

## Read Before Installing

---

### Summary (HP Part No. 30216-90247R3748)

This packet contains the following information related to Express 4 PowerPatch (C.55.04) based on MPE/iX Release 5.5:

- Information You Need to Know Before Installing
- Other Information for Hardware and Software Products
- Disk Space Requirements
- Product Tables and Dependent Patches
- New Communicator 3000 Articles
  - Year 2000 Enhancements for MPE/iX
  - New Date Intrinsic for MPE/iX
  - COBOL II/iX Enhancements
  - Inform/V
  - HP ALLBASE/BRW

---

### Information You Need to Know Before Installing

The following articles are included in this section:

- Support for the HP 3000 Corporate Business Server 997
- HP Patch/iX
- HP Stage/iX
- For Experienced Installers Only who are Running AUTOINST
- Resolving Busy XL.PRED.SYS
- A3716A 144 GB DDS-3 Autoloader for MPE/iX
- New Disk Devices Supported
- 9 GB Disk Drive—Special Installation Instructions
- Using SQLINSTL for ALLBASE/SQL
- Handling Pseudo-Mapped Files for ALLBASE/SQL
- TurboSTORE II or TurboSTORE/iX 7x24—Special Installation Instructions

#### Support for the HP 3000 Corporate Business Server 997

This release contains the necessary support for the HP 3000 Corporate Business Server 997, for both new 997 servers as well as upgrades to the 997 from previous 99x servers. The 997 uses the PA-8000 processor. Refer to the *HP 3000 Corporate Business Server 997 (1 to 5 Way) Technical Data Sheet (5965-8426E)* for more information.

The 997 server is not a customer installable product. Customer engineers should refer to the 997 installation manual for details, and in addition for upgrades, refer to the software procedure provided with the 997 upgrade package.

## HP Patch/iX

HP Patch/iX is available with the Fundamental Operating System (FOS) software. HP recommends using HP Patch/iX to customize and install HP supported patches for these reasons:

- Provides qualification of all patches applied to the system.
- Reduces the likelihood of replacing a previously installed patch with an older patch.
- Saves as much as three hours per system by eliminating a system UPDATE/REBOOT.
- Allows for using Stage/iX to quickly back out a patch when Patch/iX was used to install the patch.

The new version of Patch/iX has an “info” option to determine the version of Patch/iX. The version is B.00.10 for this Express 4 release. Use the following command to display the version:

```
:patchix;info=version
```

### Installation of Enhancement Patches

When Patch/iX detects patches which are enhancements rather than “must install,” it marks these patches as NOT QUALIFIED. To install these patches, you must FORCE these patches.

### Patch Disqualification Due to Checksum Mismatch

Patch/iX may disqualify a patch as “Naturally Disqualified” due to checksum mismatches. In this case, the patch may be FORCED after consulting with the Response Center.

For detailed information on using HP Patch/iX, refer to the *HP 3000 MPE/iX System Software Maintenance Manual Release 5.5* (30216-90223R3715).

## HP Stage/iX

HP Stage/iX is available with the FOS software. Use Stage/iX to place PowerPatch and/or reactive patches into staging areas on disk while the system is up, then choose a staging area to use at boot time to apply the patches. You can back out at any time through a reboot to the Base operating system. Stage/iX is targeted for larger HP3000 systems with adequate extra disk space on LDEV1.

For detailed information on using HP Stage/iX, refer to the *HP 3000 MPE/iX System Software Maintenance Manual Release 5.5* (30216-90223R3715).

### For Experienced Installers Only who are Running AUTOINST

This is only for experienced system installers who elect to use AUTOINST to install PowerPatches. Start with the normal checklist for your installation, then use alternate steps on page 4-21 of the *HP 3000 MPE/iX System Software Maintenance Manual Release 5.5* (30216-90223R3715). In the alternate steps on page 4-21, after “Restarting the System’ on page 5-44” insert the step, “Rerunning AUTOINST’ on page 5-48.”

## Resolving Busy XL.PRED.SYS

This is the resolution to a problem that may cause the installation process to terminate in an error state when AUTOINST attempts to restore from the CSLT tape. It is caused by the file XL.PRED.SYS being "busy."

After you complete the update with the CSLT tape, and after booting the system from disk, use LISTF,2 to check the status of XL.PRED.SYS:

```
:LISTF XL.PRED.SYS,2
```

If the file is busy (there is an \* to the right of the filename), you must abort the Predictive Support Monitor.

To abort the Predictive Support Monitor:

1. Run the Online Diagnostic System:

```
:SYSDIAG
```

2. At the DUI prompt, list the active processes.

```
DUI>DS SHOWACTIVE
```

3. Find the PID number of the PSMON process and abort it. For example:

```
CURRENTLY ACTIVE DIAGNOSTIC SYSTEM PROCESSES
```

Program name	PID
MEMLOGP	48
PSMON	49

```
DUI>DS ABORT 49
```

```
The requested diagnostic system process has been aborted.
```

```
DUI>E
```

```
END OF PROGRAM
```

This will terminate PSMON, and XL.PRED.SYS should no longer be busy.

4. Confirm that it is not busy by issuing the LISTF,2 command again.
5. Restart the AUTOINST tool, and type Y to continue.

## A3716A 144 GB DDS-3 Autoloader for MPE/iX

The A3716A is a high-speed, high-capacity tape product combined with an autoloader. The tape drive conforms to the DDS-3 standard. The autoloader houses six cartridges in the magazine. The DDS-3 tape allows 12 GB of uncompressed data to be stored on 125m DDS-3 tapes. It ensures backward compatibility in reading and writing existing DDS tapes. The sustained data rate on uncompressed data tape is 1 MBPS.

The A3716A is a single-ended SCSI-2 Device. The A3716A is supported on MPE/iX Release 5.5 with the following patches which are included in Express 4 (C.55.04):

- ODIJXJ0 (or later)
- MPEJXP9 (or later)

---

**Note**            The A3716A is NOT supported on MPE/iX 5.0, and there are currently no plans for 5.0 support of this device.

---

The A3716A is supported only on MPE/iX NIO platform. It is not supported on the MPE/iX CIO platform. The A3716A can be configured on single-ended NIO SCSI card. The autoloader option switch settings should be set to 6 (Stacker mode). Other mode settings are NOT supported.

### Example

On an NIO machine, Path 52 is configured as follows:

```
io>lp 52
PATH: 52                LDEV:
ID: A1703-60003-SCSI    TYPE: DA
PMGR: SCSI_DAM         PMGRPRI: 6
LMGR:                  MAXIOS: 0
```

If the SCSI ID of the device is 3, and path 52.3 is not configured:

```
io>addpath path=52.3 id=pseudo
io>ad ldev=31 path=52.3.0 id=hpc1557A
```

Verify the configuration using the LDEV and LP commands.

## New Disk Devices Supported

New 2 GB and 4 GB disk drives are supported starting with MPE/iX 5.5 Express 3. For product numbers refer to the table, "Peripherals and Dependent Patches," later in this *Read Before Installing*.

---

**Note** It is critical that you connect the disk drive to the proper channel. A single-ended disk drive should be connected to only a single-ended SCSI DAM. A fast/wide differential disk drive should be connected to only a FWSCSI\_DAM.

---

The single-ended narrow and fast-wide mass storage enclosures modules report the same product ID suffix of "WC"—not "N" or "WD." For example, if you run mapper on the new 2 GB SE narrow mass storage enclosures disk, you get the product ID of ST32272WC. However, use ST32272N to configure your disk in SYSGEN. For the correct configurations to use in SYSGEN, refer to the table in the article, "New Disk Devices Supported," in the *Communicator 3000 MPE/iX—Express 3 (PowerPatch Tape C.55.03) Based on Release 5.5 (30216-90245)*.

## 9 GB Disk Drive—Special Installation Instructions

The 9 Gigabyte (GB) single-ended or fast-wide disk drive is supported starting with C.55.02 PowerPatch. Before installing this drive, you should be aware of the following considerations:

- The patch must be installed **prior** to connecting the 9 GB disk drive to the system.
- The 9 GB drive is not supported as LDEV 1.

This drive is the largest single mechanism disk drive from Seagate and Quantum. The 9 GB drive has the same performance and drive characteristics as the recently introduced 2 GB and 4 GB drives from Seagate and Quantum. With this new large capacity 9 GB disk drive, MPE/iX will cross an Operating System limitation in the Secondary Storage Manager (SSM). An SSM patch is required to address the full 9 GB range. This disk drive is not supported on systems still on release 5.0.

Following is a short checklist to ensure that the 9 GB disk drive installation is a success.

### 9 GB Disk Drive Installation Checklist

- Complete the entire PowerPatch installation process including the system UPDATE from the patched CSLT created during the process.
- Power-down the system and connect the 9 GB disk drive with the appropriate SCSI ID and disk drive configurations.
- Power-on the system and boot the system with the appropriate ISL START command options.
- Add the new 9 GB disk drive to the system configuration using SYSGEN.
- Reboot the system one more time, and the drive will be available.
- Run VOLUTIL to prepare the drive for use in a volume set.

## Using SQLINSTL for ALLBASE/SQL

If you are updating from an earlier release of ALLBASE/SQL, you must perform the ALLBASE/SQL migration using SQLINSTL or SQLMigrate. Use the SQLINSTL script to migrate between versions of a release (such as from G1.14 to G1.15) or minor releases (such as from G.1 to G.2). Use SQLMigrate to convert the DBEnvironment between major releases such as from F.0 to G.0.

For more information on SQLMigrate, refer to Appendix A, "Manually Installed Products," in the *System Software Maintenance Manual Release 5.5* (30216-90223R3715) or the *ALLBASE/SQL Database Administration Guide* (36216-90005).

