HP 3000

Release Notes for PowerPatch C.55.01 Based on MPE/iX 5.5 with Communicator 3000 Articles

HEWLETT PACKARD

HP Part No. 32650-90853R3638
Printed in U.S.A. 1996

E1096

Release Notes for PowerPatch 1 (C.55.01)

These Release Notes contain critical information for customers who are installing the PowerPatch Tape C.55.01 based on MPE/iX Release 5.5. This information includes the following:

- 5.0 Diagnostics for New 1996 Disk Drives Produce a Warning but Still Work
- 9 Gigabyte Disk Drive—Special Installation Instructions
- RESTOREing Files STORED to Magneto-Optical Media
- MPE/iX 5.5 Shell Scripts Starting with "r" with the First Line #!/bin/sh Run in Restricted Mode
- Network Printing
- TIO Dynamic Configuration Causes System Abort 1458 After Copy Subtree is Done in NMMGR
- Telnet/iX Server
- Addendum to Communicator 3000 for MPE/iX 5.5
  - Telnet/iX Server—Full Functionality Release
  - ALLBASE/SQL—CAST Function

5.0 Diagnostics for New 1996 Disk Drives Produce a Warning but Still Work

New 1996 SCSI 2GB and 4GB disk drives (not 9 GB) are available for installation on MPE/iX 5.0 and 5.5 systems. However, patches to upgrade the 5.0 diagnostics and 5.0 diagnostic patches to recognize the 1996 disk drives are not yet available.

This means that when you execute the 5.0 SCSIDSK2 diagnostics (in SYSDIAG), a warning is displayed that indicates that the SCSIDSK2 diagnostic does not support the new product ID. However, the 1996 disk drives will function correctly with MPE/IX 5.0 when installed with the correct 5.0 MPE/iX patches (see the "Product Lists and Dependent Patches" section at the end of this notice). If you respond yes to the warning, the SCSIDSK2 diagnostic proceeds as normal and all disk tests work correctly, and all inquiry/capacity information reports correct disk drive information.

This is not a problem on 5.5 with the C.55.01 PowerPatch installed.

9 Gigabyte Disk Drive—Special Installation Instructions

A new 9 Gigabyte(GB) single-ended or fast-wide disk drive is not supported with this PowerPatch. It will be supported in the very near future.

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.

Before installing this drive, you should be aware of the following considerations:

- The correct patch must be installed **prior** to connecting the 9 GB disk drive to the system. This includes if the drive is to be used:
  - As an addition to the system volume set
  - As an addition to an existing user volume set
  - To create a new user volume set
- The 9 GB drive is not supported as LDEV 1.

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

9 GB Disk Drive Installation Checklist

- ☐ Install the patch ordered from the Response Center.
- ☐ Complete the entire patch process including the system UPDATE from the patched CSLT created during the patch 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.

RESTOREing Files STOREd to Magneto-Optical Media

(SR# 4701-328609)
(SR# 4701-330126)

Note **DO NOT** use magneto-optical media until you install patch MPEJX61.

In MPE/iX Release 5.5, there are two known problems with RESTOREing files STOREd to Magneto-Optical media:

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

Solution:

These two problems are fixed in a reactive patch, MPEJX61. If you need this patch, order it from your Response Center.

MPE/iX 5.5 Shell Scripts Starting with "r" with the First Line #!/bin/sh Run in Restricted Mode

(SR# 1653-179119)

MPE/iX Shell scripts with `#!/bin/sh` in the first line run in restricted mode when the script has a name that starts with an "r". You get an error that you are invoking a feature that isn't allowed in restricted mode. Restricted mode in the shell limits the script from certain features. (See the man page for the sh command for specifics of the "-r" option restrictions.)

For example, you get an error if you invoke a script named `restricted_script`:

```
#!/bin/sh
who am i 2>/dev/null
```

Solution:

There are three possible solutions to making the script succeed:

1. Rename the script to something that doesn't start with "r", such as `non_restricted`.
2. Use the "." command. For example,

   ```
   shell/iX> . restricted_script
   ```

3. Remove the #!/bin/sh from the first line of scripts that have a name that begins with the letter "r".

Network Printing

(SR# 4701-328658)

If you are using the C.55.00 network printing spooler, it will 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 is %61

A CCTL %61 on the first line of a prespace mode file should not cause a page eject. This is an error in the conditional-top-of-form logic in the network printing spooler.

