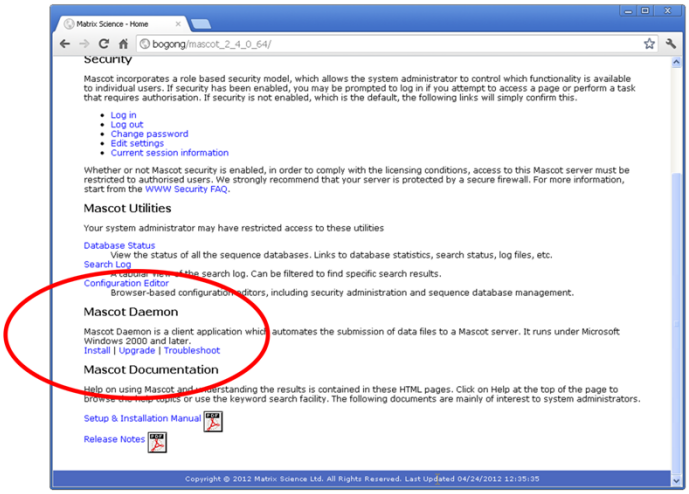
Automation with Mascot Daemon

MASCOT

MATRIX SCIENCE

Installing Mascot Daemon

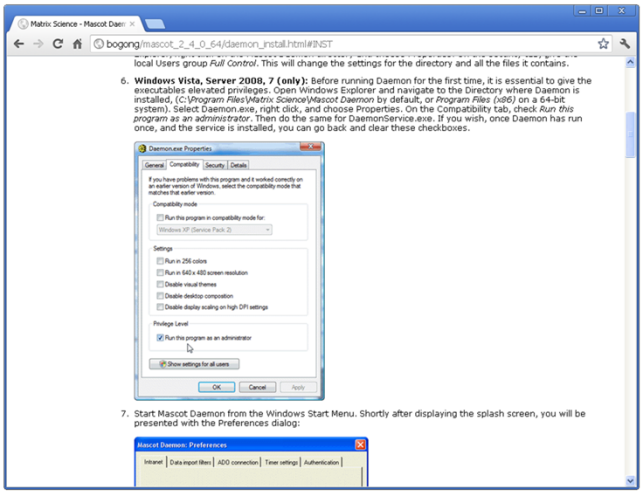


Mascot Daemon is our automation client. It's a part of the Mascot package that is only available when you have a Mascot server in-house.

It is only available for Windows because all of the mass spectrometry data systems are Windows packages, so this is always the platform where the data files originate.

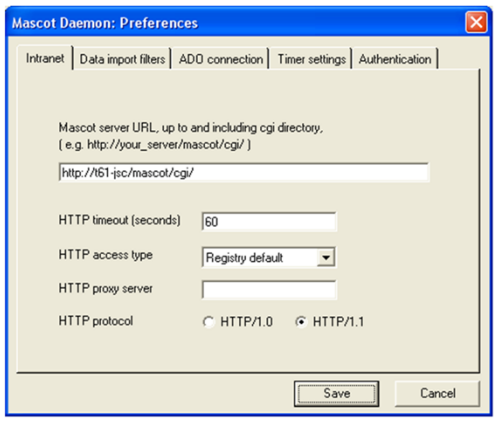
Links to install and update Daemon can be found on your local Mascot home page, so that anyone with access to the server can easily get a copy of Daemon. You should try and make sure that you have the same version of Mascot Server and Mascot Daemon installed. An older version of Daemon will probably work, but it won't support features in later versions of Mascot Server.

Installing Mascot Daemon



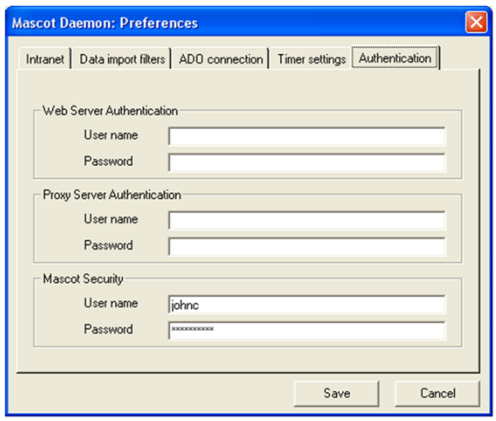
If installing on Windows Vista or later, it is essential to follow the instructions and change the properties of both executables so that they ‘Run as Administrator’. Once the Mascot service is installed and the settings saved to the registry, you can revert to running as a standard user, if you wish.

Installing Mascot Daemon



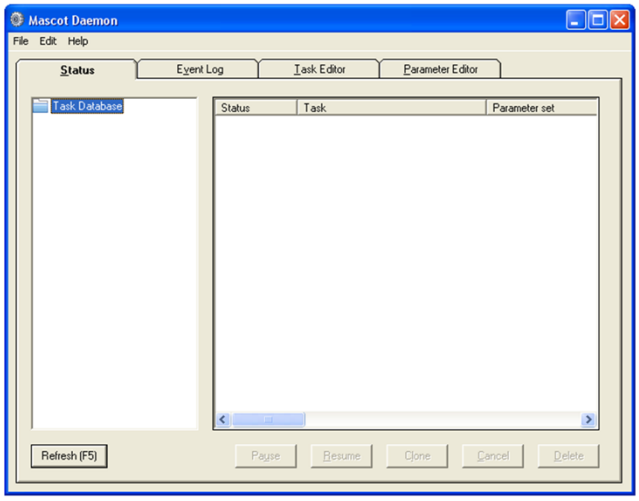
When Mascot Daemon is first launched, this dialog will be displayed and you have to enter the URL of your local Mascot server to proceed. Note that you cannot enter the public web site URL … it simply won't work. If Mascot security is enabled, you will also need to enter a valid user name and password on the authentication tab.