You must use the SQLINSTL script whenever a new version of ALLBASE/SQL (G.1 or later) is installed. Using SQLINSTL ensures the most recent version of the System and Catalog views, and it also uses VALIDATE FORCE statements to revalidate all stored sections to be compatible with the new release. If SQLINSTL is not executed, errors may result when stored sections are executed due to compatibility problems.

---

**Warning**      **SQLINSTL or SQLMigrate drop all system catalog views. If any view has been created upon a system catalog view, that view will also be dropped. Before executing SQLINSTL, use the GENERATE VIEWS command in SQLGEN to create a script to recreate the user view and avoid dropping views.**

---

Read the SQLINSTL.PUB.SYS file on your system for the command to execute it.

If you are using ARCHIVE MODE LOGGING, you must make a backup of the DBEnvironment after using SQLINSTL. This backup must be used if rollforward recovery is to be performed at some time in the future.

## Handling Pseudo-Mapped Files for ALLBASE/SQL

---

**Note**            ALLBASE/SQL no longer supports pseudo-mapped files. If you are migrating to a new version, you must convert all pseudo-mapped DBEFiles to mapped DBEFiles before you upgrade.

---

Use the SQLUtil SHOWACCESS command to determine if any files are pseudo-mapped. Then use the SQLUtil MOVEFILE command to convert each pseudo-mapped file to mapped. For more information, refer to the *Communicator 3000 MPE/iX—Express 3 (PowerPatch Tape C.55.03) Based on Release 5.5* (30216-90245).

**TurboSTORE II or TurboSTORE/iX 7x24—Special Installation Instructions**

This information is only for TurboSTORE/iX II or TurboSTORE/iX 7x24 customers who did not install MPE/iX 5.5 Express 2 or Express 3.

Users of TurboSTORE/iX II or TurboSTORE/iX 7x24 should have installed the PowerPatch and SUBSYS (or the appropriate XL) from the MPE/iX 5.5 Express 2 or Express 3 (C.55.02 or C.55.03) SUBSYS. However, if you did not do this, you may instead install MPE/iX 5.5 Express 4 PowerPatch and the SUBSYS from Express 3. Failure to install the SUBSYS from Express 3 (or the appropriate XL) will remove TurboSTORE/iX II or TurboSTORE/iX 7x24 from your system, leaving you with FOS STORE functionality.

If you do not want to install the entire SUBSYS from Express 3, you may install only the appropriate XL from the SUBSYS tape for Express 3. Users of TurboSTORE II (B5151AA) will install STORXL51, and users of TurboSTORE/iX 7x24 (B5152AA) will install STORXL52 using the following procedure.

Logon as MANAGER.SYS and issue the following commands:

```
:NEWGROUP USL
:FILE TAPE;DEV=TAPE
:RESTORE *TAPE;STORXL5#.USL.SYS;SHOW
:COPY STORXL51.USL.SYS,STORXL51.PUB.SYS;YES
      or
:COPY STORXL52.USL.SYS,STORXL52.PUB.SYS;YES
```

## Other Information for Hardware and Software Products

The following articles are included in this section:

- 100VG-AnyLAN Network Adapter
- 100Base-T Network Adapter
- Legato NetWorker Client for MPE/iX
- ODBCLink/SE
- B-Tree Indices
- TurboIMAGE/XL Dynamic Detail Dataset Expansion (DDX) Problem
- CI LISTF Enhancements—Note on Formats 8 and 9
- Telnet/iX Server—Full Functionality Release
- RESTOREing Files STOREd to Magneto-Optical Media
- Network Printing Patch
- EMC Symmetrix Disk Arrays
- Single-Ended DLT4000 Magnetic Tape—Support Announcement

### 100VG-AnyLAN Network Adapter

100VG-AnyLAN Network Adapter (B5426AA) is a high-speed network link that provides HP3000 Series 900 systems that have HP-PB (HP Precision Bus, also known as NIO) backplanes, with a UTP cable RJ-45 connection to a 100VG local area network. The 100VG adapter connects to the 100Mb/s network through any 100VG hub or switch meeting the IEEE 802.12 standards.

100VG-AnyLAN Network Adapter includes the 100VG network adapter, 100Mb driver components, and hardware product manuals. TCP/IP Transport is already included with the HP 3000/iX fundamental operating system (FOS).

For more information on 100VG-AnyLAN Network Adapter, see the *Communicator 3000 MPE/iX—Express 3 (PowerPatch Tape C.55.03) Based on Release 5.5 (30216-90245)*.

### 100Base-T Network Adapter

100Base-T Network Adapter (B5427AA) is a high-speed network link that provides HP3000 Series 900 systems that have HP-PB (HP Precision Bus, also known as NIO) backplanes, with a UTP cable RJ-45 connection to a 100Base-TX local area network. The 100Base-TX adapter connects to the 100Mb/s network through any 100Base-TX hub or switch meeting the IEEE 802.3u standards.

100Base-T Network Adapter includes the 100Base-TX network adapter, 100Mb driver components, and hardware product manuals. TCP/IP Transport is already included with the HP 3000/iX fundamental operating system (FOS).

For more information on 100Base-T Network Adapter, see the *Communicator 3000 MPE/iX—Express 3 (PowerPatch Tape C.55.03) Based on Release 5.5 (30216-90245)*.

## Legato NetWorker Client for MPE/iX

The Legato NetWorker Client for MPE/iX is now shipping. The Legato NetWorker ClientPak product number is B5475AA.

The Legato NetWorker ClientPak is only supported with a patch. After you update to MPE/iX Express 4, you need to obtain the Legato patch MPEKX21 from your HP Response Center or patch machine.

## ODBCLink/SE

Starting with Express 3 for 5.5, IMAGE/SQL is bundled with a new 16-bit and 32-bit ODBC driver, ODBCLink/SE, which is a replacement for the 16-bit ODBC driver, HP PC API. ODBCLink/SE is an implementation of Microsoft's Open Database Connectivity (ODBC) interface that enables Microsoft Windows based applications and tools to access TurboIMAGE/XL and ALLBASE/SQL data on the HP3000 in a client/server environment. The client can run under Microsoft Windows 3.1 or 3.11, Windows95, or WindowsNT V3.51 or V4.0. Connection via Winsock is available in 16-bit and the new 32-bit version. The driver can be used by direct calls from a Windows program, or through an ODBC-compliant application such as Microsoft Access and Visual Basic.

A data migration tool is provided to migrate data sources created for the current HP PC API to ODBCLink/SE. Once the data sources have been identified, the translation is performed automatically.

To facilitate the transition to the new driver, HP PC API is also included in the product. HP PC API with Gupta can co-exist with ODBCLink/SE but will no longer be enhanced.

For more information on ODBCLink/SE, see "Introducing ODBCLink/SE" in the *Communicator 3000 MPE/iX—Express 3 (PowerPatch Tape C.55.03) Based on Release 5.5 (30216-90245)*.

## B-Tree Indices

Starting with Express 3 for 5.5, TurboIMAGE/XL, IMAGE/SQL, and Query have been updated to support B-Tree indices. For more information on B-Tree indices, see "IMAGE/SQL with TurboIMAGE/XL Enhancements" in the *Communicator 3000 MPE/iX—Express 3 (PowerPatch Tape C.55.03) Based on Release 5.5 (30216-90245)*.

## TurboIMAGE/XL Dynamic Detail Dataset Expansion (DDX) Problem

(SR# 5003-367607)

This article is only for customers using the TurboIMAGE/XL feature Dynamic Detail Dataset Expansion (DDX).

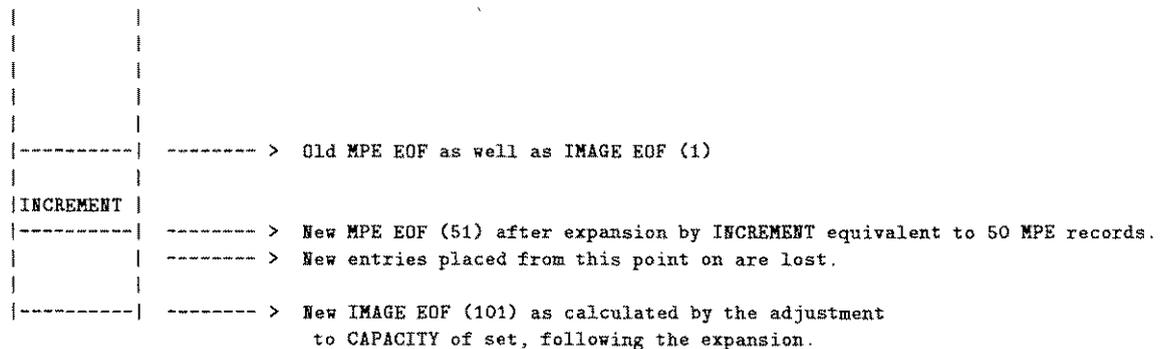
The DDX feature of TurboIMAGE/XL has a known problem which *may* cause you to lose *some* of your new data entries added following the dynamic detail dataset expansion. This defect has been in the feature since its introduction, but has surfaced recently and has been reported by a few customers. The problem has been fixed both for MPE/iX 5.0 and MPE/iX 5.5, and the patch information is as follows:

MPE/iX Release	Patch ID	TurboIMAGE/XL Version
5.0	TIXKX11	C.06.23
5.5	TIXKX13	C.07.07

These patches are not contained in this release. Contact your HP Response Center to obtain the correct patch. If you want to know more about the problem and ascertain if you may be susceptible to it, continue reading.

### Problem Description

The problem is that, following a dynamic expansion, the addition of the new data entries (DBPUTs), which exceed the INCREMENT in number, are written to the data set beyond the physical end-of-file (MPE EOF). When the database is closed and reopened, those new entries exceeding the INCREMENT are inaccessible. If a program attempts to read that data or add more data entries in that area, it gets an error -212, "Database is Corrupt." The diagram below illustrates the problem.



