TimeWarp/3000

User Manual

Release: 1999-05-22

Table of Contents

TimeWarp Overview

TimeWarp is designed to do just one thing: to allow the testing of user programs and CI scripts within any desired Date and Time, without requiring you to change your system date globally. TimeWarp allows you to specify that a given Job or Session operate under a particular date and Time in one of three ways:

1) **User specified.**

Use the supplied WARP UDC from the CI in any Job or Session, at any time. This immediately switches the Job or Session in which it is issued to any desired Date and Time.

2) **Rule based.**

You can specify that a particular Job or Session automatically operate in a given Date and Time, by entering one (or more) rules via a supplied Manager program. These rules (which may use wildcards) allow you to control what Date and Time any Job or Session will operate in, before it even logs on. Rules remain active until explicitly removed.

3) **Automatic inheritance.**

Any Job that is streamed from within a Job or Session that is currently 'switched' to a TimeWarp-controlled Date and Time, will automatically inherit the same Date and Time. This greatly simplifies testing of complex environments where online applications stream off one or more background jobs to complete processing.

TimeWarp provides advanced facilities to allow you to determine where your applications make calls that look up system Date or Time. TimeWarp can display (or log) just the names of the external Date or Time-related external procedures called by your applications; it can also display (or log) a summary stack trace that shows you exactly where your application makes calls to retrieve the system Date and Time (including calls from SL or XL-based procedures). It works with both CM and NM applications, and can be used by any user (no special MPE capabilities are required).

All TimeWarp features are configured and monitored using a single command (WARP) installed as a system-wide UDC. TimeWarp is supplied with a Manager program that can show exactly which Jobs and Sessions are currently running in TimeWarp-controlled Dates and Times, and gives online access to their log file contents.

TimeWarp and 'External' DateTime Calls

Whatever programming language your applications are written in, when they need to retrieve the current system date or time, they will ultimately call system routines located in one of three system libraries: `SL.PUB.SYS`, `XL.PUB.SYS` or `NL.PUB.SYS`. TimeWarp intercepts these calls, and can modify the Date and Time information returned to your applications so that they 'see' a different Date and Time to the 'real' current system Date and Time.

TimeWarp is able to display the names of all 'external' routines (located in any of the system libraries) called by your applications, to aid you in determining how your applications retrieve the system date and time. More importantly, TimeWarp can display the stack-trace leading up to every date and time-related external call; this stack trace identifies the names of (and offsets within) every procedure that your applications call in order to retrieve the system date and time. This can be of significant help in determining where applications reference dates, particularly when they make use of code located inside user libraries (SLs and XLs located in arbitrary MPE groups).

TimeWarp also intercepts attempts by CI jobs, command files and UDCs to retrieve the system date and time, and correctly modifies the values returned so that your command scripts can be tested at the same time as your application programs. TimeWarp also intercepts date and time references performed via the COMMAND and HPCICOMMAND intrinsics, as well as attempts to programmatically retrieve date and time information from system variables (HPYEAR, HPDATE, etc).

TimeWarp integrates smoothly with VESOFT's MPEX enhanced command interpreter, allowing date and time references within MPEX command files to be tested at the same time as your application programs.

Finally, TimeWarp also correctly intercepts attempts by Posix applications to reference date and time information from their Posix environment, and correctly simulates virtual date references from within the Posix Shell.

Taken together, TimeWarp's extremely thorough coverage of date and time related functionality means that you can rely on TimeWarp to run applications in virtual date and time environments that perfectly mimics the way they'll run when the system date is really set to these virtual dates and times.

Supported MPE/iX Versions

Before installing TimeWarp, please note the following:

TimeWarp has been tested within MPE/iX 5.0, 5.5, and 6.0 environments.

You should not attempt to install or use the product under other MPE/iX releases without first checking with us.

Once TimeWarp has been installed, you will NOT have to reinstall after updating your MPE/iX version, or installing MPE patches.

Technical Support

If you require technical support during testing or use of this product, please contact the Technical Support Staff of our local distributor:

United States, Canada or Mexico:

OmniSolutions, Inc. (800) 935 0101	`http://www.OmniSolutions.com`
MiniSoft, Inc. (800) 682 0200	`http://www.MiniSoft.com`

Europe:

Gainsborough Software (UK) +44 0121 352 0707	`http://www.Gainsborough.com`

You may also contact SMGA directly:

SMGA (714) 271 1360	`http://www.smga3000.com`

Installing (or Updating) TimeWarp

1) Logon as `MANAGER.SYS` and restore the contents of the TimeWarp DAT:

```
:RESTORE ;@.@.@;CREATE
```

2) Stream the supplied installation job to setup the correct accounting structure and complete the installation procedure:

```
:STREAM SETUP.INSTALL.TIMEWARP
```

3) Logon again, so that the new WARP UDC is available to you.

Notes:

- All files are installed into the TimeWarp Account.

- The TimeWarp Account should be password-protected immediately after the SETUP job has completed. Subsequent re-installation of TimeWarp will not remove any password(s) that you've assigned to the Account.

- The SETUP job also installs the WARP UDC system-wide.

The WARP command

All aspects of TimeWarp configuration and management are performed using the `WARP` command, which is automatically installed as a System-level UDC. Typing `:WARP` with no parameters displays the following help screen, summarizing all available WARP keywords:

```
WARP command syntax:

  :WARP STATUS | HELP | ENABLE | MANAGER | PRINTLOG
  :WARP TURNON <LogTrace> <DateTime>
  :WARP TURNOFF <LogTrace>
  :WARP EXCLUDE | INCLUDE <Program>
  :WARP DATA <File>

Specify Absolute <DateTime> using any of these formats:

  YYYY/MM/DD HH:MM ... Absolute Date & Time
  YYYY/MM/DD       ... Absolute Date
  YYYY/MM          ... Absolute Year and Month
  YYYY             ... Absolute Year

Specify Relative <DateTime> using this syntax:

  +/-<number> YEARS | MONTHS | WEEKS | DAYS | HOURS | MINUTES

Available <LogTrace> keywords:

  TRACE            ... Display DateTime calls to STDLIST
  LOG              ... Log DateTime calls to Logfile
  STACK            ... Include Stack leading to DateTime calls
  SINGLE           ... Only show first occurrence of each DateTime call
  ALL              ... Show all DateTime calls
```

The primary WARP functions are summarized below. They are explained in more detail in the sections that follow.

`:WARP DATA`	... Accesses (optional) DataWarp functionality.
`:WARP ENABLE`	... Enables TimeWarp, system-wide.
`:WARP EXCLUDE`	... Excludes Program File from TimeWarp processing.
`:WARP HELP`	... Prints the `WARP` command syntax description.
`:WARP INCLUDE`	... Reverses effect of the `WARP EXCLUDE` command.
`:WARP MANAGER`	... Runs the TimeWarp Manager.
`:WARP PRINTLOG`	... Prints the local Job/Session TimeWarp log file.
`:WARP TURNOFF`	... Turns-off any TimeWarp virtual Date and Time, locally.
`:WARP TURNON`	... Turns-on a TimeWarp virtual Date and Time, locally.

Most WARP keywords are available at any time; some are not available when you're currently in BREAK, or within a program other than the CI. Any such restrictions are detailed in the sections that follow.

