COMMUNICATOR 3000
MPE/iX-PowerPatch 5
(PowerPatch Tape C.55.05)
Based on Release 5.5

HP 3000 MPE/iX Computer Systems

Volume 9, Issue 4

Customer Order Number 30216-90257

HEWLETT®
PACKARD

Notice

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for direct, indirect, special, incidental or consequential damages in connection with the furnishing or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

Acknowledgements

Microsoft Windows ™, Windows 95 ™, Windows NT ™, Microsoft Access ™, Visual Basic ™, Visual C ++ ™, Visual FoxPro ™, and MS-Query ™are U.S. registered trademarks of Microsoft Corporation. ODBCLink/SE ™is a registered trademark of M. B. Foster Software Labs, Inc. Dr. DeeBeeSpy© 1995 Syware, Inc., all rights reserved. Axiant ™ and Impromptu ™ are registered trademarks of Cognos. Foxbase ™ is a registered trademark of Fox Software. Lotus ™ is a registered trademark of Lotus Development Corporation. Jetform is a registered trademark of Jetform Corporation. Paradox ™ is a registered trademark of Borland International Inc. PowerBuilder ™ is a registered trademark of Powersoft Corporation.

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013. Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19 (c) (1,2).

Hewlett-Packard Company
3000 Hanover Street
Palo Alto, CA 94304 U.S.A.

© Copyright and 1997, 1998 by Hewlett-Packard Company

Contents

Contents

1   Overview—MPE/iX Release 5.5 PowerPatch 5

This *Communicator* includes information about this release. We usually do not issue a *Communicator* for PowerPatch releases, however, we have important information for our products including year 2000 readiness that all HP 3000 customers should receive. This chapter contains general information about the MPE/iX-Express software release process, even though this is not an Express release.

Communicator Summary

Following is a summary of the articles in this document.

Chapter 2 System Information

Upgrading a Pre-997 Corporate Business Server to a 997

Gives an overview of the recommended upgrade process when upgrading to a 997 Corporate Business Server from a 990, 991, 992, 995, or 996 computer.

Predictive Support

Describes the new peripherals and SPUs supported by Predictive.

Chapter 3 Networking/Client-Server

Introducing 100VG-AnyLAN Network Adapter

Describes the high-speed network link that provides HP 3000 systems with a connection to a 100VG local area network.

Introducing 100Base-T Network Adapter

Describes the high-speed network link that provides HP 3000 systems with a connection to a 100Base-T local area network.

The REMSH Client Functionality

Describes the new REMSH Client that allows a user to connect to the remote system, execute a command on that remote system, and see the results of that command.

Hewlett-Packard NetWorker Client for MPE/iX

Explains that MPE/iX Express 3 for 5.5 (or later version) or a patch is required for the Hewlett-Packard NetWorker Client.

Introducing ODBCLink/SE

Describes the new product for ALLBASE/SQL links.

New Functionality Supported in DTS/TIO

Describes the following new functionality that was added in the release of C.55.02:

- Support of 400 DTCs

- Support of 2000 device classes

- Three new modem types: US modem In Out, European modem In Out, and Five wires modem.

- Two more parity types: 8 bits data length with even parity and 8 bits data length with odd parity.

These enhancements significantly increase the number of DTCs, supported modem types, and parity setting for HP 3000 machines.

Telnet/iX Server—Full Functionality Release

Describes the functionality added with Release 5.5 PowerPatch (C.55.01). Updated files are included with this patch.

Chapter 4 Peripherals

9 GB Disk Drive

Announces the 9 Gigabyte (GB) disk drive which is available for use with 5.5 (or later) systems, however, it cannot be used for LDEV 1, the boot disk.

New Disk Devices Supported

Describes the new 2 GB and 4 GB disk drives that are now supported.

High Availability Disk Array (Nike)

Describes the support for the 9 GB and 18 GB disk drives.

A3716A 144 GB DDS-3 Autoloader for MPE/iX

Gives an overview of the A3716A high-speed, high-capacity tape product combined with an autoloader.

EMC Symmetrix Disk Arrays

Overviews the SYM4 version of the EMC Symmetrix Disk Array supported starting with Express 3 for 5.5.

Chapter 5 Technical Articles

Year 2000 Web Page

Gives the URL for the HP web page, "HP Cure2000 Solution."

Year 2000 Enhancements for MPE/iX

Describes the enhancements to the operating system for Express 4 for 5.5 (and later releases) for the year 2000 and beyond. Gives the URL for the HP web page for up-to-date information on the state of year 2000 readiness for specific HP products.

New Date intrinsics for MPE/iX

Describes these enhanced date intrinsics and new date formats for MPE/iX introduced with Express 4 for 5.5:

- `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.

Year 2000 and Other Enhancements in VPLUS

Describes enhancements in B.06.07 version of VPLUS for the following functionality:

- Turn-of-century enhancement

- Ability to define default Field Type and Data Type for a Form

- Ability to define more than 52 single character fields per Form

- New intrinsics to get and set Save Field values programmatically

- Ability to renumber a Form with interactive FORMSPEC

CI Enhancements

Describes the new functionality added to the CI software to improve the ease of use and overall functionality:

- LISTF, LISTFILE now shows accessors to files.

- PAUSE can now wait for jobs to complete.

- PRINT can display all data in a file, even if the data appears to be line numbers.

- FLABELINFO and FINFO() return additional file information.

- INPUT allows specifying the number of characters to read from $STDIN.

- HPLASTSPID and HPSPOOLID are new variables.

FTP Enhancements

Describes the new functionality added to the FTP software:

• Support for non-standard MPE/iX file types.

• Ability to configure the server time-out.

• Hash command permits user to receive progress status.

Transact/V and Transact/iX Enhancements

Describes these new enhancements to Transact/V and Transact/iX:

• Literal strings and numeric constants can now be passed as parameters in the PROC verb.

• Transact now provides a method for increased decimal precision in arithmetic expressions.

Dictionary/3000

Describes these new enhancements to Dictionary/3000:

• The capability of copying entities with the new COPY command.

• The ability to enter elements multiple times within an HP Inform group.

COBOL II/iX Enhancements

Describes the COBOL II/iX enhancements in Express 3 and Express 4 for 5.5:

• Internal data structures of the compiler have been expanded to permit compiling significantly larger programs.

• The compiler is now able to detect when a source file is in Qedit format, and the Qedit product was not properly installed.

• The run-time library has been enhanced with a set of procedures to simplify bit manipulation in COBOL II/iX.

Inform/V

Announces that Inform/V is enhanced to support the Dictionary/3000 version A.03.00 enhancements.

HP ALLBASE/BRW

Describes the enhancements and defect repairs in this release of HP ALLBASE/BRW.

QUERY/iX Enhancements

Describes the enhancements made to QUERY to utilize the new B-tree index features in TurboIMAGE/XL.

IMAGE/SQL with TurboIMAGE/XL Enhancements

Explains these highlights of this release of IMAGE/SQL and TurboIMAGE/XL:

- Dynamic data set expansion for master sets (MDX or DDXM)

- Performance enhancement for databases with third-party indices

- 80 GB data sets

- Predicate locks when updating manual masters

- B-Tree indices enhancement

- New 16-bit and 32-bit ODBC driver, ODBCLink/SE, bundled with IMAGE/SQL

- Support for third-party composite indices in IMAGE/SQL

- Scalability for DBPUT, DBDELETE, and DBUPDATE (CIU on)

- Dynamic detail data set expansion (DDX) fix

New Features in ALLBASE/SQL Version G2

Describes several major enhancements in version G2:

- New authorities for security

- Year 2000 compatibility

It gives an overview of the CAST function that was added in the G1.15 release of ALLBASE/SQL.

Pseudo-Mapped Files Removed from ALLBASE/SQL

Describes the process of moving pseudo-mapped files to mapped files before upgrading ALLBASE/SQL to G2.

TurboSTORE/iX II, TurboSTORE/IX 7x24 Patches

Explains how to remove True-Online Backup if you do not want this feature with TurboSTORE II or TurboSTORE/iX 7x24 and do not have a license for it.

Chapter 6 Product List — How to Order

This chapter provides information on how to order the products and manuals.

Chapter 7 Product Release History

This chapter provides information on the currently supported Commercial Systems MPE/iX Releases and products. It also includes a matrix of the currently supported releases, the SPUs the releases are supported on, and the factory support termination dates.

Chapter 8 Catalog of User Documentation

This chapter contains a list of customer manuals for the HP 3000 Computer System that were introduced or updated at the time of this release.

MPE/iX-Express General Information

by MPE/iX System Release Team

This section is included for your information, even though this is a PowerPatch release instead of an Express release.

MPE/iX-Express is a quicker and easier process for receiving and installing new and updated systems and products for your MPE/iX General Release 5.5 based system. You can install Release 5.5 subsystem products with FOS dependencies without installing a new operating system.

Features and Benefits of MPE/iX-Express

- **New and updated software.** HP 3000 products such as systems, peripherals, databases, networks, software, and support products are now available more frequently.

- **Minimal impact to your system during installation.** New functionality can be added to your MPE/iX Release 5.5 system without updating the base system. System integrity is preserved. You do not need to recertify your system after adding functionality because your base operating system has not changed.

- **Reduced downtime.** System availability is increased. The end-user realizes increased functionality without the downtime impact required to do a system update. Software stability is maintained at the same time functionality is increased.

- **Unchanged software ordering process.** There is no change in the software ordering process.

- **Improved ordering structure for peripherals.** Software is ordered automatically when the peripheral order is placed. You do not have to place a separate order for software for peripherals.

- **Communicator.** New or enhanced product availability is announced through this *Communicator 3000*. Usually in conjunction with Express releases.

What You Receive with MPE/iX-Express

When you order new or updated software or hardware, you receive the following:

- **Customized SUBSYS tape.** The customized SUBSYS tape contains the ordered subsystem products (for optionally purchased products only). Although Express releases usually include a SUBSYS tape, 5.5 Express 4 did not include a SUBSYS tape.

- **PowerPatch tape.** The operating system patch tape containing system changes that are required for subsystem and peripheral product functionality.

- **HP 3000 MPE/iX System Software Maintenance Manual.** The manual required to perform an install, update, add-on, or Patch installation.

2 System Information

This chapter includes the following articles on system information:

- Upgrading a Pre-997 Corporate Business Server to a 997

- Predictive Support

Upgrading a Pre-997 Corporate Business Server to a 997

by Hank Cureton
Commercial Systems Division

Introduction

The I/O configuration on a 997 server has one additional hardware layer (the HP-PB I/O bus converter) compared to the earlier Corporate Business Servers (990, 991, 992, 995, and 996). As a result, the system I/O configuration on a 997 needs to be updated to reflect this additional hardware layer.

Example:

A typical SCSI disk device I/O configuration on a 995 would be described by the following I/O path:

```
2/52.3.0
```

The same SCSI disk device I/O configuration on a 997 would be described by the following I/O path:

```
2/28/52.3.0
```

NOTE   The HP-PB bus converter always has the fixed slot address 28 (decimal).

The utility packages `IOCNVRT` and `JADETOOL` are used to convert the OS software I/O configuration of a pre-997 Corporate Business Server to the new 997 I/O configuration format.

Order of Upgrade Process

The recommended order for upgrading a pre-997 Corporate Business Server (990, 991,992, 995, or 996) to a 997 server follows:

1. Make a backup copy of the `CONFIG.SYS` group and the file `NMCONFIG.PUB.SYS`.

2. Convert the `SYSGEN` configuration files using the command file `IOCNVRT`.

3. Convert the file `NMCONFIG.PUB.SYS` using the tool `JADETOOL`.

4. If RJE links are used on the system, run `NMMGR` to update any RJE links.

5. Shut down the system.

6. Upgrade the hardware to a 997.

7. Reboot the system using the newly created configuration files.

8. If the 997 SYSGEN configuration files are not in the group
    CONFIG.SYS, copy them into the CONFIG.SYS group.

9. Run NMMGR and validate DTS.

10. Shut down and reboot the system.

IOCNVRT.PUB.SYS

The command file IOCNVRT.PUB.SYS is used to convert the system
SYSGEN I/O configuration from a pre-997 Corporate Business Server
configuration to a 997 configuration, or vice versa.

Syntax:

:IOCNVRT {*source group*}, {*target group*}

source group   is the name of a valid SYSGEN configuration group in
the SYS account, containing the "source system" I/O
configuration. This is a required parameter. The files in
the *source group* will remain untouched unless the *target group* name is the same as the *source group*. The group
must exist, and its files must be valid for IOCNVRT to
work.

target group   is the name of the SYSGEN configuration group in the
SYS account into which the "target system"
configuration files will be stored. This is a required
parameter. If the group does not exist, it will be created
by the program.

Examples:

:IOCNVRT CONFIG, CONFIG

This command converts a pre-997 configuration in the group
CONFIG.SYS into a 997 configuration in the same group CONFIG.SYS.

:IOCNVRT CONFIG, NEWCONFG

This command converts a pre-997 configuration in the group
CONFIG.SYS into a 997 configuration and places it into the group
NEWCONFG.SYS.

JADETOOL

The JADETOOL utility is used to convert a pre-997 Corporate Business
Server data communications configuration to a 997 data
communications configuration.

Example:

```
:JADETOOL
Jade Tool     Version A0000006

  W A R N I N G ! ! ! ! !
This program is only to be used after authorization from HP

In a failure situation, use of this program could corrupt
the configuration file which could result in a system abort

It is recommended that this program be only used on a copy
of the NMCONFIG configuration file

Do you understand and agree to the above warning? (Y/N) y

Enter file name to convert: NMCONFIG.PUB.SYS
```

Restrictions

IOCNVRT requires that its user has SM (System Manager) capability, as a safety measure to prevent unauthorized or inadvertent changes to the system configuration file groups.

JADETOOL does not automatically convert RJE link I/O paths. This means that any users of RJE must reconfigure those links manually in NMMGR.

Recommended Sequence of Steps for Upgrading from a Pre-997 to a 997

1. Update the system with C.55.05.

2. Reboot the system.

3. Back up the S995 CONFIG.SYS group and the file NMCONFIG.PUB.SYS.

   Note: In the event that you need to convert back to the pre-997 configuration, these files will be needed.

4. :IOCNVRT CONFG995,CONFG997

   where your current 995 SYSGEN configuration files are located in the group CONFG995.SYS and the new 997 SYSGEN configuration files will be placed in the group CONFG997.SYS.

5. :RUN JADETOOL

   Convert NMCONFIG.PUB.SYS.

6. :RUN NMMGR.PUB.SYS

   Update any RJE links in NMCONFIG.PUB.SYS.

7. Shutdown the system.

8. Upgrade the hardware configuration to a 997.

9. Reboot the system using the new configuration group.

10. Move the `997 SYSGEN` configuration to `CONFIG.SYS` group—if not already there.

11. `:RUN NMMGR.PUB.SYS`

    Validate DTS.

12. Shut down and reboot the system from `CONFIG.SYS` configuration group.

Predictive Support

Product Overview

HP Predictive Support provides hardware support and helps increase the uptime of your systems by monitoring system memory and disk/tape drives.

When the HP Predictive Support software detects a potential problem, it sends a message to the HP Response Center. The Response Center portion of the system screens the data and forwards problems requiring further analysis to a Response Center Engineer. If action is needed at your site, the Response Center Engineer and the account Customer Engineer will work with you until the problem is resolved.

This proactive hardware support is provided as part of the HP Hardware Support Agreement.

Predictive Support Understands the Year 2000

The Predictive Support software has been modified to correctly handle dates in the Year 2000 including correctly displaying the dates as appropriate and being able to understand code words that expire in the year 2000 or beyond.

Upgrading Predictive Support Software

The HP Predictive Support software is upgraded each time you install a new version of the MPE/iX Operating System. The Predictive Support Monitor (PSMON) runs as a system process so this program file and its message catalog file is on the SLT. The rest of the files needed to make up the Predictive Support software is on the FOS tape and is automatically updated by using the AUTOINST process of updating the operating system software.

Support for New Peripherals

HP Predictive Support now supports the following new peripherals:

Product ID	Description
ST32272N/WC	2 GB SCSI Disc Drive
ST34572N/W/WC	4 GB SCSI Disc Drive
ST34573N/W/WC/WD	4 GB SCSI Disc Drive
XP34361W	4 GB SCSI Disc Drive

Product ID	Description
ST39173N/W/WC/WD	9 GB SCSI Disc Drive
ST118273W/WC/WD	18 GB SCSI Disc Drive
DGHS04Y	4 GB SCSI Disc Drive
DGHS09Y	9 GB SCSI Disc Drive

Before You Update HP Predictive Support

Before you install the update, you should run HP Predictive Support to ensure that any data indicating potential problems is transmitted to the HP Response Center.

If you have an HP Hardware Support Agreement, and need to have HP Predictive Support activated, contact Hewlett-Packard and a Customer Engineer will perform the initial configuration and activation.

3   Networking/Client-Server

This chapter contains the following articles related to networking and client-server applications:

- Introducing 100VG-AnyLAN Network Adapter
- Introducing 100Base-T Network Adapter
- The REMSH Client Functionality
- Hewlett-Packard NetWorker Client for MPE/iX
- Introducing ODBCLink/SE
- New Functionality Supported in DTS/TIO
- Telnet/iX Server—Full Functionality Release

Introducing 100VG-AnyLAN Network Adapter

by Randy Pawell
Commercial Systems Division

Overview

100VG-AnyLAN Network Adapter (B5426AA) is a high-speed network link that provides HP 3000 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 (Megabits per second) network through any 100VG hub or switch meeting the IEEE 802.12 standards.

100VG-AnyLAN takes advantage of collision-avoidance and noise-immunity characteristics of the 100VG protocol to provide high network reliability. It provides an evolutionary path for HP 3000 customers needing increased throughput over their existing 10Mb LAN links.

100VG-AnyLAN Network Link 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).

Product Features

The product features include:

- Supports ARPA, NS, NetIPC, and a subset of 4.3 BSD Sockets over TCP/UDP/IP.

- Provides tools compatible with those of ThinLAN, Token Ring, and FDDI 3000/iX.

- Has online diagnostics for status information (VGPBA).

- Has support software configurable with NMMGR.

- Is compatible with the IEEE 802.12 standard.

- Provides two traffic priorities: normal and demand.

- Provides 100Mb high-speed connectivity over CAT-5 or CAT-3 voice-grade unshielded twisted pair (UTP), RJ-45 terminated cabling.

- Provides 10Mb connectivity compatible with existing 10Base-T UTP networks.

- Supports a maximum cable length of 100 meters in 100Mb mode.

Introducing 100Base-T Network Adapter

by Randy Pawell
Commercial Systems Division

Overview

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 (Megabits per second) network through any 100Base-TX hub or switch meeting the IEEE 802.3u standards.

100Base-T Network Adapter provides an evolutionary path for HP 3000 customers needing increased throughput over their existing 10Mb LAN links.

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).

Product Features

The product features include:

- Supports ARPA, NS, NetIPC, and a subset of 4.3 BSD Sockets over TCP/UDP/IP.

- Provides tools compatible with those of ThinLAN, Token Ring, and FDDI 3000/iX.

- Has online diagnostics for status information (using VGPBA).

- Provides software configurable with NMMGR.

- Is compatible with the IEEE 803.2u standard.

- Provides 100Mb high-speed connectivity over CAT-5 unshielded twisted pair (UTP), RJ-45 terminated cabling.

- Provides 10Mb low-speed connectivity compatible with existing 10Base-T UTP networks, also with CAT-5 cabling.

- Supports a maximum cable length of 100 meters in 100Mb mode.

The REMSH Client Functionality

by Cas Caswell
Commercial Systems Division

An Overview of the REMSH Service

The remote shell, or REMSH, service is used to connect to a specified host and execute a command on that remote host. This article describes:

- Verifying installation

- Configuring the REMSH client

- Using REMSH

- Troubleshooting REMSH

- Implementation differences between REMSH on MPE/iX and REMSH for HP-UX

- Available documentation

REMSH is the same service as rsh on BSD Unix systems. The name was changed due to a conflict with the existing command rsh (restricted shell) on System V Unix systems.

REMSH allows a user to connect to the remote system and execute a command on that remote system. Output from the remote command is sent to the REMSH standard output so the user can see the results of the command.

Verifying Installation

The REMSH client is part of the Internet Services product with release C.55.03. To verify that REMSH is available on your system, use NMMAINT to verify versions of the Internet Services.

```
hawaii(PUB): nmmaint,73
NMS Maintenance Utility 32098-20014 B.00.09 (C)Hewlett Packard Co. 1984

MON, MAR 17, 1997, 10:35 AM
Data comm products build version: N.55.15

Subsystem version ID's:

Internet Services for the HP3000 module versions:

NM program file: INETD.NET.SYS               Version:  A0010003
NM program file: BOOTPD.NET.SYS              Version:  A0010003
NM program file: BOOTPQRY.NET.SYS            Version:  A0010002
NM program file: TFTPD.NET.SYS               Version:  A0010002
NM program file: REMSH.NET.SYS               Version:  A0010001
XL procedure:     INSVXL_SECURE_VERS         Version:  A0010004
XL procedure:     INSVXL_IPCSEC_VERS         Version:  A0010002
XL procedure:     INSVXL_NSRW_VERS           Version:  A0010003
XL procedure:     INSVXL_NETOF_VERS          Version:  A0010002
XL procedure:     INSVXL_SYSLOG_VERS         Version:  A0010003
XL procedure:     INSVXL_SIGNAL_VERS         Version:  A0010002
XL procedure:     INSVXL_GETTIME_VERS        Version:  A0010003

Internet Services for the HP3000 overall version = A.00.10
```

Configuring the REMSH Client

There is only one file on the MPE/iX system that you will need to change in order to allow use of the REMSH client. That is the file services.net.sys. However, there are some files that will need to be configured on the remote unix systems.

Editing the Services File

The services file associates official service names and aliases with the port number and protocol the services use. To enable REMSH use, you must edit the services file.

To edit the services file:

1. Open the services file with a text editor. You may edit the /etc/services file from the POSIX shell or the SERVICES.NET.SYS file from MPE/iX, whichever you prefer. Both names should point to the same file.

2. Verify that the following line exists in the file.

   ```
   shell   514/tcp  cmd   # remote command, no passwd used
   ```

   a. Add the line if it does not exist.

   b. If the line already exists in the file and is preceded by a pound symbol (#), delete the # and any spaces before the service name to enable the service.

