

# Activity**SCHEDULER**

## Activity**SCHEDULER** Overview

This document gives a brief overview of Activity**SCHEDULER** and its facilities. The later sections describe the user interfaces in more detail. For more information, see the website at [www.activityscheduler.co.uk](http://www.activityscheduler.co.uk), or email the BadgerNT Service Desk at [servicedesk@badgernt.com](mailto:servicedesk@badgernt.com).

© Copyright 2008, Badger Network Technology Ltd.

### Contents

---

|   |           |
|---|-----------|
| <b>Introducing Activity<b>SCHEDULER</b></b> .....                   | <b>2</b>  |
| An intelligent tool to automate your IT workloads.....              | 2         |
| Easy to manage, simple to use, for basic to complex scheduling..... | 2         |
| <b>Powerful functionality</b> .....                                 | <b>2</b>  |
| Hierarchical design.....  | 2         |
| Resource management.....  | 3         |
| Full condition handling.....  | 3         |
| Conditional notification.....                                       | 3         |
| Security model.....   | 3         |
| Reusable templates.....   | 3         |
| Devolved processing.....  | 3         |
| Drag-and-drop design.....   | 3         |
| Policy Management.....  | 4         |
| <b>Basic functional components</b> .....                            | <b>4</b>  |
| Activities and Policies.....  | 4         |
| Principle user interfaces.....                                      | 4         |
| Related facilities.....   | 4         |
| <b>System Architecture</b> .....                                    | <b>5</b>  |
| <b>Installing Activity<b>SCHEDULER</b></b> .....                    | <b>6</b>  |
| <b>Configuration Options</b> .....                                  | <b>6</b>  |
| <b>Predefined policies</b> .....                                    | <b>6</b>  |
| <b>The Policy Designer</b> .....                                    | <b>7</b>  |
| The Policy Explorer.....  | 7         |
| Policy variables and constants.....                                 | 7         |
| Resources and dependencies.....                                     | 8         |
| Timings.....  | 8         |
| Completion codes and linkages.....                                  | 8         |
| Notifications and output files.....                                 | 9         |
| <b>The Policy Manager</b> .....                                     | <b>10</b> |
| Managing policies.....  | 10        |
| Monitoring policies.....  | 10        |
| Crystal Reports.....  | 11        |
| <b>Where to go from here</b> .....                                  | <b>11</b> |

## Introducing ActivitySCHEDULER

---

ActivitySCHEDULER from BadgerNT is a full job-scheduling product for comprehensive IT workload automation. It is ideal software for automating routine tasks and end of day or month business applications such as invoicing, dispatch, inventory, Crystal reports, back-ups, cleaning up log and temp files, checking disk space, etc.

ActivitySCHEDULER is a job scheduler that allows the most complex of inter-dependent activities to be automated, so that they can run unattended across your network.

ActivitySCHEDULER is the professionals' choice as a network product that allows you to develop sophisticated and flexible schedules for reliably performing routine and complex tasks throughout your network. Intuitive interfaces allow you to design and manage your automated schedules with ease.

ActivitySCHEDULER allows you to 'set and forget' your IT workload schedules, notifying you only as intervention becomes necessary.

In most cases, all you will see is the confirmation report as jobs or groups of jobs are completed, giving you much-needed peace-of-mind.

### **An intelligent tool to automate your IT workloads**

In today's distributed computing environment, jobs need to be co-ordinated across multiple locations. It is no longer enough just to put a job in a queue and trust that it will run.

There are many job schedulers on the market today. Some come embedded within applications and others are free standing. But can they really handle the complex demands of true automation?

Computer jobs are frequently dependent not only on 'whether' the previous job or jobs terminated, but on 'how' they terminated. When a job fails to start, or fails while running, the right people need to be notified, and where possible, corrective action needs to be taken automatically.

But whatever happens, the log file will tell you what went wrong (if anything) – traceability that is absolutely essential.

