

ActiveQ v1.0

Copyright (C) 2002 Active Up SPRL

1. License

Active Up grants you a non-exclusive license to install and use a copy of the software on a single server held or controlled by you. You may transfer the software between servers only if it is first removed from the old computer prior to being installed on the new one. You may only install and use the software within your own organization. You shall not reverse engineer, disassemble or decompile all or part of the Program. You are permitted to copy the documentation only for your own use. You are permitted to make only one back-up copy. The sale and sub-licensing of the software are forbidden. This license shall not be transferred or assigned without the prior written agreement of Active Up or an approved distributor.

2. Freeware version

The standard software version is free of charge. It may be freely distributed on the sole condition that it is not altered in any way. ActiveQ Standard version can be used in any environment, commercial or not.

3. Professional version (with [ActiveMail](#))

You can register [ActiveMail](#) to activate mail features of the software. This registration will allow you to use the software with all its capabilities. Active Up will accept it to also be installed in a test environment if it can be accessed by only one person.

4. Warranty disclaimer

THE SOFTWARE IS SUPPLIED "AS IS" WITHOUT WARRANTY OF ANY KIND. ACTIVE UP SPRL, ITS LICENSORS AND THEIR EMPLOYEES: (1) DISCLAIM ANY WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OR MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT, (2) DO NOT ASSUME ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, OR USEFULNESS OF THE SOFTWARE, (3) DO NOT REPRESENT THAT USE OF THE SOFTWARE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS, (4) DO NOT WARRANT THAT THE SOFTWARE WILL FUNCTION UNINTERRUPTED, THAT IT IS ERROR-FREE OR THAT ANY ERRORS WILL BE CORRECTED.

5. Limitation of liability

IN NO EVENT WILL ACTIVE UP SPRL OR ITS LICENSORS BE LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL OR PUNITIVE DAMAGES OF ANY KIND OR NATURE, INCLUDING BUT NOT LIMITED TO LOSS OF PROFITS OR LOSS OF DATA, FOR ANY REASON WHATSOEVER, WHETHER SUCH LIABILITY IS ASSERTED ON THE BASIS OF CONTRACT, TORT (INCLUDING NEGLIGENCE OR STRICT LIABILITY), OR OTHERWISE, EVEN IF ACTIVE UP SPRL HAS BEEN WARNED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGES. IN NO EVENT SHALL ACTIVE UP SPRL'S LIABILITY FOR DAMAGES ARISING FROM OR IN CONNECTION WITH THIS AGREEMENT EXCEED THE AMOUNT PAID BY YOU FOR THE SOFTWARE.

6. Indemnity

The Licensee shall indemnify, defend, and hold harmless Active Up SPRL, the Software developers, the Software sponsors, and their agents, officers, and employees, against any and all claims, suits, losses, damage, costs, fees, and expenses arising out of or in connection with this Agreement. The Licensee shall pay all costs incurred by Active Up SPRL in enforcing this provision, including reasonable attorney fees.

7. Term and termination

The license granted to the Licensee under this Agreement will continue perpetually unless terminated by Active Up SPRL in accordance with this Agreement. If the Licensee breaches any term of this Agreement, and fails to cure such breach within thirty (30) days of the date of written notice, this Agreement shall immediately terminate. Upon such termination, the Licensee shall immediately cease using the Software, return to Active Up SPRL, or destroy, all copies of the Software, and provide Active Up SPRL with written certification of your compliance with the foregoing. Termination shall not relieve the Licensee from its obligations arising prior to such termination. Notwithstanding any provision of this Agreement to the contrary, Sections 2 through 9 shall survive termination of this Agreement.

8. Export controls

The Licensee shall observe all applicable Belgian and foreign laws and regulations (if any) with respect to the export, re-export, diversion or transfer of the Software, related technical data and direct products thereof, including, without limitation, the International Traffic in Arms Regulations (ITAR) and the Export Administration Regulations.

9. Applicable law - jurisdiction

This license contract is governed exclusively by Belgian law. In the event of a dispute arising in relation to the conclusion, interpretation or execution hereof, the Belgian courts, and more specifically those of the judicial district of Liege shall have sole jurisdiction.

ActiveQ is a free mail queuing service with extended task scheduling support written in 100% managed C#.NET. This service will allow you to send thousands of [queued emails](#) in a few minutes and also [schedule mail](#) for later delivery. Standard queuing is becoming required when sending a newsletter to thousands of subscribers.