The real problem is that, for certain expansions, the new CAPACITY is inadvertently adjusted by adding the INCREMENT *more than once* instead of adjusting only *once*. This results in incorrect calculation of IMAGE EOF and subsequently allows new data entries to be placed in the area which really does not belong to the data set file. Therefore, when the data set is closed, the new entries added beyond the MPE EOF are not retained as part of the data set file by the MPE file system and are subsequently lost.

The circumstances in which such an erroneous adjustment could take place are as follows:

While an actual DDX is being performed by DBPUT for one process, a second process (user) accessing the same data set for the first time causes execution of IMAGE's "open data set" module. In a narrow timing window, the result will be incorrect extraneous adjustment of the CAPACITY by adding the INCREMENT! Specifically, the CAPACITY has been incremented *twice*, while the MPE EOF has been correctly incremented—only *once*.

**How to tell if you already have this problem.** If you use DDX, check each detail data set enabled for DDX. Remember, the problem is associated only with the detail data sets which have undergone dynamic expansion.

You can check for this condition as follows:

1. Use the FORM SETS command of QUERY, which gives the Current Capacity (CC) and Blocking Factor (BF) of each data set.

```
:query
HP32216D.03.11  QUERY/3000  TUE, NOV  4, 1997,  3:20 PM
COPYRIGHT HEWLETT-PACKARD CO. 1976

>b=dbusa
PASSWORD = >>
MODE = >>1
>form sets

DATA BASE: DBUSA                      TUE, NOV  4, 1997,  3:20 PM

DATA BASE LANGUAGE ATTRIBUTE: NATIVE-3000

          ITEM  CURRENT  ENTRY  ENTRY  BLOCKING
SETS:     TYPE COUNT  CAPACITY  COUNT  LENGTH  FACTOR

          PEPU01 D     3    1010    11    12     10
```

2. Calculate the IMAGE EOF of a DDX data set using the following formula:

$$(CC + (BF-1))/BF$$

For example:

$$\text{IMAGE EOF} = (1010 + (10-1))/10 = 101$$

3. Verify this calculated EOF with the MPE EOF given by the command :LISTF *dbnamenn*,2 where *dbnamenn* is the file name of that DDX data set.

```
:listf dbusa02,2

ACCOUNT= TESTACCT  GROUP= DDX

FILENAME CODE  -----LOGICAL RECORD-----  ----SPACE----
          SIZE TYP      EOF      LIMIT R/B  SECTORS #X MX
DBUSA02  PRIV  256W FB      51      200  1    112  2  4
```

**Part No. 30216-90247R3748**

4. If the two EOFs do not match, there is a problem.

In the example, MPE EOF of 51 and IMAGE EOF of 101 do not match—the problem exists.

The SHOW DBUSA CAPACITY command of DBUTIL gives capacity parms:

Data Set Name	Type	No. of Entries	%Max Cap	Capacity			Dyn Exp	
				Max	Current	Initial Increment		
PEPU01	D	11	1	2000	1010	10	500	YES

Based on the above formula for calculating MPE records, the initial capacity was 1 MPE record ( (10+9)/10), and the increment was 50 MPE records ( (500+9)/10). The correct physical EOF should be 51 (1+50). However, IMAGE EOF as shown by QUERY is 101 (1+50+50). This establishes that the IMAGE EOF was adjusted twice.

If you use one of the IMAGE/SQL structure maintenance tools (such as Adager, DBCPLUS, or DBGENERAL), it may have the capability to locate instances of this problem, and if you discover that one of your data sets is so afflicted, it may be possible to correct the problem using the same tool. (See the documentation for your tool.)

**How to avoid experiencing this problem on your IMAGE database.** The simplest answer is to install and use one of the TurboIMAGE/XL beta patches mentioned above. This is the option that HP recommends.

However, if you are unable to acquire and install the patch immediately, consider the following alternative:

Until you install the patch, avoid opening a data set from one process while another is expanding it!

Here are some ways you might accomplish this; each way is independent of the others:

1. Preallocate and pre-expand DDX data sets which are *nearly-full* by adding, in exclusive mode, many new dummy entries. Add more than your actual processing will add. Then delete them before your real processing begins.
2. Open all DDX-enabled data sets needed from each process which accesses them before allowing any process to add data to any of these sets.
3. Access the database exclusively, if possible, until the expansion has completed.
4. Disable DDX on any nearly-full data set, using an HP or third-party tool.

## CI LISTF Enhancements—Note on Formats 8 and 9

The “LISTF,access” portion of the 5.5 Express 3 patch has a known problem which will have low impact and is fixed in Release 6.0.

Anyone requesting the accessors for an “3000devs” spooled device file will see a CI warning indicating that some data may have been lost, followed only by the summary line for the listed file. For example:

```
:listfile /3000devs/DEVICES/@ ,8 ;seleq=[access=inuse]
```

```
Some data may be missing, internal escape. (CIWARN 9171)
*****
FILE: /3000devs/DEVICES/00000101
2 Accessors(O:O,P:O,L:O,W:O,R:O),Share
```

This example shows CIWARN 9171 followed by the summary line for the file pertaining to LDEV 101. The fix for this problem eliminates the extraneous CI warning and shows an accurate open count (O:) of one rather than zero.

---

**Note** This problem only applies to the new accessor formats 8 and 9 when the LDEV file has a spooler process running and is spooled.

---

A SHOWDEV on the LDEV corresponding to the above file looks like this:

```
:showdev 101
```

LDEV	AVAIL	OWNERSHIP	VOLID	DEV	ASSOCIATION
101	SPOOLED	SPOOLER OUT			

## Telnet/iX Server—Full Functionality Release

(SR# 1653-204313)

The version of the Telnet/iX Server included with this PowerPatch (C.55.04) is fully functional as was the version with the two preceding PowerPatches (C.55.02 and C.55.03). In addition, patch PTDEDP1A fixes a problem with handling VPLUS mode correctly in a Telnet/iX session. For more information, refer to “Telnet/iX Server—Full Functionality Release” in the *Communicator 3000 MPE/iX—Express 3 (PowerPatch Tape C.55.03) Based on Release 5.5* (30216-90245).

### Telnet Version Numbers

While the overall *V.uu.ff* of Telnet available in PowerPatch C.55.04 has not changed, internal version numbers of parts of Telnet changed with C.55.03. You can obtain the Telnet version numbers on your system by using this command:

```
:NMMAINT.PUB.SYS,72
```

The following table documents the Telnet version numbers.

**Telnet Version Numbers**

Part of Telnet	(type)	C.55.00	C.55.01	C.55.02	C.55.03 C.55.04
TELNET.ARPA.SYS	pgrm file	A5500000	A5500000	A5500000	A5500000
PTD_SM_VER	NL proc	A5500000	A5500100	A5500100	A5500100
PTD_HANDLER_VER	NL proc	A5500000	A5500101	A5500101	A5500102
PTD_PTID_VER	NL proc	A5500002	A5500102	A5500103	A5500105
PTD_PTOD_VER	NL proc	A5500002	A5500101	A5500101	A5500105
PTD_COMMON_VER	NL proc	A5500000	A5500100	A5500100	A5500102

## RESTOREing Files STOREd to Magneto-Optical Media

(SR# 4701-328609)

(SR# 4701-330126)

In MPE/iX Release 5.5 and previous PowerPatches, HP recommended that Magneto-Optical should not be used until patch MPEJX61 was installed. The fixes from this patch are included in the PowerPatches for C.55.03 and C.55.04. The full functionality of Magneto-Optical can be used with either release installed.

These two problems with RESTOREing files STOREd to Magneto-Optical media are fixed starting with Express 3 (C.55.03):

1. The 5.5 Virtual Storage Management performance enhancement to “prefetch-behind” one logical page (4KB) caused a read of a pre-erased sector and a resultant process abort.
2. The change in the size of the 5.5 STORE label caused an incorrect offset for the read of the media directory in pre-5.5 media and a resultant process abort.

## Network Printing Patch

Previously, there was a problem with the C.55.00 network printing spooler that caused it to eject a blank page at the start of the spool file if the output was created using prespace mode, and the carriage control character for the first line was %61. This problem is fixed in the patch MPEJXC1 which is available in Express 3 (C.55.03) and Express 4 (C.55.04).

## EMC Symmetrix Disk Arrays

The new SYM4 version of the EMC Symmetrix Disk Array is supported on MPE/iX 5.5 starting with Express 3. The SYM4 has the CUDA-9 9 GB disk mechanism that spins at 7200 rpm, a 20 MBs fast-wide differential bus to replace the 5MBs back-end single-ended bus, larger integrated cache, and other features to boost capacity and performance.

---

**Note** This is NOT an HP or customer installable device.

---

The following is a summary of software requirements for support of the EMC Symmetrix FW/SCSI Disk Arrays. Device installation and support is provided by EMC. For this reason, the following guidelines are provided only for general information purposes. For detailed information on the support and configuration of these devices, refer to the chapter, "HP 3000 Server Environment," in the *EMC Product Guide—Symmetrix Open Systems Host Environment* (200-999-563).

These are the MPE/iX 5.5 (C.55.04) patches available on Express 4:

- MPEJXG6—(Busy Bit) for 5x63 microcode and \$FD
- ODIJXJ0—EMCARRAY (1.2) for decoding the extended sense bytes of environmental errors

This is the minimum microcode level supported:

- 5062-61-40 (3/26/97) with Busy and Environmental bits set on the Symmetrix
- 5063-39-28 (5/27/97) with Environmental bits set on the Symmetrix
- 5063-39-29 (5/27/97) with Environmental bits set on the Symmetrix

## Single-Ended DLT4000 Magnetic Tape—Support Announcement

With the release of PowerPatch C.55.01 based on MPE/iX Release 5.5, HP announced support for the single-ended DLT4000 magnetic tape. This device represents a new performance and capacity high-end. Users can expect to store 20 GB native (40 GB compressed) on a single DLTapeIV cartridge at a rate of 10+ GB/hr.