3. Save the file and exit the editor program.

Unix Configuration

The REMSH service does not prompt for user ID and passwords. That information is handled via the command line parameters and configuration on the Unix host. See the using REMSH section for details on how the user id is determined and passed to the Unix host.

Password information is bypassed by use of a .rhosts file in the remote user's home directory or by use of the file /etc/hosts.equiv. See the man pages of the unix system for details on how to set up a /etc/hosts.equiv file. A user's .rhosts file entry will consist of the MPE/iX system name and user id.

For example, to access the HP-UX system "taltos" as user "cawti," from the MPE/iX system "jhereg" while user `MANAGER.SYS`, set up a host equivalency via the /etc/hosts.equiv file, or you will create a .rhosts file in the home directory of user "cawti" on the "taltos" machine. The .rhosts file would look like this:

```
jhereg   MANAGER.SYS
```

This will cause the REMSH daemon on the unix host to allow a connection from `MANAGER.SYS` on "jhereg" to the "cawti" user on the host "taltos." The .rhosts file for user "cawti" would contain an entry for every host and userid that you desired to access the "taltos" host as if they were the user "cawti."

NOTE   The MPE/iX equivalent of the Unix user id is the User.Account. An artifact of the MPE/iX implementation is that the MPE/iX information is usually reported in upper case. So be sure your .rhosts or /etc/hosts.equiv entries use the MPE/iX user ID information in upper case.

Using REMSH

The REMSH service is accessed by running the `REMSH.NET.SYS` program. You may do so under the MPE/iX CI or under the Posix shell. While the format of the commands will differ depending on how you run the program, the parameter list remains the same. We'll first look at the parameters and then look at running the program under the Posix Shell or from MPE/iX CI.

For the purposes of explaining the parameters, look at a sample invocation from the Posix shell. Detailed examples of both the Posix shell and MPE/iX invocations will follow later.

From the Posix shell you would invoke the REMSH by typing:

```
/SYS/NET/REMSH remotehost -l remoteuser  remotecommand
```

In all cases you must provide a *remotehost* and a *remotecommand*. The REMSH program will fail and generate an error message otherwise. Unless the remote system has MPE/iX type userids, you will also need to provide a -l *remoteuser* parameter as well. Otherwise the remote system will not allow the connection.

remotehost   is the name of the remote host you are attempting to connect to. The host name can be either the official name or an alias as understood by gethostbyname().

remoteuser   is the userid on the remote system you wish to be.

NOTE   The traditional Unix implementation of REMSH makes the -l *remoteuser* parameter optional. If you do not provide a -l *remoteuser* parameter, REMSH takes your current userID and assumes that you wish to connect to the same userID on the remote system. Since the MPE version of the userID is `USER.ACCOUNT`, and the Unix equivalent is user, it is unlikely that you will find a user on the remote system to match your id. We recommend that you always provide the -l *remoteuser* argument to REMSH.

remote- command is the command that you wish to execute on the remote machine. This command may be a CI command, a program (that meets certain criteria) or a shell script. If *remotecommand* is not specified, REMSH will terminate and provide a usage message.

NOTE   REMSH cannot be used to run commands that require a terminal interface (such as VI) or commands that read their standard error (such as `more`).

MPE/iX Examples

To run REMSH from MPE/iX prompt, type:

```
run remsh.net.sys;info="remotehost -l remoteuser [-n]
remotecommand"

jhereg(PUB): run remsh.net.sys;info="taltos -l cawti -n pwd"
/u2/home/cawti
jhereg(PUB):
```

Posix Examples

From the Posix Shell prompt, type:

```
/SYS/NET/REMSH remotehost -l remoteuser [-n] remotecommand
shell/iX> /SYS/NET/REMSH taltos -l cawti -n pwd
/u2/home/cawti
shell/iX>
```

Since you are running under the Posix shell, there are a number of shell features that you can take advantage of.

Shell metacharacters that are not quoted are interpreted on the local host; quoted metacharacters are interpreted on the remote host. The following command line appends the *remotefile* to the local file *localfile*:

```
SYS/NET/REMSH taltos -l cawti cat remotefile >> localfile
```

The following command line appends *remotefile* to the *otherremotefile*:

```
/SYS/NET/REMSH taltos -l cawti cat remotefile ">>" otherremotefile
```

The following command line runs REMSH in the background on the local system, and the output of the remote command comes to your terminal asynchronously:

```
/SYS/NET/REMSH otherhost -l remoteuser  -n remotecommand
```

The following command line causes REMSH to return immediately without waiting for the remote command to complete:

```
/SYS/NET/REMSH otherhost -l remoteuser  -n
"remotecommand 1>&- 2>&- &"
```

In REMSH, if the first parameter in its argument vector is not REMSH, it will use the value as a host name. You may symbolically link the host name to the REMSH program. A typical BSD Unix implementation will have these links under the /usr/hosts directory.

If you have made a symbolic link to the REMSH program that is the host name (that is in our examples, `ln -s /SYS/NET/REMSH taltos`), you could generate the same result as the first example with the following:

```
shell/iX>taltos -l cawti -n pwd
/u2/home/cawti
shell/iX>
```

Troubleshooting REMSH

These are the REMSH messages with an explanation of the meaning and suggested action, if any.

Message

```
remsh: MPE/iX/X version won't support rlogin or rexec
functionality
usage: remsh host -l login -n command
```

Meaning

Be sure to provide a command to execute.

Message

```
remshd: Login incorrect
```

Meaning

There is probably an invalid entry in remote .rhosts file. Be sure *host* name and *userid* are correct. *Userid* must be in upper case. Be sure you provided an -l *userid* parameter, or that the remote system has a *userid* that matches your MPE/iX logon.

Message

```
Program requires more capabilities than allowed for the group, the user
of a temporary file, or the hierarchical directory user. (LDRERR 505)
Native mode loader message 505
Unable to load program to be run. (CIERR 625)

**** EXEC FUNCTION FAILED; subsys =517; info = 48
ABORT: REMSH.NET.SYS
NM SYS   a.00aa0270 dbg_abort_trace+$24
NM UNKN  150.00366f6c
NM UNKN  2dd.0004bbd8
[1] + Done (134) REMSH hpcsyn24 -l casc -n pwd
  262204       Abort   REMSH
```

Meaning

The first message is from running REMSH from MPE/iX name space, and the second is from running under the Posix Shell. The cause is typically lack of PM capability on the group where REMSH resides. Since REMSH is in NET.SYS, it is unlikely you will see this problem unless someone changes the capability of the NET.SYS group.

Message

```
shell/tcp: Unknown service
```

Meaning

The shell service specification is not present in the services file. Edit /etc/services or SERVICES.NET.SYS to fix.

Message

```
Can't establish stderr
```

Meaning

REMSH cannot establish secondary socket connection for stderr.

Message

```
Couldn't reopen stderr
```

Meaning

The remote command tried to reopen stderror. This is not allowed under REMSH.

Message

```
 <system call>: ...
```

Meaning

Error in executing system call. Appended to this error is a message specifying the cause of the failure.

Implementation Differences

The full remote shell service typically consists of two parts: the REMSH client which allows a user on this machine to access remote hosts, and the remshd server which allows REMSH clients on other hosts to access the local host. Only the REMSH client functionality has been implemented on the MPE/iX system.

The HP-UX version of the REMSH client has an optional -n parameter that tells the client to not read from STDIN. Due to differences between MPE IO and Unix IO the -n parameter has been hard coded into the MPE/iX client.

The HP-UX REMSH client also allows rlogin and rexec functionality. Since this implementation was designed to address the needs of users attempting to access HP-UX commands and scripts from MPE/iX stream jobs, we chose not to implement any feature needing interactive input with the remote system.

Hewlett-Packard NetWorker Client for MPE/iX

by Lalitha Pejavar and Alex Early
Commercial Systems Division

Overview

Available now for HP 3000 users is NetWorker from Legato, a client/server application that provides advanced storage management capabilities to a heterogeneous environment of servers and desktop computers.

Benefits

These are the key benefits of using NetWorker:

- Complete Solution for HP Environments

  NetWorker is the most widely-implemented backup utility in open system environments. It provides HP 3000 users with better integration in a multiplatform environment, allowing them to back up data to an UNIX, NT, or Netware server.

- Support for HP's DLT Library System

  NetWorker is fully compatible with HP's Advanced DLT Library System enabling all server and desktop data to be automatically backed up, indexed, and policy-managed for either user-based file recovery or complete disaster recovery.

- Centralized Enterprise Management

  NetWorker's client/server architecture and centralized storage model allow integration into network management frameworks. Legato NetWorker integration modules allows SNMP-based administration of NetWorker servers using HP OpenView, or complete enterprise administration using HP OpenView IT/Operations.

- Superior Protection for On-Line Databases

  NetWorker supports on-line backup and restore operations for Oracle, Informix, and application data of SAP R/3, Oracle Financials and PeopleSoft HRMS.

Features

These are the key features of the NetWorker Client for MPE/iX:

- Provides an easy-to-operate graphical user interface and maintains a file index and media index to track data backed up and recovered.

- Supports a wide range of industry tape (DAT) and DLT libraries and furnishes a sophisticated media manager which allows for unattended backup and restore using autochange library.

- Provides for concurrent device support for backup and recovery operations using up to 16 backup devices simultaneously.

- Utilizes most of the features of 7x24 True-Online Backup since NetWorker uses TurboSTORE/iX 7x24 for backup. However, it will also work with FOS STORE and TurboSTORE/iX.

- Supports automatic backups of MPE volume sets. Also provides the ability to run two or more scheduled backups in parallel.

- Provides an optional Legato NetWorker Archive Application which gives the ability to take a snapshot of finished project files, directories, or MPE volume sets residing on primary media and safely stored on removable media.

Required Hardware and Software

Required hardware:

- HP 3000 any Series 9xx as a client

- HP 9000 server or Net Server

- DLT auto-library or DAT auto-changer

  - HP DLT 4/48, 2/28

  - STK 9714, 9710

- Networking connection (that is, FDDI, LAN, or 100VG-ANY LAN; 100VG ANY-LAN or FDDI LAN recommended)

Required software:

- NetWorker Client for MPE/iX

- NetWorker Server software (`LEGNETWR`), HP-UX Network Edition

- Auto-library license for the server (`LEGAUT0C`)

NetWorker Client for MPE/iX is supported on MPE/iX 5.5 with patch MPEKX21 (or supersede) and subsequent releases of all HP MPE/iX platforms.

HP Response Center Support

The NetWorker Client for MPE/iX currently has software supported through Legato, and NetWorker server software (`LEGNETWR`) has software support available through HP or Legato.

Legato NetWorker is the only third-party storage management solution supported by HP Worldwide Customer Support Operations. HP Response Center engineers have been fully trained on NetWorker operations. HP support contracts are available for NetWorker from your HP sales representative.

Documentation

The Hewlett-Packard Network Client Kit (B5475AA) contains the Legato Network Client software, *Legato NetWorker ClientPak for MPE/iX Installation Guide* (B5475-90001), and *Legato NetWorker ClientPak for MPE/iX User's Guide* (B5475-90002).

Introducing ODBCLink/SE

by Dawna Whitehead
M.B. Foster Associates Limited

Overview

ODBCLink/SE is a replacement for the HP PC API. It is an implementation of Microsoft's Open Database Connectivity (ODBC) interface that enables Microsoft Windows based applications and tools to access IMAGE/SQL on the HP 3000 and ALLBASE/SQL on the HP 3000 and HP 9000, in a client/server environment. ODBCLink/SE is a part of IMAGE/SQL and ALLBASE/SQL.

In this environment, application developers and end-users can take advantage of the PC's graphical user interface (GUI) and processing power, while relying on the security, integrity, and database management capabilities of ALLBASE/SQL and IMAGE/SQL.

HP PC API with Gupta can co-exist with ODBCLink/SE, but it will no longer be supported, and will not be distributed in MPE 6.0 and subsequent releases.

ODBCLink/SE server runs under MPE/iX 5.0 or higher on the HP 3000 server, or under HP-UX Versions 9.x and 10.x on the HP 9000 Series 7xx and 8xx servers. The client side runs under Microsoft Windows 3.1 and 3.11, Windows95, or WindowsNT V3.51 and V4.0. Connection via Winsock is available in both the 16-bit and the new 32-bit version.

The driver can be used in two ways: by direct calls from a Windows program, such as Foxbase, or through an ODBC-compliant application, such as Microsoft Access or Visual Basic.

The ODBC `LongVarBinary` data type is supported for binary large objects (BLOBs), such as compressed photographs or document images.

ODBC Level 2 Compliance

ODBCLink/SE is 95 percent ODBC Level 2 compliant. These exceptions are not supported:

- SQLBrowseConnect
- SQLExtendedFetch
- SQLSetPos
- SQLSetStmtOptions
- Scalar functions
- SQL_BIT, SQL_TINYINT, and SQL_BIGINT data types

Client/Server Interface

The networked HP 3000 Series 9xx with ALLBASE/SQL and IMAGE/SQL (or the HP 9000 Series 7xx or 8xx with ALLBASE/SQL) provide the relational database environment on the server.

Security is provided by the MPE/iX or HP-UX logon system and ALLBASE/SQL or IMAGE/SQL.

ODBCLink/SE connects the client to the server with a collection of dynamic link libraries (DLLs) on the client and runs under Microsoft

Windows 3.1 and 3.11, Windows95, or WindowsNT 3.51 and 4.0.

Client application requests are routed over the network to an ALLBASE/SQL or IMAGE/SQL database on the server, and replies are returned to the client application. A listener process establishes the connection between the client application and the target database on the server. The listener works with HP ThinLAN 3000/iX on the HP 3000 or ARPA Services on the HP 9000.

With ODBCLink/SE and application software on the client, you can develop SQL applications, generate reports, and query ALLBASE/SQL or IMAGE/SQL on the server. The tested client applications include the following as well as many more:

Cognos Axiant	Paradox
Cognos Impromptu	PowerBuilder
Jetform	Visual Basic
Lotus 1-2-3	Visual
Microsoft Access	Visual FoxPro
Microsoft Query	

Requirements

HP 3000 server requirements:

- MPE/iX Version 5.0 or 5.5 (or later)

- IMAGE/SQL Version B.G2.03 (or later)

- ALLBASE/SQL Version A.G2.00 (or later)

- ThinLANLink/XL must be configured and turned on

HP 9000 server requirements:

- HP-UX Version 9.0 or 10.0 (or later)

- ALLBASE/SQL Version A.G2.01 (or later)

- ARPA Services must be configured and turned on

Client requirements:

- 486 processor at a minimum, preferably a Pentium

- 8 MB memory at a minimum, preferably 16 MB

- 5 MB of free disk space

- Windows 3.1 or 3.11, Windows95, or WindowsNT V3.51 or 4.0

- Microsoft or WRQ TCP/IP software

Software provided:

- Server program for the HP 3000 or HP 9000 as appropriate

- Client programs for both 16- and 32-bit platforms.

  When using a 32-bit operating system, you should use the 32-bit
  driver if possible. However, if you have a 16-bit application running
  under Win95 or WinNT, you must install the 16-bit version of the
  driver. On Windows95 or WindowsNT, you can run both the 16- and
  32-bit versions of ODBCLink/SE. 16- and 32-bit data-sources are
  maintained separately and coexist.

- Dr. DeeBeeSpy from Syware Inc. for tracing calls to the ODBC driver

Migration from HP PC API

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

Performance Considerations

ODBC access to the server is very fast. You should note the following
performance exceptions:

- Reading and sorting a large table may prevent the server from
  returning control to the client until the entire database has been
  read and sorted.

- Opening a large table in MS-Access will cause a screen full of data to
  be displayed and then appear to wait for user input. In many cases,
  data is still being downloaded to the client. Careful thought needs to
  be employed before granting ODBC access to large tables on the
  server.

- Using dynamic substitution parameters speeds up your application,
  if you are writing your own SQL to do multi-row inserts or updates.

- In general, ODBCLink/SE will not complete any request faster than
  ISQL would, and could be slower due to network overhead.

Supported Functions and Commands

These are the ALLBASE/SQL and IMAGE/SQL functions and commands supported by ODBCLink/SE:

- The complete ALLBASE/SQL and IMAGE/SQL syntax dynamic SQL

- All SQL functions

You can use the ODBC SQLGetFunctions call to obtain a list of the supported ODBCLink/SE driver functions.

Areas Not Fully Implemented

Some areas of certain functions are not fully implemented. They are as follows:

- SQLColAttributes

  All functions will receive a return value of:

  ```
  Not  Implemented
  ```

  except the following supported options:

SQL_COLUMN_COUNT	SQL_COLUMN_NAME
SQL_COLUMN_TYPE	SQL_COLUMN_LENGTH
SQL_COLUMN_PRECISION	SQL_COLUMN_SCALE
SQL_COLUMN_DISPLAY_SIZE	SQL_COLUMN_NULLABLE

- SQLSetConnectOptions

  All functions will return this message:

  ```
  Driver not Capable (SQLState S1C00)
  ```

  except the following supported options:

SQL_AUTOCOMMIT	1=ON (default) 0=OFF
SQL_OPT_TRACE	supported by Driver Manager
SQL_OPT_TRACEFILE	supported by Driver Manager
SQL_TRANSLATE_DLL	supported
SQL_TRANSLATE_OPTION	supported
SQL_TXN_ISOLATION	supported
SQL_TXN_READ_UNCOMMITTED	Read Uncommitted (RU)
SQL_TXN_READ_COMMITTED	Read Committed (RC)

SQL_TXN_REPEATABLE_READ	Repeatable Read (RR)
SQL_TXN_SERIALIZABLE	Repeatable Read (RR)
SQL_TXN_VERSIONING	Not Implemented

- SQLSetStmtOptions are not supported.

  The driver will return either of these messages:

  ```
  Driver not Capable (SQLState S1C00)
  Option value changed (SQLState 01S02)
  ```

Supported Data Types

ODBCLink/SE supports ALLBASE/SQL and IMAGE/SQL data types.

The following table shows the correspondence between ALLBASE/SQL or IMAGE/SQL data types and ODBC data types.

ALLBASE/SQL or IMAGE/SQL Data Type	ODBC Data Type
CHAR, max length <255	SQL_CHAR
CHAR, max length >=255	SQL_LONGVARCHAR
VARCHAR, max length <255	SQL_VARCHAR
VARCHAR, max length >=255	SQL_LONGVARCHAR
BINARY, length <256	SQL_BINARY
BINARY, length >=256	SQL_LONGVARBINARY
VARBINARY <256	SQL_VARBINARY
VARBINARY >=256	SQL-LONGVARBINARY
LONG BINARY	SQL_LONGVARBINARY
LONG VARBINARY	SQL_LONGVARBINARY
INTEGER (32-bit)	SQL_INTEGER
SMALLINT (16-bit)	SQL_SMALLINT
DECIMAL (Internal representation is packed decimal)	SQL_DECIMAL
Image Zoned (Z)	SQL_DECIMAL
Image Packed (P)	SQL_DECIMAL
Image I3 (48-bit integer)	SQL_DECIMAL
Image I4 (64-bit integer)	SQL_DECIMAL

ALLBASE/SQL or IMAGE/SQL Data Type	ODBC Data Type
FLOAT(24), REAL, or Image E2	SQL_REAL
FLOAT(53), DOUBLE PRECISION, or Image E4	SQL_DOUBLE
Image R2 (non-IEEE float)	SQL_REAL
Image R4 (non-IEEE float)	SQL_DOUBLE
DATE	SQL_DATE
TIME	SQL_TIME
DATETIME	SQL_TIMESTAMP
INTERVAL	SQL_CHAR

ALLBASE/SQL Storage space for BLOBs should be created using a LONG VARBINARY column.

Stored Procedures

ODBCLink/SE implements all types of ALLBASE/SQL stored procedures, including those which return multiple-format results sets. You can list the stored procedures available to your ODBC session by calling SQLProcedures. Parameter information for stored procedures is obtained by calling `SQLProcedureColumns`.

Features Specific to ODBCLink/SE

The following is a list of features that are unique or specific to ODBCLink/SE and some details about them:

- Primary key name returned by SQLStatistics

  Most ODBC applications including MS Access, Visual Basic, and PowerBuilder require a unique primary key for a table to be updatable.

- Reading or writing to ALLBASE/SQL `LongVarBinary` items.

  Any file residing on the Hewlett-Packard server can be inserted into an ALLBASE/SQL `LongVarBinary` field with the `INSERT INTO` SQL command.

- Maximum number of statements

  ODBCLink/SE supports up to 50 concurrent statements, or cursors, per connection.

> To use multiple statements when you are not writing your own SQL application, tell ODBCLink/SE to have `SQLGetInfo` report more than one statement per connection.

ALLBASE/SQL Restrictions

These are the ALLBASE/SQL and IMAGE/SQL restrictions on the ODBC grammar. These are summarized in the following table:

Statement	Programming Considerations
CREATE TABLE	UNIQUE PRIMARY KEY must follow NOT NULL
DROP TABLE	ALLBASE/SQL does not provide CASCADE or RESTRICT
REVOKE	ALLBASE/SQL does not provide RESTRICT

- ALLBASE/SQL provides a `DATETIME` data type similar to `TIMESTAMP`.

- ALLBASE/SQL does not implement optimistic locking.

  Optimistic locking means that the DBMS does not lock data until just before an update is made, thus improving concurrency because locks are not held for long.

ANSI Character Set

The HP 3000 and HP 9000 both use the default ROMAN8 character set. Most applications running under Microsoft Windows use the 8-bit ANSI (ISO 8859/1) character set to provide support for Western European languages (including American). The first 127 characters are the same for both ROMAN8 and ANSI character sets, so only the extended and special characters are different.

If your client application uses these extended or special characters, then the data from the server must be converted from ROMAN8 to ANSI, and the data returning to the server must be converted from ANSI to ROMAN8. A translation DLL has been provided to accomplish this task.

Unsupported ALLBASE/SQL and IMAGE/SQL Statements

There are several reasons why some ALLBASE/SQL and IMAGE/SQL statements are not supported in the ODBCLink/SE environment:

- ODBCLink/SE does not support embedded SQL, therefore statements that work only with embedded SQL are not supported. See details that follow.