Installing Mascot Daemon

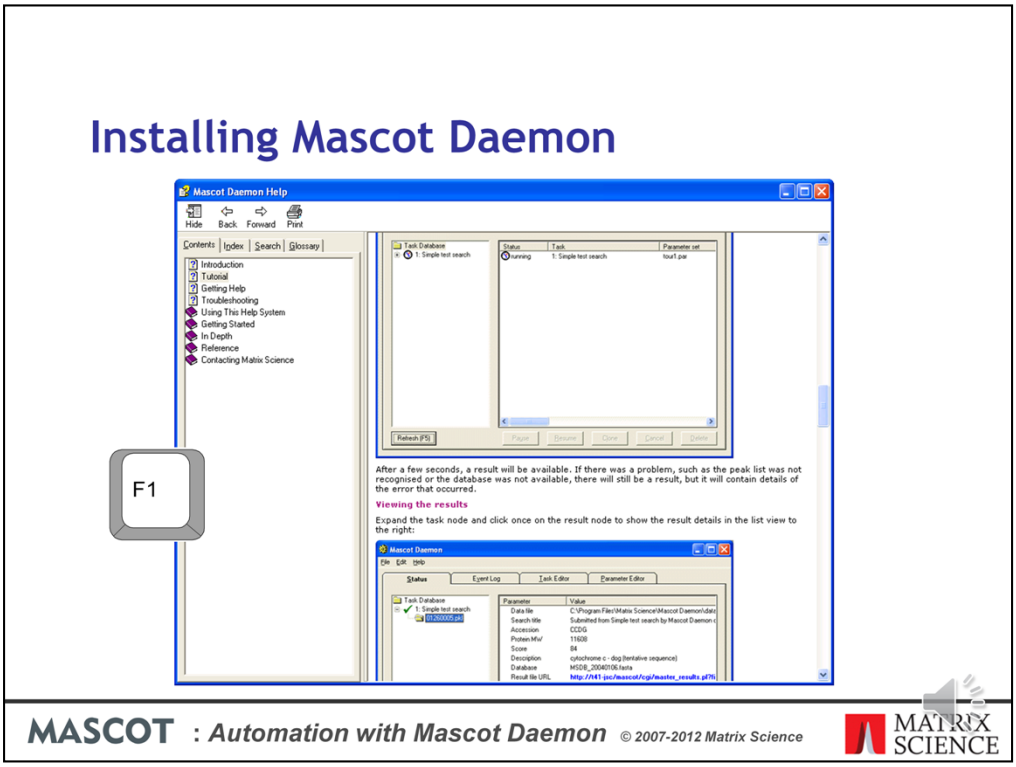


Mascot security is not the same as web server security or having to tunnel through a firewall or authenticated proxy server. There are separate fields for each of these.

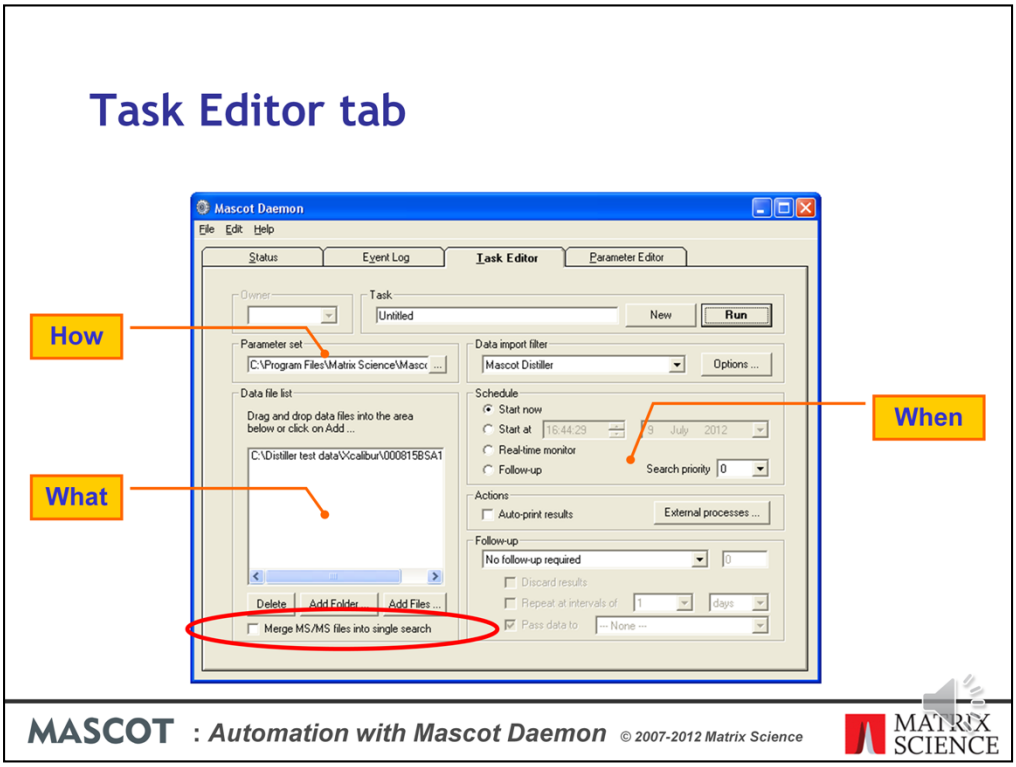
Installing Mascot Daemon



Once Daemon has connected to the Mascot server, you will see a display like this. The user interface is a simple, tabbed dialog box



Mascot Daemon has comprehensive, context sensitive help. Just press F1. There is a tutorial to get you up to speed and detailed reference material.



Lets look at the tabs in more detail.

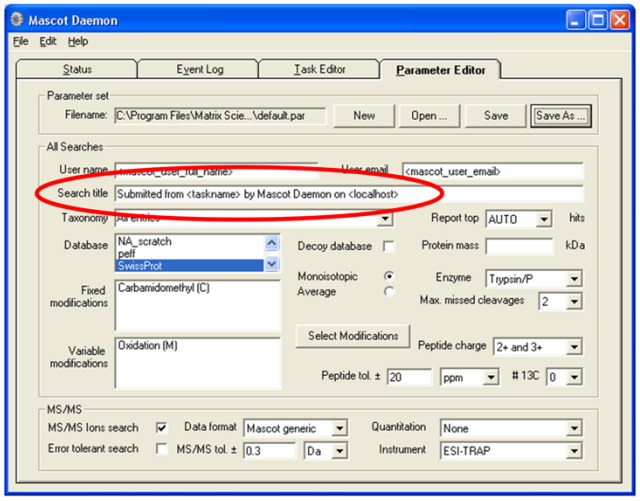
The Task Editor tab is used to define each task. A task defines *what* data will be searched and and *when* the search will take place. A Parameter set is selected that contains the search parameters. This determines *how* the data files will be searched.

Here we have a very simple batch task. A single data file has been selected, a parameter set has been chosen, and the task will run the search as soon as we press run.

The data files can be simple peak lists. In this case, we are using Mascot Distiller as a data import filter and the data file is a RAW file. More about data import filters later.

Note the checkbox at the bottom left. Merging a batch of files into a single search can be very useful. MudPIT fractions, for example

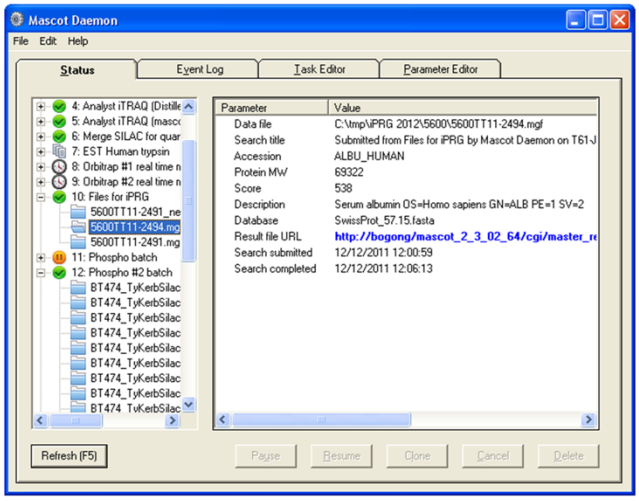
Search Parameter Editor



The parameter editor allows sets of search parameters to be defined and saved to disk, so that they can be used over and over again by tasks. The search parameters define *how* the data will be searched. As you can see, the fields are very similar to those on the web browser search form.

The search title includes some tokens, or variable, that will be replaced when the task runs. Here, for example, <taskname> will be replaced by the name of the task and <localhost> will be replaced by the name of the computer on which Daemon is running. This will make it easy to identify where the searches came from when looking at the Mascot search log on the server. There are many such tokens. You can find details in the help file.