To ensure that the device gives optimal performance, it is recommended that only one DLT4000 be configured per bus. That is, the DLT4000 should be the only device on the bus since it comes close to the single-ended bus capacity at its maximum rate. If DLT4000 shares the bus with other active devices, its performance will not be optimal.

## Disk Space Requirements

The combined System Load Tape (SLT) and Fundamental Operating system (FOS) disk space requirements shown in the table below are the minimum for any system software version. The table shows the amount of non-contiguous disk space sectors required for the three operating system components (SLT, FOS, SUBSYS) and PowerPatch of the currently supported versions of the operating system software.

During the modification process, some files are duplicated temporarily, and sometimes older versions of files are retained temporarily. As a result, the peak amount may be greater than the final amount. You must have the peak amount available to successfully modify your system.

Ensure that you have enough room on LDEV1 for all the SLT files. The FOS files do not have to go onto LDEV1.

**Note** The disk space values listed in this table are maximum estimated values only. The actual amount of disk space used on your system will vary.

For procedures for estimating the disk space required for your system, refer to "Estimating Disk Space" on page 3-20 of the *System Software Maintenance Manual* (30216-90223R3715).

### Disk Space Requirements in K Sectors for MPE/iX System Software

System Software Components	Version						
	5.0	5.5		C.55.03		C.55.04	
		Peak	Net	Peak	Net	Peak	Net
SLT only <sup>1</sup>	1,318	n/a	1,268				
SLT and FOS	2,027	2,378	2,092				
5.5 SLT, 5.5 FOS, PowerPatch-3				3,220	2,203		
5.5 SLT, 5.5 FOS, PowerPatch-4						3,415	2,171
SLT, FOS, full SUBSYS	3,226	3,637	3,170				
5.5 SLT, 5.5 FOS, 5.5 full SUBSYS, PowerPatch				4,634	3,293	4,811	3,483

<sup>1</sup> Provided for reference only. An operational system requires the SLT and FOS files at a minimum.

---

## Product Tables and Dependent Patches

This section contains the following information and tables:

- Using the Patch Tables
- Systems and Dependent Patches
- Software Products and Dependent Patches
- Peripherals and Dependent Patches

This section contains tables of the dependent patches included on the PowerPatch C.55.04 tape based on MPE/iX Release 5.5. These patches are required for the software, systems, and peripheral products to work properly.

Be sure to verify that your required patches have qualified in the patch qualification list created by AUTOINST during the PowerPatch installation process.

---

**Note** Enhancement patches are included in this PowerPatch. These patches will not qualify for automatic installation. To install these patches with HP Patch/iX, they will need to be “forced” on. The following patch tables document all the patches included in this PowerPatch. Using Patch/iX will indicate which patches automatically qualify for installation on your system.

---

### Using the Patch Tables

The product, system, and peripheral/hardware tables show the following:

- Products, systems, or peripheral/hardware that changed in each PowerPatch or Express. Products that have not changed in any of the listed releases are not included.
- New *V.uu.ff* for products that changed for the indicated release, if appropriate.
- Whether the PowerPatch (PP), SUBSYS tape (SS), or both (Both) changed for the indicated release.
- Patch IDs for the latest release. Use this table to verify that the patches qualify and are installed correctly.
- For hardware products, which Express, PowerPatch, or Mainline releases support the device.

---

**Note** Only hardware products that were introduced on the Express and PowerPatch releases are listed. It is recommended that you use the PowerPatch to install dependent patches for hardware products.

---

## Systems and Dependent Patches

### New Systems

Systems	Patch ID	Support PowerPatch
997/100, 997/200, 997/300, 997/400, 997/500	MPEJXY5 MPEJXQ0 MPEJXH2 MPEJXG5 MPEJXP2 TIXJXW2	C.55.04

## Software Products and Dependent Patches

### Year 2000 Related Patches

MPE Core Patches	Other Patches	Support PowerPatch	
MPEJXU1 MPEJXU2 MPEJXU3 MPEJXU8 MPEJXY9 NTWEDR7	BBRJXC2 BSIJXA4 DICJXE3 INFJXV7 NMSED7 OMWEDU5 OSPJXV3	RPRJXW8 RSHJXN8 SIMEDK4 SMHEDU4 SQLJXR2 TIXJXU4 VPLJXG3	C.55.04

## Software Product and MPE/iX Core Operating System Patches

	C.55.00	C.55.02	C.55.03	C.55.04	
Software Product (ID Number)	VUF	VUF	VUF	VUF in Express 4	Patch ID
ALLBASE/SQL (36216-02A)	A.G1.14	A.G1.15 PP	A.G2.00 PP	A.G2.00 PP	SQLJXR2
HP Telnet/iX <sup>1</sup> (HP32040A)	A.55.00	A.55.00 PP			
FTP/XL (36957A)	A.00.08		A.00.08 PP	A.00.08 PP	FTPEDR3
TurboIMAGE/XL with IMAGE/SQL (HP30391C)	C.06.12		C.07.04 PP	C.07.06 PP	TIXJXW2
IMAGE/SQL (HP36385B)	B.G1.10		B.G2.03 PP	B.G2.05 PP	ATCJXY6
COBOL II/iX (HP31500)	A.04.14		A.04.15 SS	A.04.16 PP	COBJXY0
Query (HP32216)	D.03.08		D.03.11 PP	D.03.11 PP	QUEJXT4
NARC Support Tools (HP35071A)	B.00.24		B.00.26 PP	B.00.26 PP	NRCJXP3
Internet Services INTERSRV	A.00.01		A.00.10 PP	A.00.10 PP	INTEDM7
100VG-AnyLAN Network Adapter (B5426AA) 100Base-T Network Adapter (B5427AA)			A.00.55 SS	C.55.00 PP	BT1EDW0 VGFEDV7 VGCEDV8 VG1EDV9
100VG-AnyLAN Network Adapter Dependent Patches					DCCEDT3 NMS EDT7 DLPEDQ5 NSTEDW4 LANEDQ6 ODIJXJ0 LSSDP4 PSIEDQ7 NMCEDP6
TurboSTORE/iX II (B5151AA) TurboSTORE/iX 7x24 (B5152AA)	C.55.00  C.55.00	C.55.08 SUBSYS			
Inform/V (HP32246)				A.11.01 PP	INFJXV7
Rapid/Dictionary (HP32244)				A.03.00 PP	DICJXE3

<sup>1</sup> See the table, "Telnet Version Numbers in C.55.04," earlier in this *Read Before Installing*.

## Software Product and MPE/iX Core Operating System Patches (continued)

	C.55.00	C.55.02	C.55.03	C.55.04		
Software Product (ID Number)	VUF	VUF	VUF	VUF in Express 4	Patch ID	
HP ALLBASE/BRW (HP35360)	A.01.55			A.01.56 PP	BRWJXX4	
Transact/iX (HP30138)		A.06.00		A.06.00 PP	TRRJXC4 TRXJXX1	
Transact/V (HP32247)	A.10.02	A.11.00		A.11.01 PP	TRVJXX2	
C/iX (HP31506A)	A.05.15			A.05.19 PP	HPCCKX05	
MPE/iX Core Operating System (HP31900) and other patches	C.55.00	C.55.02 Express 2	C.55.03 Express 3	C.55.04 Express 4	BBRJXC2 MPEJXP9 BSIJXA4 MPEJXQ0 DTCEDT5 MPEJXT0 DTSEDU3 MPEJXT2 FWSJXB6 MPEJXU2 FXLJXV2 MPEJXU8 GLAJXQ5 MPEJXW5 IACJXG0 MPEJXY5 LBCJXT6 MPEJXY9 LNKJXR8 MPEKX19 MPEJX12 NRJEDL0 MPEJX33 NTWEDR7 MPEJX35 ODUJX50 MPEJX46 OFDJXK5 MPEJX51 OMWEDU5 MPEJX54 OSPJXV3 MPEJX57 PASJXU9 MPEJX71 PMNJXF6 MPEJX79 PTDEDP1 MPEJX86 PX2JXJ3 MPEJX91 PX2JXL7 MPEJXA6 PX2JXQ2 MPEJXB4 PX2JXT8 MPEJXB5 RPGJXJ6 MPEJXF7 RPRJXW8 MPEJXF9 RSHJXN8 MPEJXJ4 SIMEDK4 MPEJXK7 SMHEDU4 MPEJXL3 SNAEDL2 MPEJXM8 SRCJXX9 MPEJXN6 STREDF4 MPEJXP2 TSMEDH6 VPLJXG3	

## Peripherals and Dependent Patches

The following peripherals are supported in Express 3 and Express 4 for MPE/iX release 5.5.