- Statements that relate to functions, such as connection, that have been replaced by functions resident on the client. See details that follow.

- Some functions are best restricted to the DBA; statements which control these functions are accessible only through Interactive Structured Query Language (ISQL) on the server.

- ODBCLink/SE internally prepares an SQL statement to be dynamically preprocessed. Therefore, you cannot use some statements such as `EXECUTE IMMEDIATE`.

Statements that Work Only with Embedded SQL

Client application software that accesses ALLBASE/SQL and IMAGE/SQL databases through ODBCLink/SE does not support embedded SQL. The application software calls the database functions directly; there is no need for preprocessing.

Because the following ALLBASE/SQL and IMAGE/SQL statements function only with embedded SQL, they are not supported by ODBCLink/SE:

CLOSE CURSOR	BEGIN DECLARE SECTION
DECLARE CURSOR	DESCRIBE
END DECLARE SECTION	EXECUTE
EXECUTE IMMEDIATE	FETCH
INCLUDE	OPEN
PREPARE	REFETCH
SQLEXPLAIN	WHENEVER

Statements Replaced by Functions on the Client

Some ALLBASE/SQL and IMAGE/SQL functions have been moved from the server to the client with ODBCLink/SE.

The following ALLBASE/SQL and IMAGE/SQL statements are not supported because they have been replaced by other functions resident on the client:

CONNECT	DISCONNECT
RELEASE	RESET
SET CONNECT	SET MULTITRANSACTION
START DBE	START DBE NEW
START DBE NEWLOG	STOP DBE

Utilities

The utilities provided with ODBCLink/SE do the following:

- Test and verify that the driver is functioning correctly.

- Determine which version of ODBCLink/SE is running.

- Delete unwanted drivers.

- Delete unwanted translation DLL's.

Tools on the Server

The following server tools are available to allow monitoring of the ODBCLink/SE listener process:

- HP GlancePlus/XL (HP 3000) Program

- HP GlancePlus/UX (HP 9000) Program

- HP-UX Process Status (ps) Command

- Monitoring ALLBASE/SQL Activity with `SQLMON`

- ODBCUTSE for testing/troubleshooting the host connection or table access

- The parameter to have error messages logged to the `ODBCLOG` file: `Trace SQL calls on the server`.

Tools on the Client

The tools on the client that you can use to monitor or troubleshoot the ODBCLink/SE client process:

- The configuration parameter, `Trace ODBC calls on the client`, is used to log information to the ODBCLink.LOG file.

- The application, Dr. DeeBeeSpy, is used to trace calls to the ODBC driver. Dr. DeeBeeSpy is shipped with ODBCLink/SE.

ODBCLink

ODBCLink/SE is a "Special Edition" of M. B. Foster Associates Limited ODBCLink option of DataExpress. ODBCLink, the full product, offers the following features in addition to the features of ODBCLink/SE:

- Support for serial and/or modem connections

- Read/Write access to KSAM files

- Read/Write access to MPE files

- Native read/write access to TurboIMAGE databases

- Support for PowerHouse dictionaries including PDL and PowerHouse subfiles

- Read/Write access to Suprtool, Dictionary/3000, HP SD, and PowerHouse Qschemac via DataExpress DataViews

- Support for TPI keyword indexes used by Omnidex and Superdex

- Access to Oracle databases on HP 3000 and HP 9000 platforms

- Access to Multiple DBEnvironments

- Access to HP intrinsics and procedures via remote procedure call

- Enhanced security via field level access controls

You may obtain an upgrade from ODBCLink/SE to the full product ODBCLink by contacting M. B. Foster Associates at 1-800-ANSWERS, (800-267-9377) or 613-448-2333, or by visiting the M. B. Foster website at `http://www.mbfoster.com`.

Documentation

For further information on ODBCLink/SE, refer to the *ODBCLink/SE Reference Manual* (36217-90406) (revised June 1998).

New Functionality Supported in DTS/TIO

by Vivek Chopra and Selva Kumar R I
Commercial Systems Division

Introduction

The Datacommunications and Terminal Subsystem (DTS) consists of host (HP 3000) based software and the DTC (Datacommunications and Terminal Controller) terminal server.

The DTC family of products supports X.25 communication and Telnet access to HP 3000 systems, and TCP/IP communication to HP 9000 and non-HP systems.

The DTS allows terminal, PCs in terminal emulation mode, serial printers, and other asynchronous devices to communicate with LAN-based hosts in HP-only and multivendor networking environments.

New Features

The DTS/TIO subsystem provides the following new functionality for MPE/iX starting with the 5.5 Express 2 release (C.55.02):

- Support of 400 DTCs.

- Support of 2000 device classes.

- Three new modem types for attached device. The new modem types are US modem In Out, European modem In Out, and Five wires modem.

- Data length/Parity list support. The DTS/TIO software has been enhanced to support two new parity types. The two new parity types are 8 bits data length with even parity and 8 bits data length with odd parity.

PowerPatch C.55.02 provided only part of the new functionality for the DTC 14.4 code support. The host DTC download files version 14.4 were installed with this patch. The host DTC download files version for MPE/iX 5.5 release was 14.2.

However, it should be noted that you do not need to run `NMMGRVER` on an existing `NMCONFIG.PUB.SYS` as no changes have been made to the `NMCONFIG` file.

Support of 400 DTCs

Users can now connect up to 400 DTCs to the HP 3000 host at a time. This is up from the earlier limit of 120 DTCs in MPE/iX 5.5 release (C.55.00). Thus the number of terminals that can be connected to a HP 3000 is significantly increased.

Support of 2000 Device Classes

A device class is a name that can be assigned to a single or a group of ldevs. This enables the user to refer to the ldev or the group of ldevs by a name. For example, the ldev associated with a tape drive is referred to as TAPE. Similarly, LP and DISC are the device classes for the system printer and disks respectively.

The number of device classes supported has been increased from 1000 to 2000. Increasing this has solved the problems of some customers who needed to add multiples of devices and were running into problem with the earlier limit.

Three New Modem Types for Attached Device

To give more flexibility, support has been added for three new modem types: US modem In Out, European modem In Out, and Five wires modem. However, these new modem types will be supported for the new types of DTCs only: DTC 72MX, DTC 16iX, and DTC 16MX.

Thus the complete list of supported modem types currently is:

- US modem In Out

- European modem In Out

- Five wires modem

- US modem

- European modem

- None (no modem)

Data Length/Parity List Support

Two more parity types have been added: 8 bits data length with even parity and 8 bits data length with odd parity. This was done as there are serial devices that need these settings. These need 8 bits which allow binary file transfer and also need to do parity checking.

These two new parity types are only supported for the newer DTCs: DTC 72MX, DTC 16MX, DTC 16iX and DTC 16RX. They are not supported for the DTC 48 or older DTCs.

The full list of supported parity settings is as given:

- 7 bits data length with even parity.
- 7 bits data length with odd parity.
- 7 bits data length with parity forced to 0.
- 7 bits data length with parity forced to 1.
- 8 bits data length with even parity.
- 8 bits data length with odd parity.
- 8 bits data length with none parity.

All the above enhancements are already supported on the 14.4 OpenView DTC manager. The required changes to support these enhancements in NMMGR were made in MPE/iX 5.5 Express 2 and later releases.

Telnet/iX Server—Full Functionality Release

by Cas Caswell
Commercial Systems Division

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 starting with PowerPatch (C.55.03) is 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.55.03), 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, AUG  7, 1997,  1:55 PM
Data comm products build version: N.55.15

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:  A5500102
NL procedure:    PTD_PTID_VER         Version:  A5500105
NL procedure:    PTD_PTOD_VER         Version:  A5500105
NL procedure:    PTD_COMMON_VER       Version:  A5500102

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 or later).

  - 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`

    1. 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.

    2. 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.

4   Peripherals

This chapter contains the following articles related to peripherals supported in the 5.5 PowerPatch 5 release:

- 9 GB Disk Drives
- New Disk Devices Supported
- High Availability Disk Array (Nike)
- A3616A 144 GB DDS-3 Autoloader for MPE/iX
- EMC Symmetrix Disk Arrays

9 GB Disk Drive

by Alex Early
Commercial Systems Division

Overview

The 9 Gigabyte (GB) disk drive has been available since March 1997 as an embedded disk and as of April 1997 in the mass storage enclosure on the HP 3000 computer systems. Due to the performance of these disks, there are special requirements for their use on MPE/iX HP 3000 computer systems. Before installing this disk drive, consider the following:

1. While the 9 GB disks are supported as an embedded disk, they cannot be used as LDEV 1 (the boot disk). Therefore, customers ordering the 9 GB disk as an internal disk must also order a smaller boot disk such as a 2 GB or 4 GB for the LDEV 1 disk. The 9 GB disk can then be just another disk in the system domain.

2. Due to a number of differences in technology, the 9 GB disk drive is only supported on MPE/iX 5.5 and later releases.

Documentation shipped with the disk to help with installation has information which incorrectly implies that the 9 GB is supported on MPE/iX 5.0. The patches referenced in the documentation are for support of the EMC disk array which also uses 9 GB disk drives. The 9 GB disk drive documentation is being changed to state that the current 9 GB disk drive is only supported on MPE/iX 5.5 and later releases.

New Disk Devices Supported

by Rakesh Saha and Sam Boles
Commercial Systems Division

Overview

The HP 3000 supports the following disk devices on MPE/iX 5.5
PowerPatch 5.

Table 4-1   **Supported Disk Devices**

Product ID	New Disk Device Supported
A3304A	ST32272N (2 GB single-ended narrow)
A3517A	ST32272WC (2 GB single-ended narrow)
A3351A	ST32272WC (2 GB differential wide)
A3318A	ST32272WC (2 GB differential wide)
A3352A	ST34572N (4 GB single-ended narrow) ST34573N (4 GB SE)
A3646A	ST34572WC (4 GB single-ended narrow) ST34573WC (4GB SE)
A3353A	ST34572WC (4 GB differential wide) XP34361W (4 GB differential wide) ST34573WC (4GB DF)
A3647A	ST34572WC (4 GB differential wide), XP34361W (4 GB differential wide) DGHS04Y (4GB DF) ST34573WC (4GB DF)
C6390A C6391A	ST34573WC (4 GB DF)
C6386A C6387A	ST34573N (4 GB SE)
A3628A	ST39173N (9 GB SE)
C6392A C6393A	ST39173N (9 GB SE)
A3629A	ST39173WC (9 GB DF)

Product ID	New Disk Device Supported
C6396A C6397A	ST39173WC (9 GB DF)
A5238A	ST39173WC (9 GB DF) DGHS09Y (9 GB DF)
A4909A	ST118273WC (18 GB DF),
A5239A	ST118273WC (18 GB DF)

Ambiguous Product ID

In the continuing effort to provide the price-performance advantage of the latest disk technology and, at the same, time protect existing customer investment, HP uses the same disk mechanism for both Single-Ended and Differential disks. However, there is a layer of transparent hardware that masks the electrical differences. This enables customers to continue to use both their existing Single-Ended and their existing Differential interface cards, but it requires some caution in connecting the disk device to the right interface.

Since an Inquiry by a utility such as ODE MAPPER returns the same Product ID (for example, ST34573WC) for both the Single-Ended and the Differential version of the disk, you must use the HP Model ID to distinguish the Single-Ended (for example, A34646A) from the Differential (for example, A3647A).

As a work-around for SYSGEN, use the "WC"-suffix ID (for example, ST34573WC) to bind to the Differential drivers; use the "N"-suffix ID (for example, ST34573N) to bind to the Single-Ended drivers.

CAUTION   It is critical that you connect the disk drive to the right bus: Single-Ended to Single-Ended; Differential to Differential. Failure to do so will result in unexpected system behavior.

HP Model	
Single-Ended	Differential
A3517A	A3318A
A3646A	A3647A

The following table details the Product IDs and the SCSI interface IDs.

Table 4-2   **Product ID for Configuring Disk Drives**

Drive Cap.	Drive Type	Product ID			Product ID for Corresponding Device Adapter
		Raw Mech	**Hot-Swap Module**	**SYSGEN**	
2 GB	SE	ST32272N	ST32272WC	ST32272N	HP27251A(CIO) HP28642A-SCSI(NIO) A1703-60003-SCSI(NIO) A2372-60003-SCSI(NIO) A2372-60004-SCSI(NIO)
2 GB	DF	ST32272WC	ST32272WC	ST32272WC	HP28696A(NIO) A2372-60003-FWSCSI(NIO) A2372-60004-FWSCSI(NIO)
4 GB	SE	ST34572N ST34573N	ST34572WC ST34573WC	ST34572N ST34573N	HP27251A(CIO) HP28642A-SCSI(NIO) A1703-60003-SCSI(NIO) A2372-60003-SCSI(NIO) A2372-60004-SCSI(NIO)
4 GB	DF	ST34572WC ST34573WC NA	ST34572WC ST34573WC DGHS04Y	ST34572WC ST34573WC DGHS04Y	HP28696A(NIO) A2372-60003-FWSCSI(NIO) A2372-60004-FWSCSI(NIO)
9 GB	SE	ST39173N	ST39173N	ST39173N	HP28642A-SCSI(NIO) A1703-60003-SCSI(NIO) A2372-60003-SCSI(NIO) A2372-60004-SCSI(NIO)
9 GB	DF	ST39173WC NA	ST39173WC DGHS09Y	ST39173WC DGHS09Y	HP28696A(NIO) A2372-60003-FWSCSI(NIO) A2372-60004-FWSCSI(NIO)
18 GB	DF	ST118273WC	ST118273WC	ST118273WC	HP28696A(NIO) A2372-60003-FWSCSI(NIO) A2372-60004-FWSCSI(NIO)

High Availability Disk Array (Nike)

With MPE/iX 5.5 PowerPatch 5 there is support for 9 GB and 18 GB disk drives. Please note the following restrictions.

1. The 9 GB and 18 GB drives are supported as RAID 1 only (that is, NOT as RAID 5, since this would exceed the 18 GB volume limit currently in effect.)

2. The 9 GB and 18 GB drives may be used as the boot device with patch MPEKX71.

NOTE   This is NOT recommended because only 4 GB is allowed by IODC in the current implementation. The 5 GB and 14 GB difference would be unusable capacity. The patch marks the space beyond 4 GB as unusable so the OS updates can occur without violating the 4 GB limitation.

3. Flare code 9.55.01 (or later) and PROM code 0173 (or later) are required for the array controller.

4. Support is for Model 20 Differential only.

5. HPC2300WD is the SYSGEN ID.

The following table shows the supported configurations:

Disk Capacity	RAID 1	RAID5	Boot
4 GB	X	X*	X
9 GB	X	NS	X**
18 GB	X	NS	X**
* Maximum 18 GB usable capacity per volume ** NOT recommended because 5 GB or 14 GB unusable			

The following table shows the product numbers and descriptions of available disk drives.

HP Product Number		Description
A3549A A3550A A3550AZ	#341	5 ea 9.1 GB DD
A3549A A3550A A3550AZ	#342	10 ea 9.1 GB DD
A3549A A3550A A3550AZ	#343	15 ea 9.1 GB DD
A3549A A3550A A3550AZ	#344	20 ea 9.1 GB DD

A3549A A3550A A3550AZ	#351	5 ea 18.2 GB HH DD
A3549A A3550A A3550AZ	#352	10 ea 18.2 GB HH DD
A3549A A3550A A3550AZ	#353	15 ea 18.2 GB HH DD
A3549A A3550A A3550AZ	#354	20 ea 18.2 GB HH DD

A3549A A3550A A3550AZ	#361	5 ea 4 GB LP DD
A3549A A3550A A3550AZ	#362	10 ea 4 GB LP DD
A3549A A3550A A3550AZ	#363	15 ea 4 GB LP DD

HP Product Number		Description
A3549A A3550A A3550AZ	#364	20 ea 4 GB LP DD

A5132A		9.1 GB single disk
A5132AM		9.1 GB multi-pack disk

A5133A		18.2 GB single disk
A5133AM		18.2 GB multi-pack disk

A5169A		4 GB single disk
A5169AM		4 GB multi-pack disk

A3716A 144 GB DDS-3 Autoloader for MPE/iX

by Aida Silva
Commercial Systems Division

Product Overview

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. 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 MB/s.

Support Information

The A3716A is a Single-Ended SCSI-2 Device. The A3716A is supported on MPE/iX Release 5.5 with following the patches:

- ODIJXJ0 (or later)

- MPEJXP1 (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.

Configuration

A3716A is supported only on MPE/iX NIO platform. It is not supported on MPE/iX CIO platform. 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.

Configuration 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 as shown in the following example:

```
io>ldev 31
LDEV:  31     DEVNAME:                OUTDEV:    0   MODE:
ID:  HPC1557A                         RSIZE:   128   DEVTYPE:  TAPE
PATH:  52.3.0                         MPETYPE:  24   MPESUBTYPE:  7
CLASS: TAPE

io>lp 52.3.0
PATH:  52.3.0                         LDEV:     31
ID:  HPC1557A                         TYPE:   TAPE
PMGR: SCSI_TAPE_DM                    PMGRPRI:  10
LMGR:   LOGICAL_DEVICE_MANAGER        MAXIOS:    0

io>lp 52.3

PATH:    52.3                         LDEV:
ID:   PSEUDO                          TYPE:   DA
PMGR: TRANSPARENT_MGR                 PMGRPRI:   6
LMGR:                                 MAXIOS:    0
```

Operating Instructions

If you need to store more than one tape per day, you should purchase multiple devices or consider other technology.

WARNING	**This device is intended for customers who need to store one tape per day (maximum 12 GB on DDS-3 tape). Writing more than one tape per day will exceed the duty cycle of this technology. This could result in premature failure of the device, and more importantly, you may face DATA LOSS due to unrecoverable errors.**

The A3716A device magazine holds up to 6 DDS tapes.

With this usage, it is recommended that you load the magazine with five DDS-3 cartridges in slots one to five and a cleaning cassette in the sixth slot. The device should then be configured with the auto tape eject feature. With this configuration and the magazine loaded with the recommended cartridges, you could then perform backups Monday through Friday, without any intervention. After backup on the fifth cartridge, the drive will be cleaned automatically by the cleaning cassette from the sixth slot.

You can then load new tapes into the magazine (or a new magazine) and start the next week's cycle.

To set the tape eject option, the autoloader switch setting must be six and the following command should be issued interactively, before backup is started (assuming your A3716A is LDEV #31):

`:run devctrl.mpexl.telesup 31 eject=enable`

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 MB fast-wide differential bus to replace the 5MBs back-end single-ended bus, larger integrated cache, and other features to boost capacity and performance.

<table>
<tr><td>NOTE</td><td>This is NOT an HP or customer installable device.</td></tr>
</table>

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.55.05 patches available on PowerPatch 5:

- MPEJXG6 (or supersede)—(Busy Bit) for 5x63 microcode and $FD

- ODIJXJ0 (or supersede)—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

5   Technical Articles

This chapter contains the following technical articles:

- Year 2000 Web Page
- Year 2000 Enhancements for MPE/iX
- New Date Intrinsics for MPE/iX
- Year 2000 and Other Enhancements for VPLUS
- CI Enhancements
- FTP Enhancements
- Transact V and Transact/iX Enhancements
- Dictionary/3000
- COBOL II/iX Enhancements
- Inform/V
- HP ALLBASE/BRW
- QUERY/iX Enhancements
- IMAGE/SQL with TurboIMAGE/XL Enhancements
- New Features in ALLBASE/SQL Version G2
- Pseudo-Mapped Files Removed from ALLBASE/SQL
- TurboSTORE/iX II, TurboSTORE/iX 7x24 Patches

Year 2000 Web Page

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

`http://www.hp.com/year2000`

Year 2000 Enhancements for MPE/iX

by M Gopalakrishnan
Commercial Systems Division

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 JOBFILE as January 21, 2001:

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

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 Intrinsics

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=[[*volid*][,[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 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 this *Communicator 3000*.

HP MPE/iX 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 this *Communicator 3000*.

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 Intrinsics for MPE/iX

by M Gopalakrishnan
Commercial Systems Division

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.

Table 5-1   **Existing Date Formats**

Storage Type	# Bytes	Explanation	Sortable ?	Y2K Ready?
longint [a]	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 [b]	6	YYMMDD date	yes	no
ASCII	6	MMDDYY date	no	no
ASCII	6	DDMMYY date	no	no
ASCII	6	YYMMDD date YY:MM3000 date [c]	yes	yes
ASCII	6	MMDDYY date YY:MM3000 date	no	yes
ASCII	6	DDMMYY date YY:MM3000 date	no	yes

a. "integer" and "longint" are binary values.
b. "ASCII" means ASCII character code.
c. MM3000 dates are represented as in the MM3000 product which used
   the ASCII letters "A" - "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."

Table 5-2   **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.

Table 5-3 **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[a]	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

a. MM3000 dates are represented as in the MM3000 product which used the ASCII letters "A" - "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.

Table 5-4　　　　**Special Date Values**

Data Type	Format	Un-known	Invalid	Never	Needed	Expired	Illegal
4	YYYYDDD	0	0000367	0000368	0000369	0000370	0000371
18	YYYYMMDD	0	00000101	00000102	00000103	00000104	0000105
38	YYYYMMDD	"blank"	"0000010"	"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 beyond 1753. The calendar followed by the intrinsics ignores the fact that calendars in different countries changed at different times and some dates were dropped from the calendar (around the year 1753). The validity of dates prior to 1753 cannot be guaranteed. All 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.

It is recommended that character array based output date parameters are initialized to blanks before passing them as arguments to these intrinsics.

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

inputcode   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."

inputdate   varies for type by reference.

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

outputcode   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."

outputdate   returns the date as per the format chosen by the *outputcode* parameter.

                  See the table, "Supported Date Formats," for the supported datecodes and their layouts.

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 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 *cutoff* 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 200 .. 2049 and those in the range 50 .. 99 will be translated to 1950..1999. If you specify the *cutoff* 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*}]

Table 5-5   **Valid Parameters for** *FormatElement*

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,…)

Table 5-6   **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. *fmtdatelen* parameter is length of *formatspec* parameter including the null termination character.

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* minus *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."

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.

offset   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.

outputdate   is the output date. The result of the date offset
operation. The interpretation depends upon the value
of *datecode*. See the table, "Supported Date Formats," for
the supported datecodes and their layouts.

