

# JUMP/3000

Updated 2026/04/01

JUMP/3000 is a new commercial product that allows HP3000 systems to continue running for a full 10 years past their current hard-coded end-of-life on 2027/12/31.

JUMP/3000 can be used by HP3000s running on HP hardware, as well as on Stromasys Charon virtual HP3000s.

## The 2027 Problem

The HP3000 operating system has a hard-coded limit beyond which the system date cannot be set: 2027/12/31. User programs typically make use of date-related intrinsic calls with the same limitation.

The limit affects systems running on HP hardware, as well as Stromasys Charon emulators. It's primarily due to the widespread use of the 'Calendar' intrinsic, which stores dates using a 16-bit format with inherent limits.

Currently the only way to continue using HP3000s after 2027 will be to set the system date to a prior value. This 'solution' is highly problematic, not only because of the complete disruption of all date-related processing, but also because it greatly complicates the near-universal practice of backing up systems using a combination of full and periodic incremental backups.

## The JUMP/3000 Solution

JUMP/3000 provides a full 10-year extension to the HP3000's lifespan. It allows the system date to seamlessly wrap from 2027/12/31 to 2028/01/01, and to continue running through 2037/12/31.

JUMP/3000 offers a totally transparent solution: MPE command interpreter (CI) commands and intrinsics are modified to work with the extended system date. Application programs and third-party utilities that rely on system intrinsics and language run-time support libraries will also keep working after the end of 2027.

Your HP3000 can be started up into any date through 2037/12/31; the system date can also be changed at any time using the :SETCLOCK command. File system timestamps will correctly reflect post-2027 system dates.

**JUMP/3000 provides the most complete solution for extending the life of your HP3000. It includes support for all date-related Intrinsics, CI commands and variables, and for the :STORE command's selection of files by access or modify date.**

**JUMP/3000 is the only post-2027 solution that is available for installation on your own HP3000 hardware, including Stromasys Charon HP3000 emulators<sup>1</sup>. It can be licensed annually, or through the end of any year between 2028 and 2037. It also doesn't force you to accept bundled hosting or support services.**

**JUMP/3000 is also available under an innovative Test license that provides great flexibility when preparing for post-2027 operations. It allows up to 50 different dates in the range 2028..2037 to be selected for testing, but imposes no limits on the *number* of times that you can use any particular test date.**

---

<sup>1</sup> Known variously as Charon-HPA/3000, Charon-PAR, Charon-PA3, and Charon-PAR/PA3.

# Implementation: Extending the Range of HP3000 Timestamps

Internally the HP3000 uses two primary mechanisms to keep track of dates:

- 1) Signed 64-bit timestamps that track microseconds-since-1970.  
This timestamp format is able to represent dates through the end of 2037 without change.
- 2) 16-bit 'Calendar' format timestamps, containing two sub-fields:
  - YEAR (years-since-1900) : 7 bits long, covering the range 0..127
  - DAY (day-of-year) : 9 bits long, covering the range 0..511

The YEAR field is able to represent years up to  $2^7$  relative to 1900, covering the range 1900 to 2027.

Changing the definition of this field allows it to represent years in the range 1910 to 2037:

Values in the range 10..127 continue to be interpreted as 'years since 1900'.

Values in the range 0..9 are now interpreted as 'years since 2028'.

Existing Calendar and 64-bit system timestamps (e.g. file creation and access timestamps) remain valid with this change.

## Implementation: HP3000 Patching Techniques

JUMP/3000 implements the modified Calendar format using several different techniques to allow it to intercept and modify all date references made by the MPE/iX operating system, and by all programs running on a system.

### Static Patch Component

Several low-level date system routines in the system library (NL.PUB.SYS) are patched to support the modified Calendar format, and to allow the system date to be set post-2027.

The patch is created using the same process used by HP when developing operating system patches.

Installing the static patches uses the standard method of creating a modified system load tape (SLT), and then updating the system from this tape<sup>2</sup>. JUMP/3000 static patches can also be also installed using Stage/iX.

### Dynamic Patch Component

Dynamic patching uses a low-level run-time interception method available on the HP3000. It uses an HP-supplied mechanism called Procedure Exits (also known as AIF:PE), and allows interception of date-related calls made by user code, without the risk and extensive work involved in patching the system library.

JUMP/3000 Dynamic patches are implemented without making any changes to library or program code. They are activated by running a supplied program once during system startup. Once activated they remain in effect until the system is next rebooted.

