NSPROTECTION

By: Bill van Melle (vanMelle@Xerox.com)

INTRODUCTION

The module NSPROTECTION provides a tool that enables you to easily change the protection of files and directories on Xerox NS file servers.

To install the module, load the file NSPROTECTION.LCOM. Also, Your NS file server must be running Services release 10.0 or later.

THE PROTECTION MECHANISM

An NS File Server maintains a protection for each file and (sub)directory on the server. In most cases, the protection is not specified explicitly, but rather is inherited from a file's parent directory, making it easy to maintain consistent protection over an entire branch of the file system hierarchy.

The protection is specified as a set of pairs <access rights, name>. The name can be the name of an individual user or a group. The name can also be a pattern of the restricted form *:domain:organization, *:*:organization, or *:*:*. The access rights granted to any particular user are the most general of those in the pairs that match the user's name (by exact match, pattern or membership).

The following five kinds of access rights are independently specified (the term "file" here can also denote a directory in the places where that makes sense):

Read — The user may read the file's content and attributes. In the case of a directory, the user may enumerate files in it.

Write — The user may change the file's content and attributes, and may delete the file. In the case of a directory, the user may change the protection of any of the directory's immediate children.

Add — (Applies only to directories) The user may create files in the directory (i.e., add children).

Delete — (Applies only to directories) The user may delete files from the directory (i.e., remove children).

Owner — The user may change the file's access list.

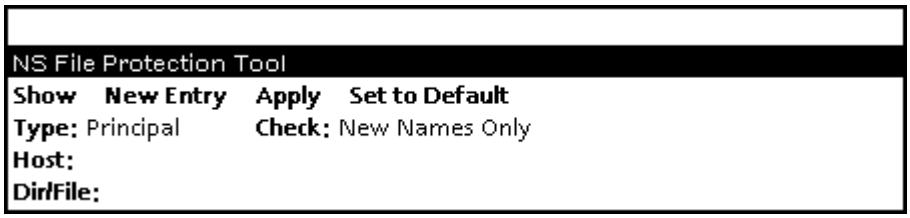
In the case of directories, it is also possible to independently specify the directory's own protection and the protection that its children inherit by default. In most cases, the latter simply defaults to the former, and it is usually best to keep it that way for simplicity. However, there might conceivably be cases where, for example, you would want a user to be able to read the files in a directory, but not be able to enumerate it, or vice-versa.

Note that there can be problems when giving a more lenient protection to a file or directory than to its parents, depending on what software is going to be used to gain access to the file. For example, if your default directory protection grants access only to you, and you want to allow a user to read a

particular file stored in your directory, then you can change the protection on just that file to allow Read access. However, the user will have to know the exact name of the file in order to read it, since she won't be able to enumerate the directory to search for the file. Specifying the exact file name works fine from Lisp, but other software that gets to a file by starting at the top and working its way down through the hierarchy would be unable to get to the file.

USER INTERFACE

To use the tool, select "NS Protection" from the background menu (if your menu has a "System" item, it's a subitem underneath it), or call the function (NSPROTECTION). You are prompted for a place to position the tool's window. Be sure to leave space below the window for the protection information that will follow.



The tool window has four command buttons across the top, two switches labeled **Type** and **Check**, and two fill-in fields for the host and file name. Holding a mouse button down over any of these items for a couple of seconds will display a help message in the prompt window.

To view or change the protection of a file or directory, first fill in the **Host** and **Dir**/**File** fields. You can edit these fields by clicking with the mouse anywhere inside the existing text (if any), or by clicking with the LEFT button on the boldface label. If you click with RIGHT on the label, then any existing text is first erased. Typing the Next or Return key moves to the next field. [See the FreeMenu documentation for more information about text editing.]

You can either enter the host and directory separately, e.g.,

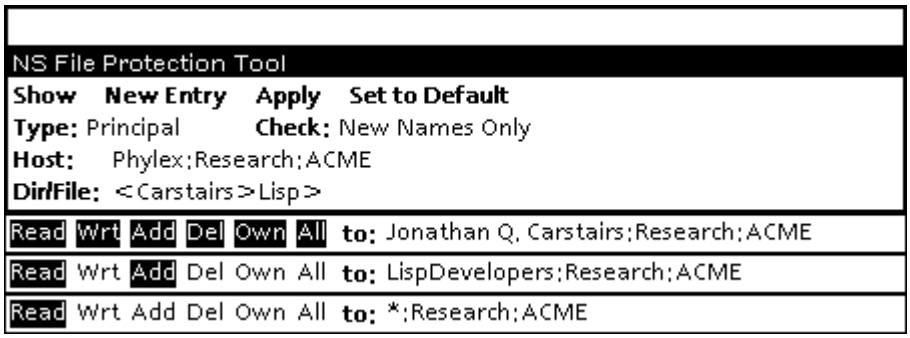
Host: Phylex
Dir/**File:** Carstairs>Lisp

or enter a file name in the usual Lisp syntax in the **Dir**/**File** field, e.g.,

Host:
Dir/**File:** {Phylex:}<Carstairs>Lisp>

This latter form is intended to make it easy to copy-select the name of the directory or file from another source, such as a FileBrowser window; the host in the full name overrides any name in the **Host** line.

