

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

The following permissions apply to Remote Agents:

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

Permission to create new agents is controlled through the Default Remote Agent Security options.

Response

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 sequentially, one at a time. 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.

See Also

Response example

Response Properties

Actions

Response Properties

A Response consists of events and 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. See the Response Event topic for more information about specifying events. The help topic for the Responses page of the job or task for which you are adding responses describes the events that are available for the job or task.

You can specify any number of events for the Response. The Response will execute each time any of its events occurs. That is, if you have selected the events "Job Begin" and "Job End," the Response will execute once when the job begins and again when the job ends.

Actions

Add, modify, remove, or reorder the Actions that will be executed when this response is triggered. See the Actions topic for information on the available actions.

The actions will be executed sequentially in the order they appear in the list. Each Action must complete before the next Action executes.

See Also

Response Overview

Response Event

A Response Event defines an event that will trigger the Response. In the Response Event properties 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.

Use a script to determine whether the Response will be activated

Use this option if you want to run a Script to determine wether the Response should execute. adTempus executes the script when the event occurs during job execution. The script then returns a True or False result to indicate whether the Response should execute.

For example, if you have selected the "Step Ends" event, adTempus will run your script when the step ends. Your script then determines whether the Response will execute.

See Also

Event Overview

Responses Overview

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.
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."
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.

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.

If a Schedule Trigger has more than one schedule, the job will be executed whenever any of the schedules is satisfied.

Multiple Schedules

A job can have more than one Schedule Trigger, and each Schedule Trigger can have any number of Schedules. The job will execute whenever any of the schedules on any of the Schedule Triggers is satisfied. However, if any Schedules or Schedule Triggers produce overlapping times, adTempus will still only run the job once. For example, even if two separate Schedules specify that the job should run at 12:30 PM on Monday, the job will only run once at that time.

For example, you can have one schedule that defines the run times for weekdays, and a second schedule for weekends.

You might want to have separate Schedule Triggers (instead of multiple Schedules on a single Schedule Trigger) if you want to use different time zone or holiday options for different schedules: all Schedules for a Schedule Trigger share the same options and holidays, but each separate Schedule Trigger for a job has its own options and holidays.

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

See Also

Triggers Overview

Schedules Page

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.

Enable this trigger

Check the **Enable** box to use this trigger. If the box is not checked, adTempus will ignore all schedules defined for this trigger.

Run according to the following schedules

Specify the Schedules for the trigger.

See Also

Schedule Trigger Overview

Schedules

Options Page

On the **Options** page you specify clock change and time zone options for this trigger.

System Clock Changes

These settings determine what adTempus will do if changes to the computer's clock affect scheduled execution times for the job.

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.

Time Zone

The time zone setting determines which time zone adTempus should evaluate this trigger in.

Local time zone on the computer where the job is triggered

The trigger will be evaluated in the time zone of the where the job is triggered. That is, if the job is scheduled to run at 11:30 AM, it will run at 11:30 AM local time on the computer triggering the job.

Note: If this job is in a Distributed Scheduling queue, this option causes adTempus to use the time zone of the Master computer for Slave and Load

Balance queues (because the Master triggers the job for these configurations). For Mirror queues, the time zone of the Agent that runs the job will be used.

A specific time zone

The trigger will be evaluated in the specified time zone. For example, if the job is scheduled to run at 11:30 AM and you select the Pacific time zone, the job will run at 11:30 AM Pacific time. If the server running adTempus is located in the Eastern time zone, the job will execute at 2:30 PM local (Eastern) time.

The time zone of the Master with which the Agent is associated

This option is only available if you are configuring the job on a Distributed Scheduling Master computer. When this option is selected, the time zone of the Master will be used to evaluate the scheduled times, even for Mirrored jobs.

See Also

Schedule Trigger Overview

Triggers Overview

Holidays Page

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.** 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 non-holiday 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 non-holiday.** The job is run on the nearest subsequent day that is not a holiday (but may be a weekend)

See Also

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.

See Also

Schedule Properties

Schedule Trigger

Date Selection Page

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 according to this schedule).

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

See Also

Day Selection Properties

Shared Schedules

Schedule Properties

Schedule Overview

Time Selection Page

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.

See Also

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.

See Also

Script Properties

Using Scripts in adTempus

Script Page

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.

> ℹ 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 the language that the script uses. The following languages are supported by default:

- VBScript
- VB.NET
- C#
- JScript

If your server has a Windows Scripting Host-compatible language (such as perlscript) that is not listed, you can enter the name of the script language and adTempus will attempt to use it.

Included Script Libraries

If your script uses procedures from any Script Libraries, select them here to make them available to the script. The libraries you use must be written in the same language as the script, therefore the list of available libraries will change when you change the Language.

Referenced Assemblies

If you have selected one of the .NET languages (VB.NET or C#), your code may reference other .NET assemblies.

Script

Provide the body of the script.

When you initially select the script language, the script body will be filled with a template script for you to use as a starting point. The contents of the template will depend on the context of the script (for example, the template for a Script Execution Task will be different than the template for a Condition Script).

In most cases, your script must return a result to adTempus to indicate the outcome of the script and tell adTempus how to proceed. See the Script Overview topic for more information about how to return a result to adTempus.

Maximum Execution Time

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

If this script is used in a Script Execution Task, which also allows you to specify a time limit, adTempus will use the most restrictive of the maximum run times specified for the Script and the Task.

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.

Quick 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.

See Also

Script Overview

Security Page

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

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 scripts is controlled through the Script Security settings.

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.

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, your adTempus job is going to be patiently waiting.

See Also

Script Action Properties

Responses Overview

Actions

Using Scripts

Script Action Properties

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.

See Also

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.

See Also

Script Condition Properties

Conditions Overview

Script Overview

Returning Results from Scripts

Script Condition Properties

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.

⚠	Your script must return a value of either **True** (condition is satisfied) or **False** (condition is not satisfied). Any other value will cause the condition to fail.

Condition Wait

This option determines whether adTempus will wait for the condition to be met:

- **Do not wait for condition to be met.** If the condition is not satisfied the first time adTempus checks it for a given instance, adTempus treats the condition as "Failed."
- **Wait until the condition is met.** If the condition is not satisfied, adTempus will wait until it is satisfied. The job or step will show a status of "Waiting for Condition."
- **Wait up to __ minutes for the condition to be met.** adTempus will wait up to the specified time limit for the condition to be satisfied. If the condition is satisfied within that time limit, job execution will continue immediately (as depending on other conditions). If the condition is not satisfied at the end of the time limit, adTempus treats the condition as "Failed."

See the Condition Polling topic for more information.