### **Easy to manage, simple to use, for basic to complex scheduling**

With ActivitySCHEDULER, high levels of automation are achieved, for both simple and complex scheduling requirements, with its sophisticated...

- hierarchical design
- devolved processing
- resource management
- full condition handling
- comprehensive notification
- drag and drop policy design
- security model adherence

## Powerful functionality

---

### **Hierarchical design**

Individual jobs can be grouped together, with groups within groups. Each group can contain a series of interdependent jobs or groups, running concurrently or consecutively according to completion status.

At each level, different dependencies and timing conditions can be imposed. Only at the highest level group do you actually define the overall scheduling parameters.

This structure allows a modular design that is much easier to understand.

## Resource management

Any job or group of jobs may require resources (dependencies) in order to run. ActivitySCHEDULER allows specification of resources such as disk space, or the presence or absence of a file or directory.

Required files can also be copied, possibly from another network location.

Resource locking is provided to limit concurrent resource usage.

## Full condition handling

ActivitySCHEDULER allows condition handling to be employed against whatever status a job

(or group of jobs) finishes with, or even whether it was able to start successfully. It is not just a case of reacting to failure or success, but also to the 'flavour' of failure or success.

## Conditional notification

ActivitySCHEDULER supports a wide variety of notification methods such as email, Event Log, 'Inform' (message box), or 'WhiteBoard' (sophisticated broadcasting). This allows you to select the most appropriate method for the condition. ActivitySCHEDULER also supports the addition of custom notification methods using the SDK.

Notification messages, whether errors or informational, can be configured within your schedule design at any relevant point.

## Security model

Regardless of the computer that a job is running on, ActivitySCHEDULER will not infringe upon or compromise the local operating system. In fact, it is specifically written to work within the security models inherent in operating systems like Windows.

Users (or groups of users) can be configured with different levels of permission within ActivitySCHEDULER. All passwords are encrypted, both in the database and when transmitted across the network.

## Reusable templates

Once you have defined an individual job, with its command line, completion codes, output files, dependencies, etc., then this can be reused as many times as you like in other schedules (or multiple times within the same schedule). This is achieved by defining a **template**, with **variables** (such as command line parameters, file names, etc.) that can be changed for each re-use.

For example, you could define a **template** for performing a backup, but change the computer name, drive letter and filename for use within different schedules simply by marking these values as **variables**.

Likewise, groups of jobs can also be defined as **templates**, for multiple re-use.

## Devolved processing

ActivitySCHEDULER schedules are controlled from the central server (or workstation), where the database and definitions are also located. The jobs themselves are controlled by a Windows service on each appropriate computer, which executes the job in the correct user context to maintain security.

All resources and interdependencies are controlled from the central server.

Jobs can also be controlled across a firewall if necessary, using a known port number.

## Drag-and-drop design

ActivitySCHEDULER uses familiar Explorer-style windows for displaying and designing the hierarchy of a schedule and its associated groups of jobs.

You can group jobs and create inter-dependencies intuitively using 'drag and drop' techniques, and in this way, develop complex schedules quickly and easily.

## Policy Management

In the production environment, ActivitySCHEDULER's management facilities allow a suitably privileged user to control the running of **policies** (schedules). If necessary, you can start and stop **policies** manually and monitor their progress. At a glance, you can see which **policies** are running, and the completion status of those that have finished.

For more detail, you can display the configurable report and the full log file for each **policy**. Details of past runs can be viewed and kept for a configurable period.

This interface is also available on the web (firewall tolerant), allowing you to control your systems truly remotely. A wizard is provided to help you set up your web interface quickly and easily.

## Basic functional components

---

### Activities and Policies

This section explains the terminology used within ActivitySCHEDULER for the jobs that are scheduled.

The basic building block of a scheduled task is the command line to be executed. In ActivitySCHEDULER, this command line is defined in an **activity**, along with the various parameters for its execution, timing conditions, completion codes, dependencies, etc.

