

## Application Monitoring using SNMPc 7.0

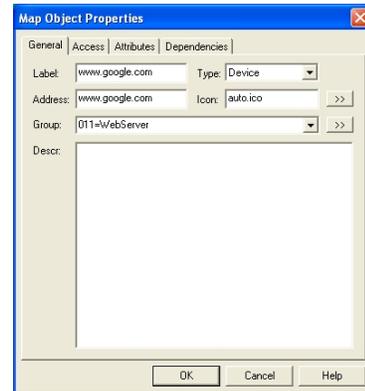
SNMPc can be used to monitor the status of an application by polling its TCP application 'port'. Up to 16 application ports can be defined per icon. You can also configure SNMPc to send a text string to the application port and then look for an 'expected reply' in the response. This allows you to provide advance monitoring that for example not only checks that a web servers HTTP port is open but also that the website homepage is loading correctly.

This first example will show how to poll a web server and ensure that the server is generating the home page correctly. The second part of this guide covers how to configure new applications to be monitored.

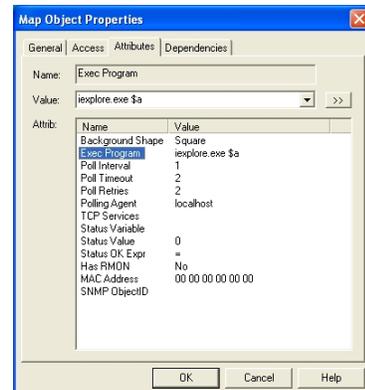
To edit an existing icon, right click on the icon and choose *Properties*. If this is a new icon then from the main window select the 'Add device' icon. 

When prompted define the device *Label* and *Icon* that you wish to use. Set the *Address* to the web URL (i.e. www.google.com).

If SNMP is not enabled on the web server (it probably will not be if this is a public server) select the *Access* tab and change the *Read Access Mode* to ICMP Ping

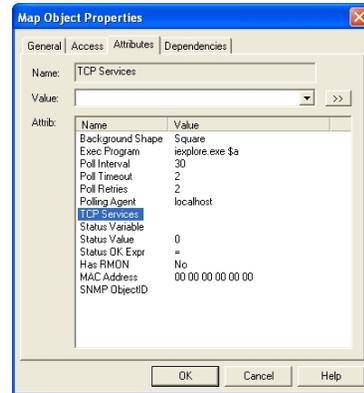


Select the *Attributes* Tab. As this is a website it would be useful to launch Internet Explorer and browse to the site when the icon is double-clicked. To configure this select the *EXEC Program* variable. From the pull down list choose *iexplore.exe \$a*.



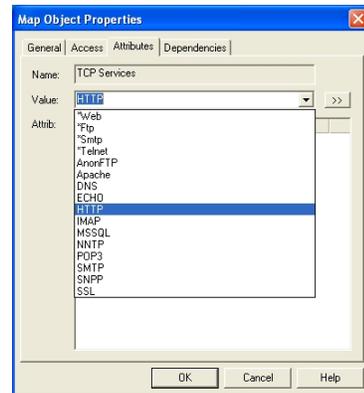
SNMPC 7.0 comes with a predefined list of services that can be polled. These cover common protocols including HTTP, FTP, DNS, POP3, SMTP etc. A full list of the pre-defined protocols and descriptions is included in *Appendix A* of this guide.

To choose one of these predefined services select *TCP Services* and click on the *Value* pull down. If you want to select multiple services to be monitored double-click on *TCP Services* and use the *Add* option on the resulting screen. Up to 16 services per icon can be configured.



In our scenario we want to ensure that the home page is being displayed correctly. Therefore we select the *HTTP* option.

Select OK to confirm the changes.



## Defining you own application ports.

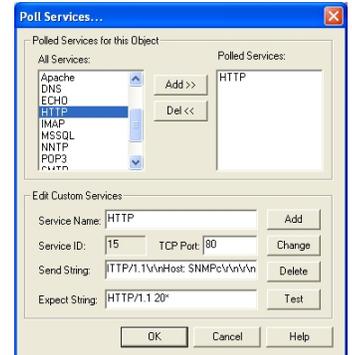
SNMPC 7.0 gives you the ability to monitor the health of any TCP based application by sending a string to the application port and monitoring the response.

To define a new application double-click on *TCP-Services*. This will open the Poll Services... window.