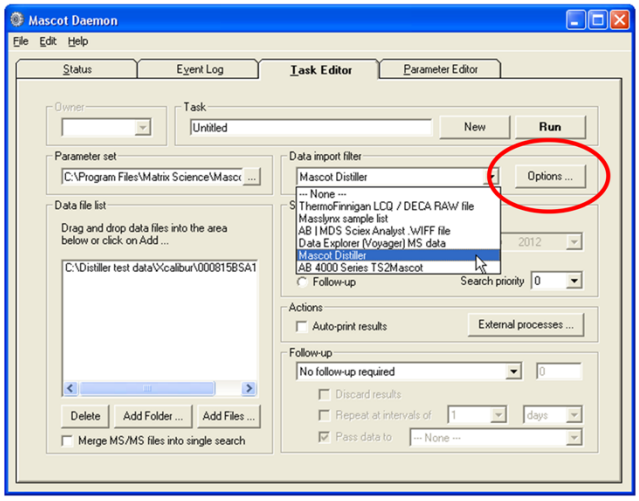
Status tab



Each search result is added to an Explorer-like tree on the Status tab. There is a minimum of summary information, and a hyperlink to the full result report. Mascot Daemon isn't intended to be an interface to browse Mascot results, it is just a tool to automate search submission and keep track of results

Data files within a particular task are run serially, but multiple tasks can be set running at the same time. This means that several data files can be processed and search the same time.

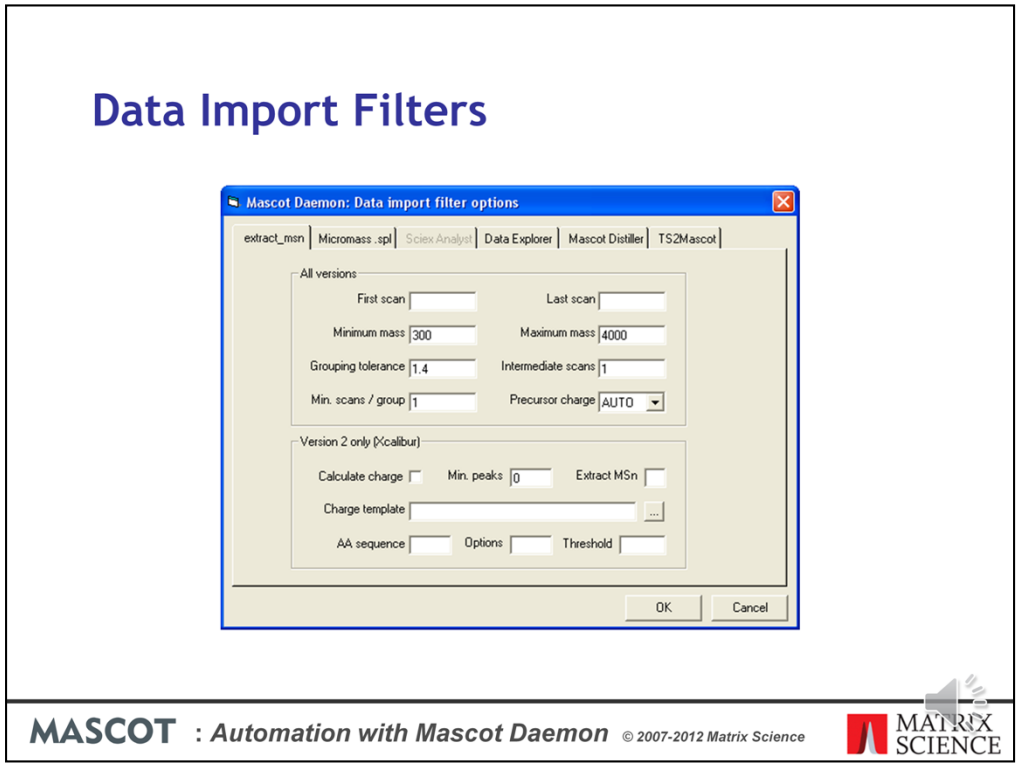
Data Import Filters



Going back to the task editor tab. For some file formats, we have data import filters. These allow Daemon to generate peak lists directly from the binary files created by the MS data system.

These data import filters are only listed if the relevant software packages are installed. For Mascot Distiller, you need to have a licence for the Daemon Toolbox

Next to the drop down list, you have a button to view the data import filter options

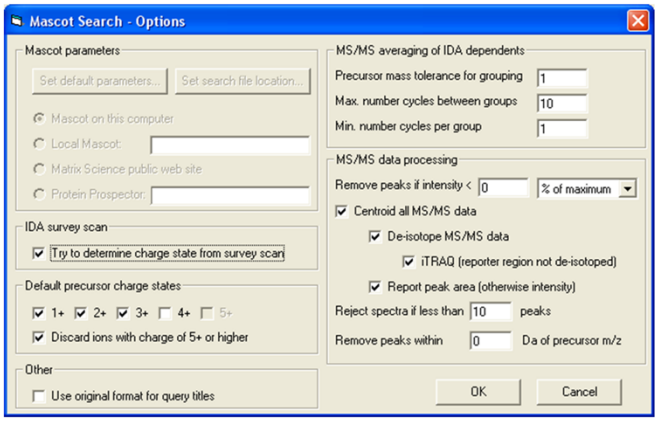


Each filter has different options, and all the options are described in the Mascot Daemon help - just press F1

The extract_msn filter is a Thermo utility that enables Daemon to create peak lists from Thermo Xcalibur .RAW files. Mascot Daemon simply calls the utility using the parameters specified in this dialog. A brief description of each argument is provided in the Mascot Daemon help. extract_msn can be downloaded from the Thermo web site, but Thermo is no longer developing it, so it cannot be used with their latest instruments, such as the Q-Exactive.

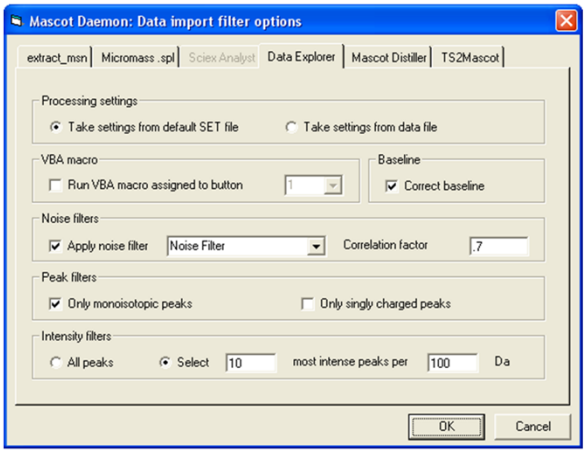
The Micromass filter was just a way of locating pkl format peak lists and is obsolete now that you can submit searches direct from ProteinLynx Global Server. It will be dropped in a future release

Data Import Filters



The AB Sciex Analyst filter uses the Mascot.dll script that is provided with Analyst and supports the QTrap and QStar families of instruments. If you are using a recent version of the mascot.dll script, the controls in the options dialog will vary slightly according to the version of Analyst.

