i-scream The future is bright; the future is blue.

Maintenance Documentation

i-scream Web Reports

The i-scream web reports provide an easy to access view of data from the i-scream monitoring system. This document provides an overview of how the web reports work and how you may add features to them.

Revision History

24/03/01	Initial creation				
		Committed by:	pjm2	Verified by:	tdb1
				Date:	24/03/01
		Committed by:		Verified by:	
				Date:	
		Committed by:		Verified by:	
				Date:	
		Committed by:		Verified by:	
				Date:	
		Committed by:		Verified by:	
				Date:	

Introduction

The i-scream web reports provide an easy to access view of data from the i-scream monitoring system. This document provides an overview of how the web reports work and how you may add features to them.

The web reports are used to display almost real-time information from the i-scream monitoring system as well as historical reports and current alerts for hosts.

Real-time viewing

Each time data is received from a host, the i-scream monitoring system writes data to files that may be used by external applications. The real-time web reports are an example of such an interface, where a user may visit a page to view the latest data about a particular host. Information is provided on a per-request basis and so is always recent.

Historical reports

The i-scream monitoring system may store data from hosts in a database. Historical reports are periodically generated from the large amounts of data in the database to produce charts of time periods for each machine. There is a huge amount of processing involved in generating historical reports, so these are not generated on a per-request basis.

Alerts

The i-scream server is capable of outputting alerts information to files. These may be used by external applications. The i-scream web alerts are one such example of these being used. The i-scream web alerts also provide a facility to construct a page suitable for displaying on a non-interactive public display screen, with automatic scrolling and updating via a web browser.

Scripting support

The i-scream web reports are written in PHP and are known to work on any web server that has support for PHP4 scripting, including Apache.

Configuration

The contents and appearance of each report may be easily configured by editing the appropriate _config.inc.php file. In most cases, this will be required to tell each section of the reports which paths to use to find the i-scream report data.

Customisation

Users are encouraged to install the i-scream web reports and to customise them to suit their needs. This typically involves editing the _config.inc.php files to customise the look and feel of each report. Other changes can include altering the contents of the bottom.inc file, which contains the copyright notice that appears at the bottom of each page. The title.inc.php file can be altered to affect the top of every report page, however, care must be taken not to harm the effect of the PHP scripting included in this file.

The above are all valid ways of customising the i-scream web reports and we would like to encourage people to do this if they want to. If you decide to go beyond this stage of customisation, we would appreciate it if you could keep a link to the i-scream homepage (http://www.i-scream.org.uk) somewhere on each report. If you feel that you have made beneficial changes to the operation of the system, then please do get in touch with us and we may include your changes in our next official build, available from the i-scream website.

Index, Headers and Footers

index.php

The main sections of the i-scream web reports are accessible from the index page. All other pages that are part of the web reports system must be located in the same directory as this index page. This page makes use of the title.inc.php and bottom.inc include files to ensure that the same look and feel is kept across all pages.

title.inc.php

The contents of this file are included on every report page, and apart from HTTP headers being sent to the browser, no HTML should be output before this file is included.

This include file declares the use of the HTML 4.01 Transitional DTD. It is recommended that all changes made to any page keep within this DTD in order to ensure browser compatibility.

The title of each page and meta tags are set using this include. The meta tags are set to instruct browsers and web caches not to cache each page, and to request that robots do not try to index any pages after this page.

The top of each page contains the i-scream logo on the right, and a timestamp on the left. The timestamp is generated when the page is issued to a browser, so it acts as a freshness indicator.

This include file is responsible for setting up the size of the main viewing area of the page by setting the width of a containing table. The default width of this table is 600 pixels, but the user may override this by specifying the *pagewidth* attribute in the HTTP request's query string, e.g. http://raptor.ukc.ac.uk/~pjm2/reports/latest.php?pagewidth=700
The *pagewidth* may be set to any value from 100-9999 pixels, or 100%. Any other values are ignored and the default of 600 pixels is used.

The base font size of a page may also be set by this include file. The *fontsize* attribute may be set in the query string to a value ranging from 1 to 7. This value is used in the basefont tag for each page. If this value is not specified, or is not specified as a number between 1 and 7, then the default size of 2 is used.

To enable some reports to be displayed in a non-interactive environment, the query string may also be used to set a value for the http-equiv meta refresh tag. This allows a value for *refresh* to be specified in the query string. This value may range from 0 to 999 seconds and causes the page to be refreshed after that time interval.

To further enhance the use of non-interactive displays, supported browsers may specify a value for *scroll* in the query string. This must be a value between 1 and 9 and determines the rate of scroll. With this attribute set, some JavaScript is included at the top of the page and triggered to run when the page has loaded (by calling *pScroll();*). This method uses *window.scrollBy* to scroll the page down by the number of pixels specified in *scroll* every 1/20th of a second. When the bottom of the page is reached, the page reloads. If the page is too small to scroll, then it shall wait ten seconds before refreshing. The JavaScript used on this page is known to work only with Opera and Microsoft Internet Explorer 4.0+ but it is ideal for publicly displaying data, as a large screen font may be used and if F11 is pressed, then the display goes to full screen mode.

bottom.inc

This include file in used to finish each page. It closes the containing table and includes static copyright information at the very bottom of the page, followed by the i-scream.org.uk banner image. This file does not contain any PHP scripting, as the filename reflects.