Once satisfied, do not reevaluate for this instance

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

See Also

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 and .NET. The script may be a script that is stored in adTempus or it may be an external script file.

Although external script files can also 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.

See Also

Script Execution Task Properties

Scripts Overview

Tasks Overview

Steps Overview

Script Page

The **Script** page defines script to be run.

Name for this step (optional)

Optionally, specify a descriptive name for the step. Otherwise the name of the target script 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 to Execute

Your script may be either an internal script (stored within adTempus and optionally shared with other jobs) or an external script (an external script file). The way the script is executed, and the options available to it, are the same either way.

Execute a script stored in adTempus

Select an existing script or create a new one. See the Scripts topic for information on working with internal scripts.

Execute a script stored in an external file

Specify the script file to execute. You must also specify the **Language** of the script. If you select a language other than one of those listed, the language name you enter must be a scripting language recognized by the Window Scripting Host on the server.

Maximum execution time

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

If you are running an internal script, adTempus will use the lesser of the value you specify here and the value specified on the Script's properties.

See Also

Script Execution Task Properties

Script Execution Task Overview

Scripts Overview

Options Page

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**. Your script will return a numeric result (see Returning a Result From a Script). The step will be reported as successful if the result meets the criterion you specify. By default, adTempus expects your script to return a value of "0" if it succeeds, consistent with the convention used by most programs.

- **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.

See Also

Script Execution Task Properties

Script Execution Task Overview

Returning Results from Scripts

Conditions Page

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.

Variables Page

The **Variables** page allows you to define Job Variables for this step. You can add new Variables, or override the values of Variables inherited from the Job. Any Variables you define or override here affect only this step of the Job.

To set Variables that apply to the entire job, use the Variables page in the Job properties.

Responses Page

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 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 Shared 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.

Script Library Code and Namespaces

The various scripting languages all handle namespaces slightly differently, and this may affect the way you are able to define and call code in your script Library.

WSH Scripting Languages

The Windows Scripting Host (WSH) languages (VBScript, JScript, and other WSH-compatible languages) do not support multiple namespaces. As a result, 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.

VB.NET

All script and library code used by adTempus is placed in the global namespace. Code for each Script Library must be contained by a public Module (by default the Script Library Editor will set the module name to match the library name, but this is not required).

Therefore you may have variables and procedures in different libraries that have the same name, but you may need to qualify a the variable or procedure name using the library name when you use it in a script.

For example, you have created a Script Library named "MyLibrary", and the module defined in it is also named "MyLibrary." Within that module you have a public function named "MyFunction."

When you reference this library from a script, you can call your function using either "MyLibrary.MyFunction" or "MyFunction".

However, if you create a second library that also contains a function named "MyFunction", and both libraries are referenced by your script, "MyFunction" by itself is ambiguous, and you will need to call "MyLibrary.MyFunction".

C#

All scripts and libraries by are created to use the "UserScript" namespace. You can change the namespace if you wish, and make references across namespaces as necessary. The "adTempus" object used for interaction with adTempus is always in the "UserScript" namespace.

See Also

Script Library Properties

Overview of using scripts in adTempus

Script Page

The Script page defines the script code for the Script 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.

Referenced Assemblies

If you have selected one of the .NET languages (VB.NET or C#), your code may reference other .NET assemblies.

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.

When you initially select the script language, the script body will be filled with a template script library for you to use as a starting point.

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.

See Also

Script Library Properties

Script Library Overview

Script Overview

Security Page

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

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 scripts is controlled through the Script Security settings.

Server Options

General 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 backed up each day.

This option is only available if you are using the SQL Server Express instance installed with adTempus. If you are using your own database server, you must make your own provisions for database backup.

Security Settings Page

The options on the **Security Settings** page are the default security settings for all adTempus objects, and also control access to adTempus.

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

> ℹ 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 to the server" permission on this page (see "Server Permissions" below).

Object Permissions

The security permissions specified here are inherited by all objects in adTempus. For example, if you grant "View" permission to a user (or group), that user (or group) will be able to view all objects (jobs, notification recipients, scripts, etc.) in adTempus.

You can grant permissions for different kinds of objects in other places in adTempus. For example, a in a common scenario you would grant "View" permission to all users here, allowing all users to view all adTempus objects. If you want to allow a group of users to create and modify jobs (but not create or modify any other objects), you would modify the sercurity settings for the root "Jobs" group to grant the appropriate permissions.

See the Security Inheritance and Security Configuration Guidelines topics for more information.

Server Permissions

This is also where you specify who is allowed to connect to this adTempus server. No matter what object-level permissions you have assigned to a user, the user cannot connect to the server unless you have assigned "Connect to the server" permission. Similarly, granting a user "Connect to the server" permission only allows that user to log in to adTempus through the adTempus Console. You still must grant object-level permissions to allow the user to view, create, and modify objects such as jobs.

See Also

Security Overview

Security Editor

Security Inheritance

Security Configuration Guidelines

Variables Page

The **Variables** page allows you to define Job Variables for this server. Any Variables you define here are inherited by all Job Groups and Job Queues (and through them, Jobs) on the server.

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.

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.

See Also

Service Control Task Properties

Tasks Overview

Steps Overview

Options Page

The **Options** page defines the basic settings for the Service Control Task.

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.

You may also type in the name of the service if necessary. For example, the service may exist on the Agents where the job will run, but not on the Master computer where the job is maintained. Since the service list would come from the Master, in this case you would have to type in the service name.

If you enter the service name yourself, you must enter the "service name," which is often not the same as the "display name" that appears in the Service Control Manager. The service name is the same name you use when controlling the service through the "net start" and "net stop" commands. If you do not know the service name, look in the Registry under key "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services". This key contains a subkey for each service; the key name is the service name.

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.

See Also

Using adTempus to automatically restart a failed service

Service Control Task Properties

Service Control Task Overview

Conditions Page

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.

Responses Page

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.

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.

To view or manage shared schedules, select the Shared Schedules folder in the adTempus Console.

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

See Also

Shared Schedule Properties

Schedule Page

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.

See Also

Shared Schedule overview

Selecting Days

Security Page

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

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 Shared Schedules is controlled through the Shared Schedule and Holiday Security window.

See Also

Shared Schedule overview

Specifying default Shared Schedule security

SMS Service Provider

SMS Network Service Provider

The SMS Network Service Provider supports notification using the Short Message Service (SMS) protocol. Messages are delivered to the carrier over a network connection.

