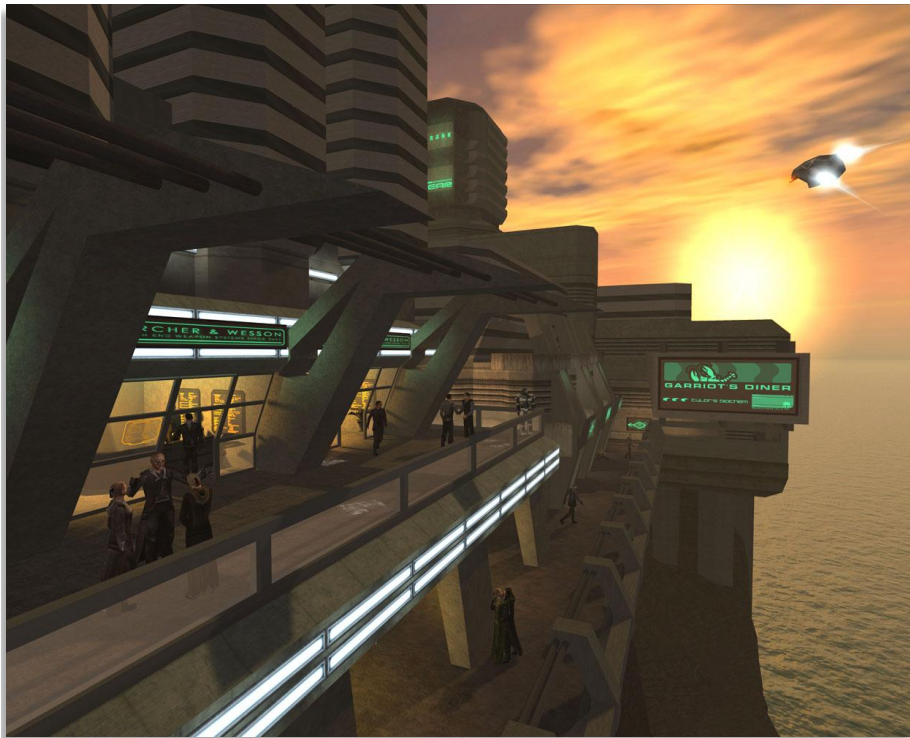Neocron Utilities

Cronalyser - User Manual

Monitor all your chat feeds in real-time!

Nick "Nidhogg" Milner

Use with Neocron 2 and any other application that outputs a text output file!

Cronalyser - User Manual

Monitor all your chat feeds in real-time!

Nick "Nidhogg" Milner

Contents

Frequently Asked Questions

Let's just cut to the chase here with the Frequently Asked Questions. More detailed information can be found in subsequent chapters.

What is Cronalyser?

Cronalyser is a tool that notifies you whenever key events happen. It does this by monitoring any number of simultaneous data streams, e.g. Neocron log, mIRC log etc. Here are some examples of what Cronalyser can do for you. With some imagination you could apply them to other games or general chat applications, but why you would want to play anyone else's game is beyond us so we'll restrict the examples to Neocron.

> You are a constructor of sublime weaponry but you don't have the time or patience to watch all the chat channels for people asking for construction services. You may even be running another character in a different client window. Cronalyser can play a custom sound and pop up a system tray alert balloon whenever anyone asks for a constructor, even if they use any one of many slang names for either the act of construction or the item they would like constructed.
>
> You already have a chat log stored in your logs folder but that file contains anything that anyone has ever said and searching it is a pain. You'd like to generate another chat log that only contains topics you're interested in or anything spoken by your clannies.
>
> You'd like a loud claxon to sound and a message to pop up from your system tray saying "OMGWTFPWN! OP H4x00red!!11oneone" when someone hacks a layer of one of your captured outposts (well, you might be annoying like that)! You *hate* it when you miss that little text warning that disappears off the top of your chat window so quickly goddammit!
>
> You'd like to keep an eye on your mIRC or other messenger system while you're playing. If someone mentions something you're interested in you want to know about it immediately.
>
> You'd like to search one or more log files that you've collected for something. You can't remember exactly what it was and a simple text search just isn't working for you.
>
> On a completely different note, you might want to monitor your system's output and error files for critical errors and be notified immediately.

How much does Cronalyser cost?

Nothing, it's freeware! All we ask is that you keep this document together with the executable if you distribute it onwards. We'd also appreciate it if you checked out our game over at http://www.neocron.com. You'll be glad you did!