### Peripherals and Dependent Patches

Product Order #	Product Name	Patch ID
A3231A	High Availability Disk Array, Model 10 with 2 GB mechanisms	ODIJXG8 ODUJX50 OFDJX47 MPEJXG6 MPEJX91
A3388A	High Availability Disk Array, Model 10 with 4 GB mechanisms	Same as A3231A
A3549A	High Availability Disk Array, Model 20 Deskside Version	Same as A3231A
S1169A, S1170A, S1171A, S1206A, S1207A, S1208A, S1209A, S1210A	HP/EMC Symmetrix 3000 Disk Array	Same as A3231A
A3550A	High Availability Disk Array, Model 20, Field Rackable Version	Same as A3231A
A3643A	High Availability Disk Array Module	Same as A3231A
A3643A	High Availability Disk Array, D-Class	Same as A3231A
A3542A	High Capacity Digital Audio Tape Device	ODIJXG8 ODUJX50 OFDJX47 MPEJXG6
A3635A	Single-Ended SCSI Digital Linear Tape	Same as A3542A
A3304A C5254R C5255R C5257U C5254RZ C5255RZ C5256RZ	ST32272N (2 GB Single-Ended SCSI)	OFDJXH9 ODIJXJ0 MPEJXP1
A3517A	ST32272WC (2 GB Single-Ended SCSI)	Same as A3304A
A3318A	ST32272WC (2 GB Fast/Wide SCSI)	Same as A3304A

## Peripherals and Dependent Patches (continued)

Product Order #	Product Name	Patch ID
A3352A C5262R C5263U C5262RZ	ST34572N (4 GB Single-Ended SCSI)	Same as A3304A
A3351A C5258R C5259R C5261U C5259RZ C5260RZ	ST32272WC (2 GB Fast/Wide SCSI)	Same as A3304A
A3353A C5264R C5266U C5264RZ C5265RZ	ST34572WC (4 GB Fast/Wide SCSI) XP34361W	Same as A3304A
A3646A	ST34572WC (4 GB Fast/Wide SCSI)	Same as A3304A
A3647A	ST34572WC (4 GB Fast/Wide SCSI) XP34361W	Same as A3304A
A3628A	9 GB (Single-Ended Disk Drive)	Same as A3304A
A3629A	9 GB WD Disk Drive	Same as A3304A
A3716A	144 GB DDS-3 Autoloader	ODIJXJ0 MPEJXP1
3330 3430	EMC Symmetrix <sup>1</sup>	MPEJXG6 ODIJXJ0

<sup>1</sup> See the chapter, "HP 3000 Server Environment," in the *EMC Product Guide—Symmetrix Open Systems Host Environment* (200-999-563).

## New Communicator 3000 Articles

---

There is no new *Communicator 3000* for Express 4 for Release 5.5. Refer to the *Communicator 3000 MPE/iX—Express 3 (PowerPatch Tape C.55.03) Based on Release 5.5* (30216-90245) for more information on the products in both release C.55.03 and C.55.04. This section includes the following new *Communicator 3000* articles which will appear in the next *Communicator 3000 MPE/iX*:

- Year 2000 Enhancements for MPE/iX
- New Date Intrinsic for MPE/iX
- COBOL II/iX Enhancements
- Inform/V
- HP ALLBASE/BRW

---

### Year 2000 Enhancements for MPE/iX

For up-to-date information on the state of year 2000 readiness for specific HP products, refer to the web page, "HP Cure2000 Solution," at the following URL:

1. [www.hp.com/go/year2000](http://www.hp.com/go/year2000)
2. Click on HP Product Status
3. Scroll down and select 2000 Readiness of MPE/iX, Subsystems and Applications.

### Overview

This version of MPE/iX provides enhancements to the operating system for the year 2000 and beyond. The enhancements include the enhancements to the operating system commands, utilities, VPLUS and databases. This article explains the enhancements.

### Background

The year 2000 issues stem from the ability to handle correctly the year 2000 and beyond. This is particularly an issue for systems and applications that are using a two-digit year to express dates. There are a few commands, utilities, and databases in MPE/iX which were designed to handle two-digit years and these had to be enhanced to support year 2000 and beyond.

The year 2000 is a leap year, and this also had to be addressed. The MPE/iX operating system uses the Gregorian calendar and its definition of leap year: any year that is exactly divisible by 400, or that is exactly divisible by 4 and not exactly divisible by 100.

## Two-Digit Years Interpretation

One of the ways for handling two-digit year input is to interpret the years in the method called fixed window method. In this method, the two-digit years map as follows:

- 00 .. 49 maps to 2000 .. 2049
- 50 .. 99 maps to 1950 .. 1999

For example, both of the following STREAM commands will schedule the JOBFIL as January 21, 2001:

```
:STREAM JOBFIL;DATE=01/21/2001
:STREAM JOBFIL;DATE=01/21/01
```

## Intrinsics

The CALENDAR intrinsic and CALENDAR date representation format are widely used in MPE/iX. There are many other intrinsics that accept or return dates in calendar format. The following discussion clarifies the interpretation of calendar format year value.

### CALENDAR Intrinsic

The CALENDAR intrinsic returns the CALENDAR date, including the day of year and the year of century. The existing CALENDAR intrinsic documentation refers to the year as "year of century." This should be interpreted as "year since 1900."

The new definition of the CALENDAR intrinsic is as follows:

#### Syntax

```
U16
date:=CALENDAR;
```

where *date* is a 16-bit unsigned integer (assigned functional return).

This returns the CALENDAR date in the following format:

Bits	Value/Meaning
7:9	day of year
0:7	year since 1900

#### Description

For the year 2000, the CALENDAR intrinsic will return the year as 100, for 2001 as 101, and so on. With this current interpretation, CALENDAR format can handle years through 2027.

All intrinsics accepting or returning dates in CALENDAR format will behave the same way as the CALENDAR intrinsic.

---

**Note** If your application uses the CALENDAR intrinsic or CALENDAR format, it may be required to verify the source for conformance to the above interpretation.

---

## HPFOPEN/FOPEN Intrinsic

HPFOPEN item 31 (labeled tape expiration parameter) and FOPEN (*formmsg* label tape parameter) will accept both two- and four-digit years as the labeled tape expiration date. This is the same as the FILE command expiration date parameter. For an interpretation of two-digit years, refer to the section, "Two-Digit Years Interpretation," in this article.

## New Date Intrinsic

This version of MPE/iX has new intrinsics added to manipulate dates in various formats. For more information, refer to the article, "New Date Intrinsics for MPE/iX," in this *Read Before Installing*.

## Commands

The following commands are enhanced to support both two- and four-digit years. For an interpretation of two-digit years, refer to the section, "Two-Digit Years Interpretation." New syntax for the commands with parameters accepting years follows:

### ■ STREAM command

```
:STREAM jobname;DATE=datespec
```

where *datespec* is the date for streaming the job in the format *mm/dd/[yy]yy*. If omitted, the current date is used.

### ■ FILE Command

```
:FILE [;LABEL=[ [valid] [, [IBM] [, [expdate] [, seq] ] ] ] ]
      [           [           [ [ANS] ] ] ] ]
```

where *expdate* is the expiration date in the format *mm/dd/[yy]yy*.

### ■ SETCLOCK command

```
:SETCLOCK ;DATE=datespec
```

where *datespec* is the local date in the form *mm/dd/[yy]yy*.

### ■ LISTSPF/SPOOLF command

```
:LISTSPF/SPOOLF ;SELEQ=[DATE=mm/dd/[yy]yy]
```

### ■ STORE Command

```
:STORE           [ {;DATE<=accdate} ]
                  [ {;DATE>=moddate} ]
```

where *accdate* and *moddate* are the accessed and modified dates in the form *mm/dd/[yy]yy*.

## CI Variables

The available HPYEAR variable is corrected to return the last two digits of the current system year, and a new CI variable is added to have the four-digit current system year.

Existing variable:

HPYEAR is a global variable used by the CI that shows the last two digits of the current year; the initial value is the year at logon. Type= R I

For example, HPYEAR will have 0 for year 2000 and 5 for year 2005.

New four-digit year variable:

HPYYYY is a global variable used by the CI that shows the the current year; the initial value is the year at logon. Type= R I

For example, for year 2000, HPYYYY will have 2000.

HPSPLITYEAR is a global variable used by the CI that shows the current century split year; the initial value is 50. Type=W I

This variable is used in the new date intrinsics to manipulate dates in various formats.

## Utilities

CLKUTIL and START utilities/commands available in ISL (Initial System Loader) are enhanced to accept dates both in two- and four-digit years.

## VPLUS

VPLUS version B.06.07 (and later) contains enhancements to handle the year 2000 and beyond in applications. Refer to the article, "Year 2000 and Other Enhancements in VPLUS," in the *Communicator 3000 MPE/iX—Express 3 (PowerPatch Tape C.55.03) Based on Release 5.5 (30216-90245R3732)*.

## Databases

The ALLBASE/SQL functions TO\_DATE and TO\_DATETIME are enhanced to handle two-digit years appropriately. For more information, refer to the article, "New Features in ALLBASE/SQL Version G2," in the *Communicator 3000 MPE/iX—Express 3 (PowerPatch Tape C.55.03) Based on Release 5.5 (30216-90245)*.

In SQLUtil, the SETUPRECOVERY command is used for roll-forward recovery. SETUPRECOVERY accepts two-digit years as part of the date specification. These two-digit years are interpreted as discussed in the section, "Two-Digit Years Interpretation," in this article.

---

## New Date Intrinsic for MPE/iX

This release of the MPE/iX operating system has enhanced intrinsics for handling date formats. These intrinsics are targeted to support widely used existing date formats in MPE/iX and three new date formats.

The date intrinsics are broadly categorized as follows:

- **HPDATECONVERT**—Converting dates from one supported format to another.
- **HPDATEFORMAT**—Converting the supported format dates to the display formats desired by the user.
- **HPDATEDIFF**—Determining the number of days that separate two given dates.
- **HPDATEOFFSET**—Adding/subtracting an offset (days) to/from the given date.
- **HPDATEVALIDATE**—Validating the given date for conformance to a supported date format.
- **HPCALENDAR, HPFMTCALENDAR**—Using the new 32-bit HPCALENDAR format.

The following sections will discuss various date formats and the syntax and semantics of new date intrinsics. COBOL and Pascal programs using these intrinsics are provided as examples.

### Existing Date Formats

Existing date formats that are widely used are summarized in the following table, “Existing Date Formats.” The column “Sortable?” indicates whether the dates can be sorted (either numerically or lexicographically depending on how they are stored). The last column “Y2K Ready?” indicates whether dates in the format under consideration can be used to represent dates beyond 1999-12-31. The column “#Bytes” represents the number of bytes required to store the date format.