At the highest level, a group of jobs is scheduled to run at a particular time or times. In ActivitySCHEDULER, this is called a **policy**, and can contain multiple **activities** and **activity sets** (linked groups of **activities**).

The **activities** and **activity sets** within a **policy** can be linked in structured relationships, to run concurrently or in series, depending on completion codes. **Activities** within an **activity set** can also be linked in the same way. At each level, different dependencies and timing conditions can be imposed.

The **activity** and **activity set** definitions used within a **policy** can be re-used in different **policies** (or even in the same **policy** with different parameters), and are thus known as **templates**.

### Principle user interfaces

The main interfaces provided with ActivitySCHEDULER are:

- The **Policy Designer** allows you design and re-use **activities**, **activity sets** and **policies**. It then allows to you **enable** policies so that they are available in the Policy Manager.
- The **Policy Manager** allows you to schedule your enabled **policies**, or to start them on an ad hoc basis. You can also manage other aspects of your operational policies and view details of past runs.
- A **Web** version of the **Policy Manager** is also available, offering the same management functionality, but with more remote flexibility
- The Policy Designer and Policy Manager (but not the web version) also provide ActivitySCHEDULER's **Configuration** menu, allowing you to specify the users and computers to be used in the product. You can also restrict the permissions for each interactive user within ActivitySCHEDULER, define custom calendars and other advanced items.

To maintain security, passwords must be configured to logon to the above interfaces, and to run activities in a user context. All passwords are encrypted.

### Related facilities

ActivitySCHEDULER also provides you with these additional utilities:

- **Report Wizard** allows you to use ActivitySCHEDULER to automate your Crystal reports. A number of sample reports for the ActivitySCHEDULER database are also supplied.
- **WhiteBoard** is a message broadcasting utility which is primarily supplied as a notification method to be used within your ActivitySCHEDULER policies, but can also be used for broadcasting messages between users and user groups, or from other applications (see ActivitySCHEDULER SDK).

- A number of pre-defined **skeleton** policies are supplied with ActivitySCHEDULER. Before you can use them, these must be customised for your own system using the *Policy Wizard*. This wizard allows you to specify which computer and user-context to use, and who to send notifications to, etc.

Some of the supplied policies perform various housekeeping tasks related to the ActivitySCHEDULER product. Others are used to demonstrate different tasks that can be automated using ActivitySCHEDULER (for example a database backup).

Once processed by the Policy Wizard, the policies are available in the Policy Manager for scheduling. You can also use them as a starting point for your own policies by editing them within the Policy Designer.

### System Architecture

The main ActivitySCHEDULER product is installed on a **Central Server**, which can be anything from Windows XP Pro upwards. A number of Windows services are installed on this machine.