There is no workaround.

TIO Dynamic Configuration Causes System Abort 1458 After Copy Subtree is Done in NMMGR

(SR# 4701-331942)

System Abort 1458 occurs if TIO Dynamic configuration is attempted after a copy subtree is done in NMMGR for a profile path.

The subtree copy function in NMMGR allows you to copy specified parts of a configuration file into either the same file or a different file.

If the path that is copied is `DTS.PROFILE.`*profile* or `DTS.PROFPC.`*profile* and then the change is made dynamically, the System Abort 1458 from Subsys 102 occurs. This occurs if the subtree copy is done in screen or maintenance mode.

Solution:

This System Abort can be avoided by the following sequence. (Use the example below as a guideline. This example uses the source path of: `DTS.PROFILE.TR1OOLD` and the destination path of: `DTS.PROFILE.TR1ONEW`.)

1. In NMMGR perform the subtree copy of the desired profile.
2. Go to the profile screen for the destination path. In this example, the profile screen is for `TR1ONEW`.
3. Press the (Save Data) key.
4. Validate the configuration file. It is now safe to initiate dynamic configuration.

```
NMMGR>openconf nmconfig.pub.sys
NMMGR>versionconf B.06.00
NMMGR>copyconf dts.profile.tr10old:nmsamp1.pub.sys dts.profile.tr10new
NMMGR>pathconf dts.profile.tr10new
NMMGR>updateconf
Saving data, please wait ...
NMMGR>validateconf dts
Searching for subsystem validation routine VALIDATEDTS
---> Validation of DTS/LINK started. <---
---> Validation of DTS/LINK finished. <---

Backup of configuration file complete.
NMMGR>:
```

Maintenance Mode Sequence Example

Telnet/iX Server

The Telnet/iX Server functionality is available in two parts:

- With MPE/iX Release 5.5, the Telnet/iX Server will allow a user to logon to the HP 3000 and use all MPE/iX CI commands as well as issue some Telnet client commands to the Server.
- Full functionality is available in PowerPatch C.55.01.

For details regarding the Telnet/iX Server functionality available with MPE/iX Release 5.5, please see "Telnet/iX Server—Full Functionality Release" in later in these *Release Notes*. Also see the *Communicator 3000 for MPE/iX 5.5* articles: "Introducing the Telnet/iX Server" (in Chapter 8, "Networking/Client-Server") and "Telnet/iX Server Functionality Details" (in Chapter 10, "Technical Articles").

Addendum to Communicator 3000 for MPE/iX 5.5

Included in this section are two additional articles for the *Communicator 3000 MPE/iX Release 5.5* (Non-Platform Software Release C.55.00, dated July 1996):

- Telnet/iX Server—Full Functionality Release
- ALLBASE/SQL—CAST Function

Telnet/iX Server—Full Functionality Release

The version of the Telnet/iX Server released with MPE iX Release 5.5 had several functionality restrictions that were documented in the MPE/iX 5.5 *Communicator* article, "Telnet/iX Server Functionality Details." The Telnet/iX Server included with this PowerPatch (C.55.01) is now fully functional. The purpose of this article is to update the information from the MPE/iX 5.5 *Communicator*.

Installation Verification

After installing this PowerPatch (C.55.01), you can verify whether your Telnet/iX Server installation was successful by running NMMAINT. A sample output follows:

```
:nmmaint,72

NMS Maintenance Utility  32098-20014 B.00.09  (C) Hewlett Packard Co. 1984

TUE, SEP  3, 1996,  1:55 PM
Data comm products build version: N.55.13

Subsystem version ID's:

HP TELNET/iX Subsystem HP32040A module versions:

NM program file: TELNET.ARPA.SYS     Version:  A5500000
NL procedure:    PTD_SM_VER          Version:  A5500100
NL procedure:    PTD_HANDLER_VER     Version:  A5500101
NL procedure:    PTD_PTID_VER        Version:  A5500102
NL procedure:    PTD_PTOD_VER        Version:  A5500101
NL procedure:    PTD_COMMON_VER      Version:  A5500100

HP TELNET/iX Subsystem HP32040A overall version = A.55.00
```

Changes from the MPE/iX 5.5 Telnet/iX Version

- FControl and FDeviceControl behavior for this version will function as documented in the *Asynchronous Serial Communications Programmer's Reference Manual* (32022-61001 Seventh Edition).