## Existing Date Formats

Storage Type	#Bytes	Explanation	Sortable?	Y2K Ready?
longint <sup>1</sup>	8	Microseconds since 1970-01-01 (MPE time-stamp)	yes	yes
integer	4	Upper 2 bytes: year next byte : month of year bottom byte: day of month	yes	yes
integer	4	Upper 2 bytes : year bottom 2 bytes: day of year	yes	yes
integer	4	Seconds since 1970-01-01 (POSIX.1 time() format; valid through 2038-01-18)	yes	yes
shortint	2	Upper 7 bits: #years since 1900 Lower 9 bits: day of the year (CALENDAR format; valid up to 2027-12-31)	yes	yes
integer	4	YYMMDD date	yes	no
integer	4	MMDDYY date	no	no
integer	4	DDMMYY date	no	no
ASCII <sup>2</sup>	6	YYMMDD date	yes	no
ASCII	6	MMDDYY date	no	no
ASCII	6	DDMMYY date	no	no
ASCII	6	YYMMDD date YY:MM3000 date <sup>3</sup>	yes	yes
ASCII	6	MMDDYY date YY:MM3000 date	no	yes
ASCII	6	DDMMYY date YY:MM3000 date	no	yes

1 "integer" and "longint" are binary values.

2 "ASCII" means ASCII character code.

3 MM3000 dates are represented as in the MM3000 product which uses the ASCII letters "A" through "Z" for decades starting with the year 2000.

## New Date Formats

Three new date formats are listed in the table, "HP Standard Formats." The first format is an integer representation of the ISO 8601 date format. The second is an ASCII representation of the same. The last date format is an extension of the existing 16-bit CALENDAR date format to a 32-bit format. These date formats will be referred to as "HP Standard Formats."

**HP Standard Formats**

Storage Type	#Bytes	Explanation	Sortable?	Y2K Ready?
integer	4	YYYYMMDD date	yes	yes
ASCII	8	YYYYMMDD date	yes	yes
integer	4	Upper 23 bits: #years since 1900 bottom 9 bits: day of the year. (extension of the existing CALENDAR format.)	yes	yes

## Supported Date Formats

The "HP Standard Formats" and "Existing Date Formats" are combined in the following table, "Supported Date Formats." Each date format is assigned a date type code. The date intrinsics support these date formats and date type codes.

## Supported Date Formats

Date Type Code	Storage Type	#Bytes	Explanation	Sortable?	Y2K Ready?
1	longint	8	MPE time-stamp (microseconds since 1970-01-01)	yes	yes
2	integer	4	Upper 2 bytes: year next byte: month of year bottom byte: day of month	yes	yes
3	integer	4	Upper 2 bytes: year bottom 2 bytes: day of year	yes	yes
4	integer	4	Upper 23 bits: #years since 1900 bottom 9 bits: day of the year. (analogous to the existing CALENDAR format.)	yes	yes
10	integer	4	Seconds since 1970-01-01 (POSIX.1 time() format; valid through 2038-01-18)	yes	yes
14	shortint	2	Upper 7 bits: #years since 1900 Lower 9 bits: day of the year (CALENDAR format; valid up to 2027-12-31)	yes	yes
15	integer	4	YYMMDD date	yes	no
16	integer	4	MMDDYY date	no	no
17	integer	4	DDMMYY date	no	no
18	integer	4	YYYYMMDD date	yes	yes
25	ASCII	6	YYMMDD date	yes	no
26	ASCII	6	MMDDYY date	no	no
27	ASCII	6	DDMMYY date	no	no
35	ASCII	6	YYMMDD date YY:MM3000 date <sup>1</sup>	yes	yes
36	ASCII	6	MMDDYY date YY:MM3000 date	no	yes
37	ASCII	6	DDMMYY date YY:MM3000 date	no	yes
38	ASCII	8	YYYYMMDD date	yes	yes

<sup>1</sup> MM3000 dates are represented as in the MM3000 product which uses the ASCII letters "A" through "Z" for decades starting with the year 2000.

## Special Date Values

Special date values to represent UNKNOWN, INVALID, NEVER, NEEDED, EXPIRED, and ILLEGAL dates are defined for dates in "HP standard formats." The following table captures this date value information. The other date formats have been in existence for some time and hence the special date values are not defined for them. These special dates, when passed to the HPDATEFORMAT intrinsic, will result in a corresponding output string. For example, passing "00000103" to HPDATEFORMAT will result in the output string: "NEEDED".

When special dates are passed to HPDATECONVERT, HPDATEOFFSET, and HPDATEDIFF intrinsics as input dates, an error status is returned and the output date is initialized to a binary zero value or a blank string, depending on the date type of the output date.

**Special Date Values**

Date Type	FORMAT	UNKNOWN	INVALID	NEVER	NEEDED	EXPIRED	ILLEGAL
4	YYYYDDD	0	0000367	0000368	0000369	0000370	0000371
18	YYYYMMDD	0	00000101	00000102	00000103	00000104	0000105
38	YYYYMMDD	" "	"00000101"	"00000102"	"00000103"	"00000104"	"0000105"

## New Date Intrinsics

The date intrinsics support dates in the range 0001-01-01 through 9999-12-31. They use the Gregorian calendar for all calculations, including the rule for leap years (even though the Gregorian calendar was not in use prior to year 1582). This calendar also ignores the fact that calendars in different countries changed at different times (around the year 1753). All the intrinsics accept byte aligned input/output date parameters, and all these intrinsics are NM callable.

On an error, the intrinsics initialize the output parameters to either a binary zero or a blank string depending on the type of the parameter.

Though the date type "4" can represent years beyond 9999, a year beyond 9999 (which needs five digits/characters) is considered an error.

New errors and warnings are documented in the System Message Catalog, SYSCAT.PUB.SYS, under the date intrinsics subsystem number 529.

## HPDATECONVERT

This intrinsic converts the dates from one supported format to another.

### Syntax

```

          I32V      *      I32V      *      I32  I32V
HPDATECONVERT(inputcode, inputdate, outputcode, outputdate, status, cutoff)

```

### Parameters

<i>inputcode</i>	is a 32-bit signed integer by value.  The value should be one of the date type codes listed in the table, "Supported Date Formats."
<i>inputdate</i>	varies for type by reference.  The interpretation depends upon the value of <i>inputcode</i> . See the table, "Supported Date Formats," for the supported date codes and their layouts.
<i>outputcode</i>	is a 32-bit signed integer by value.  The value should be one of the date type codes listed in the table, "Supported Date Formats."
<i>outputdate</i>	returns the date as per the format chosen by the <i>outputcode</i> parameter.  See the table, "Supported Date Formats," for the supported datecodes and their layouts.
<i>status</i>	is the HPE_STATUS parameter through which the error codes are returned. A value of 0 indicates no error and no warnings.
<i>cutoff</i>	is a 32-bit signed integer by value (optional).  This is used in validating the input parameter when the input date has two-digit year. This is a required parameter for dates with two-digit years. In all other cases, this parameter is ignored.  If the <i>cutoff</i> parameter is given as -1, the value of the CI environment variable HPSPLITYEAR is used as the cutoff year.  This parameter's value should be in the range 0..100. If the value of the parameter is 50, two digit years in the range 0..49 will translate to 2000..2049 and those in the range 50..99 will be translated to 1950..1999. If you specify the <i>cutoff</i> year as 70, the mapping will be 0..69 as 2000..2069 and 70..99 as 1970..1999.

### HPDATEFORMAT

You can use this routine to format the dates that can be combinations of display formats as explained below. Many of these elements are taken from ALLBASE/SQL date formats.

You can convert dates in the "Supported Date Formats" to a display string of your choice (with restrictions). The HPDATEFORMAT intrinsic will accept these format strings. The format specification strings can have the following syntax:

#### Syntax

[{*FormatElement*}{*Punctuation*}]

Valid Parameters for *FormatElement*

## Format Specification Strings

CC	Century (01 to 99)
YYYY	Year (0001 to 9999)
YY	Year of century (00 to 99) with leading zeros suppressed (0 to 99)
Q	Calendar quarter of the year (1 to 4)
MM	Month of the year (01 to 12)
ZMM	Month of the year with leading zeros suppressed (1 to 12)
DD	Day of the month (01 to 31)
ZDD	Day of the month with leading zero suppressed (1 to 31)
DDD	Day of the year (001 to 366).
ZDDD	DDD with leading zeros suppressed (1 to 366)
D	Day of the week (1 to 7 where Sunday is 1, Monday is 2, ... )
WW	Week of the year (01 to 53)
ZWW	Week of the year with leading zero suppressed (1 to 53)
Mon	Month of the year in ASCII format (Jan, Feb, ... )
Day	Day of the week in ASCII format (Sun, Mon, ... )
MON	Month of the year in ASCII (uppercase) format (JAN, FEB, ... )
DAY	Day of the week in ASCII (uppercase) format (SUN, MON, ... )

Valid Characters for *Punctuation*

-	Hyphen
/	Slash
.	Dot
' '	Blank
,	Comma
" "	Null (or no delimiter)

Thus, YYYY.MON.DAY, YY/MM/DD, DDMONYY, and DD-ZMM-YYYY are valid date formats. For example, "31 Jan 1997" when formatted through DD-ZMM-YYYY results in "31-1-1997," formatted through YYYY.MON.DAY results in "1997.JAN.FRI," while YYYYMMDD results in "19970131."

**Note** Mixing the NULL punctuation character with other punctuation characters is *not* allowed. Thus, YYYY/MM/DD is a valid format, while YYYYMM/DD is *not*.

Syntax

I32V \* CA CA I32 I32 I32V  
HPDATEFORMAT(*datecode*, *inputdate*, *formatspec*, *fmtdate*, *fmtdatelen*, *status*, *cutoff*)

Parameters

*datecode* is a 32-bit signed integer by value.

