

Chapter 8.

Migrating Applications and Data from SRM to SRM/UX Using `srmrestore(1)`

The `srmrestore(1)` command can be used to read SRM backup tapes onto the HP-UX file system on your SRM/UX server. It is the easiest way to move large numbers of files from SRM to SRM/UX.

You should use `srmrestore(1)` AFTER your SRM/UX server is installed and configured, so that the emulated SRM volumes you will be restoring to will already be set up.

There are two restrictions you must be aware of before using `srmrestore(1)`:

- `srmrestore(1)` can only read SRM backup tapes from SRM version 3.0 or later. Backup tapes from previous SRM versions have a completely different format.
- Duplinked files are not allowed to appear on different SRM/UX file system devices; that is, `srmrestore(1)` will not restore duplinked files from your SRM backup tape if you are trying to put them on separate disc drives. However, if you are restoring duplinked files to an emulated SRM volume in SRM/UX that does not itself contain any local or NFS mount points, then `srmrestore(1)` will successfully restore those duplinked files, even if the emulated SRM volume's Root Directory is NFS mounted on your SRM/UX server's file system.

To bring your SRM files over to SRM/UX, first make a backup tape of the SRM file system. Install that tape in your SRM/UX server's tape drive. Do a "catalog" on the tape to see what files it contains (if you are not sure) using the `-t` option to `srmrestore(1)`. Finally, restore the files you want on your SRM/UX system by executing the SRM/UX command `srmrestore(1)` with the appropriate options (see the table below).

The `srmrestore` Command Options

Option	What it does...
<code>-d</code>	Create directories as needed. If a directory, which appears as a component of the path name of the file to be restored, does not exist: create that directory. For example: <code>Srmrestore</code> , upon restoring the file named <code>./srmuxroot/RMB/FILE.TEXT</code> , will create the directories <code>./srmuxroot</code> and <code>./srmuxroot/RMB</code> , if they do not already exist.
<code>-m</code>	Retain the original modification dates and times on the files. Normally, the current date and time will be used when the file is restored. If this option is specified, the date the file was last modified (as determined from the tape) will be restored, instead. This option has no effect on directories as they are restored.
<code>-v</code>	Print additional information. (Verbose mode) <code>Srmrestore</code> prints the full path name to stdout after each file is restored. If combined with the <code>-t</code> option, <code>-v</code> causes the catalog to print additional information. If you abort <code>srmrestore</code> (for example: with a <code>[CTRL]-C</code>), then all of the file names that have been printed have been completely restored. This fact may help you to know the state of the restore process, i.e., which files have been restored and which haven't.

-u	<p>Run in unconditional mode.</p> <p>Normally, <code>srmdrestore</code> will only overwrite files on the disc which are older than the ones on the tape. If the <code>-u</code> option is specified, restored files will replace existing files on the disc, regardless of which files are newer.</p>
-t	<p>Catalog the tape.</p> <p>Do not restore any files. Just print a catalog of the tape. Print the catalog in the identical format that SRM uses. When combined, options <code>-t</code> and <code>-v</code> print the address of the file on the tape, flags indicating information about the file, and the tape of a target duplink.</p>
-M	<p>Set permission mask. (<code>umask</code>)</p> <p>Normally, a restored file will have its permissions set to 666 (octal) if it's a regular file and 777 (octal) if it's a directory (as modified by the current <code>umask</code>). This option allows you to set the restored file's <code>umask</code> to some non-default value. The "execute" bit will always be cleared for non-directory files being restored.</p> <p>For example: "<code>srmdrestore -M 0700</code>" will force all directories to be restored with permission mode 0700 and all regular files to be restored with permission mode 0600.</p>

Note

The examples in this chapter use `/dev/rct` as the "generic" device file for the cartridge tape device on the SRM/UX server. Check with your HP-UX System Administrator before executing **`srmdrestore(1)`** if you don't know the name of the appropriate device file on your SRM/UX server.

Cataloging a Tape

You can generate a catalog of files that are on a tape with the command:

```
srmdrestore -t /dev/rct
```

The `srmdrestore` output would look something like this:

```
Volume Backed Up:      "DISC1          "
Backup Media Number:  1
Backup File:          "dave            "
Backup Date:          7 Aug 90  11:58:36 AM
Backup Type:          Copy
Backup Log File:      "BKP080790115836 "
```

Tree SRM	Creation	Last Mod	Size
Lvl Pathname	Date	Date	(bytes)
0 .	18 Apr 89	26 Jul 90	---
1 USERS	8 Nov 84	16 Jul 90	---
2 BASIC6	12 Sep 88	12 Jul 90	---
3 B22	16 May 90	17 May 90	---
4 AM	16 May 90	17 May 90	---
5 AMD.TEXT	26 Apr 90	7 Jun 90	9216
5 AMDVR.TEXT	26 Apr 90	7 Jun 90	27648
5 AMEX.TEXT	26 Apr 90	7 Jun 90	5120
5 AMHDR.CODE	17 May 90	7 Jun 90	3072

You could place this catalog in a file named, for example: **`tape1.cat`** by executing:

```
srmdrestore -t /dev/rct >tape1.cat
```

Restoring Your Files

Files will be restored to the directory you are in when you execute **srmdrestore(1)**

Temporary Files

TM# Files

When restoring files, **srmdrestore(1)** will not overwrite a file on disc until the entire file has been read from the tape. **srmdrestore(1)** reads each file into a temporary file with the same name as the target file, but with the prefix "TM#". Thus, if you abort a restore in the middle, the only effect is to leave a temporary file; the file being restored is not created. Once the file has been completely read from the tape, the temporary file is renamed to the target file name.

