



Monitor · Manage · Control

ContentKeeper – Blockpage Customisation

Overview

This document details the configuration of ContentKeeper *Blockpage Customisation*. It includes explanations of how *Blockpage Customisation* works, how to enable and disable *Blockpage Customisation* and how to generate a customised blockpage.

Document Revision B

Date: 4th July 2003

Copyright © 2000, 2001,2002, 2003 ContentKeeper Technologies

ContentKeeper® Closed Loop Collaborative Filtering™ and *TrickleFeed™* are trademarks of ContentKeeper Technologies. Copyright © 2000 - 2003, ContentKeeper Technologies, Canberra, Australia. All Rights Reserved.

Linux is a registered trademark of Linus Torvalds, Red Hat Linux is a registered trademark of Red Hat Inc.

All other product names mentioned herein are trademarks or registered trademarks of their respective owners.

The products and specifications, configurations, and other technical information regarding the products contained in this manual are subject to change without notice. All statements, technical information, and recommendations contained in this manual are believed to be accurate and reliable but are presented without warranty of any kind, express or implied, and users must take full responsibility for their application of any products specified in this manual.

Document Author & Designer: Matthew R Richards

ContentKeeper Technologies
218 Northbourne Avenue
Braddon ACT 2612
Australia
PH +61-2-62614950
Fax +61-2-62579801
info@ContentKeeper.com
www.ContentKeeper.com

Blockpage Customisation

Table of Contents

| | |
|---|----------|
| BLOCKPAGE CUSTOMISATION | 3 |
| INTRODUCTION | 3 |
| CREATING A CUSTOMISED BLOCKPAGE | 4 |
| VIEWING EXAMPLE WEB PAGES | 4 |
| UNDERSTANDING BLOCKPAGE VARIABLES | 5 |
| CONFIGURING BLOCKPAGE CUSTOMISATION | 6 |
| TESTING YOUR CUSTOMISED BLOCKPAGES | 7 |

Introduction

ContentKeeper blockpage customisation is enabled and configured via the *Blockpage Customisation* menu in the ContentKeeper Web Interface.

Main Menu – General Settings → Blockpage Customisation

ContentKeeper provides a facility for customising its blockpages. Of the three available ContentKeeper blockpages the following two may be customised:

- *Block*
- *Coach*

To allow for the use of existing authentication mechanisms, and the integration of new authentication mechanisms, ContentKeeper must host and serve the remaining blockpage, *Authenticate*.

Customised Blockpages are served up by a web server available to your network, e.g. your local Intranet server. The ContentKeeper *Management Port* must be able to access the web server that will serve the customised blockpages.

Note: Ensure that any routing rules or network proxies necessary to allow the ContentKeeper *Management Port* to access the web server are correctly configured. Routing rules and proxies are not commonly required as the web server is usually attached to the same IP subnet as the *Management Port*.

The following is a step-by-step explanation of the process of generating a customised blockpage.

1. A user enters a request for a web page into their browser.
2. ContentKeeper intercepts the request and determines if a blockpage should be generated (in this example a blockpage is to be generated).
3. ContentKeeper determines that a customised blockpage is to be generated and creates a package in the form of an HTTP redirect that contains information about the block page to be generated and issues it to the users browser.
4. The users browser is redirected to the web server specified in the ContentKeeper *Blockpage Customisation* menu.
5. The web server uses the information in the request to generate the blockpage and then serves the blockpage to the users browser in response to its original request.
6. The customised blockpage is displayed in the users browser.

Creating a Customised Blockpage

A customised blockpage is generated by a web server when it receives a request from a users browser containing information about the URL that has been blocked. The web server must be configured to listen for a request and then generate a customised blockpage in reply. This is achieved by a special web page hosted on the server.

This is how it works. The information within the request is in the form of eleven variables. A table below shows the variables and is accompanied by an explanation of each. The web page interprets and processes the information to generate the customised blockpage, which is then sent to the computer that originally requested the URL.

There are three example web pages available in the *Blockpage Customisation* menu.

- JScript
- CGI
- Coldfusion

Use the code within these examples to help you design your own customised blockpages.

Viewing Example Web Pages

The three example web pages available in the *Blockpage Customisation* menu may be viewed by clicking on the *View Example* button located at the top of the page for the *Jscript* and *CGI* examples or by clicking on the *Coldfusion* hyperlink at the bottom of the page for the Coldfusion example.

The *Jscript* and *CGI* examples may be viewed as blockpages from a computer that is being filtered by ContentKeeper by enabling customised blockpages then clicking on the *A Jscript Example* or *A CGI Example* buttons respectively followed by the *Save* button; then entering a blocked URL into the input box next to the *Test* button and clicking the *Test* button. Refer to image on next page.

Understanding Blockpage Variables

The image below is of the example JScript customised blockpage. Within it is a table containing all of the variables passed by ContentKeeper to the web server and their values for the block page that was generated. An explanation of each variable follows.