To deliver SMS messages, adTempus must connect to an SMS server. Two kinds of servers are available:

- Your company may operate or subscribe to an SMS gateway service. The gateway accepts SMS messages and automatically routes them to the proper carrier for delivery to the device. That is, messages for both your SkyTel pager and you Cingular cell phone can go to the same gateway.
- The carrier may provide public access to its SMS server, allowing adTempus to connect directly to it. When you use this option, messages for each device must be sent by adTempus directly to the SMS server for the company that provides service for the device. That is, messages intended for your SkyTel pager must be delivered to the SkyTel SMS server; messages intended for your Cingular cell phone must be delivered to the Cingular SMS server.

Finding the Proper Settings

To configure the service provider, you will need to know the network address that adTempus should connect to, and any user information needed to connect to the server.

If you are using a gateway service, check with that service for more information.

If you want to connect directly to the carrier, you will need to check with the carrier for the connection information. adTempus has a list of carriers whose public servers we know about; you can check to see if your carrier is on that list by using the Import Settings... button.

See Also

SMS Network Service Provider Properties

SMS Service Messaging

Notification Service Providers Overview

Provider Page

This page defines general options for connecting to your SMS gateway.

adTempus has a list of settings for some carriers and gateways. Try the **Import Settings** button and see if you carrier is listed. If not, see the Overview topic for information on how to find the settings you need.

Provider Name

Provide a descriptive name for the carrier or gateway service.

Protocol

Select the network protocol used by the carrier's SMS server.

User ID

Specify the user ID/caller ID used when connecting to the SMS server.

Password

Specify the password used when connecting to the SMS server (available for SMPP servers only).

Server

Specify the name or address and port number for the SMS server.

Maximum Message Length

Most SMS providers specify a limit for the length of messages. Messages that exceed the limit may be rejected. Specify the maximum here, and adTempus will truncate messages that exceed the length.

Automatically split messages that exceed the maximum

When this option is checked, adTempus will split notification messages that exceed the maximum length into separate messages, each shorter than the limit. This allows adTempus to deliver the complete notification message to your SMS device, as a series of messages.

You can set an absolute maximum length for messages by using the limit on the Notification Address.

Import Settings

Click this button to import provider settings from a list of known providers. See the Import topic for more information.

See Also

SMS Network Service Provider Properties

SMS Network Provider Overview

Firewall Page

The options on the **Firewall** page allow you to specify the settings required to connect to the SMS server through a firewall or proxy server. These options are available only when connecting to an SMS server that uses the SNPP protocol.

Contact your network administrator to obtain the settings for your firewall.

See Also

SMS Network Service Provider Properties

SMS Network Provider Overview

Security Page

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

The following permissions apply to Messaging Service Providers:

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

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

Messaging Provider Import

Arcana Development maintains a list of SMS gateway providers that we know about. If yours is on the list, you can import the settings when setting up the provider in adTempus.

To check the list, click the **Import Settings...** button in the SMS Service Provider Properties window.

In the **Messaging Provider Import** window, select the **Download the latest list** option and click **Load**. (If your computer is not connected to the Internet, you can use the **Load from a file** option to load providers from a file installed with adTempus.)

Then look for your provider in the **Provider to Import** list and click **OK** once you find it.

If your provider is not on the list, you will need to configure the provider yourself.

SQL Server Job Task

SQL Server Job Task

The SQL Server Job task allows you to execute a SQL Server Agent job (a job defined within SQL Server) from adTempus.

This can be useful if you need to incorporate a SQL Server Agent job into a processing flow that is otherwise managed by adTempus. For example, you may want adTempus to execute a sequence like this:

1. Run a program that loads data into a SQL Server database.
2. Start a SQL Server job that processes the new data, and wait for that job to complete.
3. Run a program that produces reports based on the new data.

The SQL Server Job Task allows you to do this without the need to write your own program or database script to start and monitor the SQL Server job.

Responses can be executed based on the results of the SQL Server job.

See Also

SQL Server Job Task Properties

General Page

The **General** page contains general information about the step.

Name for this step (optional)

Optionally, specify a descriptive name for the step. If no name is specified, adTempus uses a default name based on the task settings.

Description/Notes

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

SQL Server Job Page

The **SQL Server Job Page** defines the basic settings for the SQL Server Job Task.

Server Name

Enter the name or IP address of the computer where SQL Server is running. If you need to connect to a named instance of SQL Server, include the instance name as follows:

```
servername\instancename
```

Job Name

Enter the name of the job to execute.

Monitor the job

Check this option to have adTempus wait for the SQL Server job to complete. The status of the step will then be set based on whether the SQL Server job succeeds, and Responses can be executed based on specific outcomes.

If this option is not checked, adTempus simply submits the SQL Server job and does not wait for it to execute.

Credentials

Specify the identity that adTempus should use to connect to SQL Server.

- **Use the job's credentials:** adTempus will use integrated (Windows) security to connect to SQL Server. The Windows account that the job is executing under must have permission to log on to SQL Server and submit the job.
- **Use the following SQL Server login:** adTempus will log in to SQL Server using the SQL Server user ID and password you specify. This option requires that SQL Server be configured to accept SQL Server security logins. The credentials you specify with this option are defined in SQL Server, and are not Windows credentials.

See Also

SQL Server Job Task Overview

Conditions Page

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.

Variables Page

The **Variables** page allows you to define Job Variables for this step. You can add new Variables, or override the values of Variables inherited from the Job. Any Variables you define or override here affect only this step of the Job.

To set Variables that apply to the entire job, use the Variables page in the Job properties.

Responses Page

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 SQL Server Job task:

Event	Description
Step starts	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).
Step ends	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).
SQL Server job was started	Occurs when adTempus successfully starts the SQL Server job.
The SQL Server job's outcome is unknown	Occurs if the SQL Server job finished executing, but adTempus cannot determine the outcome.
The SQL Server job does not exist or cannot be run	Occurs if adTempus was unable to start the SQL Server job.
The SQL Server job failed	Occurs if the SQL Server job completed with a failure result.
The SQL Server job succeeded	Occurs if the SQL Server job completed with a success result.
The SQL Server job was canceled	Occurs if the SQL Server job was canceled (through SQL Server).
The SQL Server job ended (regardless of outcome)	Occurs when the SQL Server job finishes executing, regardless of its status.
Task runs longer than the specified time	Occurs if the task has not completed after the specified time has elapsed.

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.

ℹ	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.

See Also

Startup Trigger Properties

Triggers Overview

Startup Trigger Properties

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.

See Also

Startup Trigger Overview

WMI Trigger

WMI Trigger

The WMI Trigger allows adTempus to receive Windows Management Instrumentation (WMI) events and trigger jobs based on them.