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 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

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.

cutoff   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 *cutoff* parameter).

result   is a 32-bit signed integer (assigned functional return).

This value will be 0 if the *inputdate* conforms to the date
format represented by *datecode*. 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.

Pascal Example of New Date Intrinsics.

```
$standard_level 'hp_modcal'$

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_yyddd_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_mmddyy_fmt        = 16;
      hp_dt_int_ddmmyy_fmt        = 17;
      hp_dt_int_yyyymmdd_fmt      = 18;

      hp_dt_ascii_yymmdd_fmt      = 25;
      hp_dt_ascii_mmddyy_fmt      = 26;
      hp_dt_ascii_ddmmyy_fmt      = 27;

      hp_dt_mm3000_yymmdd_fmt     = 35;
      hp_dt_mm3000_mmddyy_fmt     = 36;
      hp_dt_mm3000_ddmmyy_fmt     = 37;
      hp_dt_ascii_yyyymmdd_fmt    = 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           : integer;

function HPCALENDAR : integer; intrinsic;
procedure HPFMTCALENDAR; intrinsic;
function HPDATEVALIDATE : integer; intrinsic;
procedure HPDATEFORMAT ; intrinsic;
procedure HPDATECONVERT ; intrinsic;
procedure HPDATEOFFSET ; intrinsic;
procedure HPDATEDIFF ; intrinsic;

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

     date2_pac := '960121'; {The YYMMDD date in Supported format '25'.}
     date1     := 230196;   {The DDMMYY 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 YYMMDD 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);

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!
```

COBOL Example of `HPDATECONVERT`.

```
 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 NOT = 0
       PERFORM DISPLAY-ERROR
    ELSE
       DISPLAY "Convert Date in YYYYMMDD = " OUTDATE-YYYYMMDD
    END-IF
    STOP RUN.

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

Year 2000 and Other Enhancements in VPLUS

by Kumar KN and Rajesh Channabasavaiah
Commercial Systems Division

Introduction

This article provides information on the following VPLUS enhancements available in version B.06.07.

- Turn-of-century enhancement

- Ability to define default Field Type and Data Type for a Form

- Ability to define more than 52 single character fields per Form

- New intrinsics to get and set Save Field values programmatically

- Ability to Renumber a Form with Interactive `FORMSPEC`

A description of these features follows.

Turn-of-Century Enhancement

VPLUS version B.06.07 contains enhancements to handle Year 2000 and beyond in applications. These enhancements can be grouped into three segments as follows:

- Intrinsic enhancements

- `FORMSPEC` enhancements

- Processing enhancements

Intrinsic Enhancements

Two new intrinsics have been added to VPLUS to handle dates in the next century. These are `VGETYYYYMMDD` and `VPUTYYYYMMDD`. These intrinsics are similar in structure and function to the existing intrinsics `VGETYYMMDD` and `VPUTYYMMDD` respectively. The difference is that the date values handled by the new intrinsics contain 4-digit year components.

Intrinsic VGETYYYYMMDD

Converts data in the data buffer to a eight-byte character value and copies it to an application.

Syntax

`VGETYYYYMMDD` {*comarea,fieldnum,variable*}

Parameters

comarea	must be *comarea* name specified when the forms file was opened with VOPENFORMF. If not already set, the following *comarea* items must be set before calling VGETYYYYMMDD:

	cstatus	set to zero.
	comarealen	set to total number of two-byte words in *comarea*. Must be at least 70 words in length.

VGETYYYYMMDD may set the following *comarea* item:

	cstatus	set to nonzero value if call is unsuccessful or requested field has an error.

fieldnum	is a two-byte integer variable which specifies the number assigned to the field by FORMSPEC
variable	is a character string in the application into which the converted value is placed

Discussion

VGETYYYYMMDD transfers the contents of the field specified by *fieldnum* to a variable in an application. The contents of the field are stored in a data buffer from which the value is taken. This value is converted to YYYYMMDD format and the converted value is placed in the application variable. The YYYYMMDD format is an 8-byte ASCII value, for example, 19961225.

If errors occur during conversion, *cstatus* is set to an error code. If the requested field has an error, its value is moved to the variable but *cstatus* is set to an error code.

Refer to the VGETtype intrinsic description in the *Data Entry and Forms Management System VPLUS Reference Manual (32209-60002)* for related information.

Example

```
COBOL
CALL "VGETYYYYMMDD" USING COMAREA FIELDNUM VARIABLE.

SPL
VGETYYYYMMDD(COMAREA,FIELDNUM,VARIABLE);
```

These calls will convert the contents of the data buffer corresponding to the field specified by FIELDNUM and place the converted value into VARIABLE.

Intrinsic VPUTYYYYMMDD

Converts a numeric value representing a date from an application and copies the converted value to a field in the data buffer in memory.

Syntax

VPUTYYYYMMDD {*comarea,fieldnum,variable*}

Parameters

comarea   must be *comarea* name specified when the forms file was opened with VOPENFORMF. If not already set, the following *comarea* items must be set before calling VPUTYYYYMMDD:

cstatus	set to zero.
comarealen	set to total number of two-byte words in *comarea*. Must be at least 70 words in length.

VGETYYYYMMDD may set the following *comarea* item:

cstatus	set to nonzero value if call is unsuccessful or requested field has an error.
numerrs	may be decremented if the new value replaces the value of a field with an error.

fieldnum   is a two-byte integer variable which specifies the number assigned to the field by FORMSPEC.

variable   is a character string in the application which contains the value to be converted.

Discussion

VPUTYYYYMMDD converts the contents of the application variable to the date order of the field specified by *fieldnum* and copies the converted value into the corresponding field in the data buffer, right justified. The application variable should contain a numeric value in YYYYMMDD format. The YYYYMMDD format is an 8-byte ASCII value, for example, 19961225. If errors occur during conversion, *cstatus* is set to an error code.

Refer to the VPUTtype intrinsic description in the *Data Entry and Forms Management System VPLUS Reference Manual (32209-60002)* for related information.

Example

```
COBOL
CALL "VPUTYYYYMMDD" USING COMAREA FIELDNUM VARIABLE.

SPL
VPUTYYYYMMDD(COMAREA,FIELDNUM,VARIABLE);
```

These calls will convert the contents of `VARIABLE` to a date value and place the converted value in the data buffer corresponding to the field specified by `FIELDNUM`.

FORMSPEC Enhancements

`FORMSPEC` Application Ready Buffer (ARB) processing has been enhanced with the addition of a new ARB data type. This data type can be specified on the DTC Menu for Data Type Conversions from Screen Type to Application Type. The new ARB Type is `YYYYMD` and is defined as a 8-byte ASCII field containing a date value in `YMD` order with no separators. Also the year component in the date value has 4 digits. Note that `FORMSPEC` will ensure a length of 8 for the data type `YYYYMD` in the ARB. In essence, the value is of the form `YYYYMMDD` (8 bytes) even though the Type designator on the DTC Menu has been abbreviated to `YYYYMD`.

For more details on ARB Types and how to use them, refer to the "ARB Data Types" section in the *Data Entry and Forms Management System VPLUS Reference Manual (32209-60002)*.

Processing Enhancements

Date processing in VPLUS has been enhanced to handle date values with 4-digit year components. Specifically, the following additions have been made.

Date output will be formatted with 4-digit years if the target field is wide enough to receive the date, and the user requests this action. Examples of such output include the `SET` statement in `FORMSPEC`. The target field should be at least 10 characters in length to have a 4-digit year output.

VPLUS, by default, will format date output with 2-digit years. To change the behavior as described above, define a JCW called `VSETNEXTCENTURY` and set bit 15 to 1 (bits are numbered from 0 through 15). It should be noted that this option will be active for fields wide enough to hold the 4-digit year date. For other fields, 2-digit years will be used.

Existing 2-digit years can be interpreted differently to handle dates in the next century. Currently VPLUS handles year components 00-99 as 1900-1999. This scheme of interpretation can be changed. In the new scheme, 00-49 is interpreted as 2000-2049 and 50-99 is interpreted as 1950-1999. This interpretation will be valid through all processing done

by VPLUS. For example, this scheme will allow the user to enter 02/29/00 (MDY order) in an existing application and have it processed correctly.

To invoke the above method of date interpretation, define a JCW called `VSETNEXTCENTURY` and set bit 14 to 1. Note that this method is used for processing only 2-digit year dates and not 4-digit year dates. Note also that the display of 2-digit year dates is not affected by this enhancement. This means that 12/11/10 (MDY order) can refer to 1910 in one application and 2010 in another application.

NOTE   When defining the JCW `VSETNEXTCENTURY`, set all unused bits to 0.

Programmable Defaults for Field and Data Types in FORMSPEC

`FORMSPEC` version B.06.07 has been enhanced to provide the capability to set the default Field Type and Data Type of a form. This capability can be used to reduce the time taken to complete the design of a new form.

Two new fields for user input have been added to the Form Menu screen of `FORMSPEC`. One is for the default Field Type, and the other is for the default Data Type.

On the Form Menu, the user typically specifies the name of the form and other characteristics of the form. In addition now, the user can specify the default Field Type and Data Type for the form. For a list of the allowable values for the Field Type and Data Type, refer to the *Data Entry and Forms Management System VPLUS Reference Manual (32209-60002)*. All fields created newly for the form will have the Field Type and Data Type set to the default values supplied on the Form Menu.

Notes

- The Field Type and Data Type values supplied on the Form Menu are applicable only to fields created for the form thereafter. This means that if these default values are changed for an existing form, the types of the existing fields are not changed. For a new form, the default values apply to all the fields created for the form.

- Cloned forms have their screen already designed; hence, the values on the Form Menu do not apply to such forms.

- The Field Type and Data Type can be specified independently. This means that either, or both, fields on the Form Menu can be blank.

  - If the Field Type is blank, all newly created fields for the form will have a Field Type of "O."

- If the Data Type is blank, all newly created fields for the form will have a Data Type of "CHAR."

Ability to Define More than 52 Single Character Fields per Form

This enhancement allows more than 52 single character fields to be defined for a Form. Currently the field tag which is defined for a field at screen design (creation) time should begin with an alphabetic character. This enhancement will allow several other characters to be used for the field tag, thus increasing the number of single character fields that can be defined for a Form.

The following are the other characters which can be used for the field tag (defined at screen creation time):

Digits   0,1,2,3,4,5,6,7,8,9

Specials   @,#,$,%,&,*,-,+,<,>,/,\,!,|,=,?,;,_

Field Menu Initialization

In the case where one of the characters listed above (a Digit or a Special) is used in a single character field, the Field Name will be blanked out when the Field Menu is invoked for the field the first time. The user should input a new name for the field in this case. With this enhancement, a maximum of 80 single character fields can be defined for a Form.

Compatibility issues

Forms created with previous versions of FORMSPEC can be used with this version without any modifications.

Forms created with this version of FORMSPEC can be used with existing applications without any modifications.

Forms created with this version of FORMSPEC can be used with previous versions of FORMSPEC. However, some extra characters may appear near the bottom of the Form Menu screen which should be ignored. These extraneous characters appear only once, the first time the Form Menu is invoked for the form, and they do not affect the functionality in any way.

New Intrinsics to Get and Set Save Field Values

Two new intrinsics have been added to VPLUS to enable applications to access Save Fields. They are VGETSAVEFIELD and VPUTSAVEFIELD. The values of Save Fields can be accessed/changed using the new intrinsics. It is important to note that the values of the Save Fields cannot be modified in the forms file by these intrinsics.

Intrinsic VGETSAVEFIELD

Copies contents of the specified save field from save field buffer in memory to an application.

Syntax

VGETSAVEFIELD {*comarea,sfname,sfbuf,buflen,actualen*}

Parameters

comarea　　　　must be *comarea* name specified when the forms file was opened with VOPENFORMF. If not already set, the following *comarea* items must be set before calling VGETSAVEFIELD:

    cstatus　　　　set to zero.

    language　　　set to the code identifying the programming language of the calling program.

    comarealen　　set to total number of two-byte words in *comarea*. Must be at least 70 words in length.

    VGETSAVEFIELD may set the following *comarea* item:

    cstatus　　　　set to nonzero value if call is unsuccessful.

sfname　　　　is a character string in the application which contains the name of the save field.

sfbuf　　　　is a character string in the application to which the contents of the save field is copied.

buflen　　　　is a two-byte integer variable which specifies the number of bytes in *sfbuf*.

actualen　　　is a two-byte integer variable which specifies the number of bytes actually moved to *sfbuf*.

Discussion

VGETSAVEFIELD transfers the contents of the save field specified by *sfname* to a variable in an application.

If the number of bytes specified by *buflen* is less than the length of the save field, the rightmost bytes are truncated. Following a successful transfer, *actualen* is set to the actual number of bytes transferred to *sfbuf*.

VGETSAVEFIELD does not convert the data moved to the application variable.

Example

```
COBOL
CALL "VGETSAVEFIELD" USING COMAREA SFNAME SFBUF BUFLEN ACTLEN.

SPL
VGETSAVEFIELD(COMAREA,SFNAME,SFBUF,BUFLEN,ACTLEN);
```

These calls will transfer the contents of the save field specified by
SFNAME to SFBUF. If the call is successful, ACTLEN will contain the
exact number of bytes transferred.

Intrinsic VPUTSAVEFIELD

Copies data from an application to the specified save field in memory.

Syntax

VPUTSAVEFIELD {*comarea,sfname,sfbuf,buflen,actualen*}

Parameters

comarea Must be *comarea* name specified when the forms file was
opened with VOPENFORMF. If not already set, the
following *comarea* items must be set before calling
VPUTSAVEFIELD:

cstatus	set to zero.
language	set to the code identifying the programming language of the calling program.
comarealen	set to total number of two-byte words in *comarea*. Must be at least 70 words in length.

VPUTSAVEFIELD may set the following *comarea* item:

cstatus	set to nonzero value if call is unsuccessful.

sfname is a character string in the application which contains
the name of the save field.

sfbuf is a character string in the application whose contents
are copied to the save field in memory.

buflen is a two-byte integer variable which specifies the
number of bytes in *sfbuf*.

actualen is a two-byte integer variable which specifies the
number of bytes actually moved to the save field in
memory.

Discussion

VPUTSAVEFIELD transfers the contents of the application variable
specified by *sfbuf* to the save field in memory specified by *sfname*.

If the length of the save field is less than the length specified by *buflen*, the rightmost bytes are truncated. Following a successful transfer, *actualen* is set to the actual number of bytes transferred to the save field in memory.

Note that the contents of the save field as recorded in the forms file is not changed by a call to VPUTSAVEFIELD. Also VPUTSAVEFIELD does not convert the data moved to the save field in memory.

Example

```
COBOL
CALL "VPUTSAVEFIELD" USING COMAREA SFNAME SFBUF BUFLEN ACTLEN.

SPL
VPUTSAVEFIELD(COMAREA,SFNAME,SFBUF,BUFLEN,ACTLEN);
```

These calls will transfer the contents of the application buffer SFBUF to the save field in memory specified by SFNAME. If the call is successful ACTLEN will contain the exact number of bytes transferred.

Renumbering a Form with Interactive FORMSPEC

FORMSPEC provides an option to renumber forms interactively. A new command has been added to the Interactive FORMSPEC to renumber forms. This command is invoked by selecting option "N" on the Main Menu. The form to be renumbered should be specified in the field next to the keyword "N". (This field is the same as where you would specify a form when also selecting the "D" option.) This option is equivalent to the RENUMBER command in batch mode FORMSPEC.

CI Enhancements

by M Gopalakrishnan and Jeff Vance
Commercial Systems Division

Introduction

This article describes the CI enhancements introduced in MPE/iX Release 5.5. The most notable enhancement is that POSIX filenames support additional special characters. The `COPY` command has been enhanced and several new CI evaluator functions have been added.

POSIX Filenames

Prior to this enhancement, characters in POSIX filenames were restricted to upper and lower case letters, numbers, dash, underbar and dot. Now, a POSIX filename can contain the following additional characters:

A POSIX filename may begin with any supported character except a dash. There are no changes to MPE filename rules.

Example

```
BUILD ./ab$cd:ef
LISTFILE ./ab$cb:ef,6
/SYS/PUB/ab$cd:ef

COPY ./ab$cd:ef  ./~anew%
LISTFILE ./[a~]@
ab$cd:ef  ~anew%
```

Enhancements to the COPY Command

The `COPY` command has been enhanced to accept MPE-syntax and POSIX-syntax directory names for the TO= parameter. Additionally, a space is an accepted delimiter between the FROM= and TO= arguments.

For example (assume the CWD is /SYS/PUB):

```
NEWDIR mydir
NEWDIR ./mydir
BUILD ./foo

COPY ci,   boo       creates BOO.PUB.SYS, as always
COPY ci    mydir     creates /SYS/PUB/MYDIR/CI
COPY ci    ./mydir   creates /SYS/PUB/mydir/CI
COPY ci,   /SYS      creates /SYS/CI
COPY ./foo mydir     creates /SYS/PUB/MYDIR/foo
COPY ./foo ./mydir/  creates /SYS/PUB/mydir/foo
```

File equations are supported for both the FROM= and TO= parameters. However, the TO= name cannot be a back reference to a file equation who's actual filename is a directory. As before, symbolic links are accepted as the source and/or target filenames.

New Evaluator Functions

`ANYPARM()` returns any combination of characters as a string. String function.

Syntax

`ANYPARM(`*anything*`)`

This new function accepts all characters, including, commas, quotes, right parentheses, etc. and treats them as a single string value. This is handy when used in conjunction with the `ANYPARM` parameter type. The one exception is that the "!" character, which introduces a CI variable.

The `ANYPARM()` function has two important restrictions:

- It must be the last (right-most) or only function on a command line

- It may not be nested inside other function arguments. For example, `lft(anyparm(…),1)` is not supported, nor is `quote(anyparm(…))`.

These constraints are necessary since `ANYPARM()` accepts all characters as its argument, including right parentheses.

Examples

Assume the script is named "doit":

NOTE

In the last example, if one of the characters is a "!" then the CI will attempt to reference a variable that follows the "!". For example, if the "f" is replaced by a "!" then the CI will try to reference a variable named HI. If this variable is not found a CI warning is reported.

`BASENAME()`—returns the filename component of a POSIX or MPE filename. String function.

Syntax

`BASENAME(`*str*`[,`*suffix*`])`

This new function returns the base component (file part only) for the passed filename. *str* can be any filename in MPE or POSIX syntax. *suffix* is an optional string, that if supplied and the filename base ends in *suffix* then the *suffix* is removed from the `BASENAME`. *suffix* applies to MPE and POSIX names. The *suffix* portion of the `BASENAME` will not be deleted if this results in deleting the entire basename.

Examples

```
CALC basename('a.b.c')
A

CALC basename('/a/b/c')
c

CALC basename('./a/b')
b

CALC basename("./a.sl",".sl")
```

```
a

CALC basename('/')
/

CALC basename("*feq")
*FEQ

CALC basename('$null')
$NULL

CALC basename('abc.g','c')
AB

CALC basename('/usr/lib/liby.a','.a')
liby

CALC basename('/usr/lib/liby.a','liby.a')
liby.a
```

DECIMAL( )returns a string value of an integer. String function.

Syntax

```
DECIMAL(int)
```

This new function behaves analogously to the existing `hex()` and `octal()` functions. It accepts a single integer argument and functionally returns the decimal string equivalent.

Examples

```
CALC decimal(255)
255                     < - - as a string

CALC len(decimal($ff))
3, $3, %3

setvar i 0
while setvar(i,i+1) < 10 and      finfo("FILE"+DECIMAL(i), 'exists') d
o
...
```

The last example is significant since this is the only case where explicit variable substitution ('!*varname*') does not work as an alternative to decimal. If the `finfo()` line was written as:

```
finfo("FILE"+"!i", 'exists') do
```

then the first iteration would call FINFO on a file named 'FILE0' rather than a file named 'FILE1'. This is because explicit variable dereferencing is performed early by the CI, well before the `setvar()` function is called.

DIRNAME( )—returns the directory components of a POSIX or MPE filename. String function.

Syntax

```
DIRNAME(str)
```

This new function returns the directory components for the passed filename. *str* can be any filename in MPE or POSIX syntax. MPE filenames are converted to POSIX names before extracting the directory portion of the name.

Examples

```
CALC dirname('a.b.c')
/C/B

CALC dirname('/a/b/c')
/a/b

CALC dirname('./a/b')
./a

CALC dirname("./a.sl")
.