Cronalyser - User Manual

Monitor all your chat feeds in real-time!

Nick "Nidhogg" Milner

What are the system requirements?

Cronalyser needs the .NET Framework 2.0 to be installed for it to work. This can be downloaded for free from Microsoft's website:

http://www.microsoft.com/downloads/details.aspx?familyid=0856eacb-4362-4b0d-8edd-aab15c5e04f5&displaylang=en

It just says "Application cannot be initialised" or some such nonsense.

See the entry above regarding the .NET Framework.

What is Neocron?

Neocron is a sci-fi MMORPG that plays like an FPS. Come check us out at http://www.neocron.com!

What search modes are provided?

Cronalyser provides two search modes:

> One Shot – simply applies your search rules to one or more existing files. Any lines that match are displayed in the output window. Simple, but not really what Cronalyser is about…

> Free Running – this mode monitors the selected files in real-time and is the *real* power of Cronalyser. This mode allows you to watch any number of files for any number of things you may be interested in through the use of *rules*. Each rule looks for a specific thing in the input stream, and once matched causes that line to be displayed in the output file and, optionally, a custom sound to be played and a message to appear in a system tray alert balloon.

What does a rule look like?

Rules are specified as *regular expressions*. A regular expression is a powerful mechanism for describing search criteria. There are many web sites where you can learn about regular expressions but one I would recommend is: http://www.regular-expressions.info/. We take a look at some examples later on in this document.

People often call the same thing by different names. Is there any way I can simplify my rules and still cope with this?

Absolutely. Cronalyser provides the concept of *Groups* which allows you to create a list of members (themselves regular expressions if necessary) that can be referenced simply by using the group name. E.g. you could create a group called "clanny" that contains the names of all your clan members. You can then refer to any member of your clan in your rules simply as "%clanny%".

I'm monitoring a mIRC log but nothing seems to be happening.

mIRC has an option to "lock" its log files. Make sure that option is not selected.

I can't find any way of saving my stuff

There's no need. Cronalyser will persist all your configuration data to disk whenever necessary.

I've upgraded from an earlier version. Where is all my stuff?

Don't worry. The new version will be below a new version folder (see later on for details of where this is). All you have to do is to copy the files from the old folder to the new one. Cronalyser will automatically upgrade them to the latest version the first time it runs.

What about support?

Cronalyser was developed entirely by myself, primarily as a means of test driving Visual Studio 2005, Visual Basic 2005 and Word 2007. I will provide support where I can but it should be noted that this is entirely independent of Reakktor Media GmbH and neither they, nor myself, may be held liable for any damage that may be caused by the use of Cronalyser.
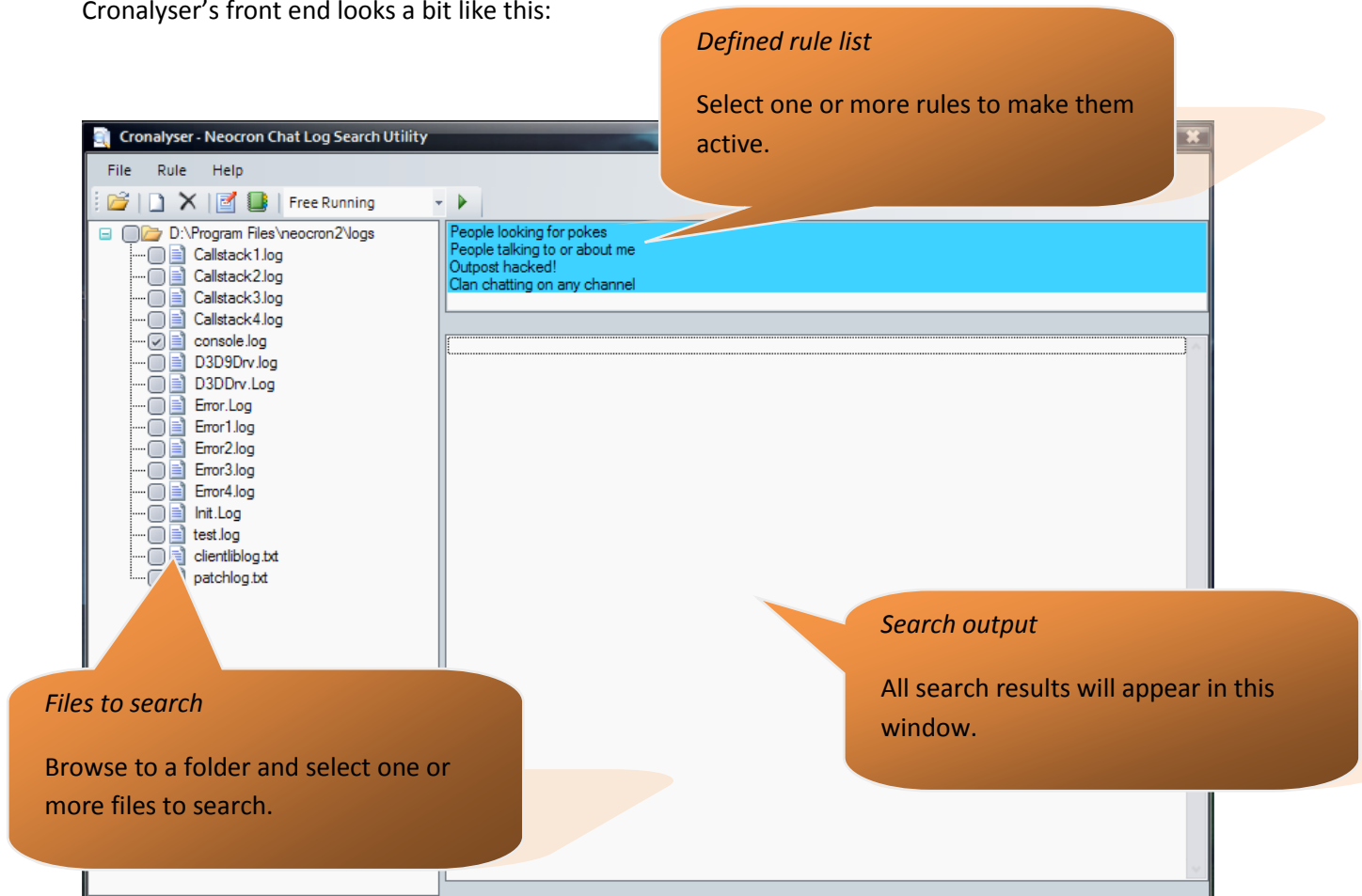
Using Cronalyser