See Also

WMI Trigger Properties

Microsoft Platform SDK documentation for WMI

Sample WMI Queries

Triggers Overview

WMI Trigger Properties

The WMI Trigger uses a WMI event query to monitor WMI events and trigger jobs in response to those events.

For information on WMI, refer to the Microsoft Platform SDK documentation at http://msdn.microsoft.com/library/en-us/wmisdk/wmi/wmi_start_page.asp?frame=true.

For sample WMI queries, visit the Arcana Development web site.

⚠	adTempus does not validate your WMI query settings at the time you save the trigger. To test your query, use the "wbemtest" system utility. If Windows returns an error when adTempus submits your query, adTempus will log an error message to the message log for the job and hold the job until you edit it to correct the query.

Optional Descriptive Name

Optionally, provide a descriptive name for this trigger. Otherwise, a description will be created automatically based on the query.

Enable this Trigger

If this box is checked, the trigger is enabled and will cause the job to run. If this box is not checked, adTempus will ignore this trigger.

Namespace

Specify the WMI namespace to use.

Query

Specify the WMI query to execute. This must be a valid WMI event query. For information on WMI event queries refer to the Microsoft Platform SDK documentation at http://msdn.microsoft.com/library/en-us/wmisdk/wmi/receiving_event_notifications.asp?frame=true.

Each time the query returns an event, adTempus will trigger the job.

Use a script to select events

adTempus will run the script you specify each time it receives a new event from WMI. The event object will be passed to the script using the JobVariables collection. Your script then evaluates the event and returns **True** if adTempus should trigger the job, or **False** if it should not.

Minimum Time Between Triggers

Specify the minimum time (in seconds) that must elapse after this trigger fires before adTempus will fire it again. Once adTempus triggers the job, it will ignore all new events until the specified time has passed.

Job Variables

The WMI Trigger sets the following Job Variables, which can be used by a selection script or by other scripts or notification messages in the job.

Parameter Name	Description
WMIEvent	An ISWbemObject interface to the event object returned by your query. For example, if your query is "SELECT * FROM __InstanceModificationEvent", the WMIEvent parameter will be an ISWbemObject to the __InstanceModificationEvent object returned by your query. If your query contains a GROUP clause, the object will be an __AggregateEvent object. See http://msdn.microsoft.com/library/en-us/wmisdk/wmi/group_clause.asp?frame=true for more information. **Note:** The event parameter is only available to the selection script called for the trigger. It is not available to scripts run later in the job, as most script parameters are. If you need information from the event later in your job, you must use the selection script to extract the information and save it in other parameters.

See Also

Using Job Variables

Web Request Task

Web Request Task

The Web Request task allows adTempus to request a page or file from a Web server.

This task can be used to simply "ping" a URL on the server (for example, if the Web application has offline processing that is triggered by requesting a special URL on the server) or to download a page or file from the Web server.

Pages and files retrieved by the task can be saved in the job's history and/or written to a specified location on the adTempus server.

The Web Request Task is used to request a URL as part of job processing. If you want to trigger a job based on the response (or non-response) of a Web server, use the Computer Monitor Trigger instead.

See Also

Web Request Task Properties

Computer Monitor Trigger

General Page

The **General** page contains general information about the step.

Name for this step (optional)

Optionally, specify a descriptive name for the step. If no name is specified, adTempus uses a default name based on the task settings.

Description / Notes

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

Web Request Page

The **Web Request** page contains the basic settings for the Web Request Task.

URL to Request

Enter the URL of the Web page or file to request. For example, "http://www.arcanadev.com". The URL may contain Job Variables.

Credentials

Specify the credentials, if any, needed for this Web resource.

- If the Web resource does not require credentials (i.e., it accepts anonymous requests), select "Use the job's credentials."
- If the Web server is configured to use Windows authentication, and the account that the job is running under has the necessary permission on the Web server, select "Use the job's credentials."
- Otherwise, you must select "Send the following credentials to the Web server" and provide the necessary credentials. Type the user name in the **User ID** box. If this is the first time this user ID has been used for a Web-based task in adTempus, you will be prompted to complete a Credential Profile for the user ID, specifying the password.

Save the page or file returned by the server

Use these options if you want to save the page or file returned by the Web server. You can choose to save the file in the job's history and/or save the file to a specified directory.

See Also

Web Request Task Overview

Conditions Page

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.

Variables Page

The **Variables** page allows you to define Job Variables for this step. You can add new Variables, or override the values of Variables inherited from the Job. Any Variables you define or override here affect only this step of the Job.

To set Variables that apply to the entire job, use the Variables page in the Job properties.

Responses Page

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 Web Request task:

Event	Description
Step starts	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).
Step ends	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).
The Web request was executed (regardless of the result)	Occurs when the Web request has been executed, regardless of the result.
The HTTP request returns the following status code	Occurs if the HTTP request returns the specified status code. For example, a status code of 404 is returned if the page or file is not found.
Task runs longer than the specified time	Occurs if the task has not completed after the specified time has elapsed.

Tools Reference

Find and Replace

The **Find and Replace** tool allows you to find jobs containing specified text, and optionally to replace text.

To jump quickly to a particular job within the Console, use the Go to Job tool instead.

To start the Find and Replace tool, select the Job Group or Job Queue you want to search in, and select **Find and Replace** from the **Tools** menu (or press **CTRL-F**). To search all jobs, select the "Jobs" node in the Console Tree.

> This tool searches for text typed within a job, and cannot find references to adTempus objects such as Notification Recipients.
>
> To find all the places where an object such as a Shared Script, Notification Recipient, etc., is used in adTempus, use the **Find References** command available on the pop-up menu for the object. For example, right-click a recipient in the Notification Recipients folder, then select **Find References** from the pop-up menu.

Fields to Search

Select the fields within each job that you want to search.

Search sub-groups of the selected group(s)

Check this option if you want to search the selected group(s) and all their descendant groups. If you do not check this option, only the selected groups are searched.

Text

Enter the text you want to search for. Optionally, enter the new text if you want to replace text within jobs.

You can choose to use Regular Expressions for more flexibility.

Find

Click the **Find** button to begin the search.

The results list will show each occurrence of the text that adTempus finds. Use the links in the **Job** and **Location** columns to edit the job or other object that contains the text.

Replace

Click the **Replace** button to begin the search operation.

The results list will show each occurrence of the text that adTempus finds and replaces. Use the links in the **Job** and **Location** columns to edit the job or other object that contains the text.

The changes made by the replace operation are not saved until you click the **Accept** button.

See Also

Go to Job Tool

Go to Job

The **Go to Job** tool allows you to jump quickly to a particular job within the adTempus Console.