Duplink Files

When files that have links to other files are restored, **srmdrestore(1)** creates a directory that contains temporary files (temporary files are symbolic links to duplink files). This directory is created in the current directory with the name **BKmmddyhhmmss**, where mmddyhhmmss is the month, day, year, hour, minute and second of the backup. To find out what exactly this name is, look at the "Backup Log File" field of the tape catalog (generated when you run **srmdrestore(1)** with the **-t** option). This directory contains duplinked files with names of the form: **DL#nnnn** where nnnn is a decimal integer (e.g., 7324). This directory and its contents are removed when **srmdrestore(1)** reaches the end of the last tape being restored.

For example, for the tape listed in the previous catalog, a duplink could be named: **BK080790115836/DL#7324** and the temporary file for **AMD.TEXT** would be named: **USERS/BASIC6/B22/AM/TM#AMD.TEXT**

Multiple Tape Restores

The **srmdrestore(1)** command is designed to allow some flexibility in how you restore files from multiple backup tapes.

In the simplest case, **srmdrestore(1)** will issue the prompt:

```
srmdrestore: type next tape device file name (Return=file_name)>
```

when it encounters the end of each tape (except the last tape). When you get the above prompt, all you need to do is insert a new tape and hit [[Return]] after the tape has completed loading.

If you have multiple tape drives, you may type the name of the device file for another tape unit in response to the prompt.

If an error occurs, or the command is aborted at this point, all you have to do to continue is to correct the error and then re-execute the **srmdrestore(1)** command.

You can restore tapes out of sequence, with the following restrictions:

- A file that is duplinked to another file on an earlier tape will not be restored, if the earlier tape has not been restored. **srmdrestore(1)** issues a warning (to stderr) for each such duplinked file.

To determine the files that will be affected, use the **-tv** option to catalog the tape. Files which have the letter "l" in the "Flags" field, are duplinked files. The number following the "l" indicates the number of the

tape that contains the first instance of the duplinked file. If that tape number has already been restored, or is the current tape, then the duplinked file will be restored.

- A file that crosses a tape boundary will not be restored if the previous tape has not been restored. **srmdrestore(1)** issues a warning (to stderr) for each file that crosses a tape boundary.

You can determine if a file crosses a tape boundary by examining the tape catalog (generated when you run **srmdrestore(1)** with the **-t** option). The files that cross tape boundaries will always be either the first or last file on the tape, and will be indicated by a ***** following the date field of that file's entry.

- All other files will be restored, even if the tapes are inserted out of sequence. This **srmdrestore(1)** feature allows you to recover selective files from the middle of a multiple backup-tape-set.

Error Logging

All errors in the restore process will be logged to **stderr**. Redirect these messages to a file if you wish to examine them later.

To redirect the error messages to a file from sh (or ksh) execute:

```
srmdrestore -dvm /dev/rct >restored.files 2>notrestored.files
```

This command sequence will log all errors to the file named **notrestored.files** and will log all successfully restored file names to the file named **restored.files**.

Any tape change requests will be logged only to your terminal, not to stderr.

Pattern Matching

To selectively restore individual files, specify their full path names in the **srmdrestore(1)** command line, exactly as they would appear in the catalog listing. You can specify wildcards to match multiple file names from the tapes. These wildcards function just like the wildcards in the HP-UX "sh" shell do. (See *regexp(7)* in the *HP-UX Reference manual* for details.)

Restore Examples

Example 1

Execute the command:

```
srmdrestore -dvm /dev/rct './USERS/BASIC6/B22/AM/AMEX.TEXT'  
'./USERS/BASIC6/B22/AM/AMD.TEXT'
```

This command will restore two files. The single quote characters are used to prevent the HP-UX shell from incorrectly interpreting characters in the file name.

Example 2

Execute the command:

```
srmrestore -dvm /dev/rct '*.TEXT'
```

This command will restore all of the files (in any directory) that have names ending in ".TEXT".

Example 3

If you cataloged a tape by executing the command:

```
srmrestore -t /dev/rct
```

and got this output:

```
Volume Backed Up:    "DISC1          "  
Backup Media Number: 1  
Backup File:        "dave          "  
Backup Date:        7 Aug 90  11:58:36 AM  
Backup Type:        Copy  
Backup Log File:    "BKP080790115836 "
```

Tree	SRM	Creation	Last Mod	Size
Lvl	Pathname	Date	Date	(bytes)
0	.	18 Apr 89	26 Jul 90	---
1	USERS	8 Nov 84	16 Jul 90	---
2	BASIC6	12 Sep 88	12 Jul 90	---
3	B22	16 May 90	17 May 90	---
4	AM	16 May 90	17 May 90	---
5	AMD.TEXT	26 Apr 90	7 Jun 90	9216
5	AMDVR.TEXT	26 Apr 90	7 Jun 90	27648
5	AMEX.TEXT	26 Apr 90	7 Jun 90	5120
5	AMHDR.CODE	17 May 90	7 Jun 90	3072

and you wanted to restore this tape to the directory **/srmuxroot**, you would execute the commands:

```
cd /srmuxroot  
srmrestore -dvm /dev/rct >restored.filelist
```

The path name of each file restored is printed to the file **restored.filelist** after it has been restored.

After you executed the above **srmrestore** command, the first file restored would be:

```
./USERS/BASIC6/B22/AM/AMD.TEXT
```

and it would have the modification date and time it had on the tape.

srmrestore(1) would create the directories:

./USERS, **./USERS/BASIC6**, **./USERS/BASIC6/B22**, and **./USERS/BASIC6/B22/AM** before restoring the file **AMD.TEXT**, if the directories didn't already exist.