Standard or scheduled queuing of complex emails

[Scheduled emails](#) can have a very powerful use in a **CRM** system. For example if a user downloads the latest version of your software, you would like to send him a confirmation email the same day, but also another email 7 days later to ask him if his evaluation is going well and if he needs assistance. You might also like to send an email to the customer 30 days later with information on how to register as well as a link to a form where they can input their impressions, suggestions etc. Using **ActiveQ**, you will be able to send these 3 emails in few seconds: the first will be delivered immediately ([Standard Queuing](#)) and the two others will be scheduled ([Scheduled Queuing](#)).

Performance and scalability

ActiveQ was designed with performance in mind, so it includes a **multi-thread processing architecture** to send multiples emails at a time. You can [configure](#) the number of threads used or you can let **ActiveQ** automatically adjust itself based on the performance/resources available for the requests.

While not required, **ActiveQ** supports a list of **SMTP servers** for redundancies. If no **SMTP server** is specified, **ActiveQ** will use the **DirectSend** technology of [ActiveMail](#) component.

ActiveQ is fault tolerant. In case of a server crash, no queued mail is lost. The system will automatically reload them at startup.

Scheduled Tasks

Another big feature of **ActiveQ** is the ability to allow you or your users to build an **XML file** containing [scheduled tasks](#). A task consists of a **URL (HTTP)** to query (POST or GET method) and a list of triggers. Triggers include *daily*, *weekly*, *monthly*, or a specific *day*, *week* or *month of year*.

This system is ideal for your [scheduled tasks](#). You will now be able to access a system like CRON without having to use SessionStart tricks and other systems that are not reliable and scalable.

You can query scripts that will:

- update currency values daily
- send monthly sales reports
- clear a temp directory
- etc...

Ideal for ISP's

ActiveQ mail queuing supports multiple pickup directories and Multiple XML task list configuration files. This is ideal for ISP's that want to allow their customers to take advantage of this without having to define special permissions for a unique directory.

Powerful service manager to monitor

The powerful **ActiveQ** manager will allow you to see what is going on with your scheduled mails and tasks. The main log window will inform you of what the service is doing for you. You will be able to add or remove pickup directories or XML task configuration files.

Availability

ActiveQ is a free application. All the email-related features require the [ActiveMail](#) component to run as well as a separate license (1 license per server). Installing [ActiveMail](#) will also allow your customers or your application to take advantage of this feature-laden email component. The [scheduled tasks](#) system, though, does not require [ActiveMail](#) and thus is completely free to use. See the [comparison chart](#) for more information.

Mail Queuing (Requires a separate license of [ActiveMail](#))

- Immediate or scheduled sending of complex emails.
- Multiple pickup directory support (ideal for ISP's).
- Automatic Thread load-balancing based on the performance and resources of the server against requests.
- Powerful service managing to monitor the running and scheduled tasks.
- Fault-tolerant. In case of a crash, nothing is lost and the system is automatically resynchronized on startup.

Scheduled Tasks

- Execute a GET or POST query on any HTTP web site or intranet.
- Triggers include Daily, Weekly, Monthly or a specific day, week and month of year. Includes hour and minute information.
- Multiple XML task list configuration file support (ideal for ISP's).

Comparison Chart

Feature	ActiveQ (free)	ActiveQ + ActiveMail
Automatic Thread load-balancing		
Scheduled Tasks (XML files)		
Service manager to monitor tasks		
Fault-tolerant		
Immediate/Scheduled sending of complex emails		
Support by email 24h		

 Included  Not Included

You can obtain information about [ActiveMail](#) pricing at our website : <http://www.activeup.com>.

ActiveQ requires the following to run properly :

- Microsoft Windows 2000, XP or 2003 : Professional, Server or Advanced Server.
- Microsoft .NET run-time, version 1.1.
- Administrator rights for service installation.
- [ActiveMail](#) component (only needed for email features)



ActiveMail is a suite of .NET components that allows users to send templated and databound emails as well as receive and parse messages, using multiple protocols such as SMTP, IMAP4, POP3 and NNTP.

ActiveMail is not just another emailing suite. It comes with **real innovations** like the **templating** feature that allows the end users of your application to modify all the parameters their email by accessing a text or XML file the component loads when executed. **ActiveMail's** advanced **databinding** feature allows the developer to bind virtually any valid datasource. This function combined with **templating** is unique and powerful.

ActiveQ uses **ActiveMail** to parse and send emails. You need to buy a separate license to use the email features of **ActiveQ**. See the [Quick Start](#) document for more details.



Title

ActiveQ v1.0

[JUN-02-2003]

- Initial release

Copyright © 2003 Active Up SPRL - All Rights Reserved - <http://www.activeup.com>

Once installed, **ActiveQ** is automatically started. A default pickup directory named “pickup/” is created as a sub-directory of the application installation directory

[INSTALLDIR]/pickup.