This tool is especially useful if your jobs are organized into many Job Groups, and you cannot remember which group a particular job is in. Rather than looking through all of the Groups in the Console, you can simply use the Go to Job tool to jump straight to the job, whatever group it is in.

To start the Go to Job tool, select **Go to Job** from the **Tools** menu or simply press **CTRL-G**). This command can be used from any folder in the Console, and will search all jobs within the currently-selected server.

In the **Go to Job** window, type the name (or part of the name) of the job you want to find. You can type the beginning of the name or text that appears anywhere in the name. As you type, the job name box will suggest jobs whose name starts with the characters you have typed.

The **Matching jobs** list will fill with the names of all jobs that contain the text you have typed. To jump to a listed job, double-click the job's name, or select the name and click OK, or simply press **Enter** to jump to the highlighted job.

The Console will then jump to the selected job in the Console, opening the containing Job Group if necessary.

See Also

Find and Replace Tool

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.

See Also

Alerts View

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.

adTempus 3.0 and later use a different export format than previous versions of adTempus. adTempus 3.0 and later cannot import data exported from a version older than 3.0, and vice-versa.

Data Format

adTempus exports data in XML format. You can use an XML or text editor to make changes to an export file before importing it. However, you must be sure that the file still validates against the XML schema referenced by the document.

Though it is technically possible to use the import facility to create and configure jobs in adTempus, the facility was not designed for this purpose. If you need an automated method to create and configure jobs, you should use the adTempus API instead.

See Also

[Exporting data]

[Importing data]

[Importing jobs from the Arcana Scheduler]

Data Export

The Data Export tool (**Tools > Import/Export > Export**) allows you to export data from adTempus. You can export data to a file to be saved for backup 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 Modify 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.

adTempus 3.0 and later use a different export format than previous versions of adTempus. adTempus 3.0 and later cannot import data exported from a version older than 3.0, and vice-versa.

Include history when exporting jobs

When this box is checked, adTempus will include the history (instance details and log events) for all exported jobs. Otherwise, history will not be included.

You must check the corresponding box when you import the data, or the history data in the file will be ignored by the import process.

Include passwords when exporting credentials

When this box is checked, adTempus will include the passwords for any Credential Profiles associated with jobs being exported. The passwords will be stored in an encrypted format.

If this box is not checked, passwords are not included, and you will be prompted for the password for each Credential Profile when the data is imported.

Export to File

Select this option to save the exported objects to a file. See the import/export overview topic for more information on the file format.

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.

See Also

Import/Export overview

Data Import

Data Import

The Data Import wizard allows you to import data that has been exported from this or another adTempus server using the Export tool.

⚠	adTempus 3.0 and later use a different export format than previous versions of adTempus. adTempus 3.0 and later cannot import data exported from a version older than 3.0, and vice-versa.

The Import Wizard comprises the following steps:

- Select Source
- Select Objects
- Import Results

See Also

Import/Export Overview

Importing data from the Arcana Scheduler

Source Page

The **Source** page of the Import Wizard allows you to select the source for the imported data, and specify import options.

Import data from file

Specify the file to import from. This file must have been created using the adTempus export tool. See the import/export overview topic for more information on the file format.

⚠	adTempus 3.0 and later use a different export format than previous versions of adTempus. adTempus 3.0 and later cannot import data exported from a version older than 3.0, and vice-versa.

It is possible to transfer data directly between adTempus servers, without exporting to a file and importing. To do this, use the **Export to Server** option in the Export tool.

Options

Import job history if present in file

When this option is checked, adTempus will import any history records found in the file for imported jobs. If this option is not checked, history will be ignored. Note that history data will not be available unless the option to export history data was checked when the data was exported.

Import security settings for objects

When this option is checked, adTempus will import the security settings for each object. If the option is not checked, the security information in the export file will be ignored, and imported objects will receive the same security settings as if they had been newly created through the Console.

Next Step

When you click **Next**, adTempus will read the file you have specified and make sure it is valid. If it is not, you will not be able to continue.

If the file is valid, the Wizard will continue to the Select Objects step.

See Also

Import/Export Overview

Select Objects Page

The **Select Objects** page of the Import Wizard lists all objects found in the source file you have selected. For each object, the State is one of the following values:

- **New.** This is a new object.
- **Existing.** This object already exists in adTempus. The data from the file will replace the current data in adTempus.
- **Duplicate.** Another object with the same name already exists in adTempus, but it has a different identity (see **More About Duplicate Objects**, below. The data from the file will replace the current data in adTempus.

Select the objects that you want to import by checking the box next to them.

Note: Some objects may be imported even if you don't select them in the list, if they are referenced by objects that you have selected to import. For example, suppose you have selected to import Job "Job A," and Job A uses Credential Profile "freduser." Because this Credential Profile is needed by Job A, it will be imported even if you don't select it in the list.

When you have finished making your selections, click **Next** to import the selected objects and continue to the Import Results step.

More About Duplicate Objects

Each object in adTempus has an internal identifier (or "key") that you do not see. This identifier never changes, even if the object's name or other properties change. This allows adTempus to always uniquely identify each object.

If you export an object and then re-import it into the same adTempus server, the identifier in the import file will match the object's identifier in the database, so adTempus will know that the object is "Existing." This is true even if you have changed the name of the object. For example, suppose that you export job "Job A" to a file, then rename it to "Job B." When you import that file, adTempus will know that "Job A" in the file is the same object as "Job B" in adTempus. If you choose to import the job, the values from the file will replace the values currently in adTempus.

However, support you independently created jobs named "Fred's Job" on Server A and Server B. Since these jobs were created separately, they will have different internal identifiers even though they have the same name. If you now export "Fred's Job" from Server A and import it to Server B, adTempus will detect that there is already a job named "Fred's Job," but the existing job has a different internal identifier. In this case adTempus will list the state of "Fred's Job" as "Duplicate," indicating to you that there is already another job with that name.

When a duplicate object is found, there are two possibilities:

- You want the object in the file to replace the object already in adTempus. In this case, simply check the box to import the object.
- These are really two separate objects, and you don't want the existing object to be replaced. For example, suppose "Fred's Job" from Server A does something completely different from the "Fred's Job" on Server B, and you want to keep both of them. In this case you need to cancel the import and either
    - Rename the object on the original server and re-export it, or
    - Modify the import file to rename the object there. See the import/export overview topic for more information on the file format.