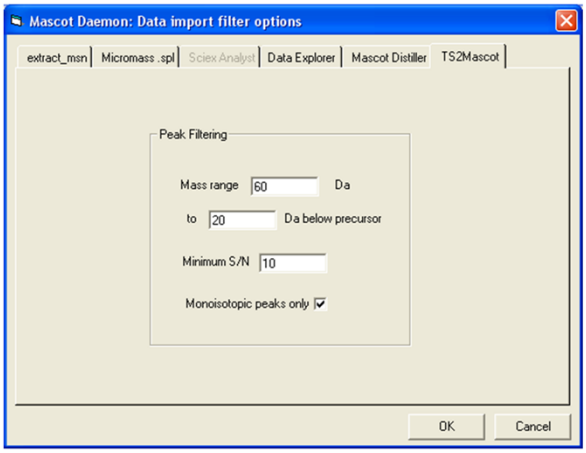
Data Import Filters



MASCOT : *Automation with Mascot Daemon* © 2007-2012 Matrix Science MATRIX SCIENCE

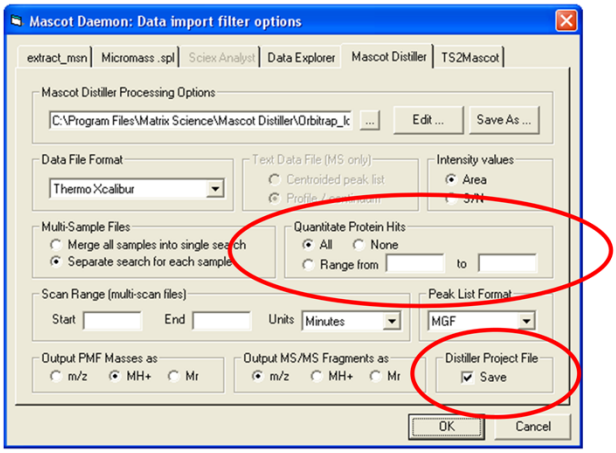
The Data Explorer filter can be used to process data files from AB Sciex Voyager systems or T2D files exported from TOF/TOF data.

Data Import Filters



TS2Mascot imports AB Sciex TOF/TOF peak lists direct from the Oracle database.

Data Import Filters



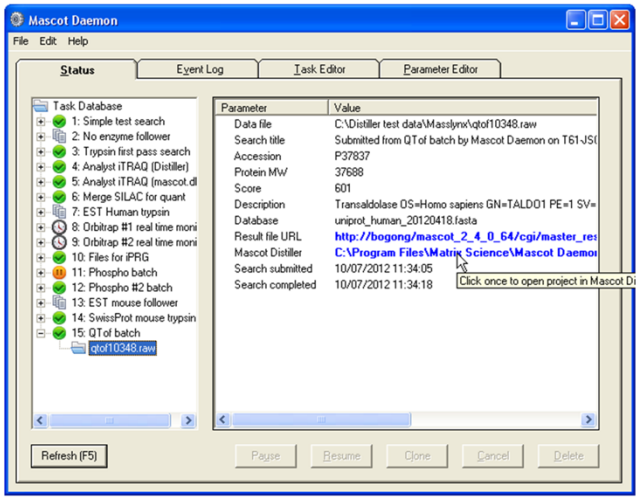
The most powerful data import filter is Mascot Distiller, because it can import all of the major raw file formats. The dialog is simple because most of the peak picking options are contained in the selected processing options file.

The idea is that you adjust the processing options by working interactively with a typical data file in Mascot Distiller Workstation. You then save these options to a file and select it here.

Notice the checkbox at the bottom right. You can ask Daemon to save a project file containing the Peak lists and Mascot search results.

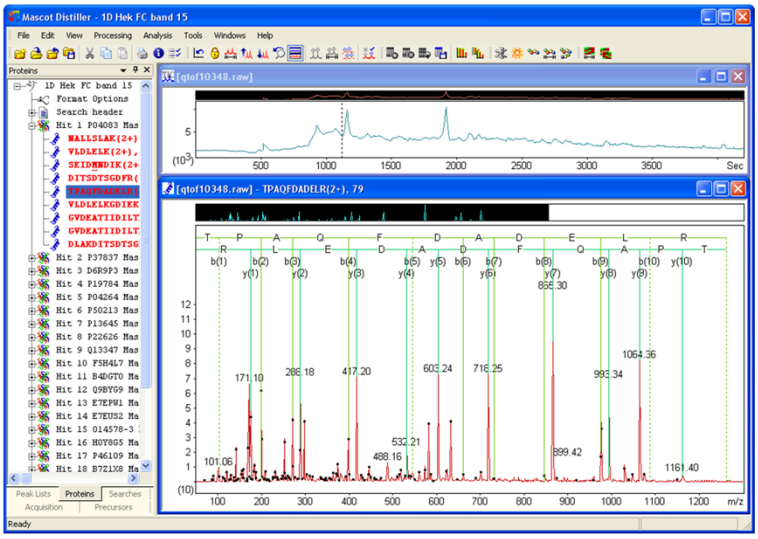
In Mascot Daemon 2.3, you can also batch automate quantitation. We'll be discussing quantitation in detail in a later talk.

Data Import Filters



If Distiller is used for peak picking, once the search has completed, you get a hyperlink to open the project in Mascot Distiller. Click on the link and …

Data Import Filters

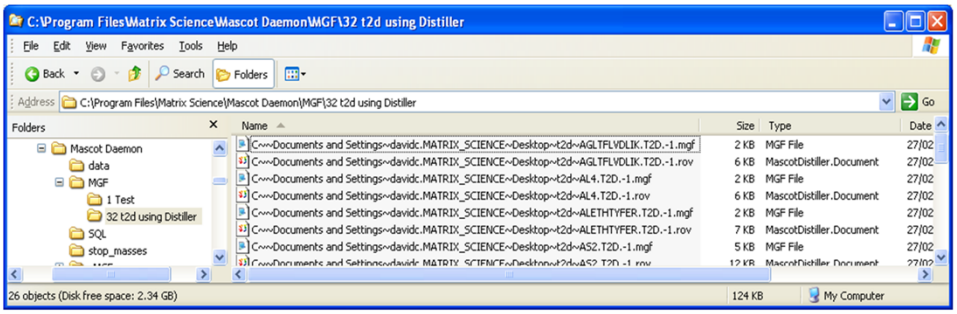


MASCOT : *Automation with Mascot Daemon* © 2007-2012 Matrix Science MATRIX SCIENCE

Shazam

Data Import Filters

- **Peak lists created by Daemon are saved to disk:**
  `<app folder>\MGF\<task name>\<data file path>`



