



White Paper: Using Time Machine[®] on an IBM SP System

1. <i>Definitions</i>	1
2. <i>Introduction</i>	1
3. <i>Getting Started</i>	2
4. <i>Nodes</i>	2
5. <i>Users and Auto-synchronization</i>	2
6. <i>Setting a Time Machine virtual clock on a single node</i>	3
7. <i>Setting a Time Machine virtual clock on more than one node</i>	4
8. <i>Summary</i>	5

Note: It is expected that the reader of this document is familiar with both the RS/6000 and the IBM SP. The document is written for developers, testers, analyst and system administrators. Detailed technical knowledge of either the RS/6000 or the SP is not necessary to configure and establish Time Machine's virtual clocks.

1. *Definitions*

The use of the words "*system(s)*" or "*SP system(s)*" will refer to a single SP computer.

The use of the word "*node(s)*" will refer to the individual RS/6000(s) which make up a single *SP system*.

The use of the words "*system clock(s)*" will refer to the single clock on a *node*, which typically is used to answer all requests for date or time. This document does not distinguish between the hardware clock and the software clock.

The use of the words "*SP system clock(s)*" will refer to the controlling, master clock of the *SP system*.

2. *Introduction*

Time Machine[™] is a powerful tool, which allows users to be more productive with their Year 2000 (Y2K) projects. Effectively, Time Machines allows up to 200 virtual *system clocks* on a single computer. One of the many benefits of Time Machine is the ability to run application tests on a production computer without interfering with production applications, or corrupting important information, including log files and file update stamps.

The IBM *SP system* is a grouping of one or more IBM RS/6000s. Within an SP, each RS/6000 is referenced as a *node*. Regardless of the number of *nodes*, the *system* enforces a matching of the *system clocks* with the *SP system clock*. Therefore, it is impossible to execute an application test with a need for a different date/time as is necessary for all Y2K testing unless **all** production is halted. Since these *systems* are specifically provided for 24x7 uptime, it is inconceivable to halt all production. This means an *SP system* generally cannot be used for date/time application testing unless it is defined as a non-production *system*.

Because Time Machine is available for many different platforms, including the RS/6000, date/time application testing can occur on an *SP system* **while** production applications continue to run. Additionally, because of the design of Time Machine, date/time testing can occur with production applications on the same *node*. Therefore, there is no requirement to modify the *node* configuration, or grouping, on an *SP system* to use Time Machine.

As with any UNIX environment, Time Machine's functions and commands are only available to the `root` user by default. However, some functions and commands can be allowed to specific non-`root` users in the typical UNIX manner. Therefore, if a non-`root` user wishes to utilize any of the following functionality without the direct involvement of the `root` user, appropriate permissions must be made.

3. Getting Started

The first step in using Time Machine is the installation of the product. The *SP system* allows administrators to shutdown single *nodes* at a time. This allows the other *nodes* in the *system* to continue processing. Therefore, Time Machine is installed one *node* at a time, allowing the administrator to schedule installs at different times and intervals. Likewise, Time Machine must be installed on all the *nodes* on which a Time Machine user will execute date/time testing.

The installation of Time Machine on a *node* is exactly the same as installing Time Machine on a normal RS/6000. Please refer to the User Manual for the specific instructions and steps.

4. Nodes

Each *node* is assigned a *node* number and name, and can be grouped with other *nodes* in the *system*. Following is an example of finding the *node* groups on the *system* and finding the *nodes* contained in a group.

```
root @ v03n07 => nglis -G
NodeGroup1
root @ v03n07 => ngresolve -G NodeGroup1
1
2
root @ v03n07 => ngresolve -G -w NodeGroup1
v03n01.spc.ibm.com
v03n02.spc.ibm.com
```

There is one *node* group on this *system*, `NodeGroup1`, which includes the first and second *nodes*. These two *nodes* are named `v03n01.spc.ibm.com` and `v03n02.spc.ibm.com`.