You have the choice to use that directory or define another one. See [Advanced Configuration](#) for more details.

When installed, **ActiveQ** does not contain the [ActiveMail](#) DLL required to use the mail features. If you own a license of [ActiveMail](#), simply copy the **DLL** into the directory where you installed **ActiveQ**. Ensure that the version of the **DLL** is **v1.4+**. If you don't own a license, you can [download](#) a free trial version of the component on our website: <http://www.activeup.com>. If you want more information, don't hesitate to contact our sales department : sales@activeup.com. We also offer volume discounts for ISP's or others big organizations.

If you haven't already done so, learn about the email queuing concept by reading [Concept \(ActiveMail\)](#) page before checking the following topics:

[Sending queued emails](#)

*This document explains how it's easy to send a queued email using **ActiveQ**.*

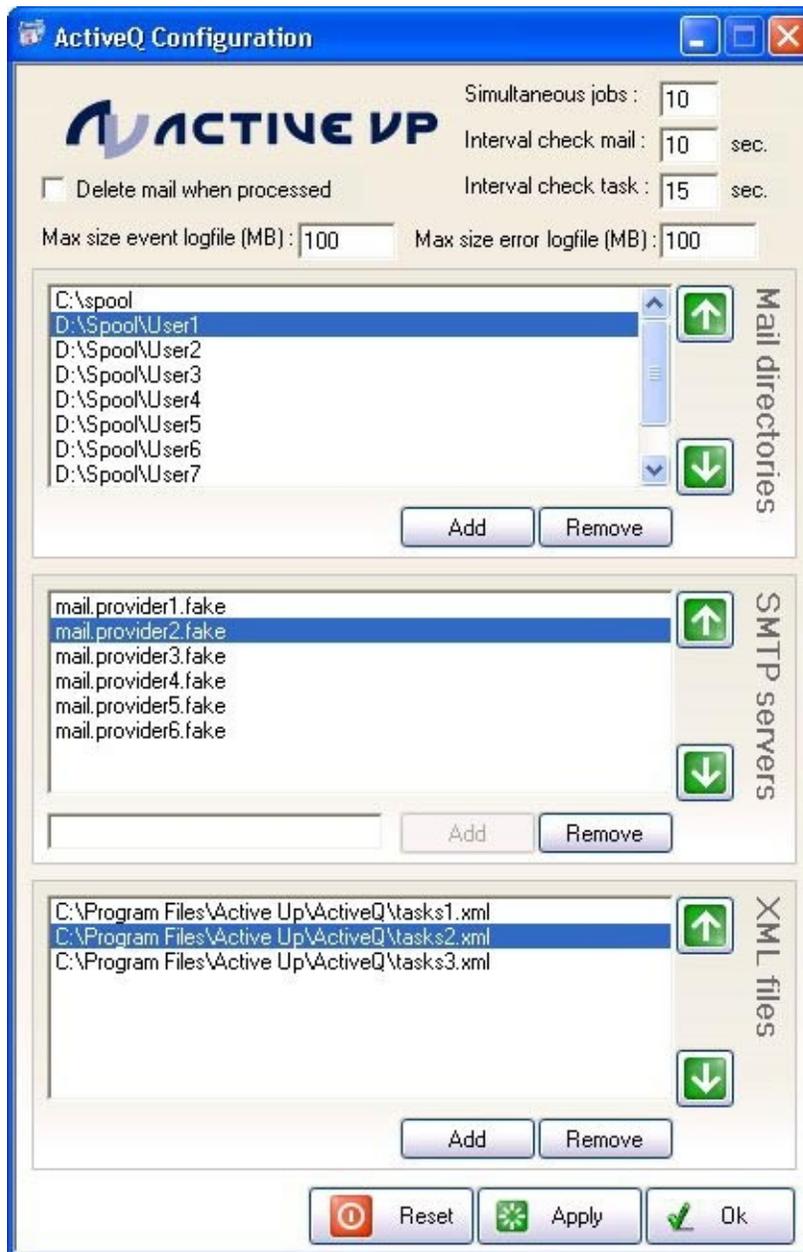
[Sending scheduled emails](#)

*This powerful **CRM** feature can be easily integrated with your **eBusiness** or **eCommerce** solutions.*

The other big feature of **ActiveQ** is [scheduled tasks](#) (not to be confused with the [scheduled emails](#)). Before learning how to create XML task files, read the page [Concept \(XML task file\)](#).

You can configure **ActiveQ** using the **ActiveQ Config Manager**.

To open the configuration panel, select **Start -> Programs -> Active Up -> ActiveQ Configuration**.