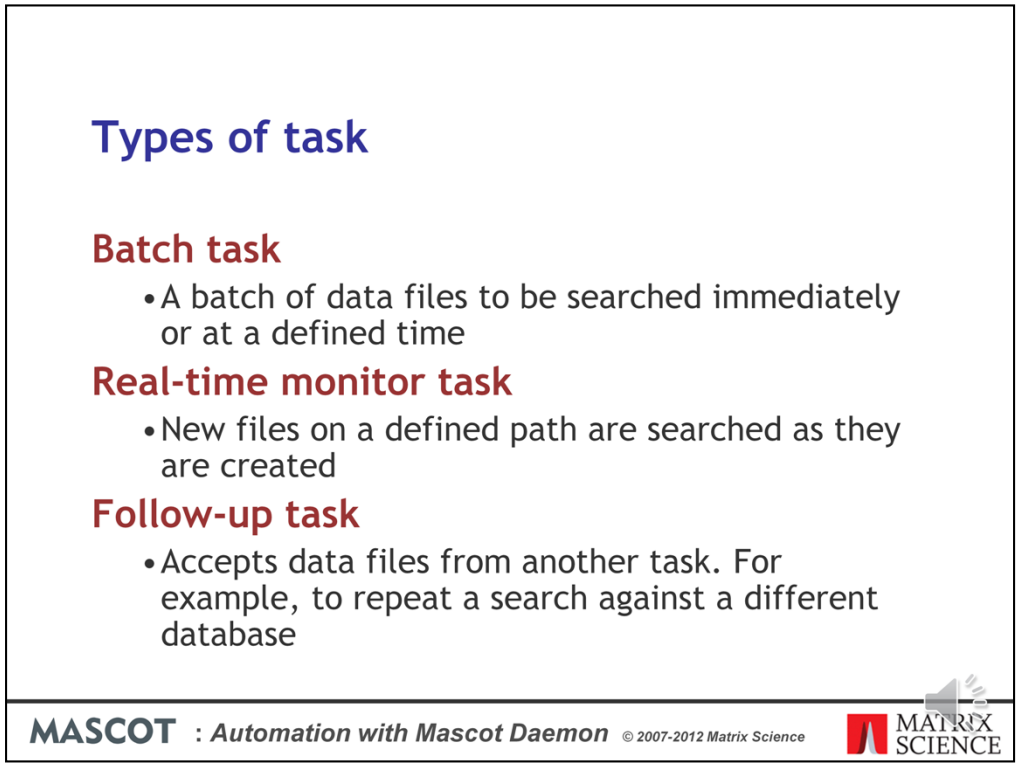
MASCOT : *Automation with Mascot Daemon* © 2007-2012 Matrix Science MATRIX SCIENCE

•Peak lists created by Daemon are always saved to disk. This is the default path, which can be changed in preferences.

•When you delete a task, any peak lists are also deleted. If you have used Mascot Distiller as in import filter, then the distiller project file is also deleted. The raw data files are never deleted.

•In this example, all the tasks between 1 and 32 have been deleted.

•Task 32 was called “t2d using Distiller” and you can see that for each raw data file, a Distiller project file and a Mascot Generic Peak list file has been created



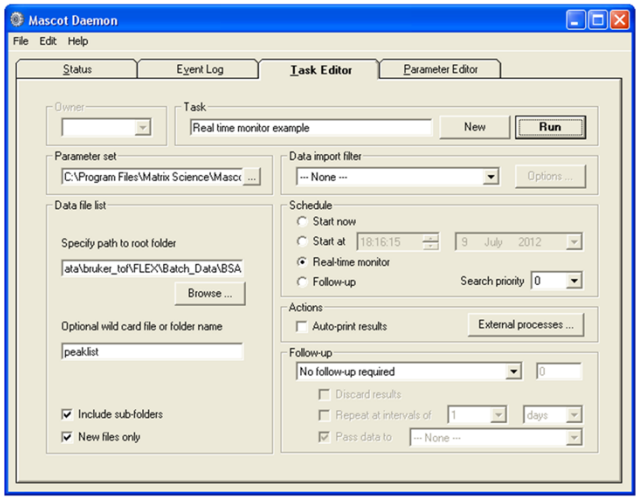
Lets talk in more detail about tasks. Mascot Daemon supports three kinds of task:

A batch task allows you to search one or more specified files immediately or at a defined date and time

A real-time monitor task is the key to real automation. You define a task using a file path for the data file which includes wild cards. Any file that matches this path is picked up and searched.

A follow-up task is a task where the data files are passed to it by another task. This is very powerful because it allows searches to be chained together to implement complex workflows. For example, a batch of data files might be screened against a contaminants database containing entries for keratins, BSA, trypsin, etc. Those data files which fail to find a match can then be automatically searched against a non-redundant protein database. Spectra which are still unmatched can then be searched against a large EST database, etc., etc.

Real-time Monitor Tasks



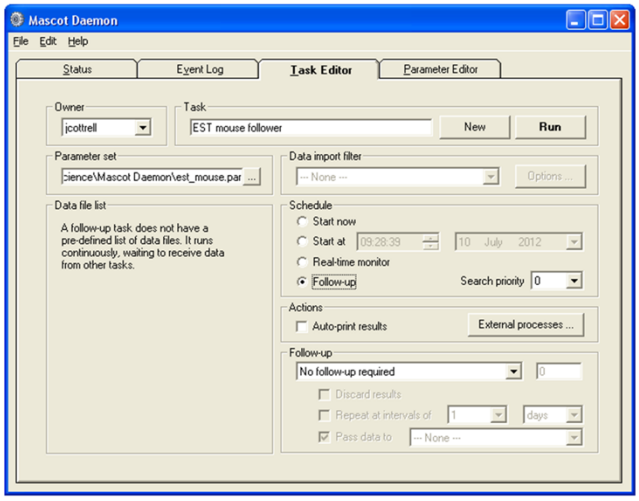
When you select real-time monitor in the schedule frame, the task editor tab looks like this. Here, we are looking in the specified directory and all its subdirectories, for any file called peaklist.

Note that the final part of the path says “file or folder name”. This is because some raw files are actually folders, not files.

If Include sub-folders is checked, then Daemon will search down through all sub-directories of the root folder

If new files only is checked, then any files that exist when the task is started will be ignored. A common problem when using real-time monitor like a batch task is to have this checked and then wonder why nothing is happening

Follow-up Tasks



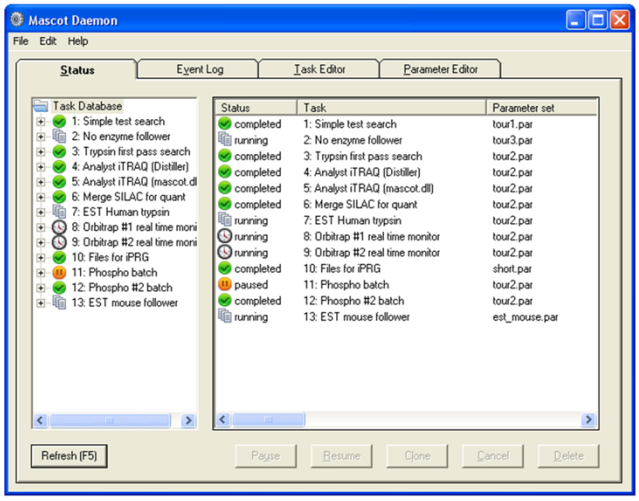
MASCOT : *Automation with Mascot Daemon* © 2007-2012 Matrix Science MATRIX SCIENCE

Follow-up tasks are a little more difficult to explain. Any task can pass data to a follow-up task to be searched again in a different way. Only follow-up tasks can receive data from other tasks in this way.

Lets illustrate with an example. Imagine we want to search our MS/MS data files against the mouse sequences plus a contaminants database as a first pass search. We then want to search all spectra that fail to get a significant match against a database of mouse EST sequences.