Latest data reports

latest.php

This page is linked from the index page. It is responsible for displaying the latest information received from a named host. The configuration for this page is obtained from *latest_config.inc.php*. A directory listing function is included from *iutils.inc.php*.

The contents of each report are defined in the *latest_display.inc.php* file.

If the page is requested with a value for *machine* set in the query string, then the report for that machine is displayed. Otherwise, the user may select from a drop-down list of all available machines. When a machine is selected using the list, that machine stays selected at the top of the list, with all other machines beneath it in alphanumerical order.

The page provides a number of functions for displaying data in various ways: -

function showStat($title, $value, $input)

This function takes the contents of the XMLPacket toString dump and looks for the specified *value*. This value is then printed out with the passes *title*.

function linkToHistory($report), global machine

This function prints a small chart image which acts as a link to yesterday's 24-hour historical report for the machine. No check is made to ensure that the 24-hour report exists. Such errors are dealt with when arriving at the historical reports page.

function showDisks($diskRoot, $units, $input)

This function shows graphical bars to illustrate the status of disk partitions. The *diskRoot* specifies the opening text of the disk fields in the XMLPacket dump, e.g. "packet.disks". A single call to this function causes all disk information in the report to be graphed with the specified *units* – typically "Kb".

function getPregMatch($value, $input)

Attempts to retrieve a named value from the *input* XML dump. The value is then returned. The value is undefined if it could not be found.

function showBar($title, $value, $max, $units, $input, $arrange='normal')

Displays a chart of a value, ranging from 0-100%, calculated from the *value* and the *max*imum value, with the specified *units*. This function is used on its own and also by the *showDisks* function. *arrange* defaults to 'normal' but if the method is called with it set to 'swap', then the chart is reversed such that a value of 30% becomes 70% and so on. This is useful, for example, in displaying the amount of used disk space when you have a value for the amount of free disk space.

latest_config.inc.php

This include file is used by *latest.php* and contains essential settings.

latest_display.inc.php

This include file is used by *latest.php* and determines which fields shall appear on each report. This file will usually contain only calls to the functions *printLine*, *showStat*, *showBar* and *showDisks*.

Historical data reports

browser.php

This page is linked from the index page. It is responsible for displaying 24-hour historical reports for a named host and report type. The configuration for this page is obtained from *browser_config.inc.php*. A directory listing function is included from *iutils.inc.php*.

A list of machines and report names only becomes visible after the user has selected a date from the *day* drop-down box. If the page is requested with a value for *day*, *machine_name* and *report* set in the query string, then the historical report for that machine is displayed.

When a historical report is viewed, a gif chart image (500x250 pixels) is shown, representing the activity of the machine for a certain report attribute over 24 hours. The charts and other data must be in a web-visible directory: $reportDirectory/$day/$machine_name/$report

Errors such as missing images are checked for and reported to the user.

browser_config.inc.php

Configuration file for the historical reports browser. This sets the location of the report directory both absolute and relative to the web root. The number of recent days shown in the *day* drop-down box may also be changed here. All config files allow for the form submission method to be altered, however, we recommend that these are all left at a value of "GET" as this allows full URLs to reports to be emailed to people more easily.

Alert reports

alerts.php

This page is linked from the index page. It is responsible for displaying alert information about either an individual machine, or all machines being monitored. The configuration for this page is obtained from *alerts_config.inc.php*. A directory listing function is included from *iutils.inc.php*.

If the page is requested with a value for *machine* set in the query string, then the report for that machine is displayed. Otherwise, the user may select from a drop-down list of all available machines. When a machine is selected using the list, that machine stays selected at the top of the list, with all other machines beneath it in alphanumerical order.

The exception to the above is if *machine* is set to a value of *ALL*, in which case, alerts for all machines are displayed.

By default, each machine's alerts appear in a table with a dark blue border. A link is provided to the latest data received by the host.

For producing aesthetically pleasing non-interactive displays for public viewing, it is possible to remove the drop-down box from this page by including *hideform=on* in the query string for the page.

For non-interactive displays that may need to be read from a distance, the amount of data shown for each alert can be reduced by setting the *small* attribute to *on*. This causes the machine name to be shown, with an alert description on each table row thereafter.

alerts_config.inc.php

This include file specifies the absolute location of the directory containing the alert data. The colours and names of each alerts level are also defined here.

Configuring Non-interactive Pages

public.php

This page is provided to assist a user in creating a display of current alerts that is suitable for non-interactive public display. It consists of a simple form, which the user may use to select various display options. The form submits to *alerts.php*.

public_config.inc.php

This simple config include is used by *public.php* and specifies the header and footer includes and form method.

Miscellaneous Files

24hour.gif

A gif image that is 500 pixels wide and used as a scale in the historical reports.

alert_logo.gif

A small icon for the alert page link on the index page.

allalerts_logo.gif

A small icon for listing alerts for all machines. Appears on the index page.

bar.gif

A horizontal blue bar that is used in charts made with the showBar function in *latest.php*.

historical_logo.gif

A small chart icon for linking to the historical reports browser from the index page.

i-scream.gif

The original i-scream blue logo.

latest_logo.gif

A small icon for linking to the latest data page from the index page.

level[0-5].gif

6 small gif images that act as an alert level scale (from 0 to 5).

minichart.gif

A very small chart image for linking to historical reports from the latest data page.

scale.gif

A 300-pixel scale for use with the *showBar* function in *latest.php*.

title.gif

The i-scream reports image that appears at the top of every page.