You can also set the configuration settings manually by editing the file **config.xml** in the **ActiveQ's** installation directory.

When you click **Apply** or **OK**, the changes are set immediately. You don't have to restart the service.

Simultaneous jobs

This defines the number of threads **ActiveQ** will use to send queued emails. The more threads you define, the more bandwidth and CPU resources will be used. You need to try different values depending of the available bandwidth and server performance.

Interval check mail

This is the number of seconds to wait between processing each pickup directory.

Interval check task

This is the number of seconds to wait between processing for each task in all the defined XML task list configuration files. For best results, you should have the maximum value here. Your users will not change these settings often.

Delete mail when processed

Let you specify if you want to delete the email when processed. If the checkbox remain unchecked, the mail will be moved in the **processed** folder.

Max size event log file

This setting let your specify the maximum number of megabytes the event log file can reach. If you specify the value **0**, it is illimited.

Max size error log file

Same as the event log file. This setting is applied for the error log file.

Mail directories

The pickup directory processor will check each defined directory in the list, one by one. Only .eml files will be processed and the others ignored.

SMTP servers

If you want better results, you can define multiple **SMTP** servers. **ActiveQ** will use the

first one by default and the others if the first one returns an error or is unavailable. If you don't specify anything, **ActiveQ** will use the powerful **DirectSend** feature of [ActiveMail](#). Please be aware that you have only 1 attempt with the **DirectSend**. This means that if the recipient's mail server is unavailable, **ActiveQ** will not try again (this feature is planned in a future version). This is why delegating to an **SMTP** server is sometimes better.

XML files

The different XML task list configuration files are defined here. Files can be anywhere on the server. Please ensure that the system has read access. Your users will be able to modify their files and **ActiveQ** will automatically update the changes after each new periodical processing.

The **ActiveQ** service processes one or more pickup directories and checks to see if there are new emails to add in its [standard queue](#) or [scheduled queue](#). If a new mail is found (a file with the extension **.eml** that is not already in one of its queues), a reference to the file is added automatically for processing.

The standard queue is processed immediately using a **FIFO** (First In First Out) scheme using the multi-thread architecture described in the [About ActiveQ](#) page. When an email is successfully sent, it is removed physically from the pickup directory where it was originally found. The file will remain in the pickup directory until it is successfully sent or has been moved to the error directory. This ensures that a message is not lost in memory in case of a server crash. On startup, **ActiveQ** will automatically detect the email in the pickup directory.

ActiveQ use [ActiveMail](#) to send its email. It can use both the **SMTP** servers list (to provide redundancy) or the **DirectSend** feature of [ActiveMail](#). It is recommended you use **SMTP** servers for better performance. You will need a separate license of [ActiveMail](#) to use the email features of **ActiveQ**. See [Quick Start](#) for more information.

Send a queued email is very simple. You only need to save or copy a valid **RFC822** representation of an email to the pickup directory. Not all email applications and components allow you to save messages in an **RFC822** format. The sample below demonstrates how to do that using [ActiveMail](#), our best selling email component.

Using C#

```
Message message = new Message();
message.From = new Address("jdoe@myhost.com", "John Doe");
message.To.Add("mjohms@otherhost.com", "Mike Johns");
message.Subject = "hey!";
message.Attachments.Add("C:\\myfile.doc");
message.BodyHtml.Text = "As promised, the requested
document.<br /><br />Regards,<br />John.";
message.StoreToFile("C:\\messages\\outbox\\tobesent.eml",
true);
```

Using VB.NET

```
Dim message As New Message
message.From = new Address("jdoe@myhost.com", "John Doe")
message.To.Add("mjohms@otherhost.com", "Mike Johns")
message.Subject = "hey!"
message.Attachments.Add("C:\\myfile.doc")
message.BodyHtml.Text = "As promised, the requested
document.<br /><br />Regards,<br />John."
message.StoreToFile("C:\\messages\\outbox\\tobesent.eml",
true)
```

We pass true to the **StoreToFile** method to ask [ActiveMail](#) to use a temp filename when writing the content of the email then rename it when it's done. This is required because **ActiveQ** can process the directory and search for **.eml** mails. If the mail is not completely written, **ActiveQ** will send an erroneous email. It's relatively impossible when sending small emails like this one, but bigger emails can cause the problem. Please note that this feature is not available in all mail components.

If you want to schedule your email, see [Sending scheduled emails](#).