We start at the end of the chain, and define the EST search as a follow-up task. We just choose Follow-up in the schedule frame and a parameter set, created earlier in the parameter editor, that contains the required search parameters.

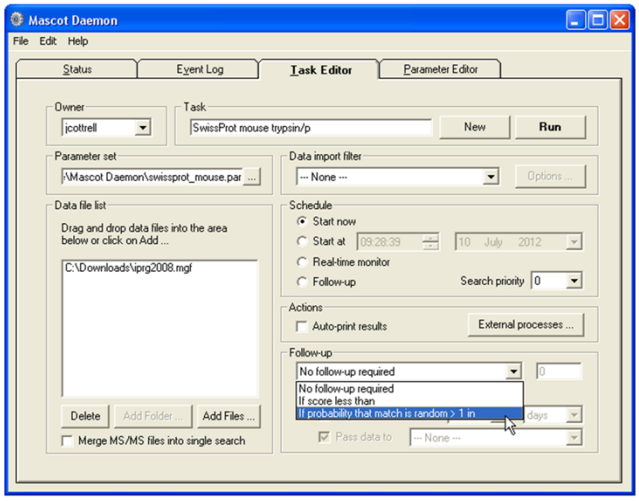
Follow-up Tasks



MASCOT : *Automation with Mascot Daemon* MATRIX SCIENCE

We press run, and we see our Follow-up task running. Although the task is running, nothing will happen until we feed some data into it

Follow-up Tasks

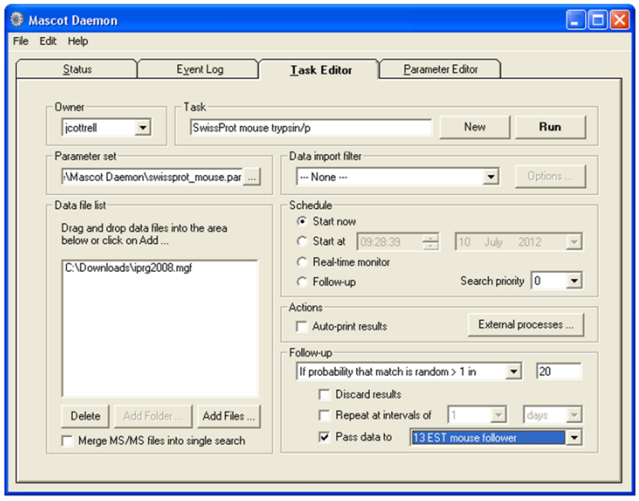


MASCOT : *Automation with Mascot Daemon* © 2007-2012 Matrix Science MATRIX SCIENCE

We go back to the task editor and create a simple batch task. There is one data file and the search parameters are a different set, in which the databases are SwissProt with a taxonomy filter of rodents plus second database of contaminant sequences.

Before pressing Run, we have to link this task to the follow-up task. We need to specify a condition for passing the data to the follow-up task. This can be score based or probability based.

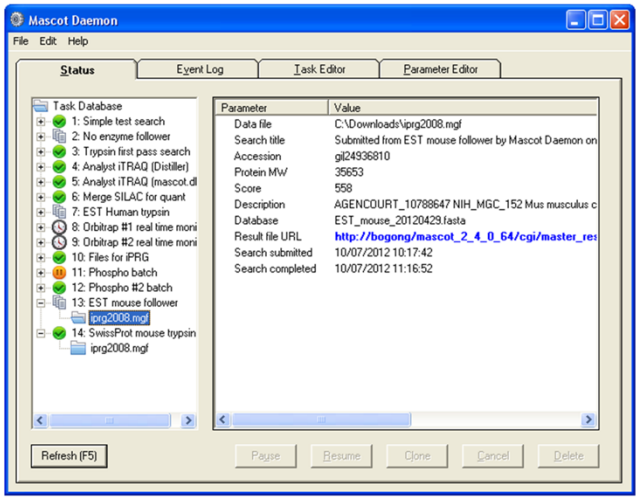
Follow-up Tasks



MASCOT : *Automation with Mascot Daemon* © 2007-2012 Matrix Science MATRIX SCIENCE

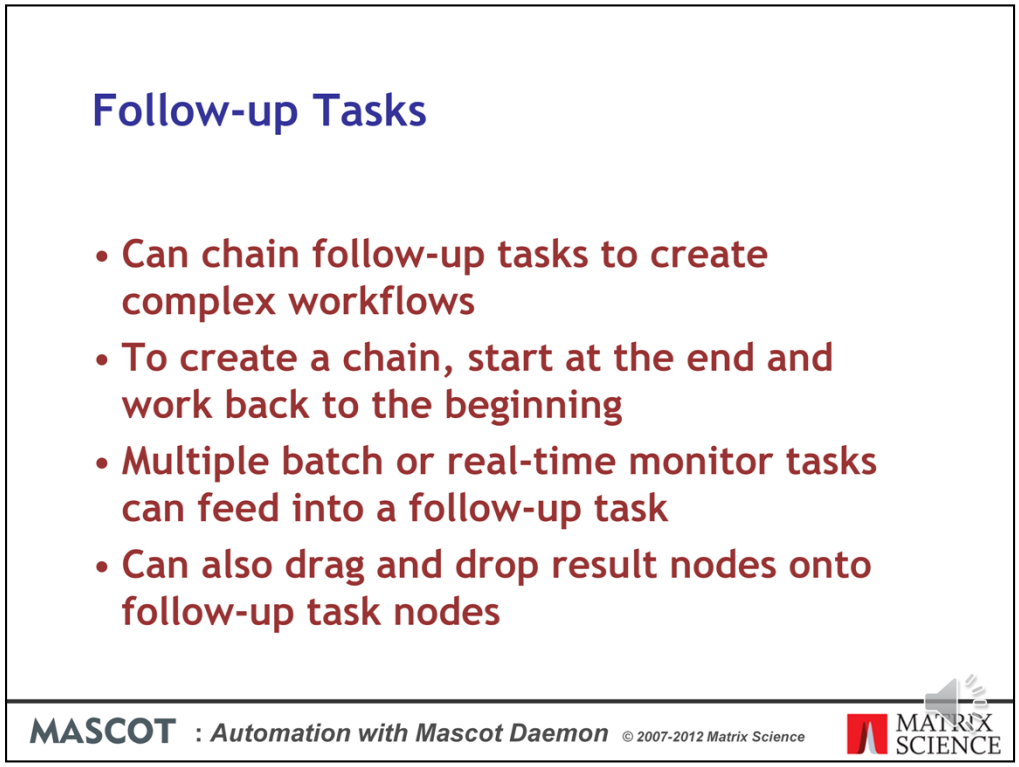
For this example, we will use the default significance threshold of 5%. If the score a peptide match in the first search is below a 5% significance threshold, then the spectrum will be passed to the follow-up task we created earlier. Note that you also have the option to repeat the search at intervals, maybe to see whether a better match can be found in some future update to the sequence database. The task is now configured, and we can press Run.