5. Users and Auto-synchronization

Auto-synchronization is important in the configuration and management of the *SP system*. Therefore, it is also important in the configuration and management of users and Time Machine. When auto-synchronization is enabled on the *system*, certain rules apply to users. One of these rules is a user that is added to more than one *node* would be assigned the same `uid` on all of the *nodes*. However, if auto-synchronization is disabled it is possible for two users with the same name to have different `uids`.

```
root @ v03n07 => dsh -a lsuser -a id test1
v03n01.spc.ibm.com: test1 id=201
v03n02.spc.ibm.com: test1 id=201
v03n03.spc.ibm.com: test1 id=201
v03n04.spc.ibm.com: test1 id=201
v03n05.spc.ibm.com: test1 id=201
v03n06.spc.ibm.com: test1 id=201
v03n07.spc.ibm.com: test1 id=201
v03n08.spc.ibm.com: test1 id=201
v03n09.spc.ibm.com: test1 id=201
v03n13.spc.ibm.com: test1 id=201
root @ v03n07 => dsh -a lsuser -a id fred
v03n01.spc.ibm.com: fred id=213
v03n02.spc.ibm.com: 3004-687 User "fred" does not exist.
v03n03.spc.ibm.com: 3004-687 User "fred" does not exist.
```

```
v03n04.spc.ibm.com: 3004-687 User "fred" does not exist.
v03n05.spc.ibm.com: 3004-687 User "fred" does not exist.
v03n06.spc.ibm.com: 3004-687 User "fred" does not exist.
v03n07.spc.ibm.com: fred id=214
v03n08.spc.ibm.com: fred id=215
v03n09.spc.ibm.com: 3004-687 User "fred" does not exist.
v03n13.spc.ibm.com: 3004-687 User "fred" does not exist.
```

In this example, one can assume that the user, `test1`, was added when auto-synchronization was enabled. When `test1` was added to the *system*, each user on the *nodes* were automatically assigned the same `uid`, 201. When `fred` was added, auto-synchronization was disabled. Therefore, `fred` was given three different `uids`, 213 on *node* 1, 214 on *node* 7 and 215 on *node* 8.

This can result in problems when using Time Machine, since the virtual clocks are assigned by `uid`, not user name. If `fred` incorrectly believed that setting a virtual clock for `uid` 213 on *node* 1, *node* 7 and *node* 8 would provide a simulated date/time for him, unwanted outcomes would occur. Therefore, users must be very careful when assigning virtual clocks when auto-synchronization is disabled.

There is good news! First, in all but the most non-typical *SP system*, auto-synchronization will be enabled, which would not allow `fred` to be created in this manner. Second, if auto-synchronization was disabled, once enabled, the user `fred` would be deleted from the *nodes* along with the `uids`. When `fred` is re-added, the `uids` will match. Additionally, the old `uids`, 213-215 in this example, would not be available for new users.

NOTE: While it is possible to test using Time Machine with production user names, SolutionSoft recommends that new user names be added specifically for testing. Because user names, and their `uid`, often span multiple nodes, using existing production user names can become confusing. Adding, and using, new test user names will help alleviate this confusion.

6. Setting a Time Machine virtual clock on a single node

Because of the typical *SP system*, it is not likely that a virtual clock is necessary for a `uid` on a single *node*. On the other hand, there are some cases where a `uid` might require a virtual clock for testing purposes. For example, an individual responsible for testing applications may exist on all *nodes*, as `test1` does. For a particular test, this individual may only be running an application on *node* 1, `v03n01`. Setting a virtual clock for all the *nodes* could cause unforeseen problems, so it may be necessary to set the virtual clock only on the one *node*.

```
root @ v03n07 => dsh -w v03n01 /etc/ssstm/tmuser -a -u 201 -x 030204002001

v03n01: Copyright (c) 1997, 1998 SolutionSoft Systems, Inc. All Rights Reserved
v03n01: Tmuser for IBM AIX 3.x and 4.x. Version 1.02
v03n01: Utility for setting TM kernel module configuration.
v03n01:
v03n01: This software is licensed to:Solution
v03n01: Your demo version software expiration date is:Thu Dec 17 14:40:22 1998
v03n01:
v03n01: Uid 201 has been added with a running virtual clock: Fri Mar  2 04:00:00 2001
```