Like [standard queued email](#), the task is pretty easy. To schedule your mail, just set the **Date** of the email explicitly (*this feature is not available in all mail components*) :

Using C#

```
Message message = new Message();
message.From = new Address("jdoe@myhost.com", "John Doe");
message.To.Add("mjohns@otherhost.com", "Mike Johns");
message.Subject = "hey!";
message.Attachments.Add("C:\\myfile.doc");
message.BodyHtml.Text = "As promised, the requested
document.<br /><br />Regards,<br />John.";
message.Date = new DateTime(2003, 12, 25, 11, 12);
message.StoreToFile("C:\\messages\\outbox\\tobesent.eml",
true);
```

Using VB.NET

```
Dim message As New Message
message.From = new Address("jdoe@myhost.com", "John Doe")
message.To.Add("mjohns@otherhost.com", "Mike Johns")
message.Subject = "hey!"
message.Attachments.Add("C:\\myfile.doc")
message.BodyHtml.Text = "As promised, the requested
document.<br /><br />Regards,<br />John."
message.Date = new DateTime(2003, 12, 25, 11, 12)
message.StoreToFile("C:\\messages\\outbox\\tobesent.eml",
true)
```

ActiveQ will load the email when available and add it to its scheduled queue for later processing. In the example above, **ActiveQ** will automatically send the email on **December 25th, 2003 at 11:12 am**

To schedule or not to schedule ?

Scheduling a task is very useful for a lot of purposes. Your task (what you need to process) can be done using a script (page) that will be executed.

Here are some ideas :

- **Update currency values daily**

If your website allows your customers to display the price of your products in more than one currency, you should take into consideration currency market fluctuations and display the most current price so your customer will not be unhappy to see a different price on the final bill. Just write a script/page (ASP, ASP.NET, ...) that updates these values and schedules it to run everyday!

- **Send monthly sales reports**

You run a great e-commerce application and you want to receive a monthly report of the online sales? Write a script that generates the report and schedule it for monthly delivery.

- **Clear a temp directory**

You need to clear a temp directory on Monday's and Tuesday's? It's easier that you think.

- **Send weekly newsletter**

You publish news on your website daily, and you want to send an email that contains all the news of the last 7 days. This can be done automatically! Schedule the script that generates and sends the email and then schedule it to run each week !

- **etc...**

You want to schedule...

You have several way to schedule the execution of your script (task) that will process what you want :

1. Using Windows' built-in task scheduler

You can add a task that will execute a script. You will be able to execute VBScript.

Plus: Works pretty well

Minus: Only the server administrator can add tasks

2. Application_Start and Session_Start trick

You can use an Application variable to store "scheduled tasks" and then write a function in Session_Start that will check if a task needs to be run: if the actual time is greater than the scheduled time, then we run the task and flag it as having been run using the Application variable.

Plus: You can use this system without any additional software.

Minus: Not reliable. Everything depends on the Session_Start procedure running that needs to be triggered by a hypothetical visitor.

3. Using ActiveQ

This is the easiest and most reliable way to schedule the execution of your processing scripts.

Plus: You have total control without having administrator privileges!

Minus: The server administrator needs to install **ActiveQ** (really a minus?)

It's easier that you think

ActiveQ will help you schedule your tasks easily by querying your page using an **HTTP GET** or **POST** method... If you are an **ISP**, **ActiveQ** is the ideal application to provide scheduling capabilities to your users. You only need to define an **XML** configuration file (in the user's web directory so they can edit it themselves) and then [add it](#) to the scheduled task files list.

An **XML** tasks configuration file (if not already created) can be generated with any text editor. Use a plain text editor only (Notepad, Ultraedit, Textpad, etc...) and not a rich text editor (Word, Wordpad, Html TextBox, etc...) to edit the file (as these rich text editors may add special formatting codes which are invalid in XML).

Don't worry if you are not familiar with XML. A sample XML configuration file can be found at the bottom of this document.

A really cool **ActiveQ XML configuration** file editor will be available soon.

Structure of the XML tasks configuration file

Here is the structure of the XML tasks configuration file :

```
<?xml version="1.0" encoding="utf-8"?>
<config xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <tasks>
    ...
  </tasks>
</config>
```

Each task have the following structure :

```
<task method="POST">
  <id>1</id>
  <address>http://www.domain.com/ActiveQ/ActiveQTestWeb/WebForm1.aspx</address>
  <datestart>2003-05-22</datestart>
  <dateend>2003-06-30</dateend>
  <recurrences>
    <recurrence>
      <type>selectedDay</type>
      <parameter>Friday</parameter>
      <hour>11:30:00</hour>
    </recurrence>
  </recurrences>
</task>
```

<task> element

The "**attribute**" *method* can only have one of two values: POST or GET.

GET

This sends the form information included in the URL. The GET method sends the information to the server in one step. It is best used for forms where the information is not secure (as it can be seen in the URL), and the amount of information passed is small.