Follow-up Tasks



MASCOT : *Automation with Mascot Daemon* © 2007-2012 Matrix Science MATRIX SCIENCE

Some time later, the status tab looks like this. The results for the first search appear under task 14. Those spectra that failed to obtain a significant match were passed to the follow-up task, task 13, where the result information appears once the search has completed

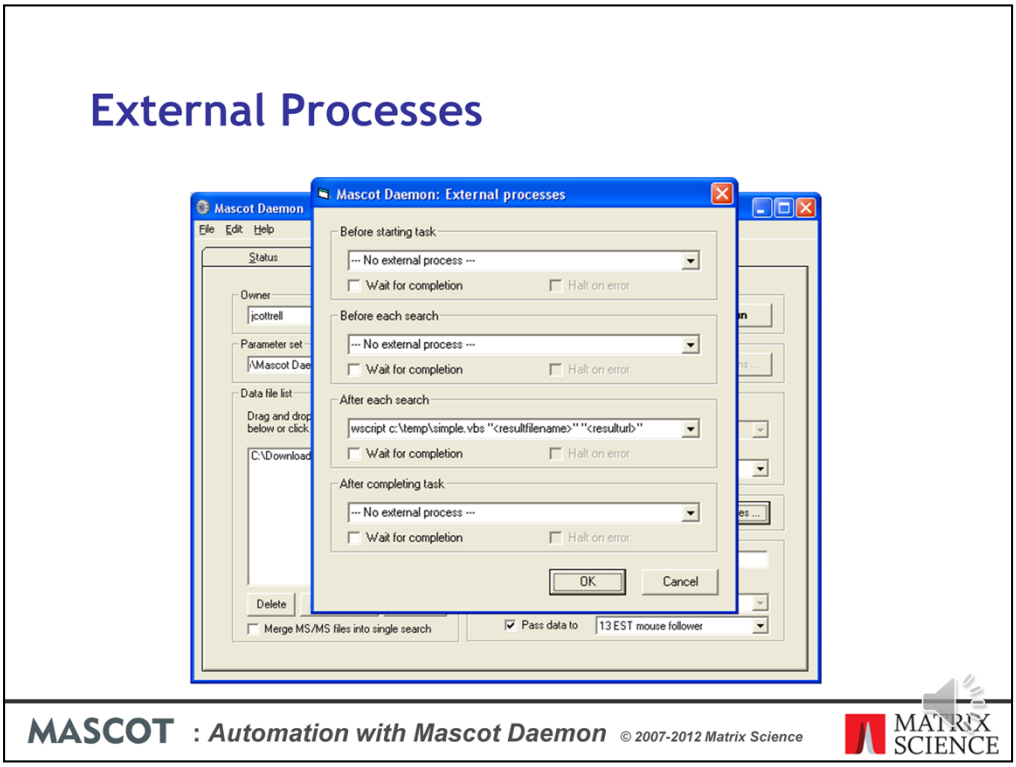


•One follow-up task can pass data to another follow-up task, creating a chain of tasks to implement a complex workflow. Thresholding applies to individual spectra. If there are 1000 spectra, and 100 get a good match, we take the 900 that didn't get a match and pass them to the follow-up task. You can think of the chain of tasks as a series of finer and finer sieves.

•Remember that, when you define the chain, you have to start at the end of the chain, with the last task, and work forwards to the first task in the chain

•Having created a follow-up task, or a chain of follow-up tasks, multiple batch or real-time monitor tasks can feed into it. It just sits there, waiting

•Note that any result node can be dragged and dropped onto a follow-up task node to repeat the search using a different parameter set



We'll briefly touch on a few other features of Daemon. The task editor tab includes a button labelled 'External processes'. A Mascot Daemon task can execute external processes at one or more of the following time points:

•At the start of a task

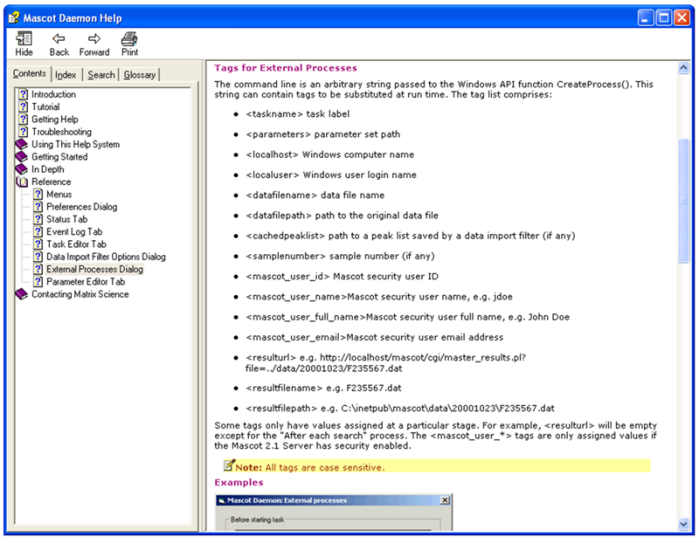
•Before submitting each search in a task

•After completion of each successful search in a task.

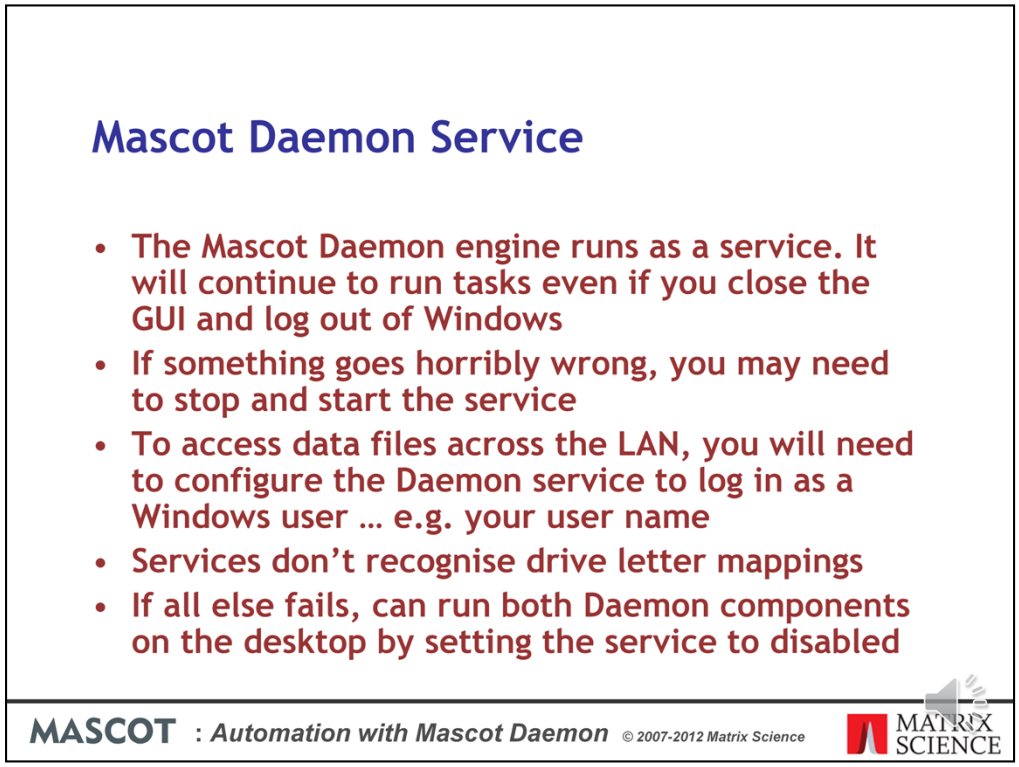
•At the end of a task

A process is a one line command. This could execute a Windows program, or it could be a batch file or script to perform some more complex procedure. You can pass arguments to the commands by including tokens, such as <taskname>

External Processes



The help file contains a list of all the available tokens. It also gives examples of how to use External processes, such as a script to build an HTML page in real-time that contains hyperlinks to the search results.