CALC dirname('/')
/

CALC dirname("*feq")
[empty string]

CALC dirname('$null')
[empty string]

CALC dirname('abc.g')
/SYS/G

CALC dirname(fqualify('./a'))
/ACCOUNT/GROUP    # when the CWD is your logon group   or
/CWD              # when the CWD is a directory
```

FQUALIFY() returns a fully qualified filename. String function.

Syntax

```
FQUALIFY(str)
```

This new function returns the qualified filename for the passed filename in *str*. *str* can be any filename in MPE or POSIX syntax. *str* is first attempted to be translated into an MPE name. If that fails it is converted into a POSIX name. POSIX names are prepended with the CWD for relative pathnames.

Examples

```
CALC fqualify('a')
A.GROUP.ACCOUNT    # when the CWD is your logon group   or
/CWD/A             # when the CWD is a directory

CALC fqualify('a.b')
A.B.ACCOUNT

CALC fqualify('a.b.c')
A.B.C

CALC fqualify('./a')
/ACCOUNT/GROUP/a

CALC fqualify('./A')
A.GROUP.ACCOUNT    # when the CWD is your logon group   or
/CWD/A             # when the CWD is a directory

CALC fqualify('/a/b/c')
/a/b/c

CALC fqualify('*a')
*A

CALC fqualify('$null')
$NULL
```

```
CALC dirname(fqualify('./a'))
/ACCOUNT/GROUP      # when the CWD is your logon group   or
/CWD                # when the CWD is a directory
```

FSYNTAX() returns the syntax of the passed filename argument. String function.

Syntax

FSYNTAX(*filename_str*)

This new function accepts an MPE or POSIX style filename and evaluates the filename's syntax, if wildcard characters are present, lockwords, back reference to a file equation, remote environment IDs and system defined files. If an error occurs the error number, following the word "ERROR" is returned. Possible return values are:

- "MPE" "MPE;WILD" "MPE;LOCK" "MPE;FEQ" "MPE;$FILE" "MPE;REMOTE" and combinations where appropriate, for example, "MPE;LOCK,REMOTE"

- "POSIX" "POSIX;WILD"

- "ERROR=nnn" for syntax error nnn

Examples

```
fsyntax('a.b.c')
MPE
fsyntax('/a/b/c')
POSIX
fsyntax('./ab@/c')
POSIX;WILD
fsyntax('$null')
MPE;$FILE
fsyntax('a.b.c.d')
ERROR=426

if word(setvar(_error, FSYNTAX(myfile))) = "ERROR" then
   echo Syntax error: ![word(_error,,-1)]
endif

if word(FSYNTAX(a_file)) = "MPE" then
   # MPE filename
elseif word(FSYNTAX(a_file)) = "POSIX" then
   # POSIX filename
else
   # error

if pos("WILD", FSYNTAX(another_file)) > 0 then
   # wildcarded MPE or POSIX filename
```

QUOTE()–returns the input string with all quotemarks doubled. String function.

Syntax

QUOTE(*str*)

This new function scans *str* and doubles all single and double quote characters found. This is useful when *str* is passed to CI parameter that requires its own quotes, like an INFO= string, or a quoted string value in the SETVAR command.

Examples

```
calc QUOTE('abc"de""f')
abc""de"""""f

calc QUOTE("xyz'z'y'")
xyz''z''y''

calc anyparm(abc'def"ghi)k)
abc'def"ghi)k
calc QUOTE(hpresult)
abc''def""ghi)k'

setvar x quote('ab"c')
showvar x
X = ab""c
setvar x "!x"
showvar x
X = ab"c
```

The last example shows that the CI does quote folding in expressions, for example, "ab""c" is folded to the string.

XWORD()—returns a string less 'word'. String function.

Syntax

XWORD(*str*[,delims][,*nth*][,*end_var*][,*start*])

This new function locates the *nth* word from *str* beginning at *start*, delimited by one of the characters in *delims*, and places the index of the delimiter that terminated the word in a CI variable named by the *end_var* argument. It returns *str* minus the located word and minus the word delimiter. It is syntactically equivalent to word(), but opposite in semantics.

The arguments for xword() are identical to the word() parameters. Please refer to word() for details. HELP word and HELP xword are available.

xword() is useful for removing a token from a command line string since the token and its delimiter are not returned.

The same examples used for the word() function in an earlier article are repeated below for xword():

```
xword('file a=bb,old;rec=40,,f,ascii')              returns:
                           'a=bb,old;rec=40,,f,ascii'

xword('file a=bb,old;rec=40,,f,ascii',,2)          returns:
                           'file bb,old;rec=40,,f,ascii'

xword('file a=bb,old;rec=40,,f,ascii',";,",,j,8) returns:
                           'file a=old;rec=40,,f,ascii' and J=10

xword('file a=bb,old;rec=40,,f,ascii',,-4,j)       returns:
                           'file a=bb,old;rec=,f,ascii' and J=18
```

It may be illustrative to execute the above examples substituting word in place of xword(), or refer the word() examples referenced above.

Summary of CI Enhancements in C.55.03

- The `LISTF` and `LISTFILE` commands allow users to discover who is accessing files on their system, including remote accessors. `LISTFILE` output can now be filtered by selecting files based on access type or file code.

- Two new CI variables were added: `HPLASTSPID` and `HPSPOOLID`. Both contain the spoolfile ID in the form: O*nnnnnnn*.

- The `FLABELINFO` intrinsic and `FINFO`() function are able to return the number of sectors and extents a file occupies, file creation time, and the number of accessors to a file.

- The `PAUSE` command can now wait for one or more jobs to complete, to change state from waiting to executing, or to begin execution.

- The `INPUT` command allows the number of characters to read from `$STDIN` to be specified.

- The `PRINT` command is now able to display all data in a file, even if this data appears to be line numbers. The new `NONUM` option enables this feature.

LISTF/LISTFILE Enhancements

Syntax

```
LISTF [fileset][,listlevel][;listfile]

LISTFILE [fileset[,fileset[,...]]]
[[;FORMAT=]format_opt]
[[;SELEQ=]select_eq indir]
[[;NAME=] pattern]
[;PASS]
[;{PERM}{TEMP}{PERMTEMP}]
[;USENAME][;TREE][;NOTREE]
```

The `LISTF` and `LISTFILE` commands allow users to discover who is accessing files on their system, including remote accessors. The existing syntaxes remain the same; however, new format numbers, format mnemonics, and selection equations can be specified as a result of this enhancement.

New Formats

Two new formats have been added: format 8 or "access," and format 9 or "locks." Both the `LISTFILE` and `LISTF` commands support these formats, although only `LISTFILE` is able to use the format mnemonic. Format 8 shows greater access-related details compared to the existing format 3, as well as displaying individual file accessors. Format 9 is a superset of format 8. All of the format 8 data is shown with specific details about each process accessing the file, including information about which locks a process owns or is waiting for. A detailed description of the items displayed for the two new formats and examples are shown below.

The selection equation parameter of the LISTFILE command is also enhanced. The new SELEQ= choices allow selection of only files that are currently opened, in use, locked, or opened exclusively. If a file is not opened as specified in the selection equation, it is skipped. Also, files can be selected based on their file code number or mnemonic. The file code mnemonic PRIV includes all privileged files, that is, those files with a negative file code. More details on the access SELEQ choices appear later in this article.

Format 8 . Both formats 8 and 9 show the fully qualified filename or absolute pathname followed by the accessor summary line (described below). After the accessor summary line, there is one line of output per file accessor, consisting of:

- job/session number

- user name

- number of processes accessing, locking, writing and reading the file

- IP address for remote sessions, or the $STDLIST spoolfile ID for job accessors, or the LDEV number for local session accessors

Following is a format 8 example:

```
:listfile hppxudc.pub.sys,8
********************
FILE: HPPXUDC.PUB.SYS
15 Accessors(O:15,P:15,L:0,W:0,R:15),Share
#S265    MIKEP.HPE                 P:2,L:0,W:0,R:2      LDEV: 49
#S263    JEFFV,MGR.JVNM            P:3,L:0,W:0,R:3      LDEV: 47
#S261    KROGERS.MPENT             P:2,L:0,W:0,R:2      LDEV: 50
#S231    SUSANC.MPENT              P:2,L:0,W:0,R:2      LDEV: 46
#S219    FAIRCHLD.MPENT            P:2,L:0,W:0,R:2      LDEV: 39
#S214    CATHY,MGR.BOSS            P:2,L:0,W:0,R:2      REM : 15.14.16.19
8
#J434    FTPMON,FTP.SYS            P:2,L:0,W:0,R:2      SPID: #O21905
```

Analyzing the above format 8 example:

1. The filename shown is always fully qualified as it appears in format 6.

2. The line beginning with "15 Accessors…" is the accessor summary line, which shows the total number of accessors of the file across all jobs and sessions. This total includes all processes that have formally opened the file and all processes that have pseudo-opened the file. This is the same number seen in the FLAGS field when executing LISTF,3. The total number of accessors may not necessarily be derived by adding up the individual accessors displayed. This is because certain file accesses are done without using the file system and these accesses are transient and difficult to detect. An example of this kind of access is when STORE pseudo-opens a file to read its contents to backup.

   The "O:" field is the actual number of processes that formally opened the file (for example, called FOPEN, HPFOPEN, OPEN, or others). The O: number can be lower than the total number when the file has

been pseudo-opened, as done by `STORE`. Another factor is that the system protects certain files at boot time by pseudo-opening them, as is done on `NL.PUB.SYS`. Also, a `LISTF,8` done at the same time on the same file as a `LISTF,2` (for instance) will show the `LISTF,2` users as an accessor that has not formally opened the file. Typically, however, the O: count and total number of accessors match.

The total number of processes (P:) accessing the file should generally match the O: count.

The number of processes that have the file locked (L:) is seen next. The number of lockers includes all file system locks (semaphores) held by all detectable processes that are accessing the file. A single process can obtain multiple locks, and each time the count is incremented. When a process releases a lock, the count is decremented. The locks that are considered are the FLOCK semaphore (used by the FLOCK intrinsic), the OPEN semaphore (used by the file system), and the GUFD semaphore (used to control I/O).

The number of writers (W:) and readers (R:) are displayed next. The writer and reader counts can be greater than the number of accessors since a single process can open the same file multiple times. Also, when a process opens a file for append, update, or save (read/write) access both the writer and reader counts are incremented.

After all of the counts are shown, the file-sharing mode is displayed. This is either "Share" for files opened shared, or "Exclusive" for files opened exclusively. Typically when a file is opened exclusively there is only a single accessor; however, the POSIX `fork()` and `dup()` functions are exceptions.

The last item displayed in the access summary is the backup state. If the file is being stored, then "Storing" is shown. If the file is being restored, then "Restoring" is seen. If the file is being concurrently backed up, then "ConcurrBackup" is displayed. Otherwise, nothing additional is shown.

3. The lines after the accessor summary line reflect each job and session accessing the file. There is one line displayed for each job and session accessing the file, even if there are multiple processes within the job accessing the same file.

The first field shown is the job/session number. If the accessor is a system process then "system" is seen instead of a job ID. Following the job ID is the user name associated with the job. This consists of the `USER.ACCOUNT` preceded with an optional job name. The logon group name is not seen.

The same counts displayed in the accessor summary line are shown for each individual job accessor. The definitions are identical. Unlike the summary line, the reader and writer fields are truncated if the

process and locker counts are extremely high. In this event the right-most field is the first to be truncated and an "$" indicates that truncation occurred.

The last field displayed depends on whether the accessor is a job, a remote session or a local session. For jobs, the spoolfile ID is shown. For local sessions, the LDEV number is displayed. If the accessor is remote, the IP address is shown.

Format 9 . In addition to the format 8 data, format 9 shows more details such as:

- PIN number

- program name

- access method (for example, read, execute, or others)

- share mode (for example, `multi`, `gmulti`)

- current record number being accessed

- file number

- locks the accessing process owns and is waiting on

Following is a format 9 example:

```
:listfile hppxudc.pub.sys,9
********************
FILE: HPPXUDC.PUB.SYS
5 Accessors(O:5,P:5,L:5,W:0,R:5),Share
#S263   JEFFV,MGR.JVNM            P:3,L:3,W:0,R:3      LDEV: 47
 #P147   (LFCI.PUB.SYS)
   ACCESS: R-excl          REC#: 0                 FNUM: 13
    LOCKSOwner   Waiter
        FLOCK
        OPEN
 #P154   (CI.PUB.SYS)
   ACCESS: R-excl          REC#: 0                 FNUM: 13
    LOCKS: none
 #P86    (JSMAIN.PUB.SYS)
   ACCESS: R-excl          REC#: 336               FNUM: 16
    LOCKSOwner   Waiter
                 FLOCK

#J434   FTPMON,FTP.SYS           P:2,L:2,W:0,R:2      SPID: #O21905
 #P79    (CI.PUB.SYS)
   ACCESS: R-excl          REC#: 0                 FNUM: 14
    LOCKS: none
 #P47    (JSMAIN.PUB.SYS)
   ACCESS: R-excl          REC#: 336               FNUM: 15
    LOCKSOwner   Waiter
        OPEN        FLOCK
```

Analyzing the above format 9 example:

1. All of the format 8 data is seen.

2. The PIN (Process ID Number) begins the next line, followed by the process filename.

3. The next line shows the type of access which includes: "R" for read, "W" for write, "X" for execute, "A" for append, "L" for lock, "S" for save, "U" for update, "RD" for directory read access. The file-sharing mode is next. Values include: "sysexc" for system exclusive (rarely seen), "excl" for exclusive, "ear" for semi-exclusive (exclusive-allow-read), "shr" for shared, "multi" for shared job, or "gmulti" for system shared. Next the current record number is seen, and last is the file number.

4. If the process does not have any locks then "NONE" is displayed. If the PIN owns or is waiting on one or more of the relevant semaphores, then the semaphore name is shown under either the "Owner" or "Waiter" column. There are three semaphores (locks) considered.

    *   The `FLOCK` semaphore is obtained when a processes calls the `FLOCK` intrinsic. This lock is exclusive; meaning that only one process at a time can own the `FLOCK` semaphore for a file. Occasionally the `FLOCK` semaphore is used by the file system to synchronize directory access.

    *   The `OPEN` semaphore is locked by the file system when a file is being formally opened. In this usage the `OPEN` semaphore is locked exclusively. It is also used by the operating system to protect certain file system data structures when they are being read. This usage allows the `OPEN` semaphore to be locked in a shared fashion. The `OPEN` semaphore can be locked shared by multiple processes, but once it is locked exclusively all other PINs trying to lock it will wait.

    *   The `GUFD` semaphore is locked by the file system when it does I/O on the target file. It is locked exclusively.

Phantom Accessors . It was mentioned earlier about processes that pseudo-open (call `sm_open`) a file. `STORE` and `PROGEN` are examples of processes that bypass the file system by not calling `FOPEN` or `HPFOPEN` to open a file. This type of access is usually only evident by noticing that the total accessors count is greater than the open count (O:), and format 9 output typically will not show these phantom accessors.

There are also cases where a process does not open the file in any way at all, yet this process can lock one or more of the semaphores, and thus needs to be accounted for. `LISTFILE` and MPEX (when calling `AIFSYSWIDEGET` item 2065 to obtain a file's accessors) are examples of these kinds of phantom accessors. In this case the total accessor count is lower than what may be revealed by counting individual accessors shown in format 9.

If any of these phantom accessors lock one of the three important semaphores, then `LISTFILE` can detect the process and report reduced information about the access. If the phantom accessor process does not lock either the `FLOCK`, `OPEN`, or `GUFD` semaphores, then its access is

undetected. Following is a format 9 example of a file that another user is also executing a `LISTF,3` against. Notice that the total accessor count is one, yet we see two accessors. PIN 58 is the process that is doing the `LISTF,3` and thus is seen as a phantom accessor.

```
FILE: GOPI.PUB.SYS
1 Accessor(O:1,P:2,L:2,W:0,R:1),Share
#S3     JVREM,MANAGER.SYS           P:1,L:1,W:0,R:1    REM : 15.28.88.18
 #P62    (LOCKP.PUB.SYS)
   ACCESS: R-excl          REC#: 0                FNUM: 11
     LOCKSOwner    Waiter
          FLOCK

#S2     JVREM,MANAGER.SYS           P:1,L:1,W:0,R:0    REM : 15.28.88.18
 #P58    (CI.PUB.SYS)
   ACCESS: n/a            REC#: n/a              FNUM: n/a
     LOCKSOwner    Waiter
          OPEN
```

Restrictions . Formats 8 and 9 are available to all users, however only more privileged users get to see individual accessor information. The user having `SM` or `OP` capabilities can see all accessors of a file—assuming the user has `TD` access to the file itself. If the user does not have `SM` or `OP` capabilities, but does possess `AM` capability, and the file's GID matches the user's GID, then all accessors are displayed. If the user lacks `SM`, `OP` and `AM` capabilities, but is the owner of the file being listed, then the individual accessors are displayed. Otherwise only the accessor summary line is visible.

Even if the user is able to see individual accessors, certain format 9 fields have additional restrictions for security concerns: the program name and IP address are protected.

To see the program name the user must meet the same rules that apply to the `SHOWPROC` command. Specifically, one or more of the following applies:

- process is within the user's logon job/session.

- process' user and account names match the user's user and account names and the system's `JOBSECURITY` is set to `LOW`.

- user has `OP` or `SM` capability.

To see the IP address the user must meet the `SHOWCONN` utility rules, namely: the user must have `SM`, `OP`, `PM`, `NA`, or `NM` capabilities.

Selection Equations for LISTFILE

Selection equations for the `LISTFILE` command have been expanded. It is now possible to select files by whether and how a file is accessed, and by file code. The `ACCESS=` criteria supports 4 types of access, as shown below:

```
SELEQ=[ ACCESS = INUSE | OPEN | LOCK[ed] | EXCL[usive] ]
```

INUSE   is true if any processes are accessing the file. This
                  includes normal FOPEN-like accessors and sm_open
                  accessors. Phantom accessors (like LISTF,2) that don't
                  open the file at all are not noticed and thus INUSE
                  evaluates to false. INUSE examines the total number of
                  accessors field described above.

OPEN   is true if there are any processes that have formally
                  opened the file by calling FOPEN, HPFOPEN, RESET,
                  OPEN, or others. OPEN uses the same field that the "O:"
                  count is derived from.

LOCK   is true if any process has locked (even if they are
                  waiting) the FLOCK, OPEN or GUFD semaphores, or has
                  called the HPFLOCK pseudo intrinsic (for POSIX record
                  level locking).

EXCL   is true if the file was opened for exclusive access. EXCL
                  is true when LISTF,3 shows EXCL or LISTFILE,8
                  shows "Exclusive."

The CODE= criteria supports file code numbers and mnemonics as
shown below:

```
SELEQ=[ CODE = number | mnemonic | PRIV ]
```

The number can be positive or negative. PRIV will match all files with
negative file codes.

NOTE   A new FINFO item 61 ("accessors") returns the number of accessors for
            a file. This represents the total number of accessors, including all
            processes that open the file in any manner. This is the first value seen
            in the accessor summary line, which is described under the Format 8
            description. It is also the same value shown in the FLAGS field of a
            LISTF,3.

New Variables

Two new variables were added to MPE/iX CI:

- HPLASTSPID

- HPSPOOLID

HPLASTSPID is a read-only string variable. It contains the $STDLIST
spoolfile ID for the job defined by the HPLASTJOB variable, typically the
job most recently streamed in the session or job. Since HPLASTJOB
allows write access, it can be set to any job number, and then
HPLASTSPID will contain that job's $STDLIST spoolfile ID.

HPSPOOLID is a read-only string variable. It contains the $STDLIST spool ID of the currently streamed job. This variable is useful only in jobs—in a session it is an empty string.

FLABELINFO and FINFO Enhancements

The FLABELINFO intrinsic returns information from the file label of a disk file and the FINFO function is a CI interface to FLABELINFO. These APIs have been enhanced to return four new items:

- number of sectors occupied by a file

- number of extents occupied by a file

- file creation time

- number of accessors to a file

FLABELINFO

Table 5-7

Item Number	Type	Item Description
58	I16	Number of sectors occupied by the file.
59	I16	Number of extents occupied by the file.
60	I32	File creation time (CLOCK format).
61	I32	Number of accessor of the file.

FINFO accepts the following numbers and aliases:

Table 5-8

FINFO Number	Mnemonic	Return Type
58	SECTORS NUM SECTORS	Integer
59	EXTENTS NUM EXTENTS	Integer

FINFO Number	Mnemonic	Return Type
60	CREATETIME FMTCREATETIME CREATION TIME	String
-60	INTCREATETIME CREATION TIME INTEGER	Integer
61	ACCESSORS NUM ACCESSORS	Integer

Footnote: this is the same value as the total number of accessors in the LISTF format 8 summary line.

Examples

```
finfo('CI.PUB.SYS',"sectors")    = 192
finfo('CI.PUB.SYS',"extents")    = 2
finfo('CI.PUB.SYS',"createtime") = 11:08 AM
finfo('CI.PUB.SYS', -60)         = 110808
finfo('CI.PUB.SYS', "accessors") = 124
```

PAUSE Enhancement

Syntax

```
PAUSE [num_seconds]
[[;JOB=]jobid]
[[;INTERVAL=]interval_secs]
[;EXIST | WAIT | NOTEXIST]
```

The PAUSE command allows the current task to be suspended or "sleep" for a specified number of seconds. PAUSE now supports sleeping until one or more jobs reach a certain state. For example, a script can pause while selected jobs or sessions are executing (EXIST). Or, a job can sleep while another job is suspended or waiting (WAIT), and as soon as the job starts executing or terminates the pause completes. A session can pause while no jobs exist (NOTEXIST) and wake up as soon as the first job is launched.

In its simplest form, the PAUSE command sleeps for *num_seconds*, or less if **Break** is pressed. In this simple case no *jobid* is specified and all other command arguments are ignored. If the *jobid* parameter is specified, then *interval_secs* and the remaining command parameters have meaning. When *jobid* is supplied, PAUSE typically sleeps until the jobs or sessions matching *jobid* have terminated.

Options

EXIST   (default) means to pause while all jobs and sessions matching *jobid* exist. These jobs can be scheduled, waiting, executing, and other states; but, as long as the

system's global job table (JMAT) contains an entry for any of the selected jobs and sessions, the `PAUSE` command will continue to sleep.

WAIT

means to pause while the selected job or jobs are waiting. As soon as all the matching jobs are no longer waiting (meaning all the job states are no longer "introduced," "waiting," or "scheduled") the pause ends. The life cycle of a job is typically:

```
[sched or waiting->] intro-> initializing->
exec-> [susp-> exec->] terminate
```

Waiting jobs are considered all job states left of and excluding "initializing." Non-waiting jobs are all jobs right of and including "initializing."

NOTEXIST

means to pause while the matching job or jobs do not exist. As soon as any jobs matching *jobid* exist (in any state) the pause completes. `PAUSE` might miss finding very fast jobs. This is particularly true for a match on a single job/session number. A more practical use might be:

```
PAUSE job=@J;notexist
```

which means to sleep while no jobs exist. As soon as the first job is streamed the above pause stops.

Collectively `EXIST`, `WAIT`, and `NOTEXIST` are referred to as the "while_state," since `PAUSE` sleeps "while" the specified state condition is true.

Parameters

num_ seconds

if *num_seconds* is specified without *jobid*, `PAUSE` sleeps for that many seconds, or until the process issuing the `PAUSE` is interrupted by the break signal.

If *jobid* is also supplied then *num_seconds* has a different meaning. In this case it indicates the maximum duration for the `PAUSE` command, such that `PAUSE` should continue while the selected jobs are in their "while_state" or when *num_seconds* has expired, WHICHEVER IS SHORTEST. Thus, *num_seconds* represents the maximum length of the pause. If `PAUSE` completes but the one or more jobs are still in their "while state" a CIWARN is reported. Note: to pause while a job is in its "while_state" or until *num_seconds* has expired, whichever is LONGEST, execute the following two commands:

```
PAUSE x
PAUSE job=y ;z
```

If after X seconds job Y is still in state Z, then the
second `PAUSE` continues while state Z applies. On the
other hand, if after X seconds job Y is not in state Z
then the pause is complete.

jobid   can be one of: [#]Jnnn, [#]Snnn, [*jobname,*]*user.acct*, @,
@J, @S. Note if *jobname* is included then the *jobid* must
be quoted since the comma is a command token
delimiter.

If the `JOB=` parameter is specified then `PAUSE` sleeps
while *jobid* is in its "while_state." *jobid* can be an
executing, waiting, scheduled job, or a session. *jobid* can
also name many jobs or sessions. Wildcarding is
supported, and a non-wildcarded [*jname,*]*user.acct* can
match several jobs or sessions. The job name can be ","
or "@," to match all jobs or sessions without a job name.
When more than one job or session matches *jobid*,
`PAUSE` sleeps while all matching jobs are in their
"while_state." If the job executing `PAUSE` matches *jobid*
it will not be selected.

interval_ secs   if specified, `PAUSE` sleeps for this many seconds
between attempts to see if *jobid* is still in its
"while_state." Otherwise, `PAUSE` sleeps a variable
amount of seconds depending on the job state and the
number of previous times a particular job has been
polled. This computed method favors executing jobs
that terminate quickly.

Examples

The following pauses while job #J20 is in any of the JOB states such as
`INTRO`, `SCHED`, and `EXEC` until the job terminates:

```
:PAUSE JOB=#J20;EXIST
```

The following pauses while job #J24 is in `INTRO`, `WAIT`, or `SCHED` state.
`PAUSE` ends when #J24 is no longer in any of these states.