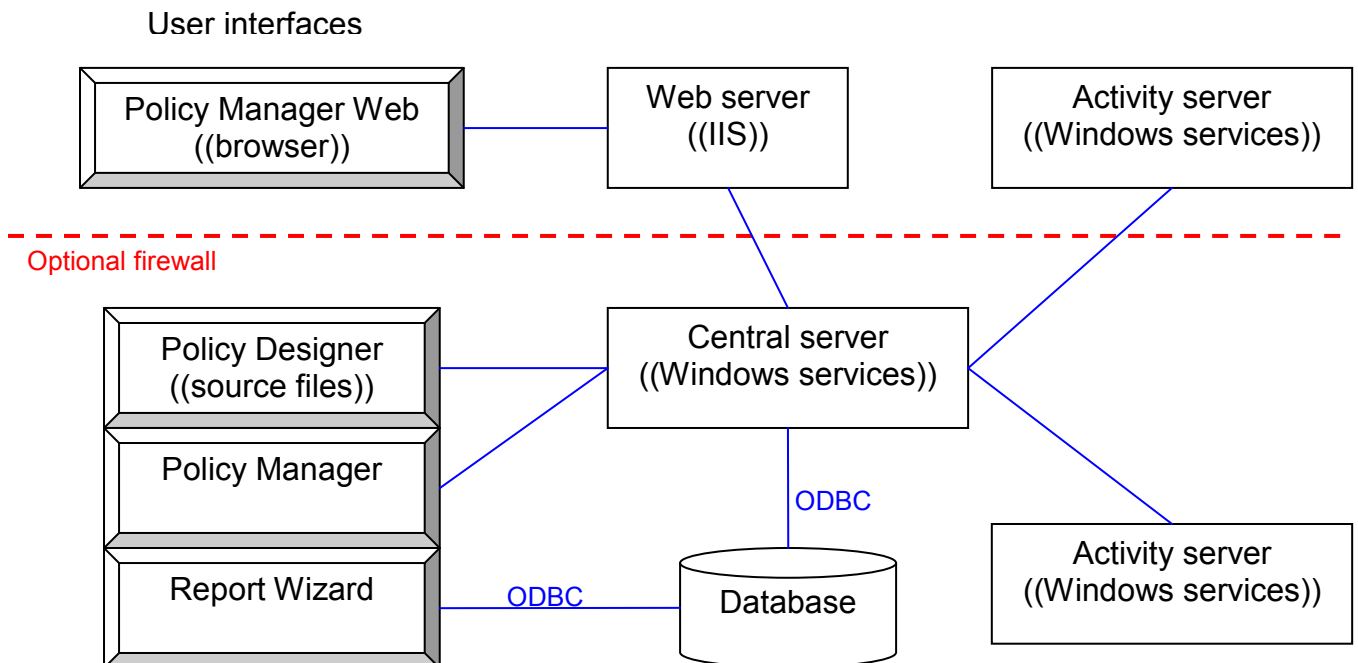
If the web interface is required, the **Web Server** can optionally be separate from the **Central Server**, with an intervening firewall if necessary.

ActivitySCHEDULER also uses a small database (accessed via ODBC) which can be either MS SQL Server (recommended) or Oracle. Access is also supported but is usually recommended for evaluation only or for the Lite edition.

Additional Windows services are installed on any machines where **activities** are to be executed. The user interfaces (see above) can be installed on any Windows machine(s) in the network.

The Web version of the Policy Manager does not require any additional installation on your PC once you have installed and configured your Web Server – you just need a browser.

The diagram below shows the basic configuration of the various components that make up the ActivitySCHEDULER product:



The diagram above shows only the minimum configuration requirements for ActivitySCHEDULER. Different components can of course be combined on a single server or workstation, for example:

- The Web Server can be on the Central Server, if a firewall is not required between them.
- Activities can be (and usually are) also run on the Central Server.
- The Policy Designer and Policy Manager can be installed on the Central Server, if you have direct or remote access and want to completely manage the system from there.

---

## Installing ActivitySCHEDULER

---

As mentioned above, ActivitySCHEDULER uses a small database, which must either be on the Central Server, or be accessible via ODBC. If you use Microsoft SQL Server or Oracle, you must create your database and ODBC connection before installing ActivitySCHEDULER – full instructions are given in the *ActivitySCHEDULER: Database Setup Guide*. If using an Access database (using the drivers installed with Windows), it will be created automatically by the installation.

When installing on the Central Server, you would typically select all components, except possibly the Web Server if that is to be separate. You can then install selected components on other computers in the network, according to where you want to run activities, and where you want to design and manage your policies. If you want to use the Web version of the Policy Manager and have a separate Web Server, you must install the relevant components there.

The simplest installation (e.g. for evaluation) has all components on a single server or workstation using an Access database, as described in the *ActivitySCHEDULER: Getting Started Guide*, delivered with the product download.

Full details of installing the product are given in the *ActivitySCHEDULER: Installation Guide*, which you can access from the BadgerNT Members Area after purchase.