ContentKeeper Customisable Blocking Page
(JavaScript Example)

Use the following variables (passed as form variables) from ContentKeeper to customise you page

| Description | Form Variable | Value |
|--|---------------|--|
| Passed String | passed | PGE=http%3A%2F%2F192.9.200.67%2Fjava-script-eg.html&URL=www.playboy.com&CAT=Adult+Content%2CShopping%2CEntertainment+%2CList+Blocked%29&ACC=matthew%2Fope&WHY=Policy%3Dmatts-policy%3B+NT%2FIP+Address%3D192.9.200.32&MOD=2&IPA=192.9.200.67&STR=1%2C0%2C1%2C874732987%2C137672775%2Cwww.playboy.com%2Cwww.playboy.com%2F&SRV=http%3A%2F%2F192.9.200.67%2Fcgi-bin%2Fck%2Fu.cgi&RAW=http%3A%2F%2Fwww.playboy.com%2F |
| Redirected URL (This Page) | PGE | http://192.9.200.67/java-script-eg.html |
| Blocked or Coached URL | URL | www.playboy.com |
| Category(ies) Blocked | CAT | Adult Content,Shopping,Entertainment (List Blocked) |
| Blocking Mode (2=BLOCK, 3=COACH) | MOD | 2 |
| Reason Blocked | WHY | Policy=matts-policy; NT/IP Address=192.9.200.32 |
| ContentKeeper Server | IPA | 192.9.200.67 |
| Reclassification String | STR | 1,0,1,874732987,137672775,www.playboy.com,www.playboy.com/ |
| Reclassification Server (Ignore if null) | SRV | http://192.9.200.67/cgi-bin/ck/u.cgi |
| Account/UserName | ACC | matthew/ope |
| Raw URL | RAW | http://www.playboy.com/ |

Example of URL Reclassification using above variables
Note : You must check for SRV=null, when Reclassification is off!

Ensure **192.9.200.67** address bypasses proxy server!

For Coaching (Case MOD=3 Only), you might want to continue to requested URL as follows:
www.playboy.com

- **passed** – Passed String: This variable contains the string of data ultimately used to generate the customised blockpage that was passed to the web server from ContentKeeper as a result of the HTTP redirect.
- **PGE** – Redirected URL: This variable contains the URL of the customised blockpage.
- **URL** – Blocked or Coached URL: This variable contains the URL that caused the blockpage, with its HTTP formatting removed.
- **CAT** – Categorie(s) Blocked: This variable contains the categorie(s) under which the URL had been categorised.
- **MOD** – Blocking Mode: Mode 2 represents a *Block* page being generated; Mode 3 represents a *Coach* page being generated.
- **WHY** – Reason Blocked: The ContentKeeper policy that was associated with the blockpage and the policy component that triggered the blockpage.
- **IPA** – ContentKeeper Server: The IP Address of the ContentKeeper *Management Port*.
- **STR** – Reclassification String: String passed to reclassification server when Submit button is clicked.
- **SRV** – Reclassification Server: The URL to which *URL reclassification* requests will be sent.
- **ACC** – Account/UserName: The resolved username and if available, associated domain account information, belonging to the user logged in to the browser or computer from where the URL request came.
- **RAW** – Raw URL: The URL that was blocked without the HTTP formatting or arguments removed.

Note: Some of the blockpage variables are only passed when certain conditions exist. If *Proxy Header Username Resolution* and *NTLM Authentication Scheme* are not enabled then the **ACC** variable will not be included, also if the *URL Reclassification* facility is not enabled then the **SRV** variable will be set to null.

Configuring Blockpage Customisation

ContentKeeper blockpage customisation is enabled and configured via the *Blockpage Customisation* menu in the ContentKeeper Web Interface.

Main Menu – General Settings → Blockpage Customisation

To enable blockpage customisation, browse to the *Blockpage Customisation* menu and click this check box at the top of the page to switch between the default and customised blocking pages.

Customised blockpages are enabled when the box is unchecked and the options below are not greyed out.

Customised blockpages are disabled when the box is checked and the options below are greyed out. Refer to the image below.

Blockpage Customisation

Use ContentKeeper Default Blockpages by checking here
otherwise uncheck
and enter URL of your custom page below.

| | | | |
|---|--|-------------------|--------------|
| <input type="radio"/> Use GET | <input type="text" value="http://192.9.200.73/java-script-eg.html"/> | A JScript Example | View Example |
| <input checked="" type="radio"/> Use POST | <input type="text" value="http://192.9.200.73/cgi-bin/cgi-script-eg.cgi"/> | A CGI Example | View Example |

Note: You will need to decide whether to process Form Variables in your page as GET or POST.
Click on Buttons for examples.

To test, first click **Save**, then enter a blocked URL and click **Test**

| | |
|----------------------|-------------------------------------|
| <input type="text"/> | <input type="button" value="Test"/> |
|----------------------|-------------------------------------|

Other Examples :[Coldfusion](#)

Once customised blockpages have been enabled, you must specify whether ContentKeeper will use the HTTP GET or POST method when sending requests to the web server that will generate the customised blockpages (this corresponds to step 3 above). Make your selection by clicking on the radio button to the left of the desired option.

After selecting the method, you must specify the location of the web server and web page that interprets and generates the final customised blockpage. Enter the web server address into the input field to the right of the method that you chose in the previous step. The address must be in the form of `http://<IP Address>/<File Path>`. An example of this will already be in the input field. Overwrite the example with the address of your customised blockpage. Finally click *Save* to apply your changes to ContentKeeper.

Testing your Customised Blockpages

To test a customised blockpage, enable and configure ContentKeeper *Blockpage Customisation* as described above, then from a computer that is being filtered by ContentKeeper enter a blocked URL into the input box next to the *Test* button at the bottom of the screen and clicking the *Test* button.

This will cause your customised blockpage to be displayed in your browser. You may then check that your customised blockpage functions and that variable are being interpreted properly.

If you have chosen to include the feature, you may also test the URL reclassification facility by clicking on the submit button and following the prompts. Login if necessary and reclassify a URL. Then verify the URL reclassification from the *View/Edit Reclassified URLs* menu.