See Also

Import/Export Overview

Import Results Page

The **Import Results** page of the Import Wizard shows the status of each object you selected for import:

- **Saved.** The object was successfully imported and saved.
- **Ready to save.** The object was imported, but informational or warning messages were issued when adTempus tried to save the object. Review the messages by clicking the link in the **Messages** column. You can review and save the objects individually by clicking the **Edit...** button, or click **Finish** to save all "Ready to Save" objects.
- **Errors found.** The object could not be saved due to errors. You can review the error messages by clicking the link in the **Messages** column. You must then review and save the objects individually by clicking the **Edit...** button next to each object.

If you click **Cancel** on this page, any objects with a status of "Saved" will remain in adTempus. Objects with a status of "Ready to save" or "Errors found" will be discarded.

See Also

Import/Export Overview

Arcana Scheduler Import Wizard

Arcana Scheduler Import

The Arcana Scheduler Import tool (**Tools > Import > Arcana Scheduler Import**) 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.

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

See Also

Import Wizard

Select Jobs

This page lists all Arcana Scheduler jobs that you have access to. Select the jobs you wish to import.

Click **Next** to set import options.

Import Options

The Import Options Page contains options that affect the import process.

Use the following set of holidays

Arcana Scheduler jobs that have the "Do not run on holidays" option checked will be configured to recognize the holiday set you specify. If you do not select a holiday set, the imported jobs will *not* be configured to skip holidays.

Disable imported jobs in the Arcana Scheduler

When this option is checked, adTempus will disable each job in the Arcana Scheduler after it is imported into adTempus.

If you do not select this option, be sure that you manually disable or delete the jobs, or disable or uninstall the Arcana Scheduler. Otherwise you will end up with the jobs being run by both schedulers.

Hold imported jobs in adTempus

Check this option to hold all imported jobs in adTempus. Otherwise jobs will be imported with the same status that they had in the Arcana Scheduler.

Regardless of whether you check this option, no imported job is saved until you have had a chance to review it.

Click **Next** to continue the import process.

Successful Imports

The Successful Imports Page lists all jobs that were imported successfully from the Arcana Scheduler.

To review an individual job, double-click the job's name in the list.

To save the jobs, select one or more jobs in the list and click **Save Selected Jobs**.

> You must select and save jobs to save them in adTempus. Any jobs that you do not save on this page will be discarded when you leave it.

Failed Imports

The Failed Imports Page lists any jobs that had errors. To save these jobs in adTempus, you must review each job, fix the problems, and save it. To edit a job, double-click its name in the list.

Server Options

General 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 backed up each day.

This option is only available if you are using the SQL Server Express instance installed with adTempus. If you are using your own database server, you must make your own provisions for database backup.

Security Settings Page

The options on the **Security Settings** page are the default security settings for all adTempus objects, and also 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 to the server" permission on this page (see "Server Permissions" below).

Object Permissions

The security permissions specified here are inherited by all objects in adTempus. For example, if you grant "View" permission to a user (or group), that user (or group) will be able to view all objects (jobs, notification recipients, scripts, etc.) in adTempus.

You can grant permissions for different kinds of objects in other places in adTempus. For example, a in a common scenario you would grant "View" permission to all users here, allowing all users to view all adTempus objects. If you want to allow a group of users to create and modify jobs (but not create or modify any other objects), you would modify the sercurity settings for the root "Jobs" group to grant the appropriate permissions.

See the Security Inheritance and Security Configuration Guidelines topics for more information.

Server Permissions

This is also where you specify who is allowed to connect to this adTempus server. No matter what object-level permissions you have assigned to a user, the user cannot connect to the server unless you have assigned "Connect to the server" permission. Similarly, granting a user "Connect to the server" permission only allows that user to log in to adTempus through the adTempus Console. You still must grant object-level permissions to allow the user to view, create, and modify objects such as jobs.

See Also

Security Overview

Security Editor

Security Inheritance

Security Configuration Guidelines

Notification Setup Window

The Notification Setup Window (**Tools > Server Options > Notification Setup**) contains the server settings used for sending notification messages (e.g., for Notification Actions).

The settings are divided among the following pages:

- Primary SMTP Server (for e-mail notification settings)
- Backup SMTP Server (for e-mail notification settings)
- Notification Service Providers (for other notification services, such as SMS and instant messaging)

Job Status Descriptions

The following table describes the statuses that each job or step may display.

Status	Meaning
Abandoned	The adTempus service was shut down while the job was running, so adTempus does not know the status of this job or step. The program launched by adTempus may still be running.
Aborted	The job was terminated by a user.
Agent Queued	The job is waiting to be dispatched to a Remote Agent.
Condition Failed	An error occurred while evaluating one or more conditions for the job or step. Check the Job Log for more information.
Condition Not Met	One or more conditions for the job or step were not met.
Dispatched to Agent	The job has been dispatched to the Remote Agent and the Master is waiting for a status update from the Agent.
Failed	The job or step failed. Check the Job Log for more information.
Killed	The step was terminated by adTempus because it exceeded the maximum execution time specified in the step's properties.
Not Run	The job or step was not executed.
Queued	The job has been triggered and is waiting to be executed.
Resource Failed	One or more resources for the job could not be loaded.
Resubmitted	The job or step was restarted as a result of a Job Control Action.
Running	The job or step is running.
Skipped	The job or step was skipped. For a job, this will occur if another instance is already running, and the job is configured to not start a second instance. For a step, this will occur if the Program Execution Task is configured to not execute if the program is already running.
Starting	The job or step is starting.
Succeeded	The job or step completed successfully.
Trigger Failed	An error occurred in one or more of the job's triggers. Check the Job Log for more information.
Waiting for Condition	The job or step is waiting for a condition to be met.
Waiting for Previous Instance	The job is waiting for a previous instance of the job to complete.
Waiting for Resource	The job is waiting for a resource to be available.
Waiting for Restart	The job or step is being restarted as a result of a Job Control Action, but the wait time has not yet elapsed.

Waiting for Trigger	The job is waiting to be triggered.

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.

