NAME

nistbladm – NIS+ table administration command

SYNOPSIS

nistbladm **–a** | **–A** [**–D** *defaults*] *colname=value* . . . *tablename*

nistbladm **–a** | **–A** [**–D** *defaults*] *indexedname*

nistbladm **–c** [**–D** *defaults*] [**–p** *path*] [**–s** *sep*] *type colname*=[*flags*][,*access*] . . . *tablename*

nistbladm **–d** *tablename*

nistbladm **–e** | **–E** *colname=value* . . . *indexedname*

nistbladm **–m** *colname=value* . . . *indexedname*

nistbladm **–r** | **–R** [*colname=value* . . .] *tablename*

nistbladm **–r** | **–R** *indexedname*

nistbladm **–u** [**–p** *path*] [**–s** *sep*] [**–t** *type*] [*colname*=*access* . . .] *tablename*

AVAILABILITY

SUNWnisu

DESCRIPTION

The **nistbladm** command is used to administer NIS+ tables. There are five primary operations that it performs: creating and deleting tables, adding entries to, modifying entries within, and removing entries from tables.

Though NIS+ does not place restrictions on the size of tables or entries, the size of data has an impact on the performance and the disk space requirements of the NIS+ server. NIS+ is not designed to store huge pieces of data, such as files; instead pointer to files should be stored in NIS+.

NIS+ design is optimized to support 10,000 objects with a total size of 10M bytes. If the requirements exceed the above, it is suggested that the domain hierarchy be created, or the data stored in the tables be pointers to the actual data, instead of the data itself.

When creating tables, a table type, *type*, and a list of column definitions must be provided.

type is a string that is stored in the table and later used by the service to verify that entries being added to it are of the correct type.

Syntax for column definitions is:

colname=[*flags*][,*access*]

flags is a combination of:

- **S** Searchable. Specifies that searches can be done on the column's values (see **nismatch**(1)).
- **I** Case-insensitive (only makes sense in combination with **S**). Specifies that searches should ignore case.
- **C** Crypt. Specifies that the column's values should be encrypted.
- **B** Binary data (does not make sense in combination with **S**). If not set, the column's values are expected to be null terminated ASCII strings.

X XDR encoded data (only makes sense in combination with **B**).

access is specified in the format as defined by the **nischmod**(1) command.

When manipulating entries, this command takes two forms of entry name. The first uses a series of space separated *colname=value* pairs that specify column values in the entry. The second is a NIS+ indexed name, *indexedname*, of the form:

[*colname***=***value***, . . .],***tablename*

OPTIONS

−a | A Add entries to a NIS+ table. The difference between the lowercase ‘a’ and the uppercase ‘A’ is in the treatment of preexisting entries. The entry’s contents are specified by the *column=value* pairs on the command line. Note: Values for *all* columns must be specified when adding entries to a table.

Normally, NIS+ reports an error if an attempt is made to add an entry to a table that would overwrite an entry that already exists. This prevents multiple parties from adding duplicate entries and having one of them get overwritten. If you wish to force the add, the uppercase ‘A’ specifies that the entry is to be added, even if it already exists. This is analogous to a modify operation on the entry.

−c Create a table named *tablename* in the namespace. The table that is created must have at least one column and at least one column must be searchable.

−d *tablename*
Destroy the table named *tablename*. The table that is being destroyed must be empty. The table’s contents can be deleted with the **−R** option below.

−e | E Edit the entry in the table that is specified by *indexdname*. *indexdname* must uniquely identify a single entry. It is possible to edit the value in a column that would change the indexed name of an entry.

The change (*colname=value*) may affect other entries in the table if the change results in an entry whose indexed name is different from *indexedname* and which matches that of another existing entry. In this case, the **−e** option will fail and an error will be reported. The **−E** option will force the replacement of the existing entry by the new entry (effectively removing two old entries and adding a new one).

−m A synonym for **−E**. This option has been superseded by the **−E** option.

−r | R Remove entries from a table. The entry is specified by either a series of *column=value* pairs on the command line, or an indexed name that is specified as *entryname*. The difference between the interpretation of the lowercase ‘r’ versus the uppercase ‘R’ is in the treatment of non-unique entry specifications. Normally the NIS+ server will disallow an attempt to remove an entry when the search criterion specified for that entry resolves to more than one entry in the table. However, it is sometimes desirable to remove more than one entry, as when you are attempting to remove all of the entries from a table. In this case, using the uppercase ‘R’ will force the NIS+ server to remove all entries matching the passed search criterion. If that criterion is null and no column values specified, then all entries in the table will be removed.