### NM and CM Code

The HP3000 supports two code environments. Native Mode (NM) code uses the native 32-bit system libraries, while Compatibility Mode (CM) supports the legacy 16-bit environment provided by the original 'Classic' HP3000 architecture. The static and dynamic patch components allow JUMP/3000 to intercept date references made from both NM and CM code.

---

<sup>2</sup> Stomasys Charon emulators support virtual tape drives; this is the preferred way to update these systems.

## Efficiency and Supportability

The combination of interception methods provides the simplest, lowest overhead, most reliable, and most comprehensive solution. JUMP/3000 Static patch replacement routines are highly efficient, and execute at the same speed (or faster) than the original date routines they replace.

JUMP/3000 Dynamic patches have a marginally higher run-time overhead, but are not used to intercept internal date calls made by the operating system. Their overhead is acceptable in view of their much lower typical usage.

JUMP/3000 is designed so that it has no visible effects on a system, other than enabling dates after 2027 to be used. It can also be installed on systems prior to 2028, with no problem.

## Specifying Post-2027 Dates During System Startup

JUMP/3000 allows post-2027 dates to be specified during system startup:

```
Initialize_genesis - Ver bld1: <<pci 2.1601>>
Initialize_genesis - Relocating pdc...
Initialize_genesis - Relocate pdc from 0xffff800000 to 0x9443000 completed.
THU, FEB 13, 2025, 4:01:04 PM (y/n)? N
Date (M/D/Y)? 02/13/37
Time (H:M:S)? 16:01
FRI, FEB 13, 2037, 4:01:00 PM (y/n)? Y
Initialize_genesis = Ready to configure I/O
```

The year can also be specified using 4 digits.

## Demonstrating JUMP/3000 Features

JUMP/3000 has a built-in demonstration mode that exercises many date-related MPE commands, programs, and intrinsics. This can be used to quickly verify all functionality.

Demo mode output is shown below – it shows annotated examples of MPE commands and error messages that have been modified to function post-2027. HP3000 Command Interpreter (CI) commands are highlighted. It also includes examples of CM and NM intrinsic calls, and running CM and NM programs.

Demo mode shows the real, unmodified output of these commands and calls; it is not a simulation.

```
:jump demo

JUMP DEMO will temporarily set your system date to 2036.

OK to continue? (y/N): y

JUMP DEMO (c) Paul Taffel Consulting (PTC), 2025

:JUMP STATUS
JUMP traps enabled

:SHOWTIME
SUN, MAY 25, 2025, 9:44 AM
```

```
SETCLOCK with out-of-range year:
:SETCLOCK DATE=04/06/2039; TIME=12:34; NOW
The specified DATE is out of range. Max year is 2037. (CIERR 11711)
```

```
:SETCLOCK DATE=04/06/2036; TIME=12:34; NOW
```

```
:SHOWCLOCK
```

```
SYSTEM TIME: SUN, APR 6, 2036, 12:34:06 PM
CURRENT TIME CORRECTION: 0 SECONDS
TIME ZONE: 8 HOURS 0 MINUTES WESTERN HEMISPHERE
```

```
:SHOWTIME
```

```
SUN, APR 6, 2036, 12:34 PM
```

```
:SHOWME
```

```
USER: #S23,MGR.DEV,JUMP (IN PROGRAM)
RELEASE: C.75.00 MPE/iX HP31900 C.45.05 USER VERSION: C.75.00
CURRENT: SUN, APR 6, 2036, 12:34 PM
LOGON: SUN, MAY 25, 2025, 9:43 AM
CPU SECONDS: 1 CONNECT MINUTES: 1
$STDIN LDEV: 4 $STDLIST LDEV: 4
```

```
:RUN SHOWCLKS.PUBXL.TELESUP
```

```
SHOWCLKS/XL A.10.00
```

```
*****
*** Greenwich Mean Time : SUN, APR 6, 2036, 7:34 PM ***
*** GMT/MPE offset : -7:00:00 ***
*** MPE System Time : SUN, APR 6, 2036, 12:34 PM ***
***
*****
```

```
*** C Library Information ***
```

```
Current value of Time Zone(TZ) variable : PST8PDT
CTIME function return : Sun Apr 6 12:34:07 2036
```

```
:SHOWVAR HPDATE@, HPDA@, HPY@, HPMO@, HP@YEAR
```

```
HPDATE = 6
HPDATEF = SUN, APR 6, 2036
HPDATETIME = 20360406123407500
HPDAY = 2
HPLEAPYEAR = TRUE
HPMONTH = 4
HPSPLITYEAR = 50
HPYEAR = 36
HPYYYY = 2036
HPYYYYMMDD = 20360406
```

```
:BUILD TESTFILE
```

```
:CALC FINFO ("TESTFILE","CREATED")
```

```
SUN, APR 6, 2036
```

```
:CALC FINFO ("TESTFILE","INTCREATED")
```

```
20360406, $136ACD6, %115526326
```