- All Block Modes and Binary Mode are now supported.
- Type-ahead is now supported.

- The maximum number of concurrent Telnet Sessions has been raised from 250 to 2000.
- The TLNETDOC.ARPA.SYS file has been updated with information for the Telnet/iX Server, and is included in this patch.
- The Telnet/iX Server now supports all of the Telnet commands that require interaction with the Server.

  In addition to the commands `open`, `send break`, `send AYT`, `send IP`, `send EC`, `send EL`, `quit`, and `close` that were supported with the MPE/iX 5.5 version, this version of the Telnet/iX Server also supports the `toggle` and `mode` commands.

  You should be aware of the following points when using the `toggle` and `mode` options:

  - `toggle`

    The Server was written to assume the user is always right. This means that if you do a `toggle echo` to allow the client to do all the echoing, the Server stops echoing and leaves all echoing to the client. However, if you then run an application on the HP 3000 that enables echoing, there will be a double echo on your display.

    `<*mode`

    a. Line mode for the HP 3000 server was implemented to be consistent with the HPUX 9.0 Telnet product. Neither of these line modes comply with the RFC-1116 line mode standard.

    b. We recommend that you do not use line mode when running block mode applications on the HP 3000. This may cause the applications to operate poorly.

ALLBASE/SQL—CAST Function

What is the CAST Function?

With the G1.15 release, the CAST function enhancement for ALLBASE/SQL and IMAGE/SQL is now available to customers. The CAST function is used to explicitly convert data from one data type to another. The CAST function not only allows conversion between compatible data types, such as between CHAR and BINARY or between INTEGER and DECIMAL, but it will also allow conversion between certain normally incompatible types, such as between CHAR and INTEGER.

The CAST function is defined in the ANSI SQL2 standard. CAST in ALLBASE/SQL and IMAGE/SQL complies with that standard. In addition, several extensions to the standard specification have been added to make CAST even more powerful.

The CAST function can be used anywhere a general expression is allowed. Also, as a part of this enhancement, the SQL parser has been enhanced to allow general expressions in more of the SQL syntax. For example, general expressions including nested functions are now allowed in all the date/time functions and string functions. Therefore, CAST will be supported inside functions that support expressions including aggregate functions. CAST will also take general expressions including nested functions as input.

CAST Specification

CAST Syntax

```
{ CAST ( { Expression } { AS } DataType[ , FormatSpec] ) }
       ( { NULL       } { ,  }                          )
```

Parameters

Expression	Column, USER function, host variable, local variable, AddMonths function, aggregate function, date/time conversion function, dynamic parameter, or procedure parameter, constant, current function, long column function, string function, or any combination of these in an arithmetic or concatenation expression.
DataType	ALLBASE/SQL data type: CHAR(n), VARCHAR(n), DECIMAL(p[,s]), FLOAT, REAL, INTEGER, SMALLINT, DATE, TIME, DATETIME, INTERVAL, BINARY(n), VARBINARY(n). The LONG BINARY(n) and LONG VARBINARY(n) cannot be used in the CAST operations.
FormatSpec	Format specification used for DATE, TIME, DATETIME, INTERVAL conversions. *FormatSpec* is the same as that used in the date/time conversion functions. See the *ALLBASE/SQL Reference Manual* (36216-90001) for more details.

Description

The following table shows what the CAST function supports:

source data type	target data type: EN	AN	VC	C	FB	VB	D	T	DT	I	TD
EN	+Y	+Y	*Y	*Y	*E	*E	N	N	N	N	N
AN	+Y	+Y	*Y	*Y	*E	*E	N	N	N	N	N
VC	*Y	*Y	+Y	+Y	+Y	+Y	+Y	+Y	+Y	+Y	*E
C	*Y	*Y	+Y	+Y	+Y	+Y	+Y	+Y	+Y	+Y	*E
B	*E	*E	+Y	+Y	+Y	+Y	*E	*E	*E	*E	*E
VB	*E	*E	+Y	+Y	+Y	+Y	*E	*E	*E	*E	*E
D	+E	+E	+Y	+Y	*E	*E	+Y	N	N	N	N
T	+E	+E	+Y	+Y	*E	*E	N	+Y	N	N	N
DT	+E	+E	+Y	+Y	*E	*E	N	N	+Y	N	N
I	+Y	+E	+Y	+Y	*E	*E	N	N	N	+Y	N
TD	N	N	*E	*E	*E	*E	N	N	N	N	+Y

```
EN = Exact Numeric(SMALLINT, INT[EGER], DEC[IMAL][(p[,s])],
     NUMERIC[(p[,s])])
AN = Approximate Numeric(DEC[IMAL][(p[,s])], NUMERIC[(p[,s])],
     FLOAT[(p)] or DOUBLE PRECISION, REAL)
C  = CHAR(n)
VC = VARCHAR(n)
B  = BINARY(n)
VB = VARBINARY(n)
D  = DATE
T  = TIME
DT = DATETIME
I  = INTERVAL
TD = TID
E  = ALLBASE/SQL Extension (not a part of ANSI standard)
N  = No
Y  = Yes
*  = Newly implemented with Cast
+  = Partially or Entirely implemented in ALLBASE/SQL already.
```

- If input to CAST is NULL, then the result of the CAST operation is NULL.
- ALLBASE/SQL supports implicit data conversion between:
  - Numeric data types to numeric data types
  - Character data types to character data types
  - Binary data types to binary data types
  - Binary data types to character data types
  - Character data types to binary data types.

  When CAST is used to do these conversions, all existing rules are applied.

- When a number of greater precision is converted to a number of lesser precision, and the number does not fit within the target precision, an overflow error occurs.
- When converting from an approximate numeric to an exact numeric or from an exact numeric to an exact numeric with less scale (integers have a scale of 0), the extra digits of scale beyond the target scale are dropped without rounding the result.
- If both source and target data type are character strings, the language of the result string is the same as the source.
- If the source data type is a character string and the target data type is a numeric, then the source value must only contain a character representation of a number. The result of the conversion is that the numeric string becomes a numeric type.

  If the source value is not a numeric string, an error occurs.
- If the target data type is CHAR(n), and the source data type is an exact numeric, the result is a character representation of that exact numeric. If the source value is less than zero, the first character of the result is a minus sign. Otherwise, the first character is a number or a decimal point.

  If the length of the resulted string is less than n, then blanks are added on the right. If the length of the resulted string is greater than n, an error occurs. The same algorithm applies if the target data type is VARCHAR(n), except that there is no need to pad the numeric string if its length is less than n.
- If the target data type is CHAR(n) and the source data type is an approximate numeric, then the number is converted to a character representation in scientific notation.

  If the length of the resulted string is less than n, then blanks are added on the right. If the length of the resulted string is greater than n, then an error occurs. The same algorithm applies if the target data type is VARCHAR(n), except that there is no need to pad the numeric string if its length is less than n.
- Conversion between character and binary data types was already supported before the CAST enhancement. The same rules still apply with CAST. If a target is shorter than the source, truncation occurs. If the target is larger than the source, the target is zero-filled in the case of BINARY(n), and blank-filled in the case of CHAR(n).
- When converting a non-character data type to BINARY(n) or VARBINARY(n), the data is not modified. Only the type changes so that the data is treated as binary data. The size of the source and the target in bytes must be equal in the case of BINARY(n), and the size of the source must be less than or equal to the size of the target in the case of VARBINARY(n). Otherwise, an error occurs.

  For decimal numbers, each digit of precision contributes 4 bits and 4 bits for the sign. The overall size is rounded up to a 4-byte boundary. The storage size for DATE, TIME, DATETIME, and INTERVAL is 16 bytes.
- When converting from BINARY(n) or VARBINARY(n) into a non-character data type, the data is not modified. Only the type changes so that the data is treated as a number of the target data type. The actual size of the source and the target in bytes must be equal, or an error occurs.
- Conversion between binary data types and numeric data types is an ALLBASE extension and is not allowed according to the ANSI SQL2 standard.