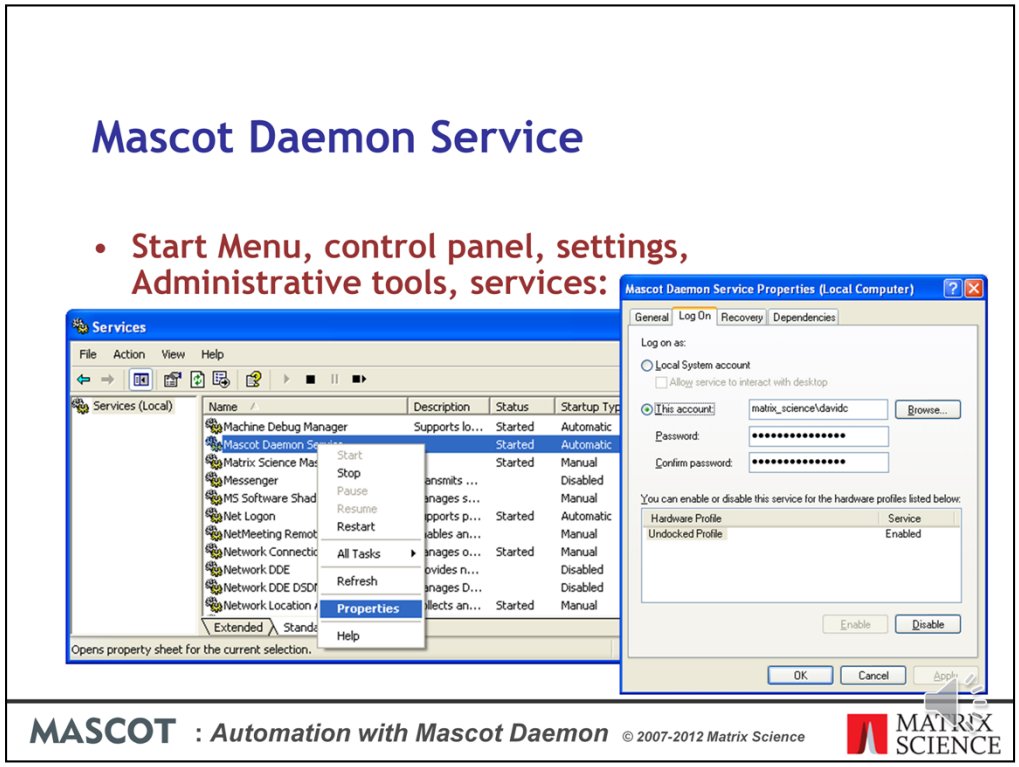
Mascot Daemon is divided into two components: a Graphical User Interface (GUI) and a Service. We have been looking at the GUI, which is used to configure Daemon, edit tasks, view search results, etc. The service is responsible for executing the tasks, in the background, whether the GUI is running or not. It will continue to run tasks even if you close the GUI and log out of Windows

If something goes horribly wrong, you may need to stop and start the service to clear the problem

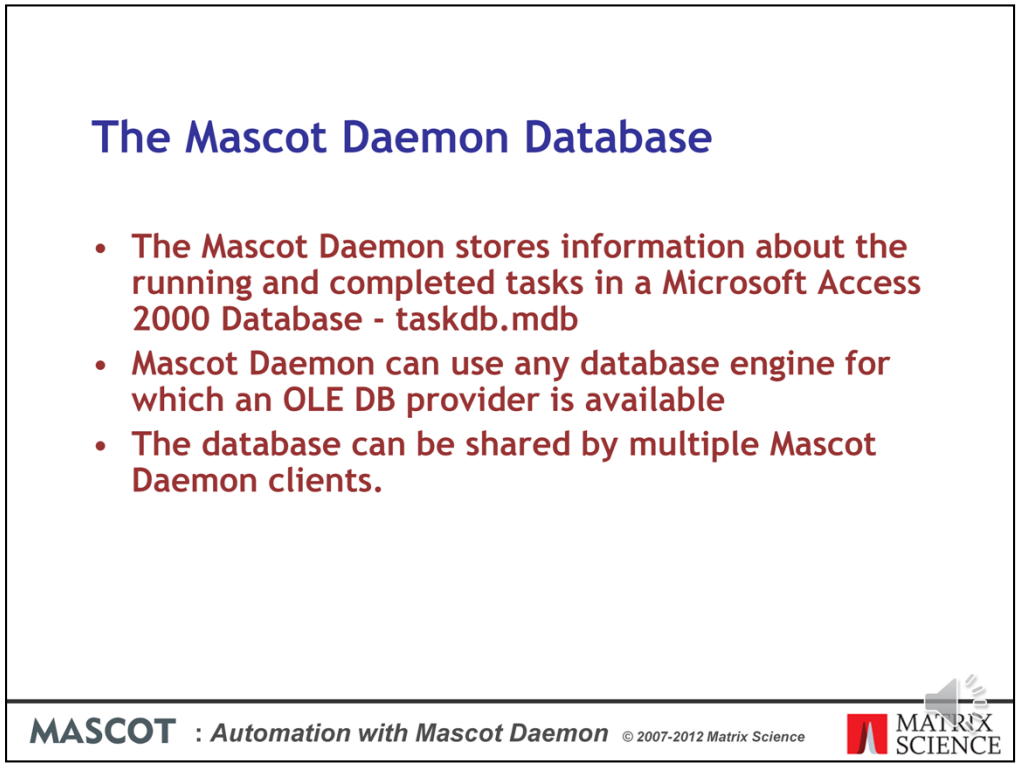
To access data files across the LAN, you will need to configure the Daemon service to log in as a Windows user. In most cases, you will want to have the service use the same Windows log-in as you do. Full details in the help file.

Remember that services don't recognise drive letter mappings. For files on remote computers, Daemon will use UNC paths and must be able to access these files without being challenged for a password

If this proves impossible, for example, if the files are on a SAMBA server, you can run both Daemon components on the desktop by setting the service to disabled



To change the service login you need to use the services control panel. Find and select the MascotDaemonService in the list. You can stop or start the service using the controls at the top, or by right clicking to get the context menu. To change the user, click properties and then, on the log on tab choose “This account” and enter your user name and password. Remember that if you change your password, you will also need to come back to this screen and also change it here.



Under the hood, Mascot Daemon stores information in a Microsoft Access Database called taskdb.mdb. It is possible to use other databases, and this should be setup in the “ADO Connection” tab of the Mascot Daemon Preferences.

Multiple Mascot Daemon clients can share the same task database. Each task is "owned" by the client that created it. Clients can only Pause, Resume, Cancel, or Delete their own tasks. However, any client can Clone any task and can transfer data files to any follow-up task, either by dragging and dropping the result node or by specifying the task in follow-up criteria.