**:LISTF TESTFILE,3**

\*\*\*\*\*

FILE: TESTFILE.JUMP.DEV

```
FILE CODE : 0                FOPTIONS: BINARY, FIXED, NOCCTL, STD
BLK FACTOR: 1                CREATOR : **
REC SIZE: 256(BYTES)         LOCKWORD: **
BLK SIZE: 256(BYTES)         SECURITY--READ   : ANY
EXT SIZE: 0(SECT)           WRITE           : ANY
NUM REC: 0                   APPEND          : ANY
NUM SEC: 0                   LOCK            : ANY
NUM EXT: 0                   EXECUTE         : ANY
MAX REC: 1023                **SECURITY IS ON
                               FLAGS : NO ACCESSORS
NUM LABELS: 0                CREATED : SUN, APR 6, 2036, 12:34 PM
MAX LABELS: 0                MODIFIED: SUN, APR 6, 2036, 12:34 PM
DISC DEV #: 2                ACCESSED: SUN, APR 6, 2036, 12:34 PM
SEC OFFSET: 0                LABEL ADDR: **
VOLCLASS : MPEXL_SYSTEM_VOLUME_SET:DISC
```

**:STORE TEST@; \*T; SHOW=DATES,SHORT; DATES>=01/04/2036**

```
>> TURBO-STORE/RESTORE VERSION C.75.09 B5152AA <<
(C) 1986 HEWLETT-PACKARD CO.
```

STORE TEST@; \*T; SHOW=DATES,SHORT; DATES>=01/04/2036

SUN, APR 6, 2036, 12:34 PM

FILENAME	GROUP	ACCOUNT	VOLUME	RESTRICTIONS	SECTORS	CODE	MEDIA
		CREATED	ACCESSED	MODIFIED			
TESTFILE.JUMP	.DEV	DISC			:C	0	1
		4/ 6/2036	4/ 6/2036	4/ 6/2036			

FILES STORED : 1

TOTAL MEDIA WRITTEN : 1

**STORE with out-of-range year:**

**:STORE TEST@; \*T; DATES>=01/04/2040**

```
>> TURBO-STORE/RESTORE VERSION C.75.09 B5152AA <<
(C) 1986 HEWLETT-PACKARD CO.
```

STORE TEST@; \*T; DATES>=01/04/2040

SUN, APR 6, 2036, 12:34 PM

STORE TEST@; \*T; DATES>=01/04/2040

^

YEAR PORTION OF DATE IS INVALID. IT MUST BE GREATER  
THAN 1970 (70) AND LESS THAN 2037 (37).  
DATE SYNTAX 'mm/dd/yy[yy]' (S/R 597)

STORE aborted because of error. (CIERR 1090)

```
:STREAM xjob; DAY=SUNDAY ;AT=23:59
#J95
```

```
:STREAM xjob; DATE= 05/22/2036 ;AT=23:59
#J96
```

```
STREAM on the last legal date in 2037:
:STREAM xjob; IN=634
#J97
```

```
:CALC JINFO (HPLASTJOB, "FMTSTREAMEDBYDATE")
SUN, APR 6, 2036
```

```
:CALC JINFO (HPLASTJOB, "FMTINTRODATE")
THU, DEC 31, 2037
```

```
:CALC JINFO (HPLASTJOB, "INTRODATE")
20371231, $136D71F, %115553437
```

```
STREAM with past date:
:STREAM xjob; DATE=1/1/33
```

Expected a number between the current year and 2037 for the year specification. (CIERR 6334)

```
STREAM with out-of-range year:
:STREAM xjob; DATE=01/31/2040
```

Expected a number between the current year and 2037 for the year specification. (CIERR 6334)

```
STREAM with out-of-range ;IN= days:
:STREAM xjob; IN=635
```