Front End

Cronalyser's front end looks a bit like this:

Defined rule list

Select one or more rules to make them active.

Files to search

Browse to a folder and select one or more files to search.

Search output

All search results will appear in this window.

1) Simply browse to the folder containing your log files and select the files you're interested in. Selecting the folder itself will select all the log files below it. The following file extensions are recognised:
   .log
   .out
   .err
   .txt

2) Select the rules you wish to use during the search. Use <CTRL> to select multiple rules individually or use <SHIFT> to select a range of rules. You can right-click and "Select All" or use the menu option on the *Rule* menu to select all the rules in one go.

3) Choose the mode. "One-Shot" searches the files, presents the results, and stops. "Free Running" will constantly monitor the selected files for search hits until told to stop.
4) Hit the "Go" button on the tool bar or start it from the *Rule* menu or the system tray icon.
5) Go do something more interesting than watching chat files!
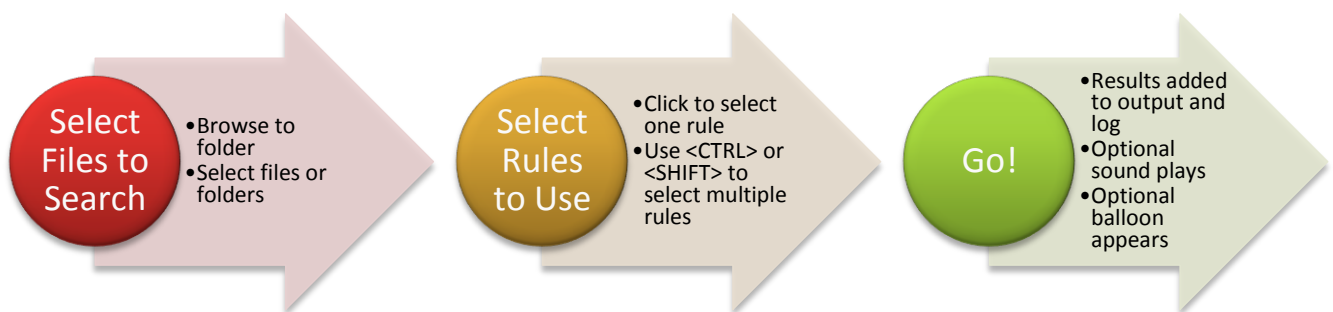
Process Flow