```
:PAUSE JOB=#J24;WAIT
```

INPUT Enhancement

The `INPUT` command was enhanced to accept the number of characters
to read from `$STDIN`. `INPUT` optionally writes *prompt* to `$STDLIST` then
reads *numchars* from `$STDIN` into *varname*, optionally waiting *seconds* for
the user to enter the data.

Syntax

```
INPUT [NAME=]varname
[[;PROMPT=]prompt]
[[;WAIT=]seconds  ]
[[;READCNT=]numchars]
```

The `READCNT` parameter is new. If *numchars* characters are specified then up to *numchars* bytes will be read from `$STDIN`. Of course the user can press the **Return** key at any time to complete the read. *numchars* is most useful for reads of only a few characters and exempts the user from being required to press the **Return** key. The default *numchars* is the maximum size of the CI's command buffer, currently 511 bytes. If *numchars* is specified and less characters are supplied as input, the user must still press the **Return** key to send the data.

Example

```
INPUT myvar,'Do you want to continue (Y/N)?', 10, 1
```

prompts to `$STDLIST`, and does a 10 second timed read on `$STDIN` of one character. When the user types the first character the read will complete and the read byte will be stored in the variable named MYVAR.

PRINT Enhancement

The `PRINT` command prints truncated records when an UNNUMbered file contains the trailing eight characters as digits. The `PRINT` command is not displaying the trailing eight characters as it assumes those digits as line numbers.

NOTE   This is the existing rule as of MPE/iX 5.5 for determining whether a file is a numbered file:

If the trailing 8 characters of at least 10 records of a file or all records (in the case of <10 records) are digits, then that file is considered as a numbered file, provided the record line numbers (trailing 8 character digits) are in ASCENDING order.

With this new option `NONUM`, the contents of the file would be displayed as it is, without assuming the trailing 8 characters as line numbers.

With this enhancement, the `PRINT` command syntax is

```
PRINT  [ [FILE=] filename]
[[;OUT=]outfile]
[[;START=] m]
[[;END=] n]
[[;PAGE=] p]
[;UNN | NUM ]
[;NONUM]
```

- With this new option, to print the line numbers as in the case of UNNUMbered files (that is, line numbers starting from 1 for the first record and so on), use this command:

  ```
  :PRINT infile;NUM;NONUM
  ```

- To consider the file as UNNUMbered file even when the file is a NUMbered file and then print the contents as it is in the file, use this command:

```
:PRINT infile;UNN;NONUM
:PRINT infile;NONUM
```

Examples

In this example, UFILEYES is an unnumbered file with trailing 8 characters as digits.

```
:PRINT UFILEYES
aaaaaaaaaaaa
bbbbbbbbbbbb
cccccccccccc
dddddddddddd
eeeeeeeeeeee
ffffffffffff
gggggggggggg
hhhhhhhhhhhh
iiiiiiiiiiii
jjjjjjjjjjjj
kkkkkkkkkkkk
llllllllllll
```

<table>
<tr><td>NOTE</td><td>The above file was considered by PRINT to be a numbered file and thus the trailing 8 bytes are truncated.</td></tr>
</table>

```
:PRINT UFILEYES;NONUM
aaaaaaaaaaaa00010001
bbbbbbbbbbbb00010002
cccccccccccc00010003
dddddddddddd00010004
eeeeeeeeeeee00020001
ffffffffffff00020002
gggggggggggg00020003
hhhhhhhhhhhh00020004
iiiiiiiiiiii00030001
jjjjjjjjjjjj00030002
kkkkkkkkkkkk00030003
llllllllllll00030004
```

In the above example, NONUM option displays all the data in the file.

FTP Enhancements

by Raghuram B.
Commercial Systems Division

The following is a description of the new features available with FTP/iX in PP5.

Support for Non-Standard MPE/iX File Types

The following types of MPE/iX files can be transferred to a remote host using FTP.

- Standard MPE files with fixed, variable or bytestream format.

- Files with either binary or ASCII data types.

In addition, the latest enhancements enable the transfer of the following types of non-standard types between two HP3000 machines.

- Privileged files (files with PRIV file code)

- Circular files

- Message files

- RIO files

- CM KSAM files

- NM KSAM files

- HPSPOOL files

The type of the file is transparent to the user. The user need not specify the file type being transferring.

Of these, SPOOL files are transferred as fixed ascii files. The above non-standard file transfers are not possible between an MPE and an non-MPE system. However, SPOOL files can be transferred as fixed ascii files from an MPE to a non-MPE system.

The client (FTP) and server (FTPSRVR) must be compatible to take advantage of this feature. The above file types can be transferred between like FTP/iX systems without the need for buildparms on the command line. Buildparms included on the command line are ignored if the file is a non-standard file type. The use of buildparms is accepted only for standard file types (file code of zero).

Non-standard file type transfers to incompatible FTP/iX versions result in a data transfer failure message. You cannot append to non-standard MPE/iX file types with the use of this new feature. Non-standard file

transfers to non-MPE/iX systems also result in a data transfer failure. Non-compatibility of client and server file transfer requests result in appropriate error messages.

Example

```
FILENAME   CODE   ------------LOGICAL RECORD-----------   ----SPACE----

               SIZE   TYP        EOF        LIMIT R/B  SECTORS #X MX

DB000     PRIV   128W   FB          6           6   1       16  1  1

ftp> get db000
200 PORT command okay
150 File: db000 opened; data connection will be opened
2816 bytes received in 0.02 seconds (130.95 Kbytes/sec)
ftp>

FILENAME   CODE   -----------LOGICAL RECORD-----------   ----SPACE----

               SIZE   TYP        EOF        LIMIT R/B  SECTORS #X MX

BIGRIO1          128B   FAR       8193        8192  3     4128  1  8

ftp> get bigrio1
200 PORT command okay
150 File: bigrio1 opened; data connection will be opened
1054934 bytes received in 3.69 seconds (279.27 Kbytes/sec)
ftp>
```

The Configurable Server Time-Out

Current patch releases of FTP/iX 5.0 and 5.5 have a hard-coded infinite time-out value applied to the data connection timer of the FTP/iX server. The FTP/iX client, which shares this data connection, has a configurable time-out which is user-controlled through the TIMEOUT command.

The latest enhancements provide a configurable server time-out. This allows the client side to control the time-out value that the server applies to its data connection timer. To set the time-out value, execute the SITE TIMEOUT command with a value from 0 (infinite) to 3000 seconds.

Syntax

```
SITE TIMEOUT [num-secs]
```

The time-out value indicates how long theFTP server should wait for a message from the remote FTP client before giving up. The allowable range is 0 to 3000. A value from 1 to 3000 indicates a time-out value in seconds. A value of 0 means no time-out (that is, wait forever). If num-secs is not specified, the current remote time-out value is displayed. Otherwise, this command sets the connection time-out to num-secs seconds.

Example

Command for the server to time out after 5 minutes (300 seconds).

```
ftp>site timeout 300
200 TIMEOUT command ok.
ftp>
```

Hash Command

While transferring files, if the user desires to receive progress status during file transmission, the hash command provides this capability. This is a user command at the client end, which allows the user to toggle state of the hash command. If the toggle state is `on' a hash symbol (#) is displayed every 1024 characters transferred. The default is set to `off'.

Example

```
ftp> hash
Hash mark printing on (1024 bytes/hash mark).
ftp> get TESTFILE
200 PORT command successful.
150 Opening BINARY mode data connection for TESTFILE (145915 bytes)

################################################################
################################################################
########
226 Transfer complete
145915 bytes received in 0.32 seconds (438.45 Kbytes/sec)
```

Transact V and Transact/iX Enhancements

by Kelly Sznaider
Support Technology Lab

Overview

Two new enhancements have been added to Transact/V version A.11.00 and also to Transact/iX version A.06.00

- Literal strings and numeric constants can now be passed as parameters in the PROC verb.

- Transact now provides a method for increased decimal precision in arithmetic expressions.

New Parameter Types

The parameter list for the PROC statement has two additional types of parameters. A programmer-defined character string can be passed by placing the string within quotes, similar to other languages. The default is to pass this literal as a 16-bit address. To change the literal to a byte address, precede the string with a "%", the same as with variables. The other parameter type is to pass a numeric value. The numeric constant is passed by value.

Examples

```
PROC ASCII (65, 10, %"abc", &(numchar));
PROC FWRITE (#FILEID(TRANOUT), "Hello", -5, 0);
```

Increased Decimal Precision

The second Transact enhancement increases decimal precision in arithmetic expressions (`IF`, `LET`, `REPEAT`, and `WHILE` verbs) via the new `!PRECISION` compiler command. A specified minimum number of decimal places is maintained for all packed-decimal intermediate results when calculating the value of an arithmetic expression. This minimum number is used only if the expression's default evaluation method would have maintained fewer decimal places.

Example

```
!PRECISION(5) maintains at least five decimal places.
```

Documentation

The detailed information for these enhancements has been included in the *HP Transact Documentation Update Notice* (32247-90028), which is also included in the latest edition of the *HP Transact Reference Manual* (32247-60003). A revised *HP Transact Quick Reference Guide* (32247-90020) is also available.

Dictionary/3000

by Kelly Sznaider
Support Technology Lab

Overview

Dictionary/3000 version A.03.00 offers two new enhancements in the DICTDBM utility:

- The capability of copying entities with the new COPY command.

- The ability to enter elements multiple times within an HP Inform group.

COPY Command for Entities

For the COPY command, a category, class, element, file, group, location, or procedure is copied to a new like entity with a different name. The entity's associated child entities and/or its related entities can also be copied if desired. This enhancement adds productivity improvements for the user.

Multiple Entries for HP Inform

The same element can now be added to an HP Inform group more than once, as long as the physical file designation or the parent file is different making the record unique. This makes elements in HP Inform groups easier to use and more straightforward. (Inform/V version A.11.00 or later is needed with this enhancement.)

Documentation

The detailed information for these enhancements has been included in the *Dictionary/3000 Documentation Update Notice* (32244-90013) which is also included in the latest edition of the *Dictionary/3000 Reference Manual* (32244-61000).

COBOL II/iX Enhancements

by Walter Murray
Support Technology Lab

Overview

COBOL II/iX with PowerPatch 5 on MPE/iX 5.5 provides several enhancements that were requested by Interex SIGCOBOL. Some of these enhancements have already been made available in various patches to MPE/iX 5.5.

Secondary Entry Points

COBOL II/iX now supports secondary entry points in a main program. To specify a secondary entry point, use the ENTRY statement, and give a nonnumeric literal with the desired entry point name. Specifying that entry point on the MPE/iX RUN command will cause execution to begin at that ENTRY statement. Any number of ENTRY statements may appear, as long as they specify different entry point names. The ENTRY statement is a nonstandard HP extension to the COBOL language.

The following sample program demonstrates the use of a secondary entry point.

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID. UPDATE-MASTER.
       DATA DIVISION.
       WORKING-STORAGE SECTION.
       01  SW-CLEAR-YTD-TOTALS    PIC X    VALUE "F".
       PROCEDURE DIVISION.
       0000-BEGIN.
      *    Enter here for routine processing.
           GO TO 1000-MAIN.
       ENTRY "YEAREND".
      *    Enter here for year-end processing.
           MOVE "T" TO SW-CLEAR-YTD-TOTALS.
       1000-MAIN.
      *    Update master file, optionally resetting
      *    year-to-date totals, etc.
           DISPLAY "Begin UPDATE-MASTER:  SW-CLEAR-YTD-TOTALS = ",
                   QUOTE, SW-CLEAR-YTD-TOTALS, QUOTE
      *    ...
           STOP RUN.
       END PROGRAM UPDATE-MASTER.
```

Suppose this program is compiled and linked into an executable program named UPDATE. The usual way to invoke it would be with this command:

```
RUN UPDATE
```

Alternately, it may be invoked through its alternate entry point, using this command:

```
RUN UPDATE,YEAREND
```

CALL by plabel

This enhancement provides an alternate way to achieve execution-time binding with the CALL statement. In standard COBOL, the CALL statement can use an identifier that specifies a nonnumeric data item containing the name of the desired subprogram. COBOL II/iX generates a call to the HPGETPROCPLABEL intrinsic to dynamically load the specified procedure, and that procedure is then executed. The XLs searched in this case are those in the binding sequence of the calling process.

There are situations where it may be desirable for the programmer to call HPGETPROCPLABEL explicitly, for example, when the XL to be searched is not known at load time or is not in the calling program's binding sequence. It is now possible to code your own call to HPGETPROCPLABEL, save the returned procedure label in a numeric data item, and then use that plabel in a CALL statement.

As an example, suppose that the following subprogram has been compiled and placed in an executable library named TESTXL.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. COBOLSUB INITIAL.
PROCEDURE DIVISION.
BEGIN.
    DISPLAY "In COBOLSUB"
    EXIT PROGRAM.
END PROGRAM COBOLSUB.
```

Using the functionality of call by plabel, this subprogram can now be dynamically loaded and invoked as follows.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. COBOL-MAIN.
DATA DIVISION.
WORKING-STORAGE SECTION.
01  PROCNAME        PIC X(10)    VALUE "-COBOLSUB-".
01  PLABEL          PIC S9(9)    COMP.
01  FIRSTFILE       PIC X(8)     VALUE "-TESTXL-".
PROCEDURE DIVISION.
BEGIN.
    CALL INTRINSIC "HPGETPROCPLABEL" USING
            PROCNAME, PLABEL, \\, FIRSTFILE
    CALL PLABEL
    STOP RUN.
END PROGRAM COBOL-MAIN.
```

The ability to use a numeric data item in a CALL statement is a nonstandard HP extension to the COBOL language.

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

- 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 that in COBOL II/iX, backslashes ("\") are used to indicate that a
parameter is passed by 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 "0000000000001111" TO BYTE-STRING.
    CALL "HP_BYTE_PACK" USING BYTE-STRING, RESULT-N, \2\.
    DISPLAY RESULT-N.
  * Results in +15.
```

Compiling Larger Programs

Internal data structures of the compiler have been expanded to permit compiling significantly larger programs. While there is no specific limit on the number of lines permitted in a source program, this version of the compiler should be capable of processing programs well in excess of 200,000 lines.

DISPLAY Index-name

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)`.

New Qedit Diagnostic

The compiler is now able to detect when a source file is in Qedit format and the Qedit product has not been properly installed on the system. Qedit is a full-screen text editor from Robelle Consulting Ltd. In the past, Qedit users have sometimes encountered problems when updating to a new release of MPE/iX and forgetting to reinstall Qedit. COBOL II/iX now detects this condition and produces error 472, `QEDIT FORMAT ENCOUNTERED FOR FILE`.

Inform/V

by Kelly Sznaider
Support Technology Lab

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

by Kelly Sznaider
Support Technology Lab

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.

QUERY/iX Enhancements

by James Overman
SSG Group R&D

Overview

The following topics in this article review the new enhancements for
QUERY/iX and some limitations on their use:

- B-Tree Enhancement

- Dynamic Dataset Capacities

- Wildcard Retrievals

- Wildcard Character Limitations

- B-Tree Limitations

- "NO MATCH" Enhancement

B-Tree Enhancement

The latest B-Tree enhancement to TurboImage/XL has required some
enhancements to the QUERY product to utilize the new features. With
the HP32216D.03.11 and N.03.11 releases of QUERY, the new B-Tree
access capabilities will be supported within QUERY on systems with
the B-Tree software release. QUERY is fully backward compatible with
the prior releases of TurboIMAGE/XL and QUERY.

New messages will be seen when B-trees are utilized by a database and
the `FORM` command is entered. The number of indexed sets, the
wildcard character, the status of the `BTREEMODE1` flag when OFF, and
the individual sets and items that have been indexed will be reported as
shown in the following example:

Figure 5-1   **FORM Example**

```
>FORM

DATA BASE: TOYDB                              MON, MAY 30, 1997, 10:48 AM
DATA BASE LANGUAGE ATTRIBUTE: NATIVE-3000
DATA BASE HAS 2 INDEXED SETS WITH WILDCARD CHARACTER= "~"    NEW
WARNING! DATA BASE BTREEMODE1 IS OFF.  WILDCARD DISABLED.    NEW

SET NAME:
   CUSTOMERS,MANUAL/INDEXED                                  NEW

      ITEMS:
          CUSTOMER-NO,        X6                             NEW
          CUSTOMER-NAME,      X30

      CAPACITY: 50              ENTRIES: 8

SET NAME:
   ORDER-MASTER,AUTOMATIC/INDEXED                            NEW

      ITEMS:
          ORDER-NO,           X8                             NEW

          CAPACITY: 503          ENTRIES: 16

SET NAME:
   INVOICES,DETAIL

      ITEMS:
          INVOICE-NO,         X6
          CUSTOMER-NO,        X6                             NEW
          ORDER-NO,           X8                             NEW
          AMOUNT,             P12

      CAPACITY: 5016         ENTRIES: 5000
   MAXIMUM CAPACITY: 5016   INITIAL CAP.: 38   INCREMENT: 38 NEW
```

Dynamic Dataset Capacities

The last line above shows another enhancement to QUERY that reflects
the Dynamic Dataset Capacity Expansion capability. When a dataset
has utilized the Dynamic Expansion feature, QUERY will display the
CURRENT, MAXIMUM, and INITIAL capacities and the
INCREMENT amount. The FORM SETS output has been slightly
modified such that the heading reflects that the Current capacity is
being reported rather than the Maximum capacity.

Wildcard Retrievals

QUERY will automatically invoke the B-Tree retrieval feature for the
FIND, LIST, SUBSET, and MULTIFIND commands when a wildcard
character is used in the search value. As QUERY may have up to ten
databases open and each could have a different wildcard, the wildcard
for the DEFINED database will be used when specified, else the first
wildcard specified by an open database will be used. QUERY uses
modes 21 or 24 for B-Tree DBFIND for best performance.

Wildcard Character Limitations

Certain possible wildcard characters are used by QUERY for syntax scanning and are limited in their usage. These special characters may be utilized when the data values are delimited by double quotes (for example, tilde, carat, comma, and parenthesis).

Figure 5-2 **Wildcard Examples**

```
>FIND CUSTOMERS.CUSTOMER-NO=01540~              Tilde is a limited wildcard

EXPECTED A CONNECTOR OR 'END'
INVALID CONNECTOR OR TERMINATOR!