Job cannot be scheduled beyond December 31, 2037. (CIERR 6346)

```
:SHOWJOB JOB=xjob,mgr.jump; SCHED
```

```
CURRENT: 4/ 6/36 12:34
```

JOBNUM	STATE	IPRI	JIN	JLIST	SCHEDULED-INTRO	JOB NAME
#J95	SCHED	13	10S	LP	4/12/36 23:59	XJOB,MGR.JUMP
#J96	SCHED	13	10S	LP	5/22/36 23:59	XJOB,MGR.JUMP
#J97	SCHED	13	10S	LP	12/31/37 12:34	XJOB,MGR.JUMP

```
3 JOBS (DISPLAYED):
JOBFENCE= 1; JLIMIT= 8; SLIMIT= 8
```

:LISTSPF @; SELEQ=[DATE>=04/01/2036 AND DATE<05/01/2036]; DETAIL

SPOOLID	JOBNUM	FILEDES	PRI	COPIES	DEV	STATE	RSPFN	OWNER	TIME
	FORMID	JOBNAME		COPSRM	SECTS	RECS	PAGES	DATE	
#I97	J97	\$STDIN			10	READY	P	MGR.JUMP	
		XJOB			16	2		04/06/36	12:34
#I96	J96	\$STDIN			10	READY	P	MGR.JUMP	
		XJOB			16	2		04/06/36	12:34
#I95	J95	\$STDIN			10	READY	P	MGR.JUMP	
		XJOB			16	2		04/06/36	12:34

INPUT SPOOL FILES

ACTIVE = 0;  
OPEN = 0;  
READY = 3;

OUTPUT SPOOL FILES

CREATE = 0;                   READY = 0;  
DEFER = 0;                   SELECTED = 0;  
DELPND = 0;                  SPSAVE = 0;  
PRINT = 0;                   XFER = 0;  
PROBLM = 0;

TOTAL IN FILES = 3;  
IN SECTORS = 48;

TOTAL OUT FILES = 0;  
OUT SECTORS = 0;

OUTFENCE = 14

:STREAM xjob  
#J98

:SPOOLF @; SELEQ=[STATE=READY AND DATE=04/06/36]; DELETE; SHOW

SPOOLID	JOBNUM	FILEDES	PRI	COPIES	DEV	STATE	RSPFN	OWNER
#011197	J98	\$STDLIST	1	1	LP	DELPND		MGR.JUMP

Sample CM and NM programs:

:RUN EDITOR.PUB.SYS

HP32201A.09.00 EDIT/3000 SUN, APR 6, 2036, 12:34 PM  
(C) HEWLETT-PACKARD CO. 1993

:RUN TDP.PUB.SYS

TDP/V (A.05.06) HP36578 Editor (c) COPYRIGHT Hewlett-Packard Co. 1998  
SUN, APR 6, 2036, 12:34 PM (DAY #97 )