Select Files to Search
- Browse to folder
- Select files or folders

Select Rules to Use
- Click to select one rule
- Use <CTRL> or <SHIFT> to select multiple rules

Go!
- Results added to output and log
- Optional sound plays
- Optional balloon appears

Figure 1 - Cronalyser process flow

Creating Rules

Cronalyser rules possess the full power of *regular expressions*. If you're familiar with regular expressions then you're already hot to trot, otherwise here are some hints. For a full explanation of regular expressions I'd recommend the site: http://www.regular-expressions.info/

Locating simple text strings

If you want to find out if someone is talking to you, simply create a rule with your name in it, like so:

```
Nidhogg
```

Case is not significant. If people call you by different names (e.g. nick names) then you can specify them all, separated by the *alternation* symbol "|". E.g.

```
Nidhogg|Nid|Niddy|Idiot!
```

While it's possible to specify alternates like this it can become a little unwieldy, especially if you want to use this in more than one rule. To make things easier Cronalyser provides the concept of *Groups*.

7

Groups

The Group dialog looks like this:

Toolbar and menus

Create or modify groups and group members using the toolbar or the menus.

Defined groups list

Select a group to see or edit its members in the member list opposite.

Member list

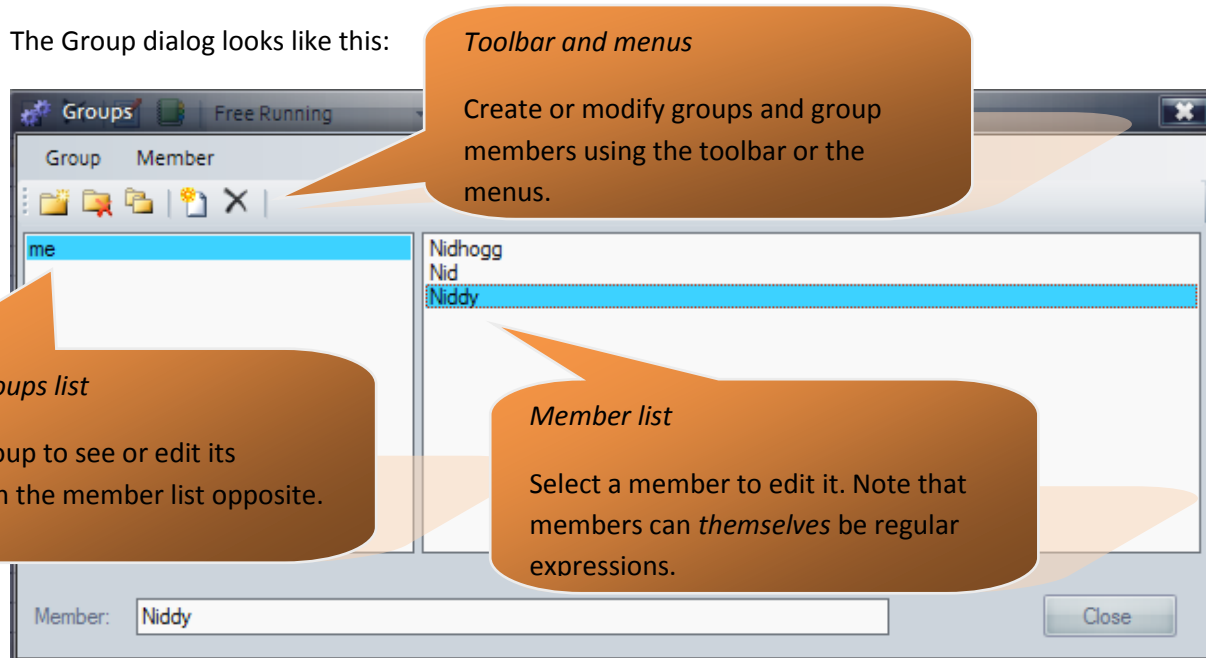Select a member to edit it. Note that members can *themselves* be regular expressions.

Figure 2 - Groups Dialog

Now, if you want to refer to yourself in a rule you can use the short form of the group name surrounded by "%" symbols, like so:

```
%me%
```

Looking for more complicated things

This time we're going to write a rule to look for any of our clan-mates cybering so that we may take suitable measures against them:

```
^.*%clanny% : .*%cyber%
```

Obviously this rule pre-supposes that you've set up two groups; *clanny*, containing the names and nick-names of our clan-mates, and *cyber*, containing any incriminating cyber-related language.

This is a slightly more complicated rule and in order to understand it we must know and understand exactly what it is that we are searching...