To use the automated reporting facilities, you should also install the Crystal Reports run-time package supplied with ActivitySCHEDULER, whether or not you already have the full Crystal Reports product on the same computer.

---

## Configuration Options

---

Before using ActivitySCHEDULER for the first time, you should run the *Configuration Wizard* on the Central Server and/or Web Server. This allows you to specify the required details for the email and web facilities, etc.

The newly installed ActivitySCHEDULER has a single user configured. You can logon as this user to access the product's Configuration menu to define additional users, and the computers (nodes) where they can access the product.

For each user, a password must be supplied, which is used both for logging onto the product, and for the user-context for executing activities (if configured). A comprehensive range of permissions can be configured for each user, for using the Policy Designer, Policy Manager, running activities, and using the configuration facilities themselves.

For each additional computer, you can configure which interfaces can be run from there, and whether activities can be executed (you must also install the required components on those computers). Your license determines how many computers (servers and workstations) you can configure to run activities.

Configuration facilities also allow you to define:

- User groups, to facilitate the setting of similar permissions for multiple users
- Concurrent resources, to limit the execution of multiple activities or policies at the same time, if they require the same notional resource (for example to limit the number of certain types of activity that will run at any one time).
- Custom calendars, to define days when particular policies should or should not run

Full details are given in the *ActivitySCHEDULER: Configuration Guide*.

---

## Predefined policies

---

After you have installed ActivitySCHEDULER, you can also install a wide range of predefined policies, which are provided as examples of the types of task that can be automated by the product. You can customise these policies for your own use by means of the supplied utility, the Predefined Policy Wizard.

Once a predefined policy has been processed by the wizard, you can further edit and rename it using the Policy Designer, to use it as the starting point for your own policy.

## The Policy Designer

---

The Policy Designer gives you an extensive range of facilities for use when defining activities, grouping and linking them within activity sets and policies. These are described in detail in the *ActivitySCHEDULER: Policy Designer Guide*; the following sections describe some of the more important facilities.

If you have received ActivitySCHEDULER as the automation facility of a separate application, you may not be licensed to use the Policy Designer. In this case, all your policies will be supplied predefined, and you only need to use the Predefined Policy Wizard to make them available for use.

### The Policy Explorer

The Policy Explorer enables you to intuitively move between the different windows within the Policy Designer, for each of the policies, activities and activity sets that you may have open. It has a standard tree-view style that shows the hierarchical structure of each of your policies and activity sets, as well as which activities and sets are being re-used.

The hierarchical structure of your policies makes it much easier to understand the dependencies and resources required at each level.

### Policy variables and constants

There are many places within the Policy Designer where you can enter information about commands to be executed, files to be manipulated, and similar values specified as text strings. To make these specifications flexible enough to meet the requirements of your organization, ActivityScheduler provides you with the means of including *policy variables* and *policy constants* within such string values:

- A *policy variable* is any entity that you may want to evaluate differently between different implementations of a template. For example, if:
  - a command line requires different options according to circumstances, you can define those options as variables
  - a file path is different on each computer on which activities are run, you can use a variable for the path or drive letter.

The template would then contain placeholder variables, and the separate activities derived from that template (when used within activity sets and policies) can then use different values for those variables.

Variables can also be used where you want to use the same value (e.g. a filename) in multiple places in a policy. Defining the value as a variable means that you only need to change the value in one place if it changes, or if you want to create multiple copies of a policy for different environments.

- A *policy constant* is a predefined entity that is given a value when the policy is run. For example, you can specify the start time of the policy or activity set in a variety of formats, the completion code of an activity, and the name of the current activity, activity set and policy.

Two special (and very useful) policy constants are the computer name and user account in which an activity is to run. Although these are called policy constants, their values must be defined by you in the policy. Values for these “constants” can be specified globally for the whole policy, or for each activity individually. They are then used in the same way as any other policy constant.