:RUN QUERYCM.PUB.SYS

HP32216D.03.17 QUERY/3000 SUN, APR 6, 2036, 12:34 PM  
COPYRIGHT HEWLETT-PACKARD CO. 1976

:RUN QUERNM.PUB.SYS

HP32216D.03.21 QUERY/NM SUN, APR 6, 2036, 12:34 PM  
COPYRIGHT HEWLETT-PACKARD CO. 1976

:RUN PASCALXL.PUB.SYS

PAGE 1 HP PASCAL/iX HP31502A.05.05 COPYRIGHT HEWLETT-PACKARD CO. 2002  
SUN, APR 6, 2036, 12:34 PM

:RUN /SYS/HPBIN/DATE

Sun Apr 6 13:34:13 PDT 2036

### Sample NM intrinsic calls:

```
AIFTIME      : 2036/04/06                ;MONTH:APR DOW:SUN
ALMANAC      : 136/04/06                ;1900 +136 = 2036
calclock     : SUN, APR 6, 2036, 12:34 PM
CALENDAR     : YEAR: 8 DAY: 97          ;1900 +8 +128 = 2036
DATELINE     : SUN, APR 6, 2036, 12:34 PM
FMTCALENDAR  : SUN, APR 6, 2036
FMTDATE      : SUN, APR 6, 2036, 12:34 PM
HPCALENDAR   : YEAR:136 DAY: 97         ;1900 +136 = 2036
HPDATECONVERT : 20360406                ;mode 14 => mode 38
HPDATECONVERT : SUN, APR 6, 2036       ;mode 4 => mode 14
HPDATECONVERT : D60406                  ;mode 4 => mode 35 (YY:MM3000)
HPDATEDIFF   : 2036/04/06 -> 2035/04/07 ;mode 14; diff -365
HPDATEFORMAT : 2036-04-06               ;mode 14, spec YYYY-MM-DD
HPDATEOFFSET : MON, APR 6, 2037         ;mode 14; offset +365 days
HPDATEVALIDATE: SUN, APR 6, 2036        ;mode 14; Valid
HPFMTCALENDAR : SUN, APR 6, 2036
NLCONVCUSTDATE: YEAR:125 DAY: 96        ;04/06/25: 1900 +125 = 2025
NLCONVCUSTDATE: YEAR: 9 DAY: 96         ;04/06/37: 1900 +9 +128 = 2037
NLCONVCUSTDATE: YEAR: 70 DAY: 96        ;04/06/70: 1900 +70 = 1970
NLFMTCALENDAR : SUN, APR 6, 2036
NLFMTCUSTDATE : 04/06/36                ;mm/dd/yy
NLFMTDATE    : SUN, APR 6, 2036, 12:34 PM ;DDD, MMM dd, yyyy
NLFMTLONGCAL : SUNDAY, APRIL 6, 2036    ;WWWWWWWWWWWW, 000000000000 dd, yyyy
```

### Sample CM intrinsic calls:

```
ALMANAC      : 136/04/06                ;1900 +136 = 2036
CALENDAR     : YEAR: 8 DAY: 97          ;1900 +8 +128 = 2036
CONVERTDATE  : YEAR:125 DAY: 96        ;04/06/25: 1900 +125 = 2025
CONVERTDATE  : YEAR: 9 DAY: 96         ;04/06/37: 1900 +9 +128 = 2037
CONVERTDATE  : YEAR: 70 DAY: 96        ;04/06/70: 1900 +70 = 1970
DATELINE     : SUN, APR 6, 2036, 12:34 PM
DATE'LINE    : SUN, APR 6, 2036, 12:34 PM
FMTCALENDAR  : SUN, APR 6, 2036
FMTDATE      : SUN, APR 6, 2036, 12:34 PM
NLCONVCUSTDATE: YEAR:125 DAY: 96        ;04/06/25: 1900 +125 = 2025
NLCONVCUSTDATE: YEAR: 9 DAY: 96         ;04/06/37: 1900 +9 +128 = 2037
NLCONVCUSTDATE: YEAR: 70 DAY: 96        ;04/06/70: 1900 +70 = 1970
NLFMTCALENDAR : SUN, APR 6, 2036
NLFMTCUSTDATE : 04/06/36                ;mm/dd/yy
NLFMTDATE    : SUN, APR 6, 2036, 12:34 PM ;DDD, MMM dd, yyyy
NLFMTLONGCAL : SUNDAY, APRIL 6, 2036    ;WWWWWWWWWWWW, 000000000000 dd, yyyy
```

### NM COBOL date constructs:

```
ACCEPT FROM DAY      : 36097
ACCEPT FROM DATE     : 360406
MOVE CURRENT-DATE TO : 04/06/36
MOVE FUNCTION CURRENT-DATE TO : 2036040613342400-0700
```

### Reset system date to original value:

```
:SETCLOCK DATE=05/25/2025; TIME=09:56; NOW
```

### :SHOWTIME

```
SUN, MAY 25, 2025, 9:56 AM
```

## The :JDATE Command

The :JDATE command is supplied to facilitate moving the system date when testing. It has a simpler and more flexible interface than the :SETCLOCK command.

JDATE allows the year, month, and/or date to be changed without also having to specify the full date and time every time. Year, month, and date can be specified in absolute or relative terms.

```
:jdate ?
```

```
Syntax:
```

```
:JDATE [[year] [, [month] [, date]]]
```

```
Specify absolute values, or relative values with +/- prefix
```

JDATE with no parameters returns to the initial date, before :JDATE was first used in your job/session.