8

Here's an example from Neocron's *console.log* file:

```
07-19-2006 21:03:12 > TRADE - NC> KK_Nidhogg: You're banned!
```

There are four main components to this line:

1. The timestamp (`07-19-2006 21:03:12`), followed by a ">"
2. The channel (`TRADE - NC`), followed by a ">". If this is missing then the line was said locally.
3. The person speaking (`KK_Nidhogg`), followed by a ":". That's me, doncha know.
4. The thing being spoken (`You're banned!`). Actually, I'm unable to impose in-game bans (contrary to popular belief), but it's the kind of thing I have the reputation of saying.

Let's review our rule, part by part:

1. `^` - this is a special character called an *anchor*. It means our pattern must start at the beginning of the line. You can also use a "$" to anchor the search string to the end of line.
2. `.*` - this allows us match any number of characters. The dot means a character and the asterisk means zero or more occurrences of the thing before it. We're not interested in what those characters are. In this case we're skipping over the timestamp and the channel because they don't interest us; anything here will be ok for a search hit. Incidentally, "`.+`"means one or more characters.
3. `%clanny%` - this is a group containing the names of our clan-mates. The system will detect if it has found one of our clan-mates, and will effectively end the reign of the ".*" from component 2.
4. "` : `" (space, colon, space) – this always denotes that whatever follows is something that has been said by someone.
5. `.*` - this is another component to skip over unwanted characters in whatever was said until we get to something more interesting
6. `%cyber%` - this is a group containing "cyber" lingo.

All 6 components of the rule need to be satisfied for the rule to generate a hit. Our clannies talking about the price of fish won't trigger a hit, nor will some random Joe cybering with his or her friend.

If we'd wanted to restrict this to people spamming TRADE chat then we could modify it slightly as follows:

```
^.*TRADE - NC> %clanny% : .*%cyber%
```

Regular expressions take some getting used to and there are far more ins and outs than I've mentioned here, but they are *extremely* powerful.

So, how do we enter these rules?

Rules

The *Rules* dialog looks like this:

Rule Name

Enter a name for the rule, or rename an existing rule.

Rule Text

Enter a regular expression representing your rule. Specify groups using percent symbols.

Sound File

Use this to specify a sound file to play whenever this rule gets a hit.

Alert Balloon

Use this to specify whether you want a balloon to appear whenever this rule gets a hit.
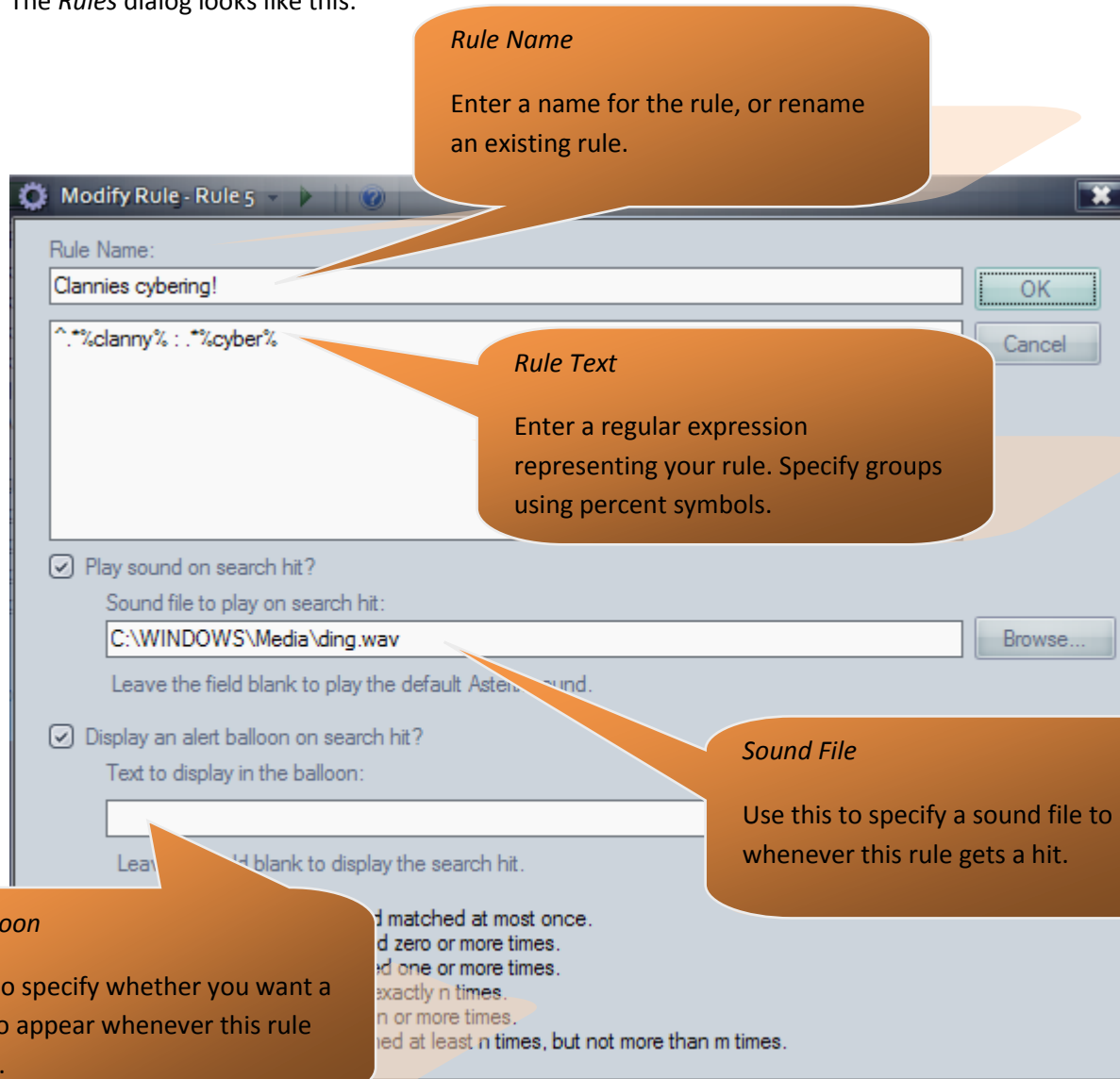
Figure 3 - Rules Dialog