Most WARP keywords may be combined with other keywords. For example:

```
:WARP ENABLE TURNON 1999/12/31 TRACE
```

has the same effect as the following sequence:

```
:WARP ENABLE
:WARP TURNON 1999/12/31
:WARP TURNON TRACE
```

Enabling TimeWarp System-Wide (WARP ENABLE)

After installation, before TimeWarp can be used to switch any Job or Session into a 'virtual' DateTime, you must 'enable' the use of TimeWarp by using the following command:

```
:WARP ENABLE
TimeWarp System-wide AIF:PE traps enabled.
```

You must be logged on as a user with SM or OP capabilities to issue this command, which is also only available when you're at the regular CI prompt.

The `:WARP ENABLE` must be performed after every system restart. You only need to perform `:WARP ENABLE` once after each restart. If you attempt to enable TimeWarp when it's already enabled, you'll see the following:

```
:WARP ENABLE
Warning: TimeWarp System-wide AIF:PE traps already enabled.
```

You can use the `:WARP STATUS` command (described below) at any time to determine if TimeWarp is currently enabled or not.

Displaying TimeWarp Status (WARP STATUS)

To display current TimeWarp status, type the following:

```
:WARP STATUS
```

This command may be issued at the CI, when in BREAK, or even from within other programs (such as MPEX or QEDIT). The `:WARP STATUS` output is also automatically displayed whenever the WARP command is used to change any aspect of TimeWarp configuration.

The output format varies according to whether TimeWarp is enabled or not, and whether a virtual TimeWarp DateTime is currently turned on or not.

Example 1: TimeWarp not currently Enabled:

```
:WARP STATUS
-------------------------------------------------------------------------
TimeWarp Status:
TimeWarp Access ... Not Currently Enabled
-------------------------------------------------------------------------
```

Example 2: TimeWarp Enabled; no local virtual DateTime turned on:

```
:WARP STATUS
-------------------------------------------------------------------------
TimeWarp Status:
TimeWarp Access ... Enabled
TimeWarp Status ... Turned OFF (WED, JAN 7, 1998, 9:18 PM)
-------------------------------------------------------------------------
```

Example 3: TimeWarp Enabled; Local DateTime set to '2002/01'; in BREAK:

```
:WARP STATUS
-------------------------------------------------------------------------
TimeWarp Status: (IN BREAK)
TimeWarp Access ... Enabled
TimeWarp Status ... Turned ON (MON, JAN 7, 2002, 9:19 PM)
Current Spec    ... '2002/01'
TRACE mode      ... FALSE
LOG mode        ... FALSE
-------------------------------------------------------------------------
```

The output of `:WARP STATUS` also displays the current LogTrace options, described in more detail in the 'LogTrace Options' Section below.

Turning-On a Virtual DateTime Locally (WARP TURNON)

To switch the DateTime in the current Job or Session to any desired Date or Time, type the following:

```
:WARP TURNON <DateTime>
```

The <DateTime> specification may be either absolute, or relative to the current system date. Absolute <DateTime> may be specified using any of the following forms:

```
YYYY/MM/DD HH:MM        ... specifies absolute Date & Time
YYYY/MM/DD              ... specifies absolute Date
YYYY/MM                 ... specifies absolute Year and Month
YYYY                    ... specifies absolute Year
```

Relative <DateTime> may be specified using the following syntax:

```
+/-<number> YEARS       ... today +/- number of years
+/-<number> MONTHS      ... today +/- number of months
+/-<number> WEEKS       ... today +/- number of weeks
+/-<number> DAYS        ... today +/- number of days
+/-<number> HOURS       ... today +/- number of hours
+/-<number> MINUTES     ... today +/- number of minutes
```

For example:

```
:WARP TURNON 1999/12/31 23:55

:WARP TURNON 1985/05/18

:WARP TURNON 2027/05

:WARP TURNON +10 years

:WARP TURNON -24 months
```

The DateTime specification should be entered exactly as shown. Use leading zeros if necessary, and specify Time using the 24-hour clock. You must include the '/' and ':' delimiters as shown, when necessary.

When using Relative DateTime specifications, you may specify any number of units of the specified 'base', for example the number of MONTHS is not limited to 12.

Relative <DateTime> specifications are always applied to the current system date to determine the 'virtual' TimeWarp DateTime. If you're currently running in a TimeWarp-controlled virtual DateTime, and you issue `:WARP TURNON` with a relative DateTime, the relative offset will be applied to the current real system date, not to your virtual TimeWarp DateTime. In other words, relative DateTimes

are not cumulative.

The <DateTime> parameter, whether specified absolutely or relatively, must lie within the year range 1970 to 2027. Dates outside this range are not currently supported by MPE/iX.

Following use of `:WARP TURNON`, all processes created within the local Job/Session will operate in the specified DateTime, and all Jobs streamed from within the local Job/Session will 'inherit' the same DateTime.

The <DateTime> specification may be omitted if `:WARP TURNON` has previously been used in the current Job/Session. In this case, the previously specified DateTime will be reapplied.

The `:WARP TURNON` command may be issued as many times as required, specifying different DateTimes, if desired.

No special MPE capabilities are required to issue `:WARP TURNON`. It may be used when in BREAK, but may not be used directly from within user programs.

Turning-Off a Virtual DateTime Locally (WARP TURNOFF)

To turn-off TimeWarp within the current Job/Session, type the following:

```
:WARP TURNOFF
```

The local DateTime will revert to the current 'real' system DateTime.

Turning LogTrace Options On or OFF (WARP TURNON / TURNOFF)

TimeWarp, in addition to allowing you to simulate virtual DateTimes, also allows you to Trace (display on the STDLIST device) and/or Log (write to an MPE file) a record of all calls to DateTime related intrinsics. Tracing and Logging are turned on using `:WARP TURNON` and `:WARP TURNOFF`:

```
:WARP TURNON <LogTrace>
```

Where <LogTrace> may be any of the following keywords:

`TRACE`	... send output to STDLIST
`LOG`	... send output to LogFile
`STACK`	... show stack trace for every event
`ALL`	... record all events
`SINGLE`	... record single event in each category

The current <LogTrace> settings are displayed by `:WARP STATUS`, but only when a virtual DateTime is active.

The TRACE and LOG options may be active at the same time, the ALL and SINGLE options are mutually exclusive.

More information about Log and Trace facilities is contained in the relevant section, below.

TimeWarp Manager (WARP MANAGER)

To access the TimeWarp Manager program:

```
:WARP MANAGER
```

The TimeWarp Manager program allows you to configure logon rules, monitor who is running, or has run TimeWarp and their current running virtual time, as well as to manage and view log files.