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
```

Command-Line Job Execution Utility (adtExec)

The Command-Line Job Execution Utility can be used to execute adTempus jobs from the command line, batch files, scripts, etc. 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.

The program is named "adtexec.exe" and can be found in the adTempus program directory. This tool is command-line driven and does not present an interactive user interface.

adtExec uses the same authentication as the adTempus Console: the user executing the utility must have Execute permission for the job.

Command-Line Syntax

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.
-agents:"agentName" [...]	Runs the job on the specified agent(s) only. Specify multiple values as show or by repeating the -agents parameter on the command line.
-checkpoint:"checkpoint"	The checkpoint value to pass to the job
-force[{+|-}]	Forces a new instance of the job even if an instance is already running and the job's settings specify otherwise. Default value: True.
-forcemaster[{+|-}]	Forces the job to run on the Master even if it is not configured to run there. Default value: False.
-ignoreheld[{+|-}]	Forces the job to run even if it is held. Default value: True.
-ignoreheldqueue[{+|-}]	Forces the job to run even if the queue to which it belongs is held. Default value: True.
-ignorejobconditions[{+|-}]	Ignores job-level conditions. Default value: True.
-ignorestepconditions[{+|-}]	Ignores step-level conditions. Default value: False.
-list	Lists all jobs that you have permission to

	execute; does not execute jobs. If you specify a partial job name, all jobs matching that name will be listed. For example adtexec warehouse -list will list all jobs containing the text "warehouse".
-masteronly[{+|-}]	Run the job only on the Master even if it is configured to run on Agents. Default value: False.
-user:"userID"	The Windows user ID to use when connecting to adTempus. If not specified, the connection is made under the identity of the account running adtexec.
-password:"password"	The password to use when connecting to adTempus. If not specified, the connection is made under the identity of the account running adtexec.
-responses:value	Determines which Responses will be executed for the job. Valid values: • all: Execute all Responses • none: Do not execute any Responses • nojob: Execute Responses except Job Control actions Default value: all.
-server:"serverName"	The name or IP address of the adTempus server to connect to. To connect to the local server, do not specify a value.
-steps:"stepNumber"	Specify the step number(s) to execute. You may use ranges or lists. For example: "2-4", "3-", "1,3,4"
-variables:""name=value"" [...]	Overrides for job variables defined for the job. Specify multiple values as show above or by repeating the -variables parameter on the command line.
-wait[{+|-}]	Wait until the job finishes executing. If not specified, adtexec returns immediately after submitting the job. Default value: False.

Examples

The following command runs job "myjob" in group "group1" and waits for it to finish, using the following settings:

- Only steps 2-4 are executed.
- Job Control Responses are not executed.
- The job is not started if another instance is already running.
- The "ExecutionSource" job variable is set to the value "CommandLine".