This value should be one of the date type codes listed in the table, "Supported Date Formats."

*inputdate* is the input date.

The interpretation depends upon the value of *datecode*. See the table, "Supported Date Formats," for the supported datecodes and their layouts.

*formatspec* is a character array (required).

This should be a NULL terminated string as per the syntax explained above in the table, "Format Specification Strings."

*fmtdate* is a character array (required).

This array size should be at least that of *formatspec*. On return, it will contain the date formatted as per the *formatspec*.

If an invalid date is passed, on return from the intrinsic its contents will be "UNKNOWN." For the "HP Standard Formats," if special values in the table, "Special Date Values," are passed for date parameter, on return from the intrinsic, the value of the string will be appropriately initialized. For example, for the date type 18, the initialized values for different special date values are as follows:

Date Value	Returned Contents
00000000	"UNKNOWN"
00000101	"INVALID"
00000102	"NEVER"
00000103	"NEEDED"
00000104	"EXPIRED"
00000105	"ILLEGAL"

If the character array passed does not have enough space to hold the special values or the formatted date, the behavior is undefined.

*fmtdatelen* is a 32 bit integer by reference (required).

On input, it is the length of the *formatspec* parameter.

On return, it represents the number of characters HPDATEFORMAT placed into *fmtdate*.

*status* is the HPE\_STATUS parameter through which the error codes are returned. A value of 0 indicates no error and no warnings.

*cutoff* is a 32-bit signed integer by value (optional).

This is used in validating and converting from two-digit to four-digit years. (See HPDATECONVERT documentation for more information on the *cutoff* parameter).

### HPDATEDIFF

This intrinsic determines the number of days that separate two given dates.

#### Syntax

```

          I32V      *      *          I32      I32  I32V
HPDATEDIFF(datecode, firstdate, seconddate, diffindays, status, cutoff)

```

#### Parameters

*datecode* is a 32-bit signed integer by value.

This value should be one of the date type codes listed in the table, "Supported Date Formats."

*firstdate* is the first input date. The interpretation depends upon the value of *datecode*. See the table, "Supported Date Formats," for the supported datecodes and their layouts.

*seconddate* is the second input date. The interpretation depends upon the value of *datecode*. See the table, "Supported Date Formats," for the supported datecodes and their layouts.

*diffindays* is the number of days difference between the two dates, computed as: *seconddate* - *firstdate*. Thus, if *seconddate* is earlier than *firstdate*, *diffindays* will be negative.

*status* is the HPE\_STATUS parameter through which the error codes are returned. A value of 0 indicates no error and no warnings.

*cutoff* is a 32-bit signed integer by value (optional).

This is used in validating and converting the two-digit years to four-digit years before computing the difference. (See HPDATECONVERT documentation for more information on *cutoff* parameter.)

### HPDATEOFFSET

This intrinsic adds or subtracts a specified offset to or from the given date.

#### Syntax

```

          I32V      *          I32V      *          I32  I32V
HPDATEOFFSET(datecode, inputdate, offset, outputdate, status, cutoff)

```

#### Parameters

*datecode* is a 32-bit signed integer by value

This value should be one of the date type codes listed in the table, "Supported Date Formats."

## Part No. 30216-90247R3748

<i>inputdate</i>	is the input date. The interpretation depends upon the value of <i>datecode</i> . See the table, "Supported Date Formats," for the supported datecodes and their layouts.
<i>offset</i>	is a 32-bit signed integer by value.  The number of days to be added to the input date. A negative value will result in a subtraction.
<i>outputdate</i>	is the output date. The result of the date offset operation. The interpretation depends upon the value of <i>datecode</i> . See the table, "Supported Date Formats," for the supported datecodes and their layouts.
<i>status</i>	is the HPE_STATUS parameter through which the error codes are returned. A value of 0 indicates no error and no warnings.
<i>cutoff</i>	is a 32-bit signed integer by value (optional).  This is used in validating and converting the two-digit years to four digit ones before computing the difference. (See HPDATECONVERT documentation for more information.)

### HPDATEVALIDATE

This intrinsic checks the validity of the given date with respect to the supported formats given in the table, "Supported Date Formats."

#### Syntax

```
I32          I32V      *      I32V
result := HPDATEVALIDATE(datecode, inputdate, cutoff)
```

#### Parameters

<i>datecode</i>	is a 32-bit signed integer by value.  This value should be one of the date type codes listed in the table, "Supported Date Formats."
<i>inputdate</i>	is the input date.  The interpretation depends upon the value of <i>datecode</i> . See the table, "Supported Date Formats," for the supported datecodes and their layouts.
<i>cutoff</i>	is a 32-bit signed integer by value (optional).  This is used in validating dates with two digit years. (See HPDATECONVERT documentation for more information on <i>cutoff</i> parameter).
<i>result</i>	is a 32-bit signed integer (assigned functional return).  This value will be 0 if the <i>inputdate</i> conforms to the date format represented by <i>datecode</i> . If it is not so, its value will be positive. If an error has occurred in evaluating the conformance, its value will be negative. This return value ranges from -999 to 1.

**HPCALENDAR**

The new HPCALENDAR intrinsic returns the date in the supported date type code 4 listed in the table, "Supported Date Formats." (This is also a HP standard format.)

**Syntax**

```
I32
  date := HPCALENDAR ;
```

where *date* is the 32-bit unsigned integer (assigned functional return).

This returns the calendar date in the following format:

Bits	Value/Meaning
23:9	Day of year
0:23	Year since 1900

**HPFMTCALENDAR**

This a new routine to handle HPCALENDAR format. It does the same job as FMTCALENDAR except that it accepts the 32-bit integer returned by HPCALENDAR intrinsic.

**Syntax**

```
I32V   CA
  HPFMTCALENDAR(date,formatdate)
```

**Parameters**

*date* is a 32-bit signed integer by value

This holds the calendar date, in the same format as the HPCALENDAR intrinsic (that is, date type 4).

*formatdate* returns the formatted calendar date in a 17-character array. If the day of the month is less than 10, a blank precedes it. For example,

```
FRI, JAN 6, 1989
```

**Examples**

Following are two examples:

- The Pascal program example uses most of the new date intrinsics.
- The COBOL program example uses the HPDATECONVERT intrinsic.

```

$standard_level 'hp_modcal'$

Program dateintr(input,output);

    {Constants for the different date types.}
const
    hp_dt_mpe_time_stamp_fmt    = 1;
    hp_dt_packed_yymmdd_fmt     = 2;
    hp_dt_packed_yydddd_fmt     = 3;
    hp_dt_new_calendar_fmt      = 4;

    hp_dt_posix_time_stamp_fmt  = 10;

    hp_dt_calendar_fmt          = 14;
    hp_dt_int_yymmdd_fmt        = 15;
    hp_dt_int_mddyymmdd_fmt     = 16;
    hp_dt_int_ddmmyy_fmtd       = 17;
    hp_dt_int_yyyyymmdd_fmtd    = 18;

    hp_dt_ascii_yymmdd_fmtd     = 25;
    hp_dt_ascii_mddyymmdd_fmtd  = 26;
    hp_dt_ascii_ddmmyy_fmtd     = 27;

    hp_dt_mm3000_yymmdd_fmtd    = 35;
    hp_dt_mm3000_mddyymmdd_fmtd = 36;
    hp_dt_mm3000_ddmmyy_fmtd    = 37;
    hp_dt_ascii_yyyyymmdd_fmtd  = 38;

    {Type definitions for the program}
type
    pac_20      = packed array [1..20] of char;
    iptr_type   = ^integer;

VAR
    date2_pac   : pac_20;           {Dates in packed character arrays.}
    fmt_pac     : pac_20;
    print_pac   : pac_20;

    cutoff      : integer;         {To represent cutoff date.}
    date1       : integer;         {Working dates.}
    date1_18    : integer;
    date2_18    : integer;
    date3_18    : integer;
    i           : integer;         {temp variables.}
    inptype     : integer;
    j           : integer;
    outtype     : integer;
    print_len   : integer;         {Length of formatted date.}
    temp_date   : integer;         {Another temp variable}

    status      : hpe_status;

function EPCALENDAR : integer; intrinsic;
procedure HPFMTCALENDAR; intrinsic;
function HPDATEVALIDATE : integer; intrinsic;
procedure HPDATEFORMAT ; intrinsic;
procedure HPDATECONVERT ; intrinsic;
procedure HPDATEDFFSET ; intrinsic;
procedure HPDATEDIFF ; intrinsic;

```

```

begin
  {Initialize the variables.}
  cutoff := 50;
  inptype := hp_dt_int_ddmmyy_fmt;

  date2_pac := '960121'; {The YYYYMMDD date in Supported format '25'.}
  date1     := 230196;   {The DDMYY date in Supported format '17'.}

  {conversion from one non standard formatto another}
  HPDATECONVERT(inptype,date1,hp_dt_ascii_yymmdd_fmt,fmt_pac,
                status,cutoff);
  if( status <> 0) then
    writeln('HPDATECONVERT intrinsic returned error',status);

  {Convert 'date1' to the YYYYMMDD integer format. With 50}
  {as the cutoff year.                                     }
  outtype := hp_dt_int_yyyymmdd_fmt;
  HPDATECONVERT(inptype,date1,outtype,date1_18,status,cutoff);
  if( status <> 0) then
    writeln('HPDATECONVERT intrinsic returned error',status);

  {Convert the 'date2_pac' to a YYYYMMDD integer format.}
  {Default cutoff is assumed (i.e., 50)                }
  HPDATECONVERT(hp_dt_ascii_yymmdd_fmt,date2_pac,
                outtype,date2_18,status);
  if( status <> 0) then
    writeln('HPDATECONVERT intrinsic returned error',status);

  {compute the difference between the two dates}
  HPDATEDIFF(outtype, date1_18,date2_18,temp_date,status,cutoff);

  if( temp_date < 0) then
    writeln(date1,' is #',temp_date,
  'days later compared to ',date2_pac)
  else if (temp_date > 0) then
    writeln(date2_pac,' is #',temp_date,
  'days later compared to ',date1)
  else writeln(date1,' is same as ',date2_pac);

  {offset date1_18 by the difference to get the      }
  {value of date2_18 through HPDATEOFFSET intrinsic.}
  HPDATEOFFSET(outtype,date1_18,temp_date,date3_18,status,cutoff)
  if( (date3_18 <> date2_18 ) and (status = 0)) then
    writeln('Error Malfunction of HPDATEOFFSET intrinsic');

  {Use the flexibility in converting the date to a display }
  {string. Use '#0' to NULL terminate the format          }
  {specification string.                                   }
  fmt_pac := 'YY.ZMM.ZDD'#0;
  print_len := 11; { 11 characters including #0 in fmt_pac }
  HPDATEFORMAT(outtype,date1_18,fmt_pac,print_pac,print_len,status);