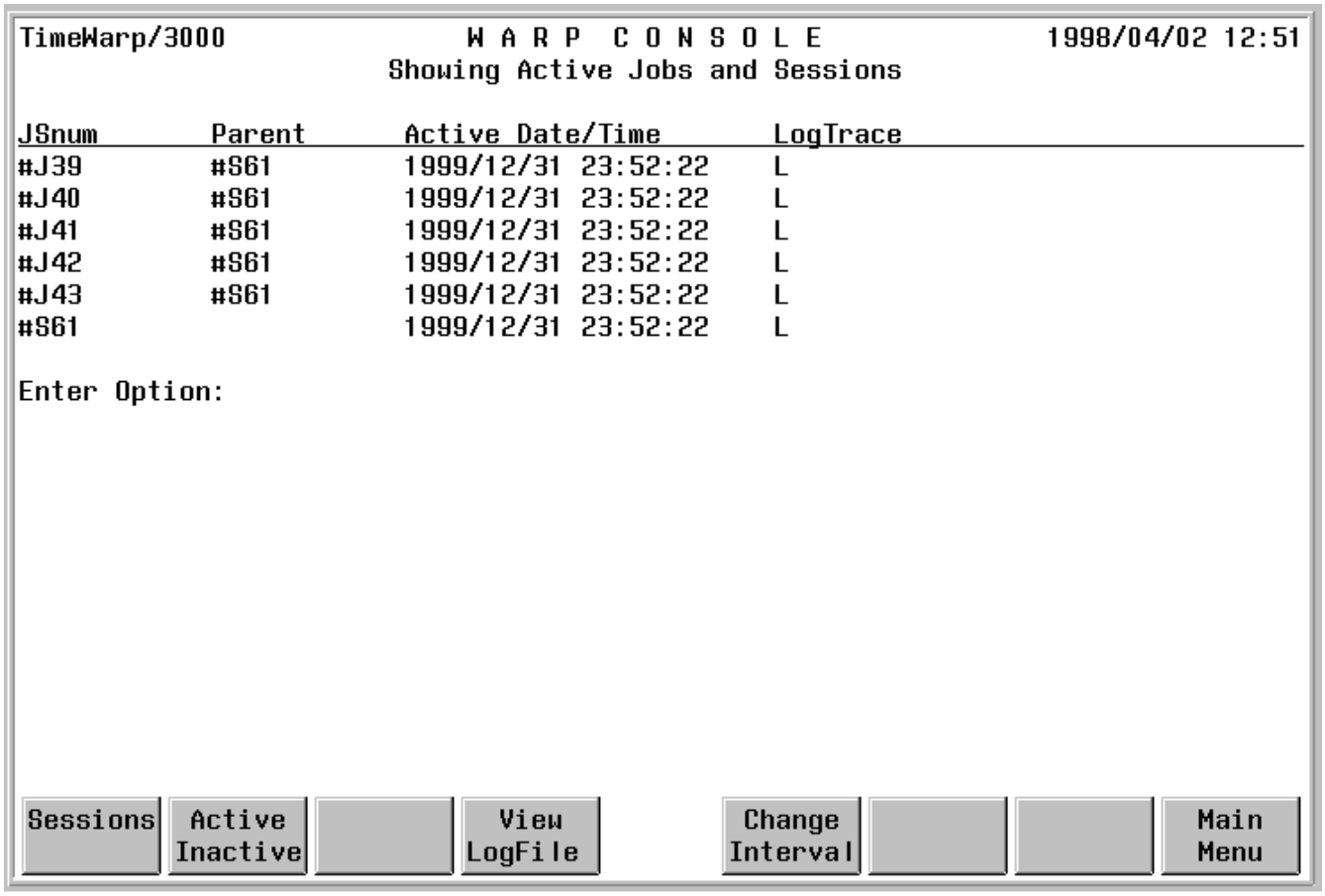
The TimeWarp Manager program uses full-screen character mode. This means that although you are presented with full-screen displays, you still press <Return>, or <Enter> on the main keyboard (rather than <Enter> on the numeric keypad) to enter data.

You will be prompted in 2 different ways. The first way is the Main Menu pop up window. This is very similar to a PC Windows-type drop down menu in that you use the up and down arrow keys to highlight an item and then press <Return>. In the second way you are presented with a screen that has various fields that require values. The cursor will highlight each field, and your options for entry will be displayed on the bottom of the screen. If function keys are displayed, whatever options you see are available to you.

Whenever an error message is displayed it will be centered and highlighted; you must press <Return> to clear it.

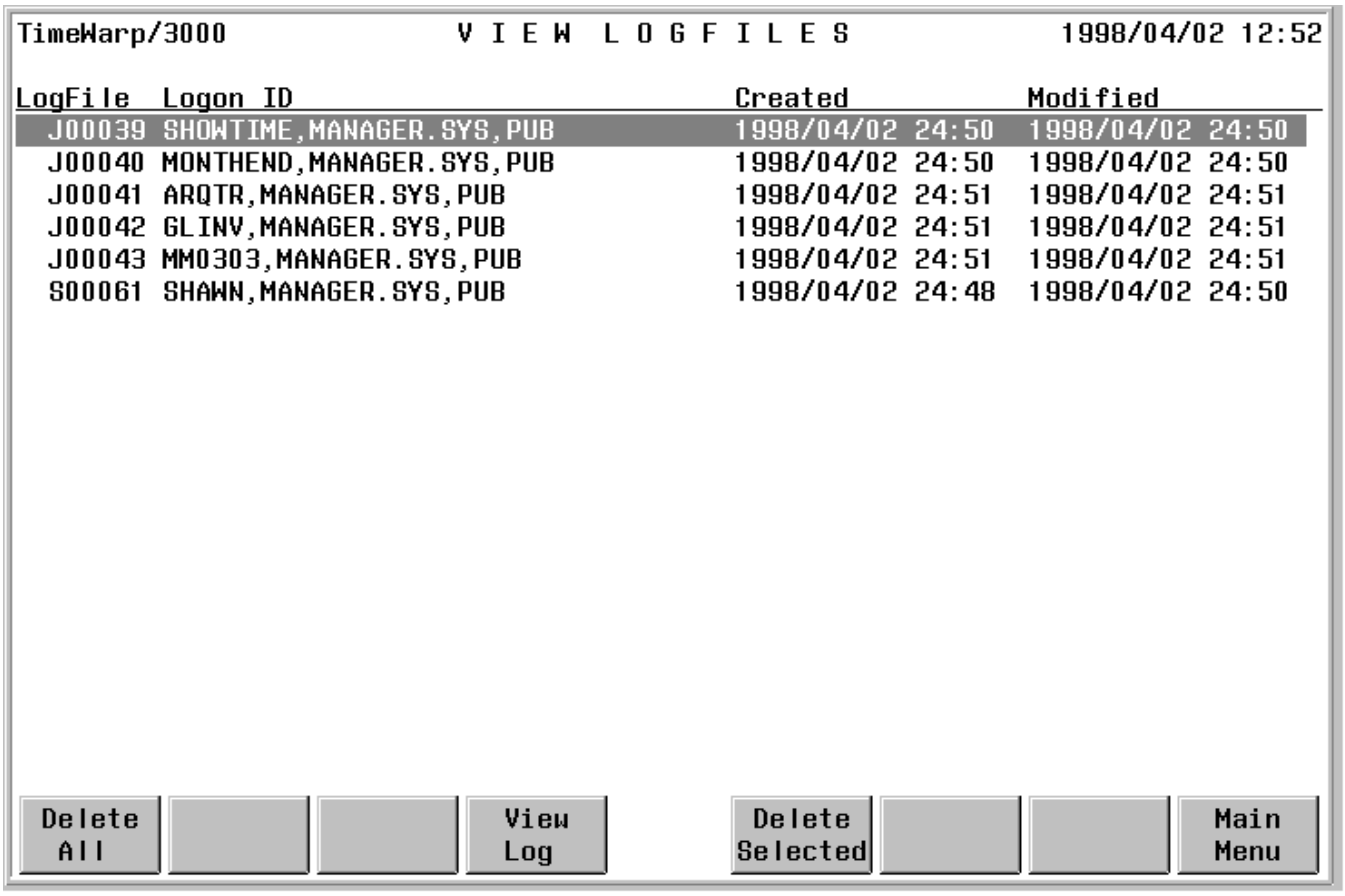
The initial :WARP MANAGER Menu screen displays the following options:

- **Warp Console**



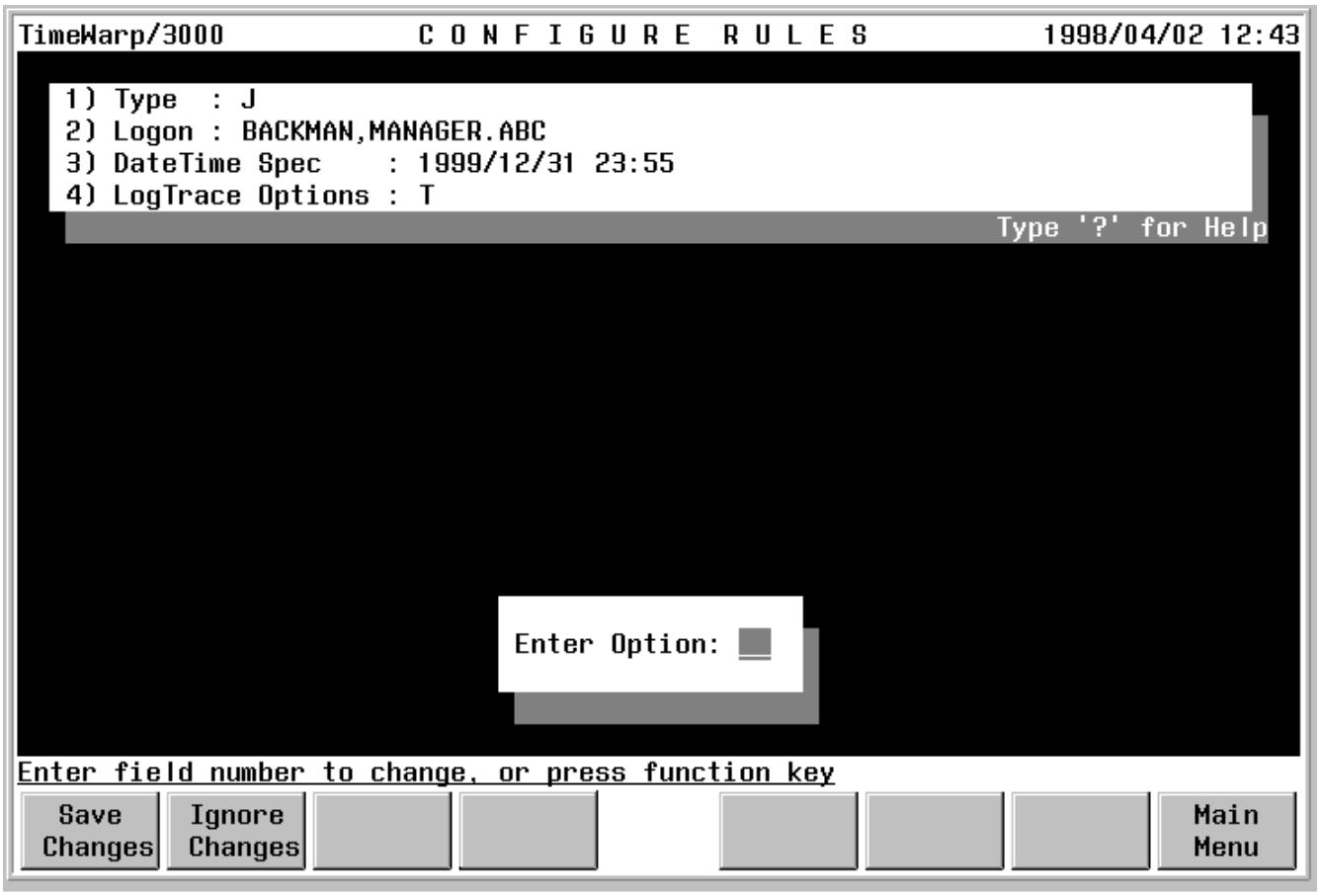
Select this option to display all users who are currently operating in TimeWarp virtual DateTime environments. Can also show users who were using TimeWarp, but have since logged off.

– View LogFiles



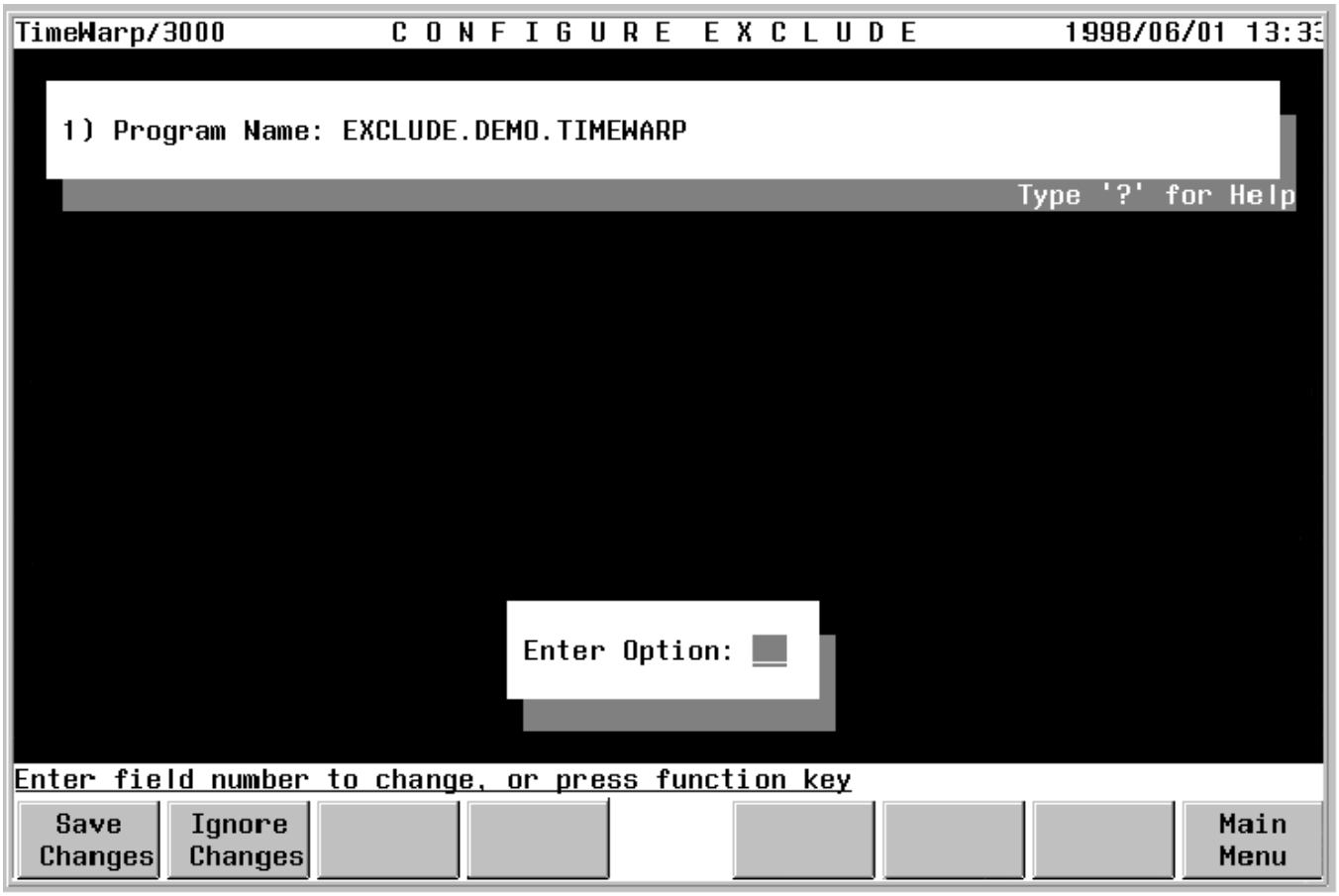
Select this option to see all currently available TimeWarp LogFiles, and to display the contents of any file. Also allows LogFiles to be deleted.