To see the protection of a file or directory, click on the command **Show**. The protection is displayed as a series of editable one-line windows beneath the main window. In each line is a set of access rights and a Clearinghouse name or pattern to which those rights are granted; for example,



The highlighted buttons indicate which of the five access rights (Read, Write, Add, Delete, Owner) are granted to the name on the right. If the displayed protection was inherited from its parent subdirectory, rather than having been explicitly set, this fact is noted in the prompt window.

To change the protection of a file or directory, set up the protection entries as desired, then click on the command **Apply**. The usual procedure is to use the **Show** command to see the current protection, then edit one or more entries. Clicking on one of the first five buttons toggles it; clicking on **All** either sets all five (if **All** was previously unhighlighted) or clears all five. In addition, setting either **Write** or **Add** also sets **Read**, since they are of little use without read access (you can, however, clear **Read** if you really meant it). The name following **to** is edited in the same manner as the **Host** and **Dir/File** items above. As with most other places in the system, the name you type can omit the domain and organization, in which case the tool will fill in the local defaults; you can also use nicknames, which will be replaced by the Clearinghouse full names (assuming checking is on).

To add an additional entry, click on the command **New Entry**. This adds a new line to the existing set of protection entries, which you can edit as appropriate. To remove a set of access rights completely for an existing name, either clear all five access buttons (most easily done by clicking once or twice on **All**), or clear the name from the **to** field (by clicking on it with the RIGHT mouse button). Any such cleared lines will be removed by the Apply command.

You can also change the protection of a file back to "default" by clicking on the command **Set to Default**. Following this command, the protection of the specified file is inherited from its parent directory. This is usually the best way to "undo" a changed protection, because then any changes to the protection of its parent, or parent's parent, etc., will have the expected effect on all its children.

For the **Apply** and **Set to Default** commands, you may also specify a group of files, rather than a single file, by giving a file pattern—a name with asterisks serving as wild cards to match zero or more characters. Any pattern acceptable to the File Browser can be used. The tool enumerates the specified set of files and applies the specified protection to each. The enumeration is made to all levels (infinite depth), so affects files both in the immediate directory and also in its subdirectories, and subdirectories of those, etc. The enumeration does not, however, include the top-level subdirectory itself; e.g., "<Carstairs>Lisp>*" matches all files (including subdirectories) anywhere in the directory <Carstairs>Lisp>, but does not include <Carstairs>Lisp> itself.

Note that applying a protection to a directory is different from applying the same protection to the files in it, because of defaulting. If you apply a protection to <Carstairs>Lisp>*, it changes the protection of every file currently in the directory, but any new files added after the change still inherit the protection of the directory <Carstairs>Lisp>. On the other hand, applying a protection to the directory <Carstairs>Lisp> itself affects all current and future files in the directory, *except* any files that already have an explicit protection currently set. To reduce confusion, it is thus preferable to apply protections

to subdirectories, rather than individual files, if you want to control a whole group of files. If you have a subdirectory containing files of miscellaneous protection that you would like to make uniform, the best procedure is to set the desired protection on the subdirectory itself, and then use the **Set to Default** command with a pattern (e.g., <Carstairs>Lisp>*) to reset all the individual files to defaulted.

The **Apply** command looks up in the Clearinghouse each of the names in the individual protection entries to make sure that they are valid, and replaces aliases (nicknames) with the canonical names. It then tells the file server to change the protection as indicated. The extent to which the **Apply** command checks names is controlled by the **Check** item in the second line of the tool window. It has four possible settings:

New Names Only	This is the default setting. The tool checks any names that you have entered or changed, but assumes that names returned by the Show command were correct.
All Names	The tool checks all names, regardless of source. You might want to do this to convert an existing protection entry into canonical form, or check that all the names are still valid.
Never	The tool never checks names; it assumes you meant exactly what you typed. You might want this setting, for example, if one of the names you are entering is registered only in a distant Clearinghouse not currently accessible.
I really mean it	Not only does the tool not check the names, it also doesn't balk if you tell it to take certain unlikely actions, such as changing a top-level directory to default protection, setting a completely null protection, or setting a protection in which nobody has Owner rights (which means the protection can only be changed by someone with Write access to the parent, if any). This setting is "one-shot"—it reverts to "New Names Only" after you issue the next command.

The **Type** item in the second line of the tool window controls which of a directory's two protection attributes is displayed or set. The initial setting is "Principal" and is the one that should normally be used (it coincides with the Lisp file attribute PROTECTION, or "Access List" in NS Filing parlance). The other setting is "Children Only". When the protection type is set this way, the tool deals with the protection that is inherited by default by the directory's children, the attribute called "Default Access List" in NS Filing parlance. Ordinarily, this attribute is defaulted, in which case the directory's principal protection is also used as its children's default protection. Using the **Apply** command changes the Default Access List to the value you specify; using the **Set to Default** command changes it back to defaulted. The **Show** command displays the directory's Default Access List if it has one; otherwise, it displays the principal protection and notes this fact in the prompt window.

The **Type** item is irrelevant for non-directory files (and, in fact, the tool sets it back to "Principal" if it has been changed). When the file is a pattern, the tool always sets the Principal protection; in the case of any subdirectories matching the pattern, it sets the Principal protection to that specified in the window and the Default Access List to "default".

As an additional convenience feature, when you request to **Show** the "Principal" protection of a top-level directory, the tool also displays in the prompt window the directory's current page usage and allocation.