- Converting a character string to a DATE, TIME, DATETIME or INTERVAL with CAST is equivalent to using the respective date/time function, TO_DATE, TO_TIME, TO_DATETIME, or TO_INTERVAL. All the same rules apply.
- Using CAST to convert numeric types directly to date/time types is not allowed. This should be done by nesting the CAST functions so that the numeric value is first converted to a character string, and then converted to the date/time data type.
- Converting a date/time data type to:
  - A character type with cast is equivalent to using the TO_CHAR date/time function. All the same rules apply.
  - An INTEGER is equivalent to using the TO_INTEGER date/time function. This function converts date/time column value into an INTEGER value which represents a portion of the date/time column. If the source data type of CAST is date/time data type, and the target data type is INTEGER, all rules for TO_INTEGER to convert date/time into INTEGER will be applied. The *FormatSpec* must be used to specify a single component of the date/time data type (i.e. HH, MM, SS, DAYS, etc.).
  - Other numeric types is also allowed using CAST. In this case, the date/time data type is first converted to an INTEGER applying all the TO_INTEGER rules, then is converted from INTEGER to the target data type.

Examples

You may notice that in many of the queries listed, none of the columns of the tables are being referenced. Normally, you would have column references in the select list, but in this case constants are used instead of columns because it is easier to show what the input data looks like.

1. Convert date string of different format to default format.

```
select cast(to_date('951023', 'YYMMDD'),
char (10)) from dummy_tab;
----------
(EXPR)
----------
1995-10-23
```

2. Do the same thing again except use CAST instead of TO_DATE.

```
select cast(cast('951023', date, 'YYMMDD'),
char (10)) from dummy_tab;
----------
(EXPR)
----------
1995-10-23
```

3. Take day of year and convert it to character.
   c_date is a DATE column.

```
select cast(to_integer (c_date, 'DDD'),
char (10)) from cast_tbl;
----------
(EXPR)
----------
```

```
241
241
293
```

4. Return the ASCII value of an "A".

```
select cast(0x00 || cast('A', binary(1)),
smallint) from dummy_tab;
------
(EXPR)
------
    65
```

5. Take minimum of short int column and convert to decimal(7,2).
   c_sint is a SMALLINT column.

```
select cast(min (c_sint), decimal(7,2))
from cast_tbl;
---------
(EXPR)
---------
-32768.00
```

6. Return the string length of a string in decimal.

```
select cast(
string_length('gimme this strings length'),
dec (10,2)) from dummy_tab;
------------------
(EXPR)
------------------
      25.00
```

7. Pull out number from inside string and convert it to decimal.

```
select cast(
substring('pull out 12.3 from this string',
10, 4), decimal(7,2)) from dummy_tab;
----------
(EXPR)
----------
     12.30
```

8. Convert an arithmetic expression to character.

```
select cast(3+4, char) from dummy_tab;
------
(EXPR)
------
7
```

9. Convert date to date string.
   c_date is a DATE column.

```
select cast(c_date, char(50), 'MM/DD/YY')
from cast_tbl order by 1;
-----------------------------------------------
```

```
C_DATE
----------------------------------------------
08/29/95
08/29/95
10/20/95
```

10. Return number of seconds in interval as integer.
    c_interval is an INTERVAL column.

```
select cast(c_interval, integer, 'SECONDS')
from cast_tbl order by 1;
-----------
(EXPR)
-----------
      15054
      86399
      86399
```

11. Return number of seconds in interval as float.
    c_interval is an INTERVAL column.

Note ISQL displays a float as a decimal if possible.

```
select cast(c_interval, float, 'SECONDS')
from cast_tbl order by 1;
-----------------
(EXPR)
-----------------
  15054.00
  86399.00
  86399.00
```

12. Convert an integer offset number of days to a date.
    dateoff is an INTEGER column.
    Values in dateoff are: 1, 60, and 2000.

Note This example is somewhat complex because the TO_INTERVAL function requires a 7-character string of digits with leading zeros for the DAYS format.

```
select to_date('1971-01-01')+
to_interval(substring(cast(10000000+dateoff,
char(8)),2,7),'DAYS') from datetab;
----------
(EXPR)
----------
1971-01-02
1971-03-02
1976-06-23
```

Manufacturing Part No. 32650-90853
R3638

HEWLETT PACKARD