```
adtexec "group1\myjob" -wait -steps:"2-4" -responses:nojob -
force- -variables:"ExecutionSource=CommandLine"
```

Database Utility (adtdbutil)

adTempus Database Utility (adtdbutil)

The adTempus Database Utility (adtdbutil) provides a means of interacting directly with the adTempus database.

To start the program, run the "adtdbutil.exe" program found in the adTempus program directory. The program will connect automatically to the adTempus database. Once the program has connected to the database, the following operations are available:

- Backing up the database
- Restoring the database
- Reinitializing (clearing) the database
- Querying and updating the database

⚠	Never use this tool (or any other) to update the adTempus database except under the direction of Arcana Development technical support personnel. Doing so may result in irreversible corruption of your data.

Backing Up the Database

The Backup command (**Tools > Backup Database**) is used to create a complete backup of your adTempus database, which can later be restored using the Restore command.

This tool is intended for use with the SQL Server Express database engine installed with adTempus. If you are using your own database server, you should use the backup and restore tools provided in the server's management software instead. If you use this tool with a remote database server, note that the file path you enter for the backup file will be evaluated on the database server, and must therefore be valid on that computer, not on the computer where you are running the tool.

ℹ	This topic covers database backups, which are used to recover your entire adTempus database in the event of a failure. Database backups cannot be used to restore selected jobs or other objects. If you want to be able to selectively restore objects, you should use the Export tool to create a data export in addition to your regular backup. You can then use the Import tool to restore selected objects if necessary.

See Also

Automatic backups

Restoring the Database

The Restore command (**Tools > Restore Database**) is used to restore your adTempus database usign a backup file created by the Backup command or by an automatic backup.

ℹ	This topic covers database backups, which are used to recover your entire adTempus database in the event of a failure. Database backups cannot be used to restore selected jobs or other objects. If you want to be able to selectively restore objects, you should use the Export tool to create a data export in addition to your regular backup. You can then use the Import tool to restore selected objects if necessary.

This tool is intended for use with the SQL Server Express database engine installed with adTempus. If you are using your own database server, you should use the backup and restore tools provided in the server's management software instead. If you use this tool with a remote database server, note that the file path you enter for the backup file will be evaluated on the database server, and must therefore be valid on that computer, not on the computer where you are running the tool.

⚠	The Restore operation will remove all existing data in the adTempus database before restoring the backup file.

See Also

Automatic backups

Reinitializing the adTempus Database

The database reinitialization command (**Tools > Reinitialize Database**) can be used to erase all data from your adTempus database and reset it to its original, "clean" state. This tool can be used in test environments where you want to clear out all test data from the database and start with an empty database.

⚠	This tool will erase all data from your adTempus database, including job definitions, history, and all configuration settings.

Querying and Updating the Database

The main screen of the Database Utility presents an empty box in which you can enter a SQL query or update statement.

Press **F5** to execute each statement.

> Never use this tool (or any other) to update the adTempus database except under the direction of Arcana Development technical support personnel. Doing so may result in irreversible corruption of your data.

Technical Information

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. They are also encrypted when exported to a file.

All data is encrypted before being transmitted between the adTempus Console and Service, so it is protected when transmitted over a network connection.

Backing Up and Restoring adTempus Data

To reduce the chances that adTempus configuration or history information is lost in the event of a hardware or software failure, it is important to back up your adTempus data frequently.

> This topic covers database backups, which are used to recover your entire adTempus database in the event of a failure. Database backups cannot be used to restore selected jobs or other objects.
>
> If you want to be able to selectively restore objects, you should use the Export tool to create a data export in addition to your regular backup. You can then use the Import tool to restore selected objects if necessary.

Backing Up Data

adTempus data resides in two locations:

- Captured files are stored in the "CapturedFiles" folder under the adTempus program directory (e.g., "c:\program files\arcana development\adtempus\CapturedFiles"). If captured file data is important, be sure your backup routine includes this folder.
- All other adTempus data (job definitions, history, etc.) is stored in the adTempus database, which should be backed up regularly as described below.

If you are using the default SQL Server Express database engine installed with adTempus, adTempus automatically backs up your database each day at about 1:30 AM. The database backup file is saved in the "backup" folder under the adTempus program directory (e.g., "c:\program files\arcana development\adtempus\backup"). You should ensure that your backup routine includes this folder.

If you are using your own database server to host the adTempus database, adTempus will not automatically create backups. Therefore you must make sure that your existing database server backup routine includes the adTempus database.

In addition to automated backups, you can use the adTempus Database Utility to manually back up the database at any time.

Restoring Data

Restoring an adTempus installation involves restoring both the adTempus database and the CapturedFiles directory. (If you do not restore the CapturedFiles adTempus will operate normally, but you will receive an error if you try to view a captured file.)

Note that restoring the adTempus database restores your configuration and history to the state they were in at the time the backup was taken; any subsequent changes are lost.

Before restoring the adTempus database and files, stop the adTempus service.

If the backup was performed on your own database server, you should use the tools on that server to restore the database.

If the backup was produced automatically by adTempus, or using the adTempus Database Utility, use that tool to restore the database.

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.

Disabling the Console Splash Screen

When the adTempus Console starts, it displays a "splash screen" while it initializes. In cases where you are remotely operating the computer on which the Console runs (using a tool such as pcAnywhere), this splash screen may take a long time to transmit over a network connection to be displayed to you.

In such cases you may want to disable the splash screen. You can do so as follows:

1. Run the Registry Editor and locate (or create if necessary) the key `HKEY_CURRENT_USER\Software\Arcana Development\adTempus\3.0\Console`.
2. Under this key, create a new DWORD value called `splashscreen`.
3. Set this value to 0 if you do not want a splash screen at all. Set it to 2 to display a simple dialog box with a "loading" message. Set the value to 1 or delete it to restore the original splash screen.

This change affects only affects the account you are logged in under when you make the change. To change the setting for all users of a computer, make the same change to the key `HKEY_LOCAL_MACHINE\Software\Arcana Development\adTempus\3.0\Console`.

Disabling the splash screen will not decrease the load time for the Console; it will only eliminate apparent delays caused by the need to transmit the graphical image across a network connection. You will instead see an hourglass (or a simple splash screen if you've chosen that option) while the Console loads.

Disabling the Live Welcome Page

The **Welcome** page in the adTempus Console provides news and other information about adTempus.

Each time you start the Console, adTempus attempts to connect to the Arcana Development web server to refresh this information. If a connection cannot be made, an alternate "static" page is displayed.

If your computer is not normally connected to the Internet but is configured to dial out when a connection is needed, adTempus will cause a dial-out connection to be made each time it starts.

You can prevent adTempus from displaying the Welcome page as follows:

1. Run the Registry Editor and locate (or create if necessary) the key `HKEY_CURRENT_USER\Software\Arcana Development\adTempus\3.0\Console`.
2. Under this key create a DWORD value called `LiveWelcomePage`. Set this value to `0` to disable the live Welcome page.

This change applies only to your user profile. To change the setting for all users of a computer, make the same change to the key `HKEY_LOCAL_MACHINE\Software\Arcana Development\adTempus\3.0\Console`.

How To...

Create a Basic Job

This example shows how to create a simple job to run the "Notepad" program from adTempus.

To create the job:

1. Select the Job Group that you want to create the job in. Right-click the group's name and select **New job** from the pop-up menu. (If you don't have any Groups defined, or want to create the job in the root group, right-click the **Jobs** node.)
2. The Job Properties window will open. Though adTempus offers many options, a simple job requires only a few things.
3. Enter a **Name**. Give the job a unique, descriptive name, such as "Run Notepad."
4. Enter the **User Account**. Type the user ID for the Windows user account the job should run under. If the account is a domain account, include the domain, for example "mydomain\myuserid". When you leave the User Account box, you will be prompted to enter the password for this account. For this example, use your own user ID.
5. Click to the Steps page.
6. Click **Add...** and select the "Execute a program, batch file, etc." task type. This creates a Program Execution Task, which runs a program or batch file and is the most commonly-used kind of task.
7. The Program Execution Task properties window will open.
8. Next to the **Target** box, click the "**...**" button to browse for the file to execute. Browse to the Windows directory (usually "c:\windows") and select "notepad.exe".
9. Click **OK** to save the task properties and return to the job properties.
10. Click **OK** to save the job.
11. The new job will now appear in the folder.

Because this job does not have any Triggers assigned to it, adTempus will run it only if you request it. To run the job:

1. Right-click the job's name and select the **Run** command. The Execute Job window will be shown.
2. Click **OK** to start the job, then click **OK** to acknowledge the informational message.
3. The job's status in the Console will be updated to show that it is running, and the Notepad program will open on your desktop.

ℹ	This example assumes you are running the adTempus Console on the same computer where the adTempus server is installed. If you are not, or if you are connected to that computer through a Remote Desktop or Terminal Services connection, you may not be able to see the Notepad program running on your desktop. In this case you will need to terminate the job by right-clicking it and selecting the **Terminate** command.

4. Close Notepad. The job's status will be updated to show that the job has finished.

Next Steps

Most jobs will have a Trigger so that adTempus will run them automatically. The most common trigger is the Schedule Trigger, which runs the job at specified days and times. We'll add a simple Schedule Trigger to the job you just created.

1. Double-click the job to display its properties, then click to the Triggers page.
2. Click the **Add...** button and select "Execute according to a schedule." The Execution Schedule Properties window will open.
3. Keep the default settings on the **Date Selection** page. These settings will cause the job to run every day.
4. Click to the **Time Selection** page.
5. Select the **Trigger at these times** option.
6. Below the list, type in a time when the job should run. For this example, enter a time that is two or three minutes in the future.
7. Click the "**+**" to add the time to the list.
8. Click **OK** to save the schedule properties. The Schedule Trigger properties window now appears, where you could set additional trigger options.
9. Click **OK** to return to the job.
10. Click **OK** again to save the job and return to the Console.
11. After a moment you will see a **Next Start** time for your job.
12. When the scheduled time arrives, the job will start.

Trigger One Job When Another Finishes

adTempus does not have a Trigger that starts one job when another finishes. Instead, this is configured using a Response.

In this example, we have already set up "Job B" and "Job A," and now we want "Job B" to run when "Job A" completes successfully. This is set up as follows:

1. Edit Job A and go to the Responses page.
2. Click **Add** to add a new Response. The Response Properties window will open.
3. In the **Events** section click **Add** to add a new Event. The Response Event Properties window will open.
4. In the **Event** selector select "Job Succeeded," then click **OK** to return to the Response.
5. In the **Actions** section click **Add** to add a new Action.
6. In the **Select Actions** window select "Control a job or job step." The Control a Job window will open.
7. Under **Action to Take** select "Run a job."
8. Under **Applies To** select "Another job on the same computer, then click the **Select** button.
9. Locate the job you want to run ("Job B" in our example) and click **OK.**
10. Set additional options if you need to (see the Control a Job topic for more information on settings).
11. Click **OK** to save the Action.
12. Click **OK** to save the Response.
13. Click **OK** to save the Job.

Configuration is complete. Now when "Job A" runs successfully, "Job B" will be started. If "Job A" fails for any reason, "Job B" will not be run.

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.aspx.

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.

Index