*1#2$3+4K5Restriction List Editor

Go to Pattern Syntax Help

Introduction

The restriction list editor provides a way to "hide" accesses from the statistics reports generated by WebStats. Generally, there are groups of accesses that you won't want to show in the statistics, and you can hide them with this facility.

Objects That Can Be Hidden

There are three types of objects that are subject to restriction:

- Documents (objects)
- Sites (client hosts)
- Authorized users (accessing restricted documents)

To restrict a particular type of object, add a match pattern to the list for that object.

Degrees of Hiding

Each match pattern in the list for a given object type can cause two types of hiding:

- Hide the object from "top 10" lists only. It still counts in totals.
- Hide the object completely. Never count it in any statistics.

The option buttons below the pattern edit box control that selection.

Examples

1 Hide all GIF files by adding "*.GIF" to the document list.

2 Hide everything in "/icons/" by adding "/icons/*" to the document list. Note that the paths are in logical (Web) space not physical (file path) space.

3 Hide all sites within the "beavis.com" by adding "*.beavis.com" to the sites list.

4 Hide user "denny" by adding "denny" to the users list

1BuildAll
2INDEX
3Index
4index:00001
5;

*6#7$8+9K10Using Match Patterns

The patterns you specify for restrictions are used in SQL statements with the Like operator to apply the restrictions. The rules for using patterns in Access databases are described below. The information was taken directly from the Access and Visual Basic SQL documentation.

General Considerations

The case sensitivity and character sort order of the Like operator depend on the setting of the Option Compare statement. Unless otherwise specified, the default string-comparison method for each module is Option Compare Binary; that is, string comparisons are case-sensitive. Built-in pattern matching provides a versatile tool for string comparisons.

The pattern-matching features allow you to use wildcard characters, such as those recognized by the operating system, to match strings. The wildcard characters and what they match are shown in the following table:

Character(s) in pattern:	Matches:
?	Any single character
*	Zero or more characters
#	Any single digit (0-9)
[charlist]	Any single character in charlist
[!charlist]	Any single character not in charlist

Character Lists

A group of one or more characters (charlist) enclosed in brackets ([ ]) can be used to match any single character in expression and can include almost any characters in the ANSI character set, including digits. In fact, the special characters left bracket ([ ), question mark (?), number sign (#), and asterisk (*) can be used to match themselves directly only by enclosing them in brackets. The right bracket ( ]) cannot be used within a group to match itself, but it can be used outside a group as an individual character.

In addition to a simple list of characters enclosed in brackets, charlist can specify a range of characters by using a hyphen (-) to separate the upper and lower bounds of the range. For example, [A-Z] in pattern results in a match if the corresponding character position in expression contains any of the uppercase letters in the range A through Z. Multiple ranges are included within the brackets without any delimiting. For example, [a-zA-Z0-9] matches any alphanumeric character.

Other Important Rules

An exclamation point (!) at the beginning of charlist means that a match is made if any character except the ones in charlist are found in expression. When used outside brackets, the exclamation point matches itself.

The hyphen (-) can appear either at the beginning (after an exclamation mark if one is used) or at the end of charlist to match itself. In any other location, the hyphen is used to identify a range of ANSI characters. When a range of characters is specified, they must appear in ascending sort order (from lowest to highest). [A-Z] is a valid pattern, but [Z-A] is not.

The character sequence [ ] is ignored; it is considered to be a zero-length string.

Special Characters

6BuildAll
7PATTERN_HELP
8Pattern Help
9index:00002
10;

In some languages, there are special characters in the alphabet that actually represent two separate characters. For example, several languages use the character "æ" to represent the characters "a" and "e" when they appear together. The Like operator recognizes that the single special character and the two individual characters are equivalent.

When a language that uses one of these special characters is specified in the WIN.INI file (sLanguage), an occurrence of the special single character in either pattern or expression matches the equivalent 2-character sequence in the other string. Similarly, a special single character in pattern enclosed in brackets (by itself, in a list, or in a range), matches the equivalent 2-character sequence in expression.

*[11]#[12]$[13]K[14]Character(s) Matches:
in pattern:

?	Any single character
*	Zero or more characters
#	Any single digit (0-9)
[charlist]	Any single character in charlist
[!charlist]	Any single character not in charlist

More Help

11BuildAll
12PATTERN_POPUP
13Pattern Popup
14;