Both policy variables and constants are very easy to use. To define a policy variable, simply enter its name in place of the relevant value and click the Variable button. A list of variables and their values is then displayed for each Activity, Activity Set and Policy – you can specify their values at any level. A policy cannot be enabled until all variables have been given a value.

Policy constants all start with the \$ character - as soon as you type it, a drop down list appears for you to select the appropriate constant.

## Resources and dependencies

You can specify that activities, activity sets and policies should only be run when suitable resources (dependencies) are available. These resources include:

- The availability of disk space on a particular drive
- The presence or absence of a specified file or folder
- The availability of a specified amount of a concurrent (virtual) resource, which can be configured to represent elements such as media drives, CPU usage or even the use of a system-resource intensive program. You can then allocate the resource to the chosen activity, activity set, or policy until it has completed. The release of a concurrent resource can be dependent on the completion status of an activity; for example, to retain the resource until any automatic recovery procedure has been performed.

Although not strictly “resources”, you can also copy or delete existing files within the network prior to executing an activity, set or policy.

As described above, policy variables and policy constants can be included within the specifications of drives, file paths, and other parameters used in the definition of resources. This enables you to easily use the same values in different parts of a policy.

## Timings

You can schedule policies in three ways:

- On request – “manual” running
- On a regular constant schedule through the day, with a fixed time interval such as one hour between runs.
- On a schedule determined by a specified calendar, so that the policy runs daily, weekly, monthly, yearly, etc.

The first of these is a straightforward ad hoc run, which causes the policy to be started whenever you want. Facilities exist for such a policy to be started by another application – please ask for details. The second method allows you to run of the policy at intervals through the day. The third offers you a range of opportunities for defining exact run times within particular days, weeks, months and years. You may also use custom calendars that you have defined for your own use.

In addition, you can specify a start time for each activity and activity set within a policy. This can be a fixed time of day, or an elapsed time (delay).

If a start time is significantly delayed for any reason, you can specify that it should not run at all.

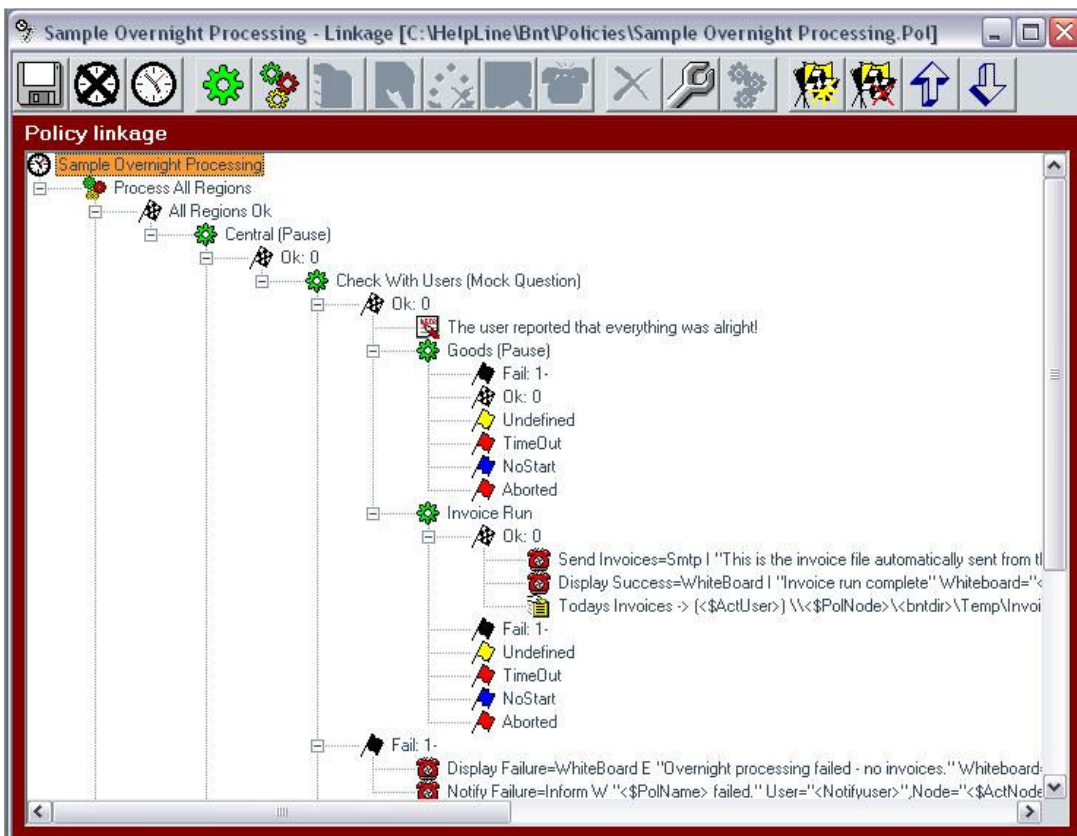
You can also specify time-outs that will interrupt activities, activity sets, and policies that have overrun their expected running times. These can also be elapsed times, or a fixed time of day. For example, abort a resource-intensive activity if it is still running at the start of the working day.