- Configure Rules



Select this option to display all current TimeWarp logon Rules, and to optionally add, delete or modify any rule.

- Configure Exclude



Select this option to display all globally excluded Program Files, and to optionally add or delete any Program File from the exclude list.

- Enable TimeWarp (Option only displayed when TimeWarp not currently enabled)

  Select this option to Enable TimeWarp System-wide (same effect as typing `:WARP ENABLE` command).

- Exit

  Select to exit WARP MANAGER.

WARP Manager: Warp Console Screen

This screen displays (by default) information on all users who are currently operating in TimeWarp controlled 'virtual' DateTime environments. The following information is shown for all displayed users:

- `JSnum` ... Job/Session Number of user
- `Parent` ... Job/Session from which DateTime was inherited
- `Active DateTime` ... Current 'virtual' DateTime
- `LogTrace` ... Current LogTrace options

The screen display is updated automatically every 5 seconds; the update interval may be changed via Function Key 4. Each active Job/Session's virtual DateTime field is updated whenever the screen is redrawn.

LogTrace options are abbreviated to single characters, interpreted as follows:

`T` ... Trace intercepted DateTime calls to STDLIST.
`L` ... Log intercepted DateTime calls to a LogFile.
`S` ... Include Stack trace leading up to intercepted DateTime call.
`1` ... Show 1 occurrence only of each intercepted DateTime call.

Within this screen, Function Keys perform the following functions:

- F1: Toggles between three display modes:

  - Display info for Jobs and Sessions (default)
  - Display info for Sessions only
  - Display info for Jobs only

- F2: Toggles between three display modes:

  - Display info for Active users
  - Display info for Active and Inactive users
  - Display info for Inactive users only

  'Inactive' refers to Jobs and Sessions that ran under a TimeWarp virtual DateTime, but are no longer logged on.

- F3: View LogFile for selected user.

- F4: Change update interval.

- F8: Return to Main Menu.

WARP Manager: View LogFiles Screen

This screen displays information on all existing LogFiles, located in the `LOG.TIMEWARP` group.

The following information is shown for all displayed files:

- `LogFile` ... Name of LogFile, identifying Job/Session number.
- `Logon ID` ... Name Job/Session logged on as.
- `Created` ... Real DateTime LogFile created.
- `Modified` ... Real DateTime LogFile last written to.

Within this screen, Function Keys perform the following functions:

- F1: Deletes all LogFiles.

- F4: View highlighted LogFile.

- F5: Delete highlighted LogFile.

- F8: Return to Main Menu.

WARP Manager: Configure Rules Screen

On entry, this screen displays all configured Logon Rules.

Logon rules allow you to setup any number of rules that are tested whenever any Job or Session logs on. If the Job or Session name matches a particular rule, the associated DateTime is automatically setup as soon as the Job or Session logs on. In addition, any specified LogTrace options are turned on.

A rule consists of both 'Type' and 'Logon ID' fields. If both fields match, then the rule's DateTime Spec and LogTrace Options are automatically applied.

The following information is shown for each Rule:

- `Type` ... 'J', 'S' or '@'.

This allows a rule to be constructed that covers only Jobs, only Sessions, or both Jobs and Sessions.

- `Logon ID` ... Job/Session pattern (including MPE wildcards).

This specifies the Job/Session name pattern used when determining if a particular rule applies to a job or Session. The Logon ID is of the form:

```
jobsession,user.account,group
```

The 'jobsession' and 'group' parts are optional. MPE wildcards can be used in any part of the Logon ID.

- `DateTime Spec` ... Absolute or Relative DateTime specification to apply if the Job/Session matches the rule.

The DateTime spec accepts the same formats as the `:WARP TURNON` command.

It's also legal to specify blanks for this field. In this case the rule specifies that qualifying users be excluded from TimeWarp processing. As rules are evaluated in order until the first match, any such 'exclusion' rules must appear before normal 'inclusion' rules (with non-blank DateTime specifications) for them to have the desired effect.

- `LogTrace Options` ... LogTrace abbreviated Options to apply if the Job/Session matches the rule.

The LogTrace options accepts the following characters, specified in any order:

```
T       ... Trace intercepted DateTime calls to STDLIST.
L       ... Log intercepted DateTime calls to LogFile.
S       ... Include Stack trace up to each DateTime call.
1       ... Show 1 occurrence only of each DateTime call.
```

Within this screen, Function Keys perform the following functions:

- F1: Add a new Rule.

- F4: Modify highlighted Rule.

- F5: Delete highlighted Rule.

- F8: Return to Main Menu.

A particular rule may be modified by selecting it, and following the dialog that appears on the next screen. New rules are defined using the same screen.

With the 'Modify Rule' screen, the following Function Keys are available:

- F1: Save changes.

- F2: Ignore changes.

- F8: Return to Main Menu.

WARP Manager: Configure Exclude Screen

On entry, this screen displays all configured globally Excluded Program files.

Excluding a Program using this screen ensures that it will not be affected by any TimeWarp virtual DateTime settings in any Job or Session on the system.

The only information that is stored is the fully-qualified name of each Program file to be excluded. You cannot use wildcards in this screen.

Within this screen, Function Keys perform the following functions:

- F1: Add a new Excluded Program.

- F4: Modify highlighted Program.

- F5: Delete highlighted Program.

- F8: Return to Main Menu.

A particular Program name may be modified by selecting it, and following the dialog that appears on the next screen. New Excluded Program names are defined using the same screen.

With the 'Modify Program' screen, the following Function Keys are available:

- F1: Save changes.

- F2: Ignore changes.

- F8: Return to Main Menu.

Displaying Local LogFile contents (WARP PRINTLOG)

To display the contents of the local LogFile:

```
:WARP PRINTLOG
```

To display contents of Logfile for any Job/Session, use the `:WARP MANAGER` command, as described previously.