POST

This sends the form information in the HTTP environment. It is a little more secure, in that your readers can't see it directly (although it is sent as clear text, so it can be "hijacked"). The post method sends the data to the server in two steps. First the browser contacts the server and, after the contact is made, it sends the information. It is best used when you have a lot of information to pass. Currently, you can't define custom fields and values to pass. However, this is planned for a future release of **ActiveQ**.

FILE

This attribute lets you execute a file on the computer. The **<address>** attribute is the path to the file.

<id> element

Each task is identified by a unique id. Be careful to use a unique id. Otherwise, only the first task defined with that id will be used. Other tasks with the same id will be ignored.

<address> element

Address used for the request. Usually the URL to the script file or the UNC path to the file you want to execute .

<datestart> element

Start date of the task. No tasks are executed before this date.

<dateend> element

End date of the task. No tasks are executed after this date.

For recurring tasks, you can combine several **<recurrence>** elements. Each recurrence must have a **<type>** and an **<hour>** child element. The **<parameter>** element is used for some types and will be described later.

<type> element

Indicates the type of recurrence and can have the following values :

- **once**: task executed only one time at the <hour> specified. The value of the <hour> element must have the following structure :

HH:MM:SS

Example : 23:21:59 specifies 23h 21min 59sec or 9:21:59 pm.

- **daily**: the task executes each day at the <hour> specified.
- **selectedDay**: the task executes each day specified by the <parameter> element at the <hour> specified. The <parameter> element must have one of the following values: "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday" and "Sunday".
- **selectedDate**: the task executes on each date specified by <parameter> at the <hour> specified. The <parameter> element value indicates the date when the task must run. The date should be in the following format:

YYYY-MM-DD

Example: 2003-05-29 specifies May 29th, 2003.

Be careful to use a valid day, for instance, if you specified 31 as day, the task will not execute on the 31st of each month because some months contain less than 31 days.

Once you have created (or modified) your XML tasks configuration file, you need to add the **ActiveQ** scheduled tasks queue. If this hasn't already been done by your administrator, you will need administrator access to the server to add your file to the queue. Read the document "[Using the XML tasks file](#)" for more information.

Sample Configuration File

Here is a sample XML tasks configuration file :

```
<?xml version="1.0" encoding="utf-8"?>
<config xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <tasks>
    <task method="post">
      <id>1</id>
```

```
<address>http://www.domain.com/ActiveQ/ActiveQTestWeb/WebForm1.aspx</addr
<datestart>2003-05-22</datestart>
<dateend>2003-06-30</dateend>
<recurrences>
  <recurrence>
    <type>daily</type>
    <hour>15:51:00</hour>
  </recurrence>
  <recurrence>
    <type>daily</type>
    <hour>15:52:00</hour>
  </recurrence>
  <recurrence>
    <type>selectedDay</type>
    <parameter>Friday</parameter>
    <hour>11:30:00</hour>
  </recurrence>
</recurrences>
</task>
<task method="get">
  <id>2</id>
  <address>http://www.google.com</address>
  <datestart>2003-05-22</datestart>
  <dateend>2003-06-12</dateend>
  <recurrences>
    <recurrence>
      <type>selectedDay</type>
      <parameter>Friday</parameter>
      <hour>11:30:00</hour>
    </recurrence>
  </recurrences>
</task>
<task method="post">
  <id>3</id>
  <address>http://www.yahoo.com</address>
  <datestart>2003-05-22</datestart>
  <dateend>2004-05-30</dateend>
  <recurrences>
    <recurrence>
      <type>daily</type>
      <hour>12:02:00</hour>
    </recurrence>
```

```
<recurrence>
  <type>selectedDay</type>
  <parameter>Friday</parameter>
  <hour>11:30:00</hour>
</recurrence>
</recurrences>
</task>
<task method="post">
  <id>3</id>
  <address>c:\system\backup.bat</address>
  <datestart>2003-02-22</datestart>
  <dateend>2005-02-30</dateend>
  <recurrences>
    <recurrence>
      <type>daily</type>
      <hour>00:00:00</hour>
    </recurrence>
  </recurrences>
</task>
</tasks>
</config>
```

If not already done by your server administrator, your configuration file needs to be added to the scheduled tasks queue. This can be achieved with the Server Manager.

Open the configuration panel from the **Start -> Programs -> Active Up -> ActiveQ -> Configuration**.

In the XML files tab, click on **Add**.

Browse and Select the task file. Click **Open**.

Click on the **Ok** button.

The tasks configuration file will be processed and loaded in the scheduling.