## Completion codes and linkages

A policy is not simply a collection of activities and activity sets. The Policy Designer allows you to use the completion codes defined in activities and activity sets to determine which further actions are to be taken.

For example, if one component activity of a policy runs successfully you can specify that another activity should then be run. However, if it fails you can specify a variety of other activities to be run, with the chosen activity depending on the reason for the failure, as given by the completion code. These could be activities derived from the same template, but with different variable values. Alternatively, a suitable email could be sent to an appropriate person if certain severe errors occur.

You define the relationships on a linkage diagram such as the one shown on the next page. The entities in a linkage can easily be manipulated by simple drag-and-drop.



This particular example also shows how you can link other operations to completion codes, such as including a message in the policy report and sending notifications.

Completion codes are defined by you, according to the codes returned by the program you are running. You can give a meaningful name to each numeric status code or group of codes.

You can also see that there are some pre-defined completion codes such as NoStart (not started because of missing dependencies, etc.) and TimeOut (did not complete in a timely manner), allowing you to perform relevant actions when these conditions occur.

## Notifications and output files

As mentioned above, you can link standard actions to completion codes, including:

- Copying and deleting any files output by a completed activity.
- Sending notifications to users by various methods, including email, the WhiteBoard broadcasting utility, and any proprietary notification methods that you wish to define.
- Writing messages to the policy report, and including excerpts from an activity output file in that report when necessary.
- Releasing allocated concurrent resources.

A policy report (text) is always produced for each run of a policy. You can specify the level of detail this report should include, as well as writing specific messages to it at various points in the policy (as mentioned above).

You can use policy variables and constants in file specifications, notifications, and report messages whenever necessary, including the completion status of the current activity or activity set.

This functionality allows you to send relevant exception notifications only when specific conditions have occurred, as well as confirmation that all is well.