The previously discussed *HTTP* service checks to see the default web page is displayed correctly. SNMPC sends the following string to the web server on port 80.

```
HEAD / HTTP1.1\r\nHost: SNMPC\r\n\r\n
```

/r = Carriage Return  
/n = Line Feed



When a web page is generated correctly it displays as part of the html response

```
HTTP/1.1 200 OK
```

Therefore the *Expect String* is configured to monitor for the response *HTTP/1.1 20\**.

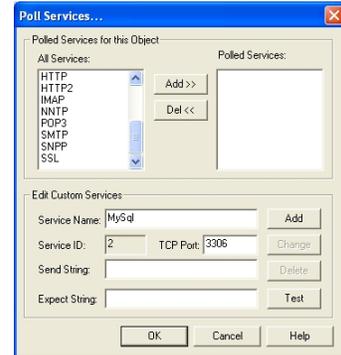
In this example we will create a custom service monitor for MySQL ([www.MySQL.com](http://www.MySQL.com)). MySQL runs on TCP port 3306. If you do not know the TCP port number for your application a quick search on Google for "Application Name' TCP Port" will normally display it.

To analyze what information the application displays when it is polled you can either Telnet to the port number or use a packet analyzer to monitor the application. Ethereal ([www.ethereal.com](http://www.ethereal.com)) is a useful free analyzer tool which runs on both Windows and Linux platforms.

To create the new service, enter a *Service Name* and the *TCP Port Number*.

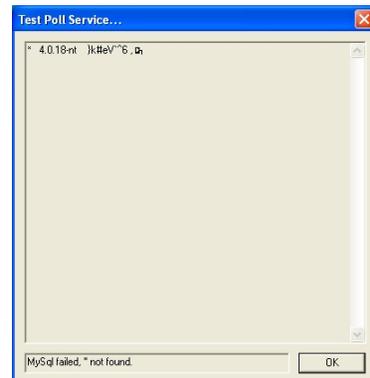
The *Send String* and *Expect String* can be defined as required. Special characters in the Send String include:

/w = Pause for 1 second before sending  
/r = Carriage Return  
/n = Line Feed



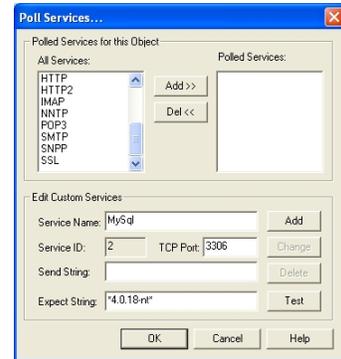
SNMPC 7.0 includes a Test option that can display what information the application is returning. This is an excellent way of evaluating your settings.

Select the *Test* button and a new Window will be presented showing the output from the application. In our scenario we can see that when MySQL is accepting connections it returns a series of characters including the MySQL version number and platform (4.0.18-nt). This is the string therefore to match on.

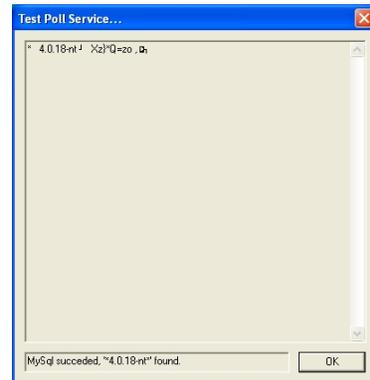


Select OK to close the Test Poll Service screen and enter the text \*4.0.18-nt\* in the *Expect String* field.

'\*' is a wildcard. By using it at the beginning and end of the text it means that the string will be matched as long as it appears somewhere in the text.



To ensure that the *Expect String* will be matched select the *Test* option again. You should see 'MySQL succeeded '\*4.0.18-nt\*' found.' displayed



You can then select *OK* and *Add* to add the service to the list of available services to be polled.

Up to 16 services can be defined per icon.

## Appendix A – Predefined Applications

Service 'Name'	TCP Port	Comment
*Web	80	Simple Poll to check service availability
*FTP	21	Simple Poll to check service availability
*SMTP	25	Simple Poll to check service availability
* Telnet	23	Simple Poll to check service availability
AnonFTP	21	Checks that an anonymous user can log into the FTP site
Apache	8080	Checks that an HTTP page can be loaded successfully
DNS	53	Simple Poll to check service availability
ECHO	7	Checks that a test string is successfully returned
HTTP	80	Checks that an HTTP page can be loaded successfully
IMAP	143	Checks that the service is ready to accept mail
Microsoft SQL Server (MSSQL)	1433	Simple Poll to check service availability
NNTP	119	Checks that the news service is active
POP3	110	Checks that the mail service is active
SMTP	25	Checks that the mail service is active