>FIND CUSTOMERS.CUSTOMER-NO="01540~"            Tilde enclosed in quotes works

8 ENTRIES QUALIFIED

>FIND CUSTOMERS.CUSTOMER-NO=01540"              Double-quote will not work…
EXPECTED A CONNECTOR OR 'END'

>>
INVALID CONNECTOR OR TERMINATOR

>FIND CUSTOMERS.CUSTOMER-NO="01540"""           Unless doubled in quotes and
                                                double-quote is wildcard
8 ENTRIES QUALIFIED

>FIND CUSTOMERS.CUSTOMER-NO=01540*              If asterisk is NOT the wildcard for this
                                                database, then no match unless
                                                the value 01540* is in dataset.
0 ENTRIES QUALIFIED
```

B-Tree Limitations

QUERY will utilize the B-Tree functionality when possible for both character and binary data types. The wildcard feature is only implemented by TurboIMAGE/XL for character data types (X and U) and not for binary data types. Neither has the Not Equal operator been implemented for B-trees and so QUERY will continue to use the older access methods (normally serial reads) for Not Equal retrievals. Note that the LIST command always uses a serial read but will apply the wildcard search character for relations.

When a database has the BTREEMODE1 flag set to OFF, QUERY will not attempt to use the B-Tree functionality for Wildcard retrievals. This condition is reported by all forms of the FORM command.

"NO MATCH" Enhancement

The FIND, MULTIFIND, and SUBSET commands have been enhanced to accept a NO MATCH alternative such that all data records NOT MATCHING the pattern provided are returned by the commands. The new form of the FIND command is:

```
FIND [#LIMIT=I;] RELATION/ITEM IDENTIFIER [NO]
M[ATCHING] "PATTERN"[AND/OR RELATION/ITEM IDENTIFIER
[NO] M[ATCHING] "PATTERN"]... [END]
```

The `MULTIFIND` and `SUBSET` command syntax are similarly enhanced.

The `NO` may be followed by additional characters such that all of the following examples are accepted:

FIND PRODUCTS.NAME NO MATCH "ABC?*"

FIND PRODUCTS.NAME NOT MATCHING "ABC?*"

FIND PRODUCTS.NAME NONE M "ABC?*"

This NO Matching enhancement is NOT related to the B-Tree functionality and does not utilize the B-Tree retrieval mechanisms.

IMAGE/SQL with TurboIMAGE/XL Enhancements

by IMAGE/SQL Team
Commercial Systems Division

Overview

This article discusses enhancements as well as other important information commencing with the Express 3 release of MPE/iX 5.5. Express 3 (and later releases) also includes the revised editions of some selected manuals.

Highlights

Highlights of this release of IMAGE/SQL, version B.G3.00 or later, and TurboIMAGE/XL, version C.07.14 or later, are the following:

- Dynamic data set expansion for master sets (MDX or DDXM)

- Performance enhancement for databases with third-party indices

- 80 GB data sets

- Predicate locks when updating manual masters

- DBSCHEMA and P type with odd sub-item length

- B-Tree indices enhancement

- New 16-bit and 32-bit ODBC driver, ODBCLink/SE, bundled with IMAGE/SQL

- Support for third-party composite indices in IMAGE/SQL

- Scalability for DBPUT, DBDELETE, and DBUPDATE (CIU on)

- Dynamic detail data set expansion (DDX) fix

Manuals Bundled with IMAGE/SQL

The following manuals are included with IMAGE/SQL:

- *HP IMAGE/SQL Administration Guide* (36385-90001)
  (Revised August 1997)

- *Getting Started With HP IMAGE/SQL* (36385-90008)

- *TurboIMAGE/XL Database Management System Reference Manual* (30391-90001)
  (Revised August 1997)

- *ALLBASE/SQL Database Administration Guide* (36216-90005)

- *ALLBASE/SQL Reference Manual* (36216-90001)

- *ALLBASE/SQL Message Manual* (36216-90009)

- *ALLBASE/SQL Performance and Monitoring Guidelines* (36216-90102)

- *ISQL Reference Manual for ALLBASE/SQL and IMAGE/SQL* (36216-90096)

- *ODBCLink/SE Reference Manual* (36217-90406) (Revised June 1998)

Updating IMAGE/SQL

If you are updating from an earlier release of IMAGE/SQL and you have databases which are already `ATTACH`ed to one or more DBEnvironments, you must `DETACH` and `ATTACH` again in order to benefit from the enhancements. Also, it is recommended that you issue the `UPDATE STATISTICS` command of ISQL for each of the tables you access frequently. This may enable you to attain performance improvement.

Furthermore, if you have created SQL data using an earlier release of IMAGE/SQL and are now updating to a latter release, you must perform the ALLBASE/SQL migration using SQLINSTL or SQLMigrate. For more information, refer to the *ALLBASE/SQL Database Administration Guide* (36216-90005) and the *Read Before Installing* for this release.

TurboIMAGE/XL Enhancements

The following are TurboIMAGE/XL enhancements, independent of SQL access:

- Dynamic data set expansion for masters (MDX or DDXM)

- 80 GB data sets

- Performance Enhancement for databases with third-party indices

- DBSCHEMA and P type with odd sub-item length

- B-Tree indices

- Scalability for DBPUT, DBDELETE, and DBUPDATE (CIU on)

Dynamic Data Set Expansion for Masters

Dynamic data set expansion feature for non-jumbo (file size within 4 GB limit) detail data sets (DDX) was first released in MPE/iX 5.0. With this 5.5 PowerPatch 5 patch, this feature is available for non-jumbo master data sets (MDX or DDXM) as well. The feature allows dynamic expansion of a data set during DBPUT when the data set has approximated its current capacity and DBPUT would fail unless the data set is expanded. As in DDX, the capacity parameters which are maximum capacity, initial capacity, and increment (optional), used for dynamic expansion, must be set prior to the actual expansion. For new databases, these parameters can be specified in the CAPACITY statement of the schema definition to be processed by DBSCHEMA. For existing databases, third-party tools which support MDX need to be employed.

The capacity parameters are:

- maximum capacity

  is a required parameter and is a maximum number of entries the data set can contain. It must be less than or equal to $2^{31}-1$ (2,147,483,647). The maximum capacity is adjusted by TurboIMAGE/XL to represent a multiple of the blocking factor.

- initial capacity

  is a required parameter and is the initial, primary, or hashing capacity for the data set. It is the number of entries for which space will be allocated and initialized when the data set is created. More importantly, the initial capacity, and not the current capacity (initial capacity with expansions included), will be used in the hashing algorithm to calculate the primary address of the new entry being added. If initial capacity is very low, there can be frequent expansions which could result into both long synonym chains which degrade the performance of some TurboIMAGE intrinsics and disk fragmentation. Therefore, this number should be derived after

giving consideration to the expected volume of current data, anticipated growth, and synonym chain lengths. It should be such that a minimum number of expansions are necessitated. The initial capacity must be between 1 and $2^{31} -1$ inclusive but must be less than or equal to the maximum capacity. If it is not specified, or if it is either zero or equal to the maximum capacity, dynamic capacity expansion is not enabled for the data set, and maximum capacity is used for the data set file creation and primary address calculation. The initial capacity is adjusted to represent a multiple of the blocking factor.

- increment

  is an optional parameter and is either the number of entries or the percentage of the initial capacity by which the data set needs to be expanded each time. If a percentage is used, the percent sign (%) must follow the incremental amount. The increment is adjusted (reduced) for the last expansion so that it does not exceed the maximum capacity. This increment parameter can only be used if a valid initial capacity parameter is also specified. This number must be 1 to 100 inclusive for percent, or 1 to $2^{31} -1$ (2,147,483,647) inclusive for number of entries. If it is very low, there can be frequent expansions leading to severe disk fragmentation and performance degradation. If it is very high, DBPUT that triggers the expansion may take long time to complete which could impact other database users.

  If the increment is not specified for the data set, or is zero, but the initial capacity is greater than zero and not equal to the maximum capacity, the increment for each expansion defaults to ten percent (10%) of the initial capacity for the data set. If the initial capacity is equal to the maximum capacity, or the initial capacity is zero, it indicates that the data set cannot be expanded and increment is ignored.

The enhanced syntax for the CAPACITY statement of DBSCHEMA is same for both DDX and MDX, and is as follows:

```
{CAPACITY: | C:} maximum capacity [(blocking factor)]
[,initial capacity [,increment]];
```

Dynamic Expansion and Placement of a New Entry

The instant when the expansion is triggered and how record address is assigned to the new entry in a master data set varies greatly from detail data set. In a detail data set, expansion takes place during DBPUT when the free entry count for the set is zero. In case of a master data set, expansion is triggered during DBPUT when there is no free (unoccupied) entry in the close proximity (within an internally-maintained number of blocks) of the calculated primary

address of the new entry and the set is expandable. This means that an expansion takes place even when the free entry count is greater than zero.

Following the first expansion, the master data set can be perceived as having two areas, original area and expanded area, managed differently. The original area is managed in the same way as in releases prior to MDX, that is, the primary address for the new entry is calculated using the hashing algorithm and the secondary address is assigned according to the availability of the free entry, but with modifications in searching for a free entry. The use of the expanded area is controlled using the pointer to a delete chain head and the last entry used in the file (high-water mark) as in detail data sets. Subsequent expansions belong to the expanded area. The original area can have primary entries and secondary entries, while the expanded area can only have the secondary entries. Note that the primary capacity (also hashing or initial capacity) used to calculate the primary address does not change. In essence, the dynamic expansion results in allocating additional space for secondary entries which implicitly makes room for additional primary entries in the original area. The primary address of a new entry is calculated using the value of the entry's key item and primary capacity in the hashing algorithm. The new entry can reside either at its primary address in the original area, or in the close vicinity of the primary address in the original area, or in the expanded area.

To elaborate further, when a `DBPUT` for a manual master, or implied `DBPUT` to an automatic master, is processed, the address where the new entry resides is based on the following:

1. If the primary address in the original area is not occupied, the new entry resides there. The new entry also becomes a synonym chain head with a count of one.

2. If the primary address is already occupied by a primary entry which hashed to the same address, TurboIMAGE/XL scans a few blocks (predetermined based on internal values, also known as quick search) in the close proximity of the primary address to find a free entry (empty entry). If found, the new entry resides at this secondary address in the original area. Otherwise, search for a free entry is directed towards the expansion area. At this point, if the set has not been expanded at all and can be expanded, expansion will take place and the new entry will reside at the first entry address in the new expanded area.

   If prior expansion has already taken place, the pointer to the delete chain head and the last entry used in the file for the expansion area are interrogated to determine the secondary address in the expansion area for the new entry. When there is room, the new entry is assigned a secondary address in the expanded area, and the new

entry becomes the last entry in the synonym chain. If the expanded area is full and can be expanded, it is further expanded to accommodate the new entry.

When there is no room in the expanded area and cannot be expanded further, however, there is room in the original area based on the free entry count, the original area is scanned once again (long search) to find a secondary address for the new entry. If a free entry is found, the new entry resides there, and it becomes the last entry in the synonym chain. Otherwise, the set is full and DBPUT will fail.

3. If the primary address is occupied by a secondary entry, the secondary entry is relocated to another secondary address in either the original area or the expansion area. A free entry is searched as described in item 2 above. The new entry (primary) becomes a synonym chain head with a count of one. TurboIMAGE/XL performs synonym chain maintenance for the secondary entry which relocates.

Following the expansion, TurboIMAGE/XL updates various fields related to the expansion such as current capacity, including the expansion, in the root file. Also, the data set user label is updated for the last entry used in the file (high-water mark), free entry count, and pointer to the put-delete chain head.

Intrinsics and utilities. The following intrinsics and utilities are enhanced to support MDX:

*   DBINFO modes 205 and 208 (new for MDX) give detailed information pertaining to dynamic expansion. The buffer layout for mode 205 remains the same as that of DDX except that it is relevant to MDX. The buffer layout for new DBINFO mode 208 is in the following table.

    Buffer   must be at least a 64-byte record and returns the following (each element is a 32-bit word):

DBINFO Mode 208

Element	Contents
1	Primary (hashing) capacity for masters, 0 for details
2	Current capacity, including expansions
3	Maximum capacity
4	Expansion threshold: -1 ... 100 percentage -1 this value means expansion is not triggered by percentage, or it is a non-expandable set.
5	Delete chain free head (0 for non-expanded masters)
6	high-water mark (0 for non-expanded masters)
7	Expansion threshold: -1 ... 2 billion blocks -1 this value means expansion not triggered by traversing 3 blocks without success, or it is a non-expandable set.
8...16	Reserved; 0 is returned

DBINFO mode 208 returns information about internals which will be meaningful to only a few customers.

- DBUTIL SHOW *database* ALL command shows the presence of MDX. The SHOW database CAPACITY command gives detailed information pertaining to capacity parameters and identifies sets enabled for dynamic expansion.

  For more information, refer to the sixth edition of the *TurboIMAGE/XL Database Management System Reference Manual.*

NOTE Select master data sets which need dynamic expansion based on anticipated growth and specify the capacity parameters (in particular the initial capacity) very carefully. Specifying low initial (primary/hashing) capacity and increment leading to frequent expansions may result into long synonym chains, which degrade the performance of some TurboIMAGE intrinsics, and disk fragmentation. Select initial capacity such that expansions will be minimal.

80 GB Data Sets

TurboIMAGE/XL version C.07.14 includes an enhancement which will allow jumbo data sets to be up to 80 GB in size, twice as big as the old limit. There are no new externals/options added for this enhancement. For new databases, DBSCHEMA will allow data sets to be up to 80 GB in

size. As for jumbo data sets, the control option `'$CONTROL JUMBO'` is required prior to specifying a data set greater than the default of 4 GB. For existing databases, use third-party tools which support this feature. If you use third-party indexing software, you may need the version of the software which supports this feature.

When creating a jumbo data set of this magnitude using `DBUTIL`, you will experience some delay.

Performance Enhancement for Databases with Third-party Indices (TPI)

TurboIMAGE/XL intrinsics `DBGET` and `DBFIND` are enhanced for improved performance in the following circumstances when the database is enabled for TPI:

- `DBFIND` and `DBGET` when the set does not have TPI and the trace facility, if applicable, is turned off.

- Serial `DBGET` (modes 2 and 3) for a set with TPI when the mode need not be promoted to other mode (possibly a chained mode) by the third-party software.

- Chained `DBGET` (modes 5 and 6) for a `DBFIND` of an IMAGE search item or a B-Tree item for a set with TPI.

In order to attain the performance gain, you must obtain the version of the third-party software which supports this enhancement. In the absence of either one of the needed software products (TurboIMAGE/XL or third-party product), `DBFIND` and `DBGET` will continue to work as they do in the absence of this enhancement. That is, TurboIMAGE/XL and the third-party products are backward compatible for this enhancement.

DBSCHEMA and P Type with Odd Sub-Item Length

`DBSCHEMA` is enhanced to generate an error when the sub-item length of the datatype P is odd, even if the product of sub-item count and sub-item length is divisible by 4. For example, `DBSCHEMA` will flag the data items 4P1 or 8P3 as invalid. Although syntactically correct, none of the major languages can use them. This change will not affect existing databases. That is, if you currently have databases with P type defined using odd sub-item length and sub-item count such that their product is divisible by 4, you may continue to use it without any problem when creating a new database. The default for `DBSCHEMA` is to generate an error in this situation. If you want to create a new database with P data type having odd sub-item length, you will need to use the new `CONTROL` option, `$CONTROL ODDPALLOWED`.

B-Tree Indices

You can create a B-Tree index on the master data set's key item and perform B-Tree index searches using the key item as well as all of its corresponding detail data set search items. The B-Tree index searches are done using `DBFIND` with a master or detail data set and its key or search item respectively. The qualified entries can be retrieved using `DBGET` mode 5 or 6.

Key Features of B-Tree Indices

Following are key features of B-Tree indices:

1. B-Tree index can be created only on the key item of the master data set.

2. B-Tree searches using the search items of all of its corresponding detail paths can be done.

3. You may create B-Tree indices for zero, one, or more master data sets.

4. `DBSCHEMA` has a new option, `INDEXED`, for the `SET` specification.

   New syntax

   ```
   NAME:setname,{M[ANUAL] | A[UTOMATIC]}[/INDEXED]
   [(read class list)/(write class list)][,device class];
   ```

5. `DBUTIL` has new commands and options.

   - New `ADDINDEX`, `DROPINDEX`, and `REBUILDINDEX` commands (to be used for one, more than one, or all masters).

     Syntax

     ```
     {ADDI[NDEX] | DROPI[NDEX] | REBUILDI[NDEX]}
     database name[/maintword] FOR {ALL | setnamelist | setnumlist}
     ```

     Parameters

setnamelist	is the list *setname*[,…]
setnumlist	is the list *setnum*[,…]
ALL	means all master data sets for the database.

   - The `SET` command has a new `BTREEMODE1` option to set `DBFIND` mode 1 access `ON` or `OFF` for a B-Tree wildcard search for X and U types. The default is `OFF`. The `ON` setting allows you to use B-Tree indices for generic search without making application changes. Also, it allows you to define your own database-wide wildcard character.

     New Syntax

     ```
     SET database name[/maintword]
     BTREEMODE1={ON | OFF}[,[WILDCARD=]c]
     ```

where *c* is any printable ASCII character, and the default character is @.

- The SHOW command has a new INDEX, INDEXES, or INDICES option.

  New Syntax

  SHOW *database name*[/*maintword*]
  {INDEX | INDEXES| INDICES}

- CREATE, ERASE, PURGE, SECURE, RELEASE, and SHOW commands include B-Tree index files in their operation.

- The MOVE command does not allow moving the index file.

6. DBCONTROL has these modes pertaining to B-Tree indices:

13   is for B-Tree index file control. That is, to ADD, DROP, REBUILD, ATTACH, or DETACH a B-Tree index file. This mode requires PRIVILEGED mode, and hence, must be used carefully as improper use may damage your system.

14   is used to obtain and control database-wide B-Tree information. This mode requires PRIVILEGED mode, and hence, must be used carefully as improper use may damage your system.

15   sets BTREEMODE1 ON, and optionally allows the wildcard character to be set for the current DBOPEN.

16   sets BTREEMODE1 OFF for the current DBOPEN.

7. DBINFO has these modes pertaining to B-Tree indices:

209   informs whether or not a B-Tree index exists for a master.

113   gives BTREEMODE1 setting and the wildcard character for the database as well as current DBOPEN.

8. DBFIND has these features pertaining to B-Tree indices:

- Can be used for details as well as masters to specify B-Tree index searches.

- Introduces a new fixed-format structured argument.

- DBFIND modes with added functionality pertaining to B-Tree indices:

1   can continue to work as it does in releases prior to the release of B-Tree index feature (C.07.00), despite the presence of a B-Tree index, or it can be used for a B-Tree search if BTREEMODE1 is ON. Uses the classic mode 1 argument format.

4	used for B-Tree index searches on numeric as well as ASCII types and, for details, it returns accurate chain (super-chain, that is, multiple qualified chains) counts. Requires a new structured argument.
10	allows you to simulate `DBFIND` mode 1 as if there were no B-Tree index. This is the same as TPI mode 10. Requires the classic mode 1 argument format.
21	is the same as B-Tree index search using `DBFIND` mode 1, except it is a faster version and does not return accurate chain counts. Requires the classic mode 1 argument format.
24	is the same as B-Tree index search using `DBFIND` mode 4, except it is a faster version and does not return accurate chain counts. Requires a new structured argument.

Allows wildcard search, as well as range search (`<`, `<=`, `>`, `>=`, `"PK"` for partial key, or `[]` for between).

9. `DBGET` modes 5 and 6 can be used for B-Tree index retrieval for masters or details. Super-chains are traversed for detail data sets.

10. KSAM/iX files are used for B-Tree index files. These are KSAM related key points:

   - There is one KSAM/iX privileged file with a filecode of -412 for each B-Tree index file. The size limit for this B-Tree index file is 4 GB. A jumbo master (larger than 4 GB) can have a B-Tree index file provided the B-Tree index file remains within its 4 GB limit.

   - KSAM/iX B-Tree index file is named using the POSIX file format with the "idx" extension (lowercase). For example:

     `/ACTSALES/GRPSALES/ORDERS03.idx`

   - The KSAM file has the Native Language Support language specified to match the language of the database, if the key is an ASCII type.

11. Third-party index can coexist with B-Tree index, that is, on the same item of the data set.

Quick Start for Using B-Tree Indices

If you are interested in generic trailing-@ searches only and want to get started quickly without making any application changes, you may use the following steps.

1. Identify the masters with ASCII key item that can benefit from B-Tree indices.

2. Create B-Tree indices using either of these methods:

    a. Use the `INDEXED` option of `DBSCHEMA` for new databases.

    b. Use the `ADDINDEX` command of `DBUTIL` for existing databases.

3. Set the `BTREEMODE1` option `ON` using `DBUTIL` as follows:

```
:Run DBUTIL.PUB.SYS
```

```
>SET database name [/maintword] BTREEMODE1=ON
```

If your database is new, you will need to add data. You are now ready to perform B-Tree index searches. You can include the wildcard character in your `DBFIND` argument and observe the results.

Scalability

Prior to this enhancement, the modification intrinsics, `DBPUT`, `DBDELETE`, and `DBUPDATE` (Critical Item Update feature `ON`) were serialized for the database. That is, only one such intrinsic could execute at a time for the database. In order to scale with the high-end machines, TurboIMAGE/XL is enhanced to increase the concurrency of these modification intrinsics.

The increased concurrency is based strictly on the database design. The database is internally grouped into independent SUB-DATABASES based on the physical relationships of master and detail data sets. Subsequently, the modification intrinsics can execute concurrently for these independent subdatabases. The throughput is dependent on the number of subdatabases, the more the better. The worst case is a database with only one subdatabase, one detail linked to 16 masters. The best case is when there are numerous stand-alone masters or details.

To use this enhancement, your database must be activated to use the Dependency Semaphore. To achieve this, the `ENABLE` and `DISABLE` commands of `DBUTIL` introduce a new option, `DSEM`. The default for `DSEM` is `DISABLED`.

New syntax

`EN[ABLE]` *database name*`[/maintword] FOR DSEM`

`DI[SABLE]` *database name*`[/maintword] FOR DSEM`

After enabling your database for `DSEM`, if you feel that your database design cannot attain expected concurrency, you may `DISABLE` the feature as there is some overhead when the database is enabled for `DSEM`.

Dynamic Detail Dataset Expansion (DDX) Fix

(SR# 5003-367607)

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. The patch ID for MPE/iX 5.5 is TIXKX62 (or superseded patch id) and the TurboIMAGE/XL version is C.07.10. This patch is contained in this release.

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 following diagram illustrates the problem.

```
|          |
|          |
|          |
|          |
|----------|-----> Old MPE EOF as well as IMAGE EOF (e.g., 1)
|          |
INCREMENT  |        New MPE EOF (e.g., 51) after expansion by
|----------|-----> INCREMENT equivalent to 50 MPE records.
|          |-----> New entries placed from this point on are lost.
|          |
|----------|-----> New IMAGE EOF (e.g., 101) as calculated by the
                    adjustment to CAPACITY of set, following the
                    expansion.
```
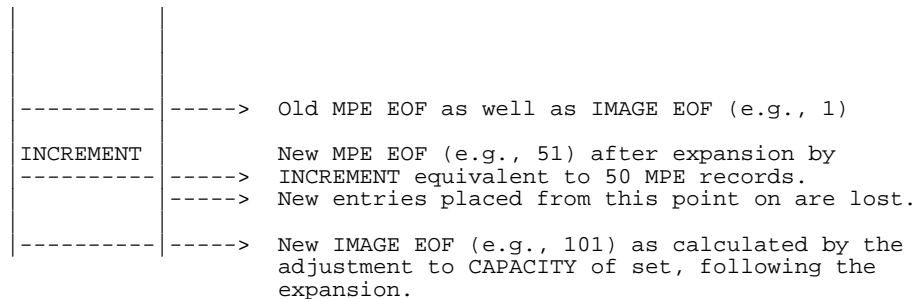
The real problem is that, for certain expansions, the new CAPACITY is inadvertently adjusted by adding the INCREMENT *more than once* instead of adjusting only *once*. This results in incorrect calculation of IMAGE EOF and subsequently allows new data entries to be placed in the area which really does 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:

```
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 dbusa01,2

ACCOUNT=  TESTACCT    GROUP=  DDX

FILENAME  CODE  ----------LOGICAL RECORD----------  ----SPACE----
                SIZE  TYP         EOF     LIMIT R/B  SECTORS #X MX

DBUSA01   PRIV  256W  FB          51       200  1   112     2  4
```

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:

```
               No. of  %Max -----------Capacity------------- Dyn
Data Set Name Type Entries Cap Maximum Current Initial Increment Exp:

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 the TurboIMAGE/XL patch 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.