## The Policy Manager

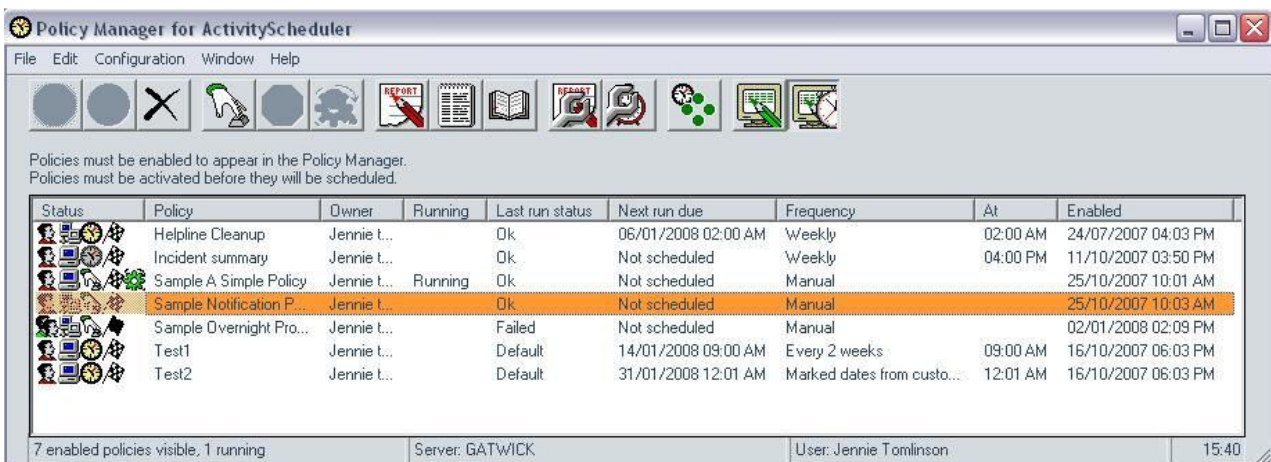
Once development staff have created your policies, then operational and administrative staff can activate them and monitor their progress by means of ActivitySCHEDULER's Policy Manager. A web version of this interface is also provided, allowing you to manage your system truly remotely. Full details of these interfaces are given in the *ActivitySCHEDULER: GUI Policy Manager Guide* and the *ActivitySCHEDULER: Web Policy Manager Guide*, available from the BadgerNT Members Area after purchase.

### Managing policies

Any user with sufficient authorization can use the Policy Manager to view enabled policies and make use of the various management functions:

- Starting a policy so that it runs immediately.
- Activating a policy, so that it runs on a predefined schedule.
- De-activating and re-activating a scheduled policy, stopping a running policy, and disabling a policy so that it can no longer be accessed from the Policy Manager.
- Redefining the timing controls for a policy, to choose another schedule.
- Redefining the level of detail included in a policy report.
- Receiving and reacting to notifications from running policies.

An example Policy Manager screen is shown below:



### Monitoring policies

Whenever a policy runs, a **policy log** and a **policy report** are written and saved on the Central Server. They can be viewed from the Policy Manager at any time.

The **policy log** contains a transcript of all actions that were taken while the policy was active on the system.

The **policy report** contains a more structured account of the policy run, summarizing the effects of its component activity sets and activities. You can control the level of data included in policy reports.

The **Policy History** facility allows you to review details of previous runs of a policy – you can choose the period retained by the history to suit your needs. For each previous run of the chosen policy you can display details of its start / end times and duration, plus its completion status, any delays that it was subject to (if run on a schedule), or the person who started it if run manually.

## Crystal Reports

To use the Crystal automation functionality in ActivitySCHEDULER, you must install the supplied Crystal run-time on each computer that is to run reports (typically just your Central Server, but possibly also your PC/workstation).

The Report Wizard is accessible from the Policy Manager or from your desktop, for running the supplied reports, such as which policies ran yesterday and which are due to run tomorrow. You can also use the Report Wizard to automate your own Crystal Reports – all you need is the report definition and an ODBC connection to the relevant database.

If you only want to schedule Crystal reports, then you should consider installing ReportSCHEDULER first. If you subsequently need more functionality, then you can easily upgrade to ActivitySCHEDULER.

## Where to go from here

---

You can download ActivitySCHEDULER from the website [www.activityscheduler.co.uk](http://www.activityscheduler.co.uk) for a FREE 30-day trial.

Also from the website, you can:

- Download the technical specification
- Contact our Service Desk for further assistance
- Fill in an order form for purchasing ActivitySCHEDULER

After purchasing the product, you will have access to the full manual set via the BadgerNT Members Area.