Tracing and Logging Facilities

TimeWarp can Trace (display on the STDLIST device) and/or Log (write to an MPE file) a record of all calls to Date or Time related intrinsics.

Tracing and Logging are turned on using `:WARP TURNON`, described above.

Trace and Log output is only generated when a virtual TimeWarp DateTime is currently turned on.

1) Standard TRACE/LOG output.

By default, both Trace and Log output record every instance of a call to any Date or Time related intrinsic by generating a single line per call:

```
---------------------------------- CALENDAR (intrinsic)
---------------------------------- CLOCK (intrinsic)
---------------------------------- HPGMTSECS (intrinsic)
---------------------------------- ctime (C library)
---------------------------------- HPCIGETVAR (intrinsic): HPYYYY
```

2) 'STACK' trace output.

When the STACK option is set ON, the stack trace leading up to every call to any Date or Time related intrinsic is also Traced/Logged, for example:

```
-------------------------------------------------------------------------
                               CALENDAR (intrinsic)
      export stub: 368.00006c00 SHOWCLKS.PUB.DEV/get_mpe_time+$14
    SP=41836208 RP=368.00006c50 get_all_times+$20
    SP=418361d0 RP=368.000074e8 PROGRAM+$428
-------------------------------------------------------------------------
                               CLOCK (intrinsic)
      export stub: 368.00006c0c SHOWCLKS.PUB.DEV/get_mpe_time+$20
   SP=41836208 RP=368.00006c50 get_all_times+$20
    SP=418361d0 RP=368.000074e8 PROGRAM+$428
-------------------------------------------------------------------------
                               HPGMTSECS (intrinsic)
      export stub: 149.00083c08 XL.PUB.SYS/time+$34
      export stub: 368.00007a48 SHOWCLKS.PUB.DEV/showctime+$10
    SP=418361c8 RP=368.00007940 PROGRAM+$880
-------------------------------------------------------------------------
                               ctime (C library)
      export stub: 368.00007a54 SHOWCLKS.PUB.DEV/showctime+$1c
    SP=418361c8 RP=368.00007940 PROGRAM+$880
-------------------------------------------------------------------------
```

When output is being sent to a log file (`:WARP TURNON LOG` has been used), all output from all processes run within the monitored Job/Session is appended to the same file. To allow output from individual processes to be distinguished from each other, each LogFile line is prefixed by the pin number of the process that generated it.

For example:

```
:WARP TURNON 2002/01 LOG STACK
:RUN COBXX.PUB
:WARP PRINTLOG

Contents of TimeWarp LogFile S00053.LOG.TIMEWARP:

>>>>>> User #S53: MANAGER.DEV,PUB                          1998/01/07 22:00:16

68 >>> Logging CI.PUB.SYS                                   2002/01/07 22:00:16
44 >>> Logging COBXX.PUB                                   2002/01/07 22:00:41
44 >>> Virtual DateTime:                                   2002/01/07 22:00:41
44 >>>
44 >>> ---------------------------- HPGMTSECS (intrinsic)
44 >>>    export stub: 150.0037d6a0 XL.PUB.SYS/COB_TZ_DATE+$24
44 >>>    export stub: 538.000052a0 COBOL.PUB.DEV/datez+$28
44 >>>
44 >>> ---------------------------- CALENDAR (intrinsic)
44 >>>    export stub: 150.00378894 XL.PUB.SYS/cr_calendar+$10
44 >>>    SP=41836288 RP=150.0037651c COB_DAY_DATE_TIME+$134
44 >>>    SP=41836250 RP=150.00375cc0 COB_DISPLAY_SR+$14
44 >>>    export stub: 538.000053d0 COBOL.PUB.DEV/datez+$158
44 >>>
44 >>> ---------------------------- CLOCK (intrinsic)
44 >>>    export stub: 150.003788bc XL.PUB.SYS/cr_clock+$10
44 >>>    SP=41836288 RP=150.0037640c COB_DAY_DATE_TIME+$24
44 >>>    SP=41836250 RP=150.00375cc0 COB_DISPLAY_SR+$14
44 >>>    export stub: 538.00005418 COBOL.PUB.DEV/datez+$1a0
44 >>>
68 >>>
68 >>> TURNOFF                                             2002/01/07 22:01:46
68 >>>
```

In this example, the CI has pin number 68, while the program `COBXX.PUB` ran with pin number 44.

The LogTrace settings are automatically inherited by jobs streamed from within the active Job or Session.

Disabling TimeWarp System-Wide

To disable TimeWarp, system-wide:

```
:WARP DISABLE
TimeWarp System-wide AIF:PE traps disabled.
```

You need SM or OP capabilities to issue this command.

This disables TimeWarp's AIF:PE traps system-wide, immediately. However, it does not affect any other product's use of AIF:PE traps on your system.

You'll need to re-enable TimeWarp using the `:WARP ENABLE` command before TimeWarp can be used again.

Note: Disabling TimeWarp should never normally be necessary: the traps have no effect for any Job/Session unless TimeWarp has been turned on via the `:WARP TURNON` command, or the Job/Session matches one of the logon Rules defined via WARP MANAGER's Configure Rules screen.

Automatic DateTime Inheritance

When a Job or Session is operating in a TimeWarp 'virtual' DateTime, any jobs that are streamed from within the Job/Session will also automatically execute in the same virtual DateTime.

This feature greatly simplifies testing of complex environments, where online applications perform processing by streaming off background jobs.

Such 'nested' jobs are said to 'inherit' the DateTime of the Job/Session that streamed them. When they log on, an extra line appears in the Job's STDLIST indicating that the DateTime has been inherited:

```
   JOB SAMPLE,MANAGER.SYS,PUB.
   Priority = DS; Inpri = 8; Time = UNLIMITED seconds.
   Job number = #j55.
   MON, NOV 17, 1997, 10:48 PM.
   HP3000 release: C.55.00 User Version: C.55.00
   MPE/iX HP31900 C.05.08 Copyright Hewlett-Packard 1987.
   All rights reserved.
   STREAMED BY ONOFF,MANAGER.SYS (#J54) ON LDEV# 10
   STREAM DATE: MON, NOV 17, 1997, 10:48 PM
>> TIMEWARP DATE: SUN, NOV 22, 1987, 11:09 AM (inherited)
```

Similarly, any Jobs streamed from within this Job will Inherit the DateTime settings currently active

within this Job. The :WARP MANAGER command's 'Console' screen identifies all TimeWarp controlled Jobs, and displays the Job/Session from which they inherited their DateTime settings.

In addition to Jobs, any Sessions that are created by use of the :STARTSESS command from within a 'Warped' Job or Session will inherit the Warped Job or Session's virtual DateTime. An identification line is displayed when such Sessions are created that indicates that the DateTime has been inherited.