10

Here's our clan cyber rule entered into Cronalyser. We've also specified that we want to hear a "ding" whenever it happens and we'd like to see the incriminating evidence pop up in a system tray alert balloon.

There are four main components to the *Rule* dialog:

1. Rule name – this is the name of the rule as you want it to appear in the rules list on the main dialog. You cannot have two rules with the same name, nor can the name be blank.
2. Rule text – this is the regular expression that makes up your rule. Reference your groups by wrapping them in percent symbols. E.g. `%MyRule%`. Note that group names are case sensitive. Rules themselves, however, are case *insensitive*. Rules are only applied to new text in the monitored files, so it doesn't "slow down" over time as the files get bigger and bigger.
3. Sound file – if you would like a sound to play whenever the rule gets a hit then tick this box and browse to a valid WAV sound file. If you leave the field blank then the *Asterisk* sound from the current theme will be used instead. In order for any sounds to be heard you must also check the *Play Sounds* option on the *Rule* menu.
4. Alert balloon – if you would like an alert balloon to appear from the Cronalyser icon in your system tray whenever this rule gets a hit just tick this box. If you leave the display text field blank then the line that matched the rule will be displayed in the balloon, otherwise the specified text will be used. In order for any alert balloons to appear you must also check the *Show Balloon Tips* option on the *Rule* menu.

Configuration Files

Cronalyser uses a number of XML configuration files. All of these files are located in the folder:

```
<system disk>

        documents and settings

                <user>

                        application data

                                Neocron Utilities

                                        Cronalyser

                                                <Cronalyser Version>
```

If the folder does not exist (and let's face it, what are the chances?) then Cronalyser will automatically create it, as well as two configuration files:

> CronalyserConfig.xml – stores all application settings between runs. If you selected "Free Running" last time you used Cronalyser then you'll get "Free Running" by default next time.
> CatSpyRules.xml – this file contains all the rules you've configured.

When you create a group Cronalyser will create a new XML file to represent it. Group files all start with the prefix "`CronalyserGroup_`". If you rename or delete a group the associated file will be affected accordingly.

Finally, a history of everything that appears in the Cronalyser output window is written to a log file called "`CronalyserLog.Log`". Think of this as a distillation of all your monitored streams containing only data that matches your rules.

Importing Groups and Rules

Cronalyser provides two mechanisms to allow people to share useful configuration:

> Simply dropping a new group XML file into the configuration folder will cause that group to be made available the next time you start Cronalyser.
> Files starting with the prefix "CronalyserRule_" are deemed to contain new rules. Cronalyser will import these rules into the main body of rules and then delete the import file afterwards.