JDATE is implemented as a command file. It uses the :SETCLOCK command with ;NOW specified, and preserves the current system time, rounded to the nearest second.

```
:showtime
```

```
MON, AUG 18, 2025, 11:18 AM
```

```
:jdate +10
```

```
:SETCLOCK DATE=8/18/2035;TIME=11:18:37;NOW
```

```
Date changed to 8/18/2035
```

```
SAT, AUG 18, 2035, 11:18 AM
```

```
:jdate -20
```

```
:SETCLOCK DATE=8/18/2015;TIME=11:18:45;NOW
```

```
Date changed to 8/18/2015
```

```
TUE, AUG 18, 2015, 11:18 AM
```

```
:jdate 2037
```

```
:SETCLOCK DATE=8/18/2037;TIME=11:18:55;NOW
```

```
Date changed to 8/18/2037
```

```
TUE, AUG 18, 2037, 11:19 AM
```

```
:jdate 28
```

```
:SETCLOCK DATE=8/18/2028;TIME=11:19:4;NOW
```

```
Date changed to 8/18/2028
```

```
FRI, AUG 18, 2028, 11:19 AM
```

```
:jdate
```

```
:SETCLOCK DATE=8/18/2025;TIME=11:19:14;NOW
```

```
Date reset to 8/18/2025
```

```
MON, AUG 18, 2025, 11:19 AM
```

JDATE can also be used to change the month and/or date. If you attempt to change the date to an illegal value the command will fail with :SETCLOCK's generic error message.

```
:jdate , +2, 15
:SETCLOCK DATE=10/15/2025;TIME=11:19:24;NOW

Date changed to 10/15/2025
WED, OCT 15, 2025, 11:19 AM

:jdate 2035, 2, 30
:SETCLOCK DATE=2/30/2035;TIME=11:19:32;NOW
The specified DATE is out of range. Max year is 2037. (CIERR 11711)

WED, OCT 15, 2025, 11:19 AM
```

## Rolling from 2027 to 2028

JUMP/3000 allows the system clock to smoothly roll over from 2027/12/31 to 2028/01/01. This display shows the output of a :WHILE loop that calls :SHOWCLOCK, followed by a short pause. It shows just how uneventful this transition will be when using JUMP/3000.

```
SYSTEM TIME: FRI, DEC 31, 2027, 11:59:11 PM
CURRENT TIME CORRECTION:          0 SECONDS
TIME ZONE:      8 HOURS  0 MINUTES WESTERN HEMISPHERE

SYSTEM TIME: FRI, DEC 31, 2027, 11:59:27 PM
CURRENT TIME CORRECTION:          0 SECONDS
TIME ZONE:      8 HOURS  0 MINUTES WESTERN HEMISPHERE

SYSTEM TIME: FRI, DEC 31, 2027, 11:59:44 PM
CURRENT TIME CORRECTION:          0 SECONDS
TIME ZONE:      8 HOURS  0 MINUTES WESTERN HEMISPHERE

SYSTEM TIME: SAT, JAN  1, 2028,  0:00:01 AM
CURRENT TIME CORRECTION:          0 SECONDS
TIME ZONE:      8 HOURS  0 MINUTES WESTERN HEMISPHERE

SYSTEM TIME: SAT, JAN  1, 2028,  0:00:19 AM
CURRENT TIME CORRECTION:          0 SECONDS
TIME ZONE:      8 HOURS  0 MINUTES WESTERN HEMISPHERE

SYSTEM TIME: SAT, JAN  1, 2028,  0:00:37 AM
CURRENT TIME CORRECTION:          0 SECONDS
TIME ZONE:      8 HOURS  0 MINUTES WESTERN HEMISPHERE
```

## Native Language Support

JUMP/3000 implements full Native Language support, using the same configuration settings used by MPE. NLS settings affect the NLxxxx intrinsics, and a small number of MPE commands, such as :SHOWTIME.

This extract shows how :JUMP DEMO adapts when CANADIAN-FRENCH has been configured:

```
Sample CM intrinsic calls (using CANADIAN-FRENCH formats):
ALMANAC      : 136/04/06                ;1900 +136 = 2036
CALENDAR     : YEAR: 8 DAY: 97          ;1900 +8 +128 = 2036
CONVERTDATE  : YEAR:125 DAY: 96         ;04/06/25: 1900 +125 = 2025
CONVERTDATE  : YEAR: 9 DAY: 96         ;04/06/37: 1900 +9 +128 = 2037
CONVERTDATE  : YEAR: 70 DAY: 96        ;04/06/70: 1900 +70 = 1970
DATELINE     : SUN, APR 6, 2036, 12:34 PM
DATE'LINE    : SUN, APR 6, 2036, 12:34 PM
FMTCALENDAR  : SUN, APR 6, 2036
FMTDATE      : SUN, APR 6, 2036, 12:34 PM
NLCONVCUSTDATE: YEAR:125 DAY: 96       ;2025-04-06: 1900 +125 = 2025
NLCONVCUSTDATE: YEAR: 9 DAY: 96       ;2037-04-06: 1900 +9 +128 = 2037
NLCONVCUSTDATE: YEAR: 70 DAY: 96     ;1970-04-06: 1900 +70 = 1970
NLFMTCALENDAR : dim, 6 avr 2036
NLFMTCUSTDATE : 2036-04-06           ;yyyy-mm-dd
NLFMTDATE     : dim, 6 avr 2036, 12h34 ;DDD, dd MMMM yyyy, 24h
NLFMTLONGCAL  : dimanche, Le 6 avril 2036 ;WWWWWWWWW, Le dd 000000000 yyyy

Reset system date to original value:
:SETCLOCK DATE=09/30/2025; TIME=13:29; NOW

:SHOWTIME
mar, 30 sept 2025, 13h29
```