Identifying Rule-Based Jobs

When Jobs log on that match logon Rules (setup using the :WARP MANAGER command's 'Configure Rules' screen), an extra line appears in the Job's STDLIST indicating that the DateTime is Rule-based:

```
   JOB SAMPLE,MANAGER.SYS,PUB.
   Priority = DS; Inpri = 8; Time = UNLIMITED seconds.
   Job number = #j55.
   MON, NOV 17, 1997, 10:48 PM.
   HP3000 release: C.55.00 User Version: C.55.00
   MPE/iX HP31900 C.05.08 Copyright Hewlett-Packard 1987.
   All rights reserved.
   STREAMED BY ONOFF,MANAGER.SYS (#J54) ON LDEV# 10
   STREAM DATE: MON, NOV 17, 1997, 10:48 PM
>> TIMEWARP DATE: SUN, NOV 22, 1987, 12:00 AM from rule)
```

If a Job would be both a candidate for a Rule-based DateTime, as well as an Inherited DateTime, the Inherited DateTime takes precedence.

Excluding Individual Program Files from being affected by TimeWarp

In the event that you wish to prevent particular programs from being affected by TimeWarp, even when the job or Session in which they're being run is operating in a virtual DateTime, you can do so using either of two available exclusion mechanisms: Local or Global.

Local Exclusion

To exclude a particular program from being affected by any TimeWarp virtual DateTime within the current Job or Session, issue the following command:

```
:WARP EXCLUDE <program>
```

where <program> is the name of the program to be excluded. The <program> name may be fully or

partially qualified, and must exist when the command is issued. You can use MPE or HFS syntax when specifying the program name.

Internally, the `:WARP EXCLUDE` command converts <program> into a SETVAR command, mapping embedded '.' and '/' characters to '_' characters.

Programs excluded using this command are only excluded for the duration of the Job or Session in which the command was issued. The exclusion is not inherited by Jobs streamed from within the current Job or Session.

To reverse the effect of a `:WARP EXCLUDE` command, use the following command:

```
:WARP INCLUDE <program>
```

Global Exclusion

To globally exclude a particular program from being affected by any TimeWarp virtual DateTime, in any Job or Session, use the `:WARP MANAGER` command, and select the Configure Exclude option on the Menu screen.

The Globally excluded program list is processed once, whenever a Job or Session logs on. Changes made do not take effect within currently running Jobs or Sessions.

Displaying all Excluded Programs

To display all currently excluded Program files (both Local and Global), issue the following command (without any parameters):

```
:WARP EXCLUDE

TimeWarp Excluded Programs:

TIMEWARP_EXCLUDE_EDITOR_PUB_SYS = TRUE
```

The display shows all active excluded programs, by internally performing a SHOWVAR on the converted program names (after '_' replacement). Excluded program names are represented by variables set TRUE.

You can also use the `:WARP MANAGER` command's Configure Exclude option to display the Global exclusion list.

TimeWarp Implementation Notes

- There are three ways that a Job/Session can operate under TimeWarp's control:

    - User issues `:WARP TURNON` command within CI.

    - User uses `:WARP MANAGER` to set up Rule within `SPEC.PUB.TIMEWARP` file, then Job/Session matching rule logs on.

    - User streams Job from within TimeWarp virtual DateTime environment.

- TimeWarp intercepts (indirectly or not) all calls to:

```
CALENDAR
HPCALENDAR
CLOCK
DATELINE
TELLOP / PRINTOP
HPCIGETVAR
COMMAND / HPCICOMMAND
AIFTIME
```

    All language specific date/time primitives, such as COBOL's TIME-OF-DAY and Posix program references to ENV variables.

- Within the CI, TimeWarp catches all references to all HP-date and time system variables wherever they're used (IF, WHILE, CALC, ECHO, etc). TimeWarp intercepts references to the following system variables:

```
HPDATE
HPDATEF
HPDATEYEAR
HPDAY
HPDOY
HPHHMMSSMMM
HPHOUR
HPLEAPYEAR
HPMINUTE
HPMONTH
HPTIMEF
HPYEAR
HPYYYY
HPYYYYMMDD
```

TimeWarp also intercepts the following CI commands:

```
:SHOWCLOCK
:SHOWME
:SHOWTIME
:SHOWJCW
:SHOWVAR (including wildcard variable patterns)
```

- Within the Posix Shell (`SH.HPBIN.SYS`), the initial environment list is set up with all DateTime related system variables correctly fixed up.

  Note that the Date and Time related env variables in the shell represent a snapshot of the Date and Time when the Shell was launched. They are not implemented (under MPE) as active functions.

- TimeWarp integrates seamlessly with VESOFT's MPEX, including the following:

```
%CALENDAR
%SHOWME
%SHOWTIME
%SHOWVAR (including wildcard variable patterns)
```

  VESOFT recommend installing version 27N or later of MPEX, SECURITY/3000 and VEAUDIT to ensure that all products correctly handle dates in the year 2000.

- TimeWarp is not intended to allow the security systems of third-party products to be circumvented.

  We will work with third party vendors to ensure that TimeWarp is not used to bypass their internal security.

TimeWarp Account Structure

The TimeWarp account contains the following Groups:

`DEMO .......`	Example programs and CI scripts that demonstrate TimeWarp capabilities.
`HELP .......`	TimeWarp Documentation and Help files.
`INTERNAL ...`	Files used internally by TimeWarp.
`INSTALL ....`	Copies of all files. May be purged following installation.
`LOG ........`	Log files generated by TimeWarp.
`PUB ........`	User accessible Programs and commands.

DataWarp Module Overview

So, you've converted all your programs to be Y2K compliant, and you've expanded date fields in all your database, MPE and KSAM files. The only problem is that you don't actually have very much, or any data in them with dates in the new century that you can test against. This is where DataWarp can help.

DataWarp is an add-on module to TimeWarp/3000 that allows you to age any date, in any file. We support dates stored in all types of 32 bit integer fields (I2, J2, K2), as well as in Z6, Z8, X6, or X8 field types. Dates are recognized when stored as named Image item fields, when embedded within Image items, or when located in MPE/KSAM files.

The following date formats are currently supported:

YYMMDD
CCYYMMDD
MMDDYY
DDMMYY
CALENDAR format
HPCALENDAR format
YYMMDD date, where YY is an MM3000-style year (2000 represented as 'A0')
MMDDYY date, where YY is an MM3000-style year (2000 represented as 'A0')
DDMMYY date, where YY is an MM3000-style year (2000 represented as 'A0')

You specify the offset to be applied to dates using the same syntax as TimeWarp uses to specify system date offsets: absolute or relative offsets specified in YEARS, MONTHS, WEEKS and/or DAYS.

Using DataWarp

DataWarp can perform up to 99 date transformations in a single execution, with the option of only selecting records that match supplied search values.

The DataWarp subsystem operates in either of two modes:

1: Interactive Mode

In this mode, DataWarp presents a number of prompts, allowing you to define the file being operated on, the location and type of the date field being manipulated, as well as the date offset to be applied.

When all prompts have been replied to, you have the option of performing the date transformation immediately, or saving the prompt replies in a flat file, known as a DataWarp script file.

Interactive mode is accessed using the `:WARP DATA` command.

2: Script-Driven Mode

In this mode, DataWarp is provided with the name of a previously-built DataWarp script file. It performs the transformation defined in the script file without prompting for any additional input.

Script-Driven mode is accessed using the `:WARP DATA <ScriptFile>` command.

DataWarp Interactive Mode

Within Interactive mode, type '`??`' at any time for context-sensitive help, describing the input currently being prompted for.