In this `dsh` command, a Time Machine virtual clock was assigned to `uid` 201 on *node* `v03n01` using the `-w` option. The date and time were set to Friday, March 2, 2001. Subsequent application date/time calls on *node* `v03n01` by `uid` 201, such as made with the UNIX commands `cal` or `date`, will see the date as March 2, 2001. Application date/time calls on the other *nodes* will continue to see the *SP system clock* time.

```
root @ v03n07 => telnet v03n01
Trying...
Connected to v03n01.
Escape character is '^]'.

telnet (v03n01)
AIX Version 4
(C) Copyrights by IBM and by others 1982, 1996.
```

```

login: test1
test1's Password:
*****
*
*
* Welcome to AIX Version 4.3!
*
*
* Please see the README file in /usr/lpp/bos for information pertinent to
* this release of the AIX Operating System.
*
*
*****
Last login: Wed Dec  2 15:04:24 CST 1998 on /dev/pts/0 from v03n07
$ cal
      March 2001
Sun Mon Tue Wed Thu Fri Sat
          1   2   3
 4   5   6   7   8   9  10
11   12  13  14  15  16  17
18  19  20  21  22  23  24
25   26  27  28  29  30  31
$ date
Fri Mar  2 04:04:05 CST 2001

```

```

root @ v03n07 => telnet v03n02
Trying...
Connected to v03n02.
Escape character is '^]'.

telnet (v03n02)
AIX Version 4
(C) Copyrights by IBM and by others 1982, 1996.
login: test1
test1's Password:
*****
*
*
* Welcome to AIX Version 4.3!
*
*
* Please see the README file in /usr/lpp/bos for information pertinent to
* this release of the AIX Operating System.
*
*
*****
Last login: Wed Dec  2 15:04:24 CST 1998 on /dev/pts/0 from v03n07

$ cal
      December 1998
Sun Mon Tue Wed Thu Fri Sat
          1   2   3   4   5
 6   7   8   9  10  11  12
13  14  15  16  17  18  19
20  21  22  23  24  25  26
27  28  29  30  31

$ date
Wed Dec  2 20:04:05 CST 1998

```

7. Setting a Time Machine virtual clock on more than one node

As can be seen from the previous example, setting a Time Machine virtual clock is completed with standard `dsh` options. Following are examples for specifying multiple *nodes*:

To set Time Machine for two specific *nodes*:

```
root @ v03n07 => dsh -w v03n01,v03n08 /etc/ssstm/tmuser -a -u 201 -x 030204002001
```

To set Time Machine for the *node* group:

```
root @ v03n07 => dsh -G -N NodeGroup1 /etc/ssstm/tmuser -a -u 201 -x 030204002001
```

8. Summary

When using Time Machine on an *SP system*, there are few things to remember:

- a) Auto-synchronization should be enabled at all times, or you must be very careful with the use of user names and uids.
- b) New user names and uids should be added specifically for testing. Utilizing existing user names can become confusing.
- c) If a user exists on more than one *node*, consideration must be made as to which *nodes* you wish to set a Time Machine virtual clock.
- d) If setting a Time Machine virtual clock for a user on more than one *node*, the appropriate `dsh` command options must be used. This ensures that the Time Machine virtual clocks are synchronized.
- e) Time Machine virtual clocks are "remembered" between log ons. If you do not manually delete a virtual clock for a user, when the user logs on again the existing virtual clock will continue to be used, unless there was a shutdown of the *node* (See the Time Machine User Manual for more information).
- f) Time Machine must be installed on all the *nodes* on which a Time Machine user will execute date/time testing.