Using the Manager



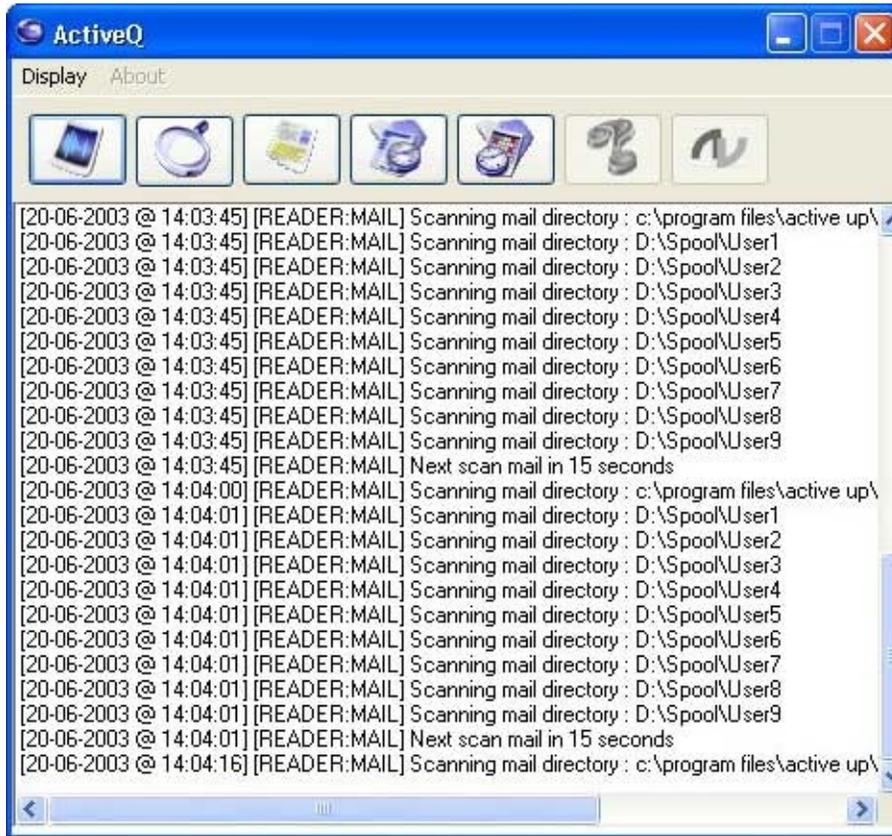
1. **Right click** on the **Manager icon** in the **task bar**.
2. **Select Start, Pause, Resume or Stop.**

Using the Windows Services control panel

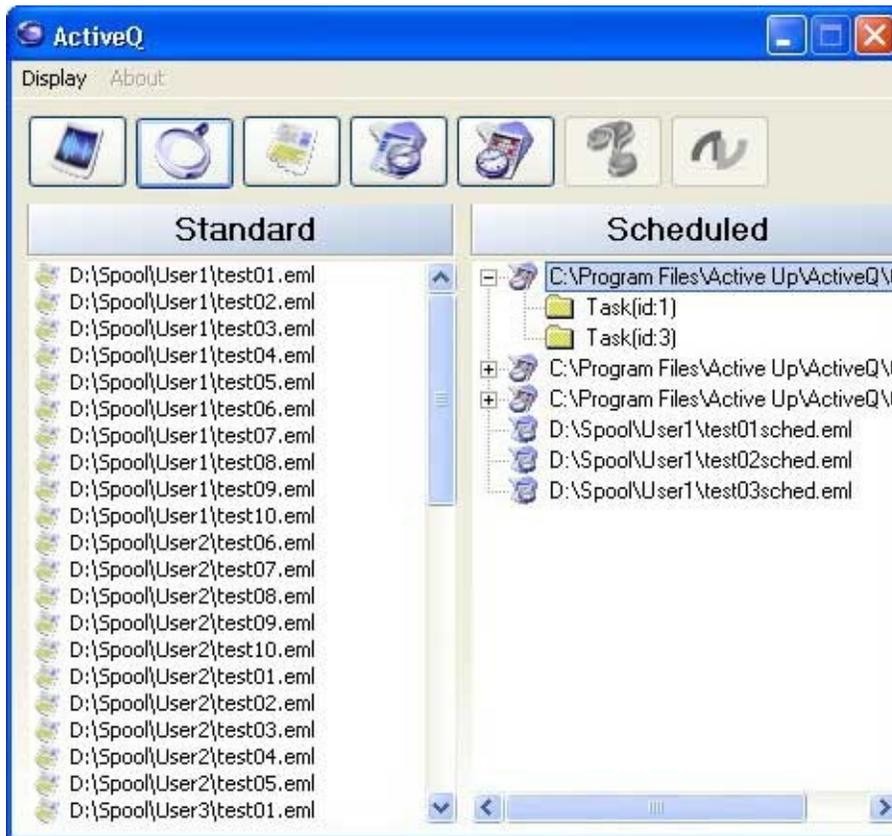
1. **Click on Start -> Settings -> Control Panel.**
2. **Double click on Administrative Tools folder.**
3. **Double click on Services icon.**
4. **Select the service named ActiveQ.**
5. On the control bar, **choose** the action you want **Start, Stop, Pause, Restart, etc.**