JUMP/3000 has been tested with ENGLISH, SPANISH, FRENCH, GERMAN, and CANADIAN-FRENCH formats.

## Technical Limitations and Caveats

Modifying the HP3000 operating system to handle post-2027 dates is a significant part of keeping your system running, but it does not guarantee that every user program and third-party product will be usable without modification post-2027.

Programs can have hard-coded date limits; they may also directly manipulate Calendar timestamps and their sub-fields. If they're not aware of the change to the Calendar format, they could interpret years in the range 2028..2037 as 1900..1909.

Unmodified software could interpret a Calendar field containing Year=9 as 1909 or possibly 2009, instead of the correct 2037 value.

Programs that sort or compare dates by treating the Calendar timestamp as a 16-bit integer will give incorrect results when comparing years in the range 2028..2037 with dates prior to 2028.

Even though JUMP's post-2027 patches are not a panacea, it does provide a stable platform for an HP3000 to be used post-2027. **Users will need to carefully test their programs and third-party applications to verify their correct functioning, and be prepared to make modifications if necessary.**

It is possible that testing may reveal some aspects of the operating system where dates are incorrectly formatted post-2027. We are not able to give a 100% guarantee that every possible command and programming interface will work correctly post-2027.

We are aware of the following situations where post-2027 dates are currently not handled correctly:

- When labelled tapes have expiration dates defined, overwrite protection may not be processed correctly.

## Appropriate Expectations

**It is crucial that you have appropriate expectations before using JUMP/3000.** You must be prepared to perform comprehensive testing of all in-house and third-party software to verify correct functioning post-2027, and to make changes if necessary.

**It is possible you will encounter problems with some third-party utilities; you may need to obtain updated versions of these utilities, or be prepared to work-around problems using these utilities.**

Testing will require temporarily setting the system date to post-2027 values. This will obviously disrupt routine system operations. Ideally you should have access to a test system.

## Third-Party Products

JUMP/3000 has been successfully tested with Adager, and with all Robelle and Vesoft products.

Current releases of Adager and Robelle products already handle dates after 2027. Vesoft products have now been modified to fully support JUMP/3000; contact Vesoft to obtain updated versions of their products.

## Supported MPE/iX Versions

JUMP/3000 has been developed and tested using MPE/iX 7.5. It works identically on HP3000s running on HP hardware, and on systems using the Stomasys CHARON emulator.

Both static and dynamic JUMP/3000 patch components are designed to be largely independent of the MPE/iX version. JUMP/3000 has also been confirmed to work under MPE/iX 6.5.

## Developer Credentials

JUMP/3000 is the work of an HP3000 developer of long-standing. He worked as a software developer for several prominent HP3000 third-party vendors, including Bradmark, Vesoft, Quest, Orbit, and Informatica. He has extensive experience using dynamic patching techniques to modify HP3000 program and operating system behavior. Products developed by him were at one time installed on several tens of thousands of HP3000 sites.

He created the TimeWarp virtual date testing product in 1999. TimeWarp simplified testing system and application behavior post-Y2K, when this was a significant issue. TimeWarp intercepted many of the same operating system date routines that are used by JUMP/3000, and formed the basis for initial work on this project.

As the Stomasys HP3000 Business Manager during the development of Charon-HPA/3000, he gained extensive experience with the Stomasys emulator.

He remains active as a developer, and will continue supporting JUMP/3000 in the future.

Contact: [ptaffel@gmail.com](mailto:ptaffel@gmail.com)

Resume: <https://ptaffel.com/Detail>