−u Update attributes of a table. This allows the concatenation path (−**p**), separation character (specified with the (−**s**)), column access rights, and table type string (−**t**) of a table to be changed. Neither the number of columns, nor the columns that are searchable may be changed.

−D *defaults*
When creating objects, this option specifies a different set of defaults to be used during this operation. The *defaults* string is a series of tokens separated by colons. These tokens represent the default values to be used for the generic object properties. All of the legal tokens are described below.

ttl=*time* This token sets the default time to live for objects that are created by this command. The value *time* is specified in the format as defined by the **nischttl**(1) command. The default value is 12 hours.

owner=*ownername*
This token specifies that the NIS+ principal *ownername* should own the created object. Normally this value is the same as the principal who is executing the command.

group=*groupname*
This token specifies that the group *groupname* should be the group owner for the object that is created. The default value is NULL.

access=*rights*
This token specifies the set of access rights that are to be granted for the given object. The value *rights* is specified in the format as defined by the **nischmod**(1) command. The default value is −−−−**rmcdr**−−−**r**−−−.

−p *path* When creating or updating a table, this option specifies the table's search path. When a **nis_list( )** function is invoked, the user can specify the flag FOLLOW_PATH to tell the client library to continue searching tables in the table's path if the search criteria used does not yield any entries. The path consists of an ordered list of table names, separated by colons. The names in the path must be fully qualified.

−s *sep* When creating or updating a table, this option specifies the table's separator character. The separator character is used by **niscat**(1) when displaying tables on the standard output. Its purpose is to separate column data when the table is in ASCII form. The default value is a space.

−t *type* When updating a table, this option specifies the table's type string.

RETURN VALUES

This command returns **0** on success and **1** on failure.

EXAMPLES

This example creates a table named **hobbies** in the directory **foo.com.** of the type **hobby_tbl** with two searchable columns, **name** and **hobby**.

example% nistbladm −c hobby_tbl name=S,a+r,o+m hobby=S,a+r hobbies.foo.com.

The column **name** has read access for all (that is, **owner**, **group**, and **world**) and modify access for only the owner. The column **hobby** is readable by all, but not modifiable by anyone.

In this example, if the access rights had not been specified, the tables access rights would have come from either the standard defaults or the **NIS_DEFAULTS** variable (see below).

To add entries to this table:

example% nistbladm −a name=bob hobby=skiing hobbies.foo.com.
example% nistbladm −a name=sue hobby=skiing hobbies.foo.com.
example% nistbladm −a name=ted hobby=swimming hobbies.foo.com.

To add the concatenation path:

example% nistbladm −u −p hobbies.bar.com.:hobbies.baz.com. hobbies

To delete the skiers from our list:

example% nistbladm −R hobby=skiing hobbies.foo.com.

Note: The use of the −**r** option would fail because there are two entries with the value of **skiing**.

To create a table with a column that is named with no flags set, you supply only the name and the equals (=) sign as follows.

example% nistbladm −c notes_tbl name=S,a+r,o+m note= notes.foo.com.

This example created a table, named *notes.foo.com.*, of type *notes_tbl* with two columns **name** and **note**. The **note** column is not searchable.

When entering data for columns in the form of a *value* string, it is essential that terminal characters be protected by single or double quotes. These are the characters equals (=), comma (,), left bracket ([), right bracket (]), and space ( ). These characters are parsed by NIS+ within an indexed name. These characters are protected by enclosing the entire value in double quote (") characters as follows.

example% nistbladm −a fullname="Joe User" nickname=Joe nicknames

If there is any doubt about how the string will be parsed, it is better to enclose it in quotes.

ENVIRONMENT

NIS_DEFAULTS This variable contains a defaults string that will be override the NIS+ standard defaults. If the −**D** switch is used those values will then override both the **NIS_DEFAULTS** variable and the standard defaults.

NIS_PATH If this variable is set, and the NIS+ table name is not fully qualified, each directory specified will be searched until the table is found (see **nisdefaults**(1)).

SEE ALSO

nis+(1), **niscat**(1), **nischmod**(1), **nischown**(1), **nisdefaults**(1), **nismatch**(1), **nissetup**(1M)

WARNINGS

To modify one of the entries, say, for example, from “bob” to “robert”:

example% nistbladm -m name=robert [name=bob],hobbies

Note that "**[name=bob],hobbies**" is an indexed name, and that the characters '[' (open bracket) and ']' (close bracket) are interpreted by the shell. When typing entry names in the form of NIS+ indexed names, the name must be protected by using single quotes.

It is possible to specify a set of defaults such that you cannot read or modify the table object later.