Using a Command Line

1. **Click on Start -> Programs -> Accessories -> Command Prompt.**
2. In the (DOS) window, type ***net start ActiveQ*** to start the service or ***net stop ActiveQ*** to stop the service, then press **Enter**.



This screen will display what **ActiveQ** is doing or has done. It will give you additional information about scheduled events and more.



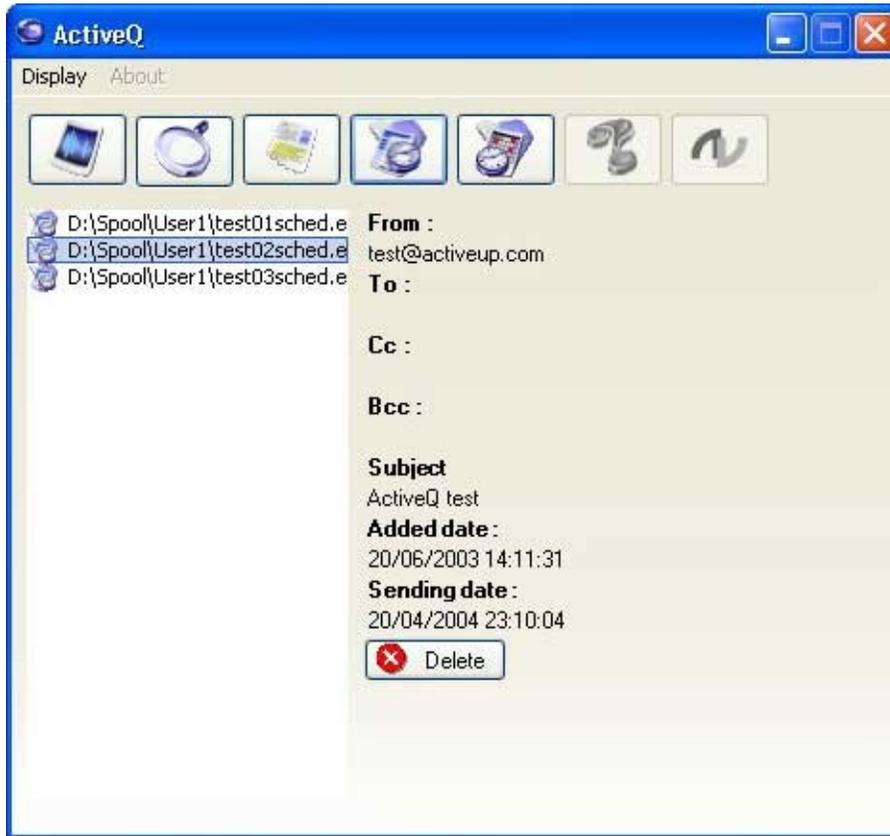
This screen display an overview of the 2 queues: Standard (immediate processing using FIFO) and Scheduled (both emails and tasks).

The information contained in these lists are updated in real time.



If you want to display information about a queued email, just select it. The From, To, Cc, Bcc and Subject fields (headers) will be displayed.

If for any reason you want to stop a specific email, select it and click on the Delete button. This will also delete the file in the pickup directory.



Like the Standard Mail Queuing window, this screen displays the emails that are actually queued, but scheduled. You can delete an email by selecting it and clicking on the Delete button.



Task files are displayed using a tree view. Each node represents a specific task in the task file.

You can delete a specific task by selecting it and clicking on the Delete button. This will not delete the XML tasks configuration file, but only the specific task contained by the file.

You can delete the whole file (and all tasks contained in that file) by selecting the master node and clicking in the Delete button.

This part will be available when we have enough feedback from our customers. If you have any problems, please go directly to the [Support Options](#).

Many solutions can be found in this documentation but you can also post your question to the [Active Up Online Community](#).

The other support options are available to trial users and registered users :

- [Live Support](#). It is generally available from 10h to 19h (GMT+1). Live support uptime is not guaranteed.
- [Support Form](#). The most direct and effective way to contact us.

If you don't want to buy [ActiveMail](#) and take advantage of our premium support, you can buy a separate support agreement. Please contact our sales department for details : sales@activeup.com.