```

## Pascal Example of New Date Intrinsics—Page 2

```
if( status <> 0) then
  writeln('HPDATEFORMAT intrinsic returned error',status);

  writeln('The converted date is: ',print_pac);

  {The following demonstrates how a byte array can be      }
  {passed in place of an integer. We are storing the      }
  {integer 230196 in a packed array of characters, which is}
  {an equivalent of a byte array. We subsequently cast the }
  {address of this array to an integer pointer to read the }
  {integer value out of it.                                }
  HPDATECONVERT( hp_dt_int_yyyymmdd_fmt, date1_18,
  hp_dt_int_ddmmyy_fmt, fmt_pac, status);

  $push, type_coercion 'storage'$
if( iptr_type (addr(fmt_pac))^ = date1 ) then
  writeln( 'Program worked fine! ')
  else   writeln( 'Error, in using byte array parameters !');
  $pop$

end.
```

The expected output from the program is:

```
230196 is #      2days later compared to 960121
The converted date is : 96.1.23
Program worked fine!
```

Pascal Example of New Date Intrinsic—Page 3

```

IDENTIFICATION DIVISION.
PROGRAM-ID. DATEINTR.
REMARKS. SAMPLE PROGRAM USING DATE INTRINSICS
DATA DIVISION.
WORKING-STORAGE SECTION.
*Date code for YYMMDD is 25, for YYYYMMDD is 38
01 INDATE-CODE          PIC S9(9) VALUE 25.
01 OUTDATE-CODE        PIC S9(9) VALUE 38.
01 SPLITYEAR           PIC S9(9) VALUE 70.
01 OUTDATE-YYYYMMDD   PIC X(8) VALUE SPACES.
01 INDATE-YYMMDD      PIC X(6) VALUE SPACES.
01 STATUS-VAR.
   05 S-INFO           PIC S9(4) COMP VALUE 0.
   05 S-SUBSYS        PIC S9(4) COMP VALUE 0.

PROCEDURE DIVISION.
CONVERT-DATE-PARA.
  DISPLAY 'Enter the date in YYMMDD format:'
  ACCEPT INDATE-YYMMDD
  CALL INTRINSIC "HPDATECONVERT" USING
    INDATE-CODE
    INDATE-YYMMDD
    OUTDATE-CODE
    OUTDATE-YYYYMMDD
    STATUS-VAR
    SPLITYEAR

  IF S-INFO 0
    PERFORM DISPLAY-ERROR
  ELSE
    DISPLAY "Convert Date in YYYYMMDD = " OUTDATE-YYYYMMDD
  END-IF
  STOP RUN.

DISPLAY-ERROR.
  DISPLAY "HPDATECONVERT FAILED. ERROR = " S-INFO.

```

### COBOL Example of HPDATECONVERT

---

## COBOL II/iX Enhancements

Version A.04.16 of HP COBOL II/iX, released in Express 4 for 5.5, makes available several additional enhancements that were requested by Interex SIGCOBOL.

1. Internal data structures of the compiler have been expanded again to permit compiling even larger programs. With Express 3 for 5.5, it was possible to compile programs in excess of 100,000 lines. While there is no specific limit on the number of lines permitted in a source program, the version of the compiler on Express 4 should be capable of processing programs well in excess of 200,000 lines.
2. The compiler has been enhanced to permit an index-name to be used as an operand of a DISPLAY statement. This is an ANSI extension, and if this feature is used and \$CONTROL STDWARN is specified, the compiler will display warning 517, "DISPLAY of index-name is nonconforming nonstandard (HP extension)."
3. The run-time library has been enhanced with a set of procedures to simplify bit manipulation in COBOL II/iX.

### New Routines for Boolean Operations

There are six new routines for performing boolean operations: HP\_BYTE\_AND, HP\_BYTE\_OR, HP\_BYTE\_XOR, HP\_BYTE\_NOT, HP\_BYTE\_UNPACK, and HP\_BYTE\_PACK. These procedures reside in the COBOL II run-time library in XL.PUB.SYS, but may be called from any program running in Native Mode.

The routines HP\_BYTE\_AND, HP\_BYTE\_OR, and HP\_BYTE\_XOR perform bitwise AND, bitwise inclusive OR, and bitwise exclusive OR. The two operands and the result may be any length, but must be the same length, and must be an integral number of bytes. The three routines have identical calling sequences. The first two parameters are the two operands, passed by reference. The third parameter is the result, also passed by reference. The final parameter is the length, in bytes, of the operands, and is passed by value. The first three parameters may not overlap, except in the case where two of them, or all three, are the same data item.

Examples:

```
CALL "HP_BYTE_AND" USING OPERAND-1, OPERAND-2, RESULT, \4\  
CALL "HP_BYTE_OR"  USING DATA-ITEM, MY-BIT-MASK, RESULT, \2\  
CALL "HP_BYTE_XOR" USING INPUT-BUFFER (J:1), RUNNING-XOR,  
                        RUNNING-XOR, \1\.
```

Note: In COBOL II/iX, backslashes ("\  
value. If the parameter is a literal, the backslashes are optional.

The routine HP\_BYTE\_NOT has the same calling sequence, except that there is only one operand rather than two. The result is computed as the bitwise complement of the operand. The operand and the result must be the same length, and must be an integral number of bytes. They may not overlap, except that the same data item may be used for both.

Example:

```
CALL "HP_BYTE_NOT" USING OPERAND, RESULT, \4\.
```

The routine HP\_BYTE\_UNPACK takes three parameters: an operand, a result, and the length of the operand in bytes. The bits of the operand are unpacked into the result, left to right. Each "zero" bit of the operand becomes an ASCII "0" byte in the result; each "one" bit becomes an ASCII "1" byte. The length specified is the byte length of the operand. The byte length of the result must be 8 times the byte length of the operand.

Example:

```

01 FIELD-A    PIC S9(4)    COMP.
01 RESULT    PIC X(16).
...
MOVE 5 TO FIELD-A.
CALL "HP_BYTE_UNPACK" USING FIELD-A, RESULT, \2\.
DISPLAY RESULT.
* Results in "0000000000000101".

```

The routine HP\_BYTE\_PACK is similar. The first parameter, the operand, is a sequence of ASCII bytes. Each byte of the operand is converted to a bit in the result, left to right. An ASCII "0" becomes a "zero" bit; anything other than an ASCII "0" becomes a "one" bit. The length specified is the length of the result in bytes. The byte length of the operand must be 8 times the byte length of the result.

Example:

```

01 BYTE-STRING PIC X(16).
01 RESULT-N    PIC S9(4)    COMP.
...
MOVE "000000000001111" TO BYTE-STRING.
CALL "HP_BYTE_PACK" USING BYTE-STRING, RESULT-N, \2\.
DISPLAY RESULT-N.
* Results in +15.

```

## Inform/V

Inform/V has been enhanced to support the Dictionary/3000 version A.03.00 enhancements. Version A.11.01 of Inform/V is now capable of handling multiple items with the same primary name in an HP Inform group. The alias names for the same primary item must be unique.

The Dictionary/3000 enhancement allows the definition of duplicate primary items in an HP Inform group within the dictionary. Whereas the Inform/V enhancement allows these group items to be selected and reported on as defined by the dictionary. Recompilation of Inform reports is not needed for this Inform/V release.

---

## HP ALLBASE/BRW

The last few revisions of HP ALLBASE/BRW have concentrated on the integration of TurboIMAGE Third Party Indexing (TPI) with BRW. BRW version A.01.54 introduced the support of TPI. In BRW version A.01.55, more TPI enhancements were provided, as well as several defect repairs. In the latest version of BRW, A.01.56, the main focus is a TPI defect repair.

This version of HP ALLBASE/BRW, A.01.56, fixes a TPI problem that was introduced in version A.01.55. When using wildcard characters for the Selection Set, the result shows that no records are found when data actually exists. The TPI tracing mechanism shows the incorrect DBFIND mode being used.

The other repair for version A.01.56, not related to TPI, corrects the continuation spoolfile number when a report is split between multiple spoolfiles.

In the previous version of HP ALLBASE/BRW, A.01.55, three enhancements and miscellaneous defect repairs were included. Among these enhancements were two new JCWs: BRWSQLTIMEOUT and BRWTPITRACE. BRWSQLTIMEOUT is given a value that is the number of minutes the BRW SELECT statement waits before timing out. A special value of 65535 causes BRW to wait indefinitely. BRWTPITRACE displays the TPI tracing information to \$STDLIST. The third enhancement in A.01.55 was to enhance the TPI selection algorithm. If more than one BRW condition matches a TPI key, the BRW condition that appears first in the compiler listing is used. If two TPI keys start with the same BRW condition, the longer key is used.



30216-90247R3748