Pressing <Enter> or <Return> at any prompt backs up to the prior prompt level. Default or available values are displayed inside '[]'s, at the end of each prompt.

Entering '/' at any prompt immediately exits DataWarp, without performing any processing.

The Filename prompt will be displayed first:

```
Enter DataBase or FileName:
```

The file can be any type of MPE or KSAM file, not including MSG files. If you enter a file (and not a DataBase) you will then be prompted for the starting byte position where the date is located in the file. The first character of the file is counted as position one (1).

If you enter a DataBase name, you will be prompted for the Database password, (default is '`;`' for creator access). You will then be prompted to enter the Image open mode to be used. Default is 1, but you may need to use 3 or 4, depending on your environment.

When a DataBase is being operated on, DataWarp next prompts for the Data Set to be transformed:

```
Enter DataSet or <??>:
```

Typing '`??`' at this prompt yields the following display:

```
Enter ? for a list of all sets.
Enter ?S for sorted ascending list of sets.
Enter ?D for sorted descending list of sets.
Press [Enter] to return to file prompt.
```

You can specify which Data Set to operate on by entering the Data Set name or number (as shown in the Help display).

When a DataBase is being operated on, DataWarp next prompts for the Data Item to be transformed:

```
(#01) Data Item to age, or <??>:
```

Typing '??' at this prompt yields the following display:

```
Enter ? for a list of all items.
Enter ?S for sorted ascending list of items.
Enter ?D for sorted descending list of items.
Enter ?string to list all items that contain 'string' in their name.
If you need to have a subset of the item then enter "ITEM(startbyte)".
Up to 99 independent Date Items may be specified.
Press [Enter] to return to DataSet prompt.
```

You can specify which Data Item to operate on by entering the Data Item name or number (as shown in the Help display).

The Data Item prompt includes a loop counter. Up to 99 different Data Items may be transformed in a single WARP DATA run. The program loops back to the Data Item prompt after you have answered the following Date Storage, Format and Offset prompts (see below).

You will next see a menu selection of supported date storage types. Select the option describing the date field that you wish to modify:

```
1. J2/I2/K2 (PIC S9(9) COMP)
2. Z6
3. Z8
4. X6
5. X8
Select Date Storage Type [1..5]:
```

Next you'll see a menu selection of supported date format types. Select the option describing the date format that you wish to modify:

```
1. YYMMDD
2. CCYYMMDD
3. MMDDYY
4. DDMMYY
5. CALENDAR
6. HPCALENDAR
7. YYMMDD date YY:MM3000 style year (2000 represented as 'A0')
8. MMDDYY date YY:MM3000 style year (2000 represented as 'A0')
9. DDMMYY date YY:MM3000 style year (2000 represented as 'A0')
Select Date Format Type [1..9]:
```

DataWarp verifies that the combination of date storage type and date format type are mutually compatible. Formats 7, 8 and 9 are provided to support the date format adopted by MM3000 and FM3000, where years after '99' are represented by the values 'A0' through 'Z9'.

Next you'll be prompted for the date offset to apply:

```
Enter Date Offset, or <??>:
```

DataWarp supports both absolute and relative values for aging. Refer to the end of this document for a detailed description of how different date types are applied.

Typing '??' at the Date Offset prompt shows the date offset formats accepted by DataWarp:

```
Enter +/-<number> YEARS | MONTHS | WEEKS | DAYS
YYYY/MM/DD          ... Absolute Date
YYYY/MM             ... Absolute Year and Month
YYYY                ... Absolute Year
```

The searching and sorting options allows you to easily look for fields within a large dataset. If you have a date that is embedded inside another field because of overloading, simply specify the byte offset, starting from one (1), of the date inside that particular item, as in FILLER-100(53).

DataWarp next prompts for a search value:

```
Data Item to search on, or <??> [none]:
```

Hit <Enter> if you don't want to limit DataWarp selection by applying a search value. If you wish to apply a search value for DataBase searching, specify the item name, optionally followed by a subset location. For MPE or KSAM files, enter the byte position and length of the search field.

For example: 13:8 specifies a date field 8 bytes long, starting at position 13 within the record.

See the section on How Searching Works for a detailed description.

At this stage, DataWarp returns to the 'Data Item' prompt, allowing another Data Item transformation to be specified. Hit <Enter> to signal that no further transformations are required.

Finally you will have the option to either Save, Save and Execute, Execute (no save), or Discard (no save, no execute) the script that you just created:

```
Select one of the following actions
1. Save
2. Save and Execute
3. Execute (no save)
4. Discard (no save, no execute)
Enter Option:
```

If you save the script, then you can use DataWarp in its Script-Driven mode at a later time to execute the actions you've just defined.

DataWarp Script-Driven mode

Script-Driven mode is accessed as follows:

```
:WARP DATA <ScriptFile>
```

where <ScriptFile> is the name of a previously-saved DataWarp script file.

There are a number of reasons why it may be valuable to save the DataWarp script. Firstly, it allows later (and repeated) execution. Saving the script also allows the transformation to be easily reversed. If you use a relative offset like +600 DAYS, then it is simple to go and edit the generated script file to -600 DAYS with all other criteria remaining the same, and thus reverse the date transformation.

How DataWarp Searching Works

The search capability within DataWarp allows control over the actual records within a DataSet or MPE file that are modified by DataWarp.

So if you had order records you wanted to test, you could take all records created on a particular date and push them forward into January 2000, and then select all records created on another date and push them forward to December 1999, and then test your application programs with a range that spanned those two months, thus crossing the century in your program logic.

Since many shops "overload" the image definitions of items, we also support the concept of "embedded" dates. This is where you have an image item named something like FILLER-100, and you have a date that you want to search on that starts at byte 31.

Let's assume that the date is stored as a standard ASCII string, then we could specify FILLER-100(31:6) as our search range, assuming a 6 character date.

Understanding how DataWarp Date Offsets Work

If you apply a transform with the intention of reversing the transformation at a later time and returning all date fields to their original values, you should specify the offset in terms of +/- DAYS or WEEKS only.

Any date transformations that are specified in terms of MONTHS or YEARS are subject to end-of-month rounding. While such transformations work, they generally can NOT be exactly reversed at a later time.

In other words, if you specify date offset of +6 MONTHS, don't expect that subsequently applying a transformation of -6 MONTHS will return all dates to their original values. A transformation of +183 DAYS, however, can be exactly reversed by applying a -183 DAYS transformation.

Why is this? Because of the uncertainty implied when taking a date like March 31 and transforming it by -1 MONTH. Do you wish to transform this date to February 28 (or 29), March 3 (or 2), or do you wish to generate an illegal date error?

Using absolute dates can also cause confusion because unless you used search criteria on the date field to restrict it to a particular value, all date fields will be set to the same value regardless of what their original values were, making later reversal impossible.

If an absolute date is used that does not specify the Month and/or Day, then DataWarp will default to the current Month and/or Day at run-time. This might be an issue if you create a DataWarp script and execute it at a later date.

DataWarp / MPE Compatibility

Internally, the DataWarp module relies on features that were added to MPE/iX release 5.5 express 4. Some of these features were 'broken' by HP in express 5, but were fixed again in express 6. All features work in MPE/iX release 6.0.