Important Information

1. The default for Critical Item Update option was changed from `DISALLOWED` to `ALLOWED` starting with Express 3 for 5.5.

2. A jumbo data set cannot be enabled for dynamic expansion.

3. `PURGE` and `ERASE` commands of `DBUTIL` are enhanced to prompt the user in session mode only (not batch) for confirmation (starting with version C.07.07).

4. If you want the TPI performance enhancement, you also need to get the third-party indexing software which supports this enhancement.

IMAGE/SQL Enhancements

Information regarding enhancements to IMAGE/SQL for SQL access is described in this section:

- Predicate level lock when updating masters

- Support for B-Tree indices

- Support for third-party composite indices

- New ODBC driver, ODBCLink/SE

Predicate Locks when Updating Manual Masters

IMAGE/SQL version B.G2.07 is enhanced to apply predicate level locks (row level locks), instead of a set level lock as in prior versions, when updating selected entries in the manual master data set.

When IMAGE/SQL is used as the SQL interface to TurboIMAGE/XL database, the responsibility of enforcing appropriate locks rests with IMAGE/SQL. These locks should conform to the locking schemes present in TurboIMAGE/XL. Accordingly, locks are set by IMAGE/SQL depending on the DBOPEN modes and isolation level requirements. At present, for a master dataset, for all modify intrinsics, that is, insert, delete and updates, and for DBOPEN modes 1 to 4, IMAGE/SQL puts a set level lock (or a pseudo set level lock covering all the rows in the set). But for update function in DBOPEN mode 1, setting a set level lock (or pseudo set lock) is not mandatory, a row level lock would be sufficient. Therefore, IMAGE/SQL will now lock at the predicate level if a 'where' clause is specified, otherwise a set lock will be given. This feature is actually transparent to the user.

Benefits

When multiple users are operating on the same data set, it is difficult to update different entries with set level lock in place. But when a predicate level lock is in effect, the unnecessary wait time is avoided, thus allowing multiple updates of different entries on the same data set.

For an example, User A has an update statement:

```
UPDATE Music.Albums set RecordingCo='ABC Music Ltd.' where Medium='ca';
```

The kind of lock that is placed can be seen by invoking DBUTIL as shown:

```
:DBUTIL

HP30391C.07.04 TurboIMAGE/XL:  DBUTIL (C) COPYRIGHT HEWLETT-PACKARD COM
PANY 1987

>>show Music locks
   For database MUSIC
                                              PIN/      PROGRAM
      LOCKED ENTITY  /  ( - waiting process )  PATH     NAME   JOBNUM

ALBUMS: MEDIUM = ca . . . . . . . . . . . . . . . 41/1    ISQL    #S18
```

Now, User B wants to update row(s) in the same data set via an update
statement:

```
UPDATE Music.Albums set RecordingCo='California Music' where
Medium='cd';
```

The kind of lock that is applied by User B (JOBNUM #S21) is shown in
the following example:

```
:dbutil

HP30391C.07.04 TurboIMAGE/XL:  DBUTIL (C) COPYRIGHT HEWLETT-PACKARD
COMPANY 1987

>> show Music locks
   For database MUSIC
                                              PIN/      PROGRAM
      LOCKED ENTITY  /  ( - waiting process )  PATH     NAME   JOBNUM

ALBUMS: MEDIUM = cd . . . . . . . . . . . . . .  56/1    ISQL    #S21
ALBUMS: MEDIUM = ca . . . . . . . . . . . . . .  41/1    ISQL    #S18
```

In this example, User A updates 2 entries and User B updates 5 entries
simultaneously.

Impact on current IMAGE/SQL applications

With this enhancement, the locking scheme in IMAGE/SQL is exactly
as required by TurboIMAGE/XL (as described in TurboIMAGE/XL
Database Management System Reference Manual, Aug 1997). All
applications conforming to this will not have any problems, and may
actually see improvements in performance, when updating manual
masters.

Support for B-Tree Indices

The B-Tree indices can be created using DBUTIL for existing databases
and DBSCHEMA for new databases. More details about creating B-Trees
indices is given under the heading, "TurboIMAGE/XL Enhancements,"
earlier in this article.

In order to realize the benefit of the B-Tree indices for SQL access, you
must first DETACH the database, if already attached, and ATTACH the
database for the DBEnvironment. IMAGE/SQL enters definitions for
the B-Tree indices in the system catalog of the DBEnvironment. The
index on the key item of the master, except for P and Z types, is entered
as 'unique' index. Other definitions, by default, are non-unique.

Although the B-Tree index on the related search item of the detail set is not explicitly created using DBUTIL or DBSCHEMA, its definition is entered in the SQL catalog. For example, if a B-Tree index is created on a key item with paths to 16 detail data sets, a definition for a B-Tree index for all 16 data sets will be entered as well. That is, this will result into 17 (1+16) definitions, one for each data set.

If your key item or its related search item is split using the SPLIT command of IMAGESQL, the definition for a B-Tree index on the split item will not be entered. For an example, if a key item is split but the related search item is not, the definition for the key item of the master data set will not be entered, but the one for the search item will be.

The B-Tree indices can be viewed in the views, SYSTEM.INDEX and CATALOG.INDEX, of the system catalog of the DBEnvironment.

Multiple index definitions on the same column can coexist and the SQL optimizer derives the access plan based on the statistics present in the system catalog. In other words, the key or search item of the set can have a maximum of three index definitions. One will be a hash index (only "=" operator permitted) automatically done by IMAGESQL at ATTACH time, another can be a B-Tree index, and the third can be a third-party index. It is recommended that both B-Tree index and third-party index be not created on the same item as it will unnecessary impact the performance (Optimizer calculates cost for each index).

The Optimizer derives an access plan and decides which index to use and the proper order of operations.

The version of third-party software that supports the new B-Tree index modes for DBFIND, DBCONTROL, and DBINFO, is required from both third parties.

Third-Party Composite Indices

IMAGESQL is enhanced to enter definitions for third-party composite indices in the system catalog of the DBEnvironment at ATTACH time. These composite indices can be on mixed data types, but must be on FULL items. Information about all third-party indices, including composite, is obtained from the third-party product, which is subsequently used during ATTACH. Both SUPERDEX and OMNIDEX do not provide information on the composite indices on substrings (partial item) of items to IMAGE/SQL.

These indices can be viewed in SYSTEM.TPINDEX along with other third-party indices.

At run-time, IMAGE/SQL may employ DBFIND mode 1 with "@;" appended to the argument, or mode 11 with start and stop values, as deemed appropriate.

Note that if you have multiple indices for the table, the SQL optimizer has the control on the type of access, and the specific index to be used.

For SQL access, the only step necessary to use the above enhancements pertaining to relational access, is to perform `DETACH`, if attached, and `ATTACH`.

Important Information

1. If you have third-party indexing (TPI) enabled for the database and want to create B-Tree indices, obtain the correct version of the third-party software which recognizes this coexistence. Otherwise, you will get run-time errors returned by third-party software for new modes of `DBFIND, DBINFO`, and other utility commands.

2. The Optimizer derives an index scan based on the current statistics for the data set. Hence, it may select an index on the column not used in the SQL statement.

ODBCLink/SE

In this release, IMAGE/SQL is bundled with a new 16-bit and 32-bit ODBC driver, ODBCLink/SE, which is a replacement for the 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 data on the HP3000 in a client/server environment. In such an environment, the application developers and end-users can take advantage of the PC's graphical user interface (GUI) and processing power, while relying on the security, integrity, and database management capabilities of IMAGE/SQL. The client can run under Microsoft Windows 3.1 or 3.11, Windows95, or Windows NT V3.51 or V4.0. Connection via Winsock is available in 16-bit and the new 32-bit version. The driver can be used in two ways: either 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.

The ODBCLink/SE server runs on MPE/iX 5.0 or later releases. ODBCLink/SE is ODBC Level 2 compliant with a few exceptions. For more information, refer to the article, "Introducing ODBCLink/SE," in this *Communicator 3000*. The *ODBCLink/SE Reference Manual* (36217-90406) is bundled with IMAGE/SQL.

New Features in ALLBASE/SQL Version G2

by Doug Myers
Commercial Systems Division

ALLBASE/SQL, version G2, contains several major enhancements providing significant benefits in the following areas:

- Security

- Year 2000 compatibility

- `CAST` function (introduced in G1.15)

In addition, the following manuals were updated to fully document G1 and G2 features:

- *ALLBASE/SQL Reference Manual* (36216-90001)

- *ALLBASE/SQL Database Administration Guide* (36216-90005)

- *ALLBASE/SQL Message Manual* (36216-90009)

Security

Two new authorities have been added to the G2 release of ALLBASE/SQL: `MONITOR` and `INSTALL`. These new authorities can be granted to users using the `GRANT` command, or can be revoked from users using the `REVOKE` command.

MONITOR Authority

`MONITOR` authority gives a user the ability to run `SQLMON`, an online diagnostic tool that monitors the activity of an ALLBASE/SQL DBEnvironment. Before this new authority, `SQLMON` could only be run by the creator of the DBEnvironment (also known as the DBECreator), someone who knows the maintenance word (maintword), or someone who has superuser capability.

INSTALL Authority

`INSTALL` authority gives a user the ability to use the SQL preprocessors to create an SQL module owned by someone other than themselves, or to use ISQL to install a module owned by someone other than themselves. This is an important capability for those needing to move modules from a development DBEnvironment to a production DBEnvironment. Before this new authority, only a DBA had this ability.

Year 2000 Compatibility

As the end of the century quickly approaches, there has been much concern about dates being handled correctly in the twenty-first century. This is particularly an issue for systems and applications that are using a two-digit year to express dates. ALLBASE/SQL has always stored dates in a four-digit year format which largely eliminates the year 2000 compatibility problem. ALLBASE/SQL applications can be coded using four-digit year dates, thus avoiding any ambiguity. However, for existing applications that may allow dates to be entered using a two-digit year format, there is still an issue as to how these dates get converted to the ALLBASE/SQL internal format using the `TO_DATE` function or `TO_DATETIME`. For example, what date value would the following functions return?

```
TO_DATE('12/31/35','MM/DD/YY')
TO_DATETIME('351231235959','YYMMDDHHMISS')
```

Would these functions evaluate the year as 1935 or 2035?

Solution

When YY is specified in the format specification of either `TO_DATE` or `TO_DATETIME`, and if its value of the year specified is less than 50, then the century part of the `DATE` or `DATETIME` defaults to 20, otherwise it is set to 19. This behavior can be overridden by setting the JCW `HPSQLsplitcentury` to a value between 0 and 100. If the year specified is less than the value of the JCW `HPSQLsplitcentury` then the century part is set to 20, otherwise it is set to 19.

For the above example,

Case 1: HPSQLsplitcentury is not set

```
TO_DATE('12/31/35','MM/DD/YY') = 2035-12-31
TO_DATETIME('351231235959','YYMMDDHHMISS') = 2035-12-31 23:59:59
```

Case 2: HPSQLsplitcentury is set to 0

```
TO_DATE('12/31/35','MM/DD/YY') = 1935-12-31
TO_DATETIME('351231235959','YYMMDDHHMISS') = 1935-12-31 23:59:59
```

Case 3: HPSQLsplitcentury is set to 70

```
TO_DATE('12/31/35','MM/DD/YY') = 2035-12-31
TO_DATETIME('351231235959','YYMMDDHHMISS') = 2035-12-31 23:59:59
```

CAST Function

With the G1.15 release, the `CAST` function enhancement for ALLBASE/SQL and IMAGE/SQL became 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 Syntax

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

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.

Pseudo-Mapped Files Removed from ALLBASE/SQL

by Poongodi R and Doug Myers
Commercial Systems Division

Introduction

This article provides information regarding the handling of pseudo-mapped files when installing ALLBASE/SQL. Pseudo-mapped files in MPE XL are disk files that are accessed without the usual file mapping by the operating system.

Pseudo-mapped files are no longer supported by ALLBASE/SQL. If the pseudo-mapped files already exist in the system, they will not be supported on the G.0 (or later) versions. Serious system problems may occur because of the use of pseudo-mapped files.

NOTE   The conversion from pseudo-mapped to mapped MUST be done on an F.0 version before updating to G.x. You cannot do the conversion after the system has been updated.

Converting Pseudo-Mapped Files to Mapped Files

ALLBASE/SQL files in pseudo-mapped mode must be converted into mapped mode using the MOVEFILE command in SQLUtil. Use the SHOWACCESS command in SQLUtil to show which files are pseudo-mapped. Use the following procedures:

1. Check for pseudo-mapped files as shown in this example:

   ```
   :sqlutil

   >> showaccess
   DBEnvironment Name: mydbe
   Maintenance Word:
   GGF0         mapped
   MYF1         pseudo-mapped
   MYF2         pseudo-mapped
   >>
   ```

2. Disable the pseudo-mapped feature using the SQLUtil MOVEFILE command as shown in this example:

   ```
   >> movefile
   DBEnvironment Name: mydbe
   File Name: MYF1
   Current Device: LDEV#2
   New Device (opt):LDEV#
   Access Mode (Mapped) (opt): mapped

   File moved.
   ```

Repeat for other pseudo-mapped files listed by SHOWACCESS. The pseudo-mapped files are converted into mapped mode. Hence, there will be no problem involved with the converted files during the upgrade.

TurboSTORE/iX II, TurboSTORE/iX 7x24 Patches

by Jim Nissen
Commercial Systems Division

When we shipped MPE/iX Release 5.5, we inadvertently included TurboSTORE/iX 7x24 True Online, P/N B5152AA. The inclusion of this version of TurboSTORE was in error. Hewlett-Packard Company apologizes for any inconvenience and confusion this may have caused you.

If you do not have a license to use this version, please use it as a short term demonstration copy. The demonstration period ended July 15, 1997.

If you are interested in purchasing a license to use this product, contact your HP Sales Representative or your Authorized HP Reseller and be sure to ask about the current High Availability Promotion (code named "Bridge").

Users of TurboSTORE/iX II or TurboSTORE/iX 7x24 should have installed either the SUBSYS or the appropriate XL from the MPE/iX 5.5 Express 2 (C.55.02) SUBSYS tape prior to July 15, 1997. Installing the MPEJXD4B patch requires the installation of the MPE/iX 5.5 Express 2 (C.55.02) SUBSYS or later. Failure to install either the SUBSYS 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, you may install only the appropriate XL from the SUBSYS tape. Users of TurboSTORE II (B5151AA) will install STORXL51 and users of TurboSTORE/iX 7x24 (B5152AA) will install STORXL52 using the following procedure:

```
Log on 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
```

Removing TurboSTORE from Your System

If you determine that this product is not required, please remove it
from your system. To remove TurboSTORE/iX 7x24, you need to remove
STOREXL.PUB.SYS from your system. The following procedure will do
this

```
:purge STOREXL.PUB.SYS
:sysgen
SYSGEN version E.02.00:catalog version E.02.00 THU,JAN 23,1997,11:51 AM

Copyright 1987 Hewlett-Packard Co. All Rights Reserved.

        **warning** no NMCONFIG file in this configuration.

        **warning** NMCONFIG.PUB.SYS is now the NMCONFIG file.

        **note** Retrieving NMMGR configuration data...


        ** First level command **

        io              log (lo)      misc (mi)       spu (sp)

        sysfile (sy)

        basegroup (ba)   keep(ke)      permyes (pe)    show (sh)

        tape (ta)

        clear (cl)(c)    exit (ex)(e)  help (he)(h)    oclose (oc)

        redo

 sysgen> sys

        ** SYSFILE configurator commands **

        aauto (aa)      aboot (ab)    acmsl (ac)      asprog (as)

        cmsl (cm)       dauto (da)    dboot (db)      dcmsl (dc)

        dsprog (ds)     lcmsl (lc)    rauto (ra)      rboot (rb)

        rcat (rc)       rcmsl (rcm)   rdcc (rd)       ripl (ri)

        rnmlib (rn)     rsprog (rs)   show (sh)

        clear (cl)(c)   exit (ex)(e)  help (he)(h)    hold (ho)

sysfile> dsprog prog=STOREXL.PUB.SYS
sysfile> hold
sysfile> exit
 sysgen> keep
        keeping to group CONFIG.SYS

        Purge old configuration (yes/no)?yes
         ** configuration files successfully saved **

 sysgen> exit
```

6　　Product List—How to Order

You can order subsystem products through your HP Sales Representatives.

- To order products in the U.S., call HP Computer Organization Order Fulfillment Center at 1-800-386-1117 option 3.

- To order learning products manuals in the U.S., call HP Parts Director Ordering at 1-800-227-8164.

- To order products and/or manuals in countries outside the U.S., call your local Trade Parts Sales Organization.

You must specify the product number and release base (for this release, Release 5.5) when ordering. Installing the PowerPatch tape updates your system with the FOS enhancements, as well as subsystems that have patches.

You may order only the PowerPatch tape. However, installing the PowerPatch ONLY, updates your system only with the FOS enhancements—the subsystem products **must** be ordered and installed via the SUBSYS tape from MPE/iX-Express 3 based on Release 5.5. Customers with Response Center Support can contact the Response Center to order the PowerPatch tape.

NOTE　　MPE/iX Express 3 based on Release 5.5 can be ordered only by mailing or faxing the order form sent to those customers with an appropriate support contract. It cannot be ordered by phone.

To install MPE/iX PowerPatch 5 based on Release 5.5, you will need the following:

Release On Your System	What You Need
MPE/iX General Release C.50.xx (any version)	FOS C.55.00 SUBSYS C.55.03 PowerPatch C.55.05
MPE/iX Release C.55.00, C.55.01, C.55.02	SUBSYS C.55.03 PowerPatch C.55.05
MPE/iX Release C.55.03 with PowerPatch 3 or 4	PowerPatch C.55.05

Product List

No subsystem products were submitted to MPE/iX-PowerPatch 5 based on Release 5.5.

7   Product Release History

This chapter contains tables that provide information on the currently supported Commercial Systems MPE/iX releases and products, and the systems supported for the 5.0 and 5.5 releases.

Product Changes by Releases

The following table provides information on the currently supported Commercial Systems MPE/iX releases and products. Included are the MPE/iX release or SUBSYS VUF and a list of products introduced. It also provides information on significant changes made to a release.

Table 7-1   **MPE/iX Product Releases**

Release	SUBSYS	Date Code	Product(s) Introduced/Added
C.50.00		R3504	MPE/iX Release 5.0 (Core Software Release) Workload Manager (B3879AA) Shared Globals ARPA Bundling
	C.50.02	R3513	MPE/iX-Express 2 based on General Release 5.0 MOVER Utility Information Access SQL/iX LASERRX/MPE ALLBASE/4GL Developer and Runtime
	C.50.03	R3538	MPE/iX-Express 3 based on General Release 5.0 Information Access Server/iX Symbolic Debug/iX ALLBASE/SQL G.1 IMAGE/SQL G.1 New Systems: HP 3000 Series 969KS/x00 (x = 1- 4) HP 3000 Series 996/80, 996/x00 (x = 1- 8) New SCSI Disks: ST31200N (1 GB Single-Ended) ST31230N (1 GB Single-Ended, Low Profile) ST32550N (2 GB Single-Ended) ST15150N (4 GB Single-Ended) ST32550W (2 GB Fast/Wide) ST15150W (4 GB Fast/Wide)

Release	SUBSYS	Date Code	Product(s) Introduced/Added
C.55.00		R3628	MPE/iX Release 5.5 (Major Software Release) HP Loader Dependent Libraries Subsystem Dump Facility HP Stage/iX HP Patch/iX TurboSTORE/iX 7x24 True-Online Backup HP Optical Disk Libraries: C1150B - 40GB C1160B - 80GB C1170B - 100GB TCP/IP Network Printer Support Telnet/iX Server DTS/TIO Dynamic Configuration Online System Device Configuration
	C.55.01	R3638	PowerPatch 1 based on Release 5.5 ALLBASE/SQL CAST function HP Telnet/iX—full functionality CCSY Access Server 979/x00KS support High Availability Disk Array support
	C.55.02	R3715	Express 2 based on Release 5.5 CI enhancements VPlus enhancements DTS/TIO new functionality

Release	SUBSYS	Date Code	Product(s) Introduced/Added
	C.55.03	R3732	Express 3 based on Release 5.5 COBOL II/iX enhancements 100VG-AnyLAN Network Adapter introduced 100Base-T Network Adapter introduced ALLBASE/SQL release G2 ODBCLink/SE ODBC driver introduced TurboIMAGE/XL B-Tree and scalability enhancements IMAGE/SQL B-Tree enhancement 9 GB Disk Drive NMMGR Node Maintenance Manager enhancements New SCSI Disks: ST32272N, ST32272WC (2 GB) ST34572N, ST34572WC (4 GB) XP34361W (4 GB)
		R3748	Express 4 based on Release 5.5 Year 2000 enhancements MPE/iX new date intrinsics 997/100-500 support Enhancements for: Transact V and Transact/iX Dictionary/3000 COBOL II/iX Inform/V HP ALLBASE/BRW QUERY/iX
		R3813	PowerPatch 5 based on Release 5.5 989/100, 989/200, 989/400, 989/600 support 997/600, 997/800 support A3699A - HP-IB I/O Bus Converter New SCSI Disks: A4909A - 18GB, WD A5239A - 18GB, WD Enhancements for Image/SQL and TurboIMAGE/XL: Dynamic data set expansion for master sets Performance enhancement for TPI Image Row level locking 80 GB Image data sets ODBCLink/SE E.56.12

Supported Releases

The naming conventions for the different types of releases have been changed slightly to clarify the type of release being discussed. The terms used to describe or refer to the releases are:

Mainline Release
A mainline release involves recompiling and reintegrating all software release components (FOS, SLT, and SUBSYS tapes). The release number is changed (for example, 5.0 or 5.5) and the update *UU* field of the *V.UU.FF* is changed. There are two types of mainline releases: *Platform* and *Non-Platform*.

Platform Release
A platform release (previously also known as a "core" release) is a subset of a mainline release. Typically, the release number ends with a "0" such as 6.0. Platform releases are *automatically distributed* to all customers with support contracts.

Non-Platform Release
A non-platform release (previously known as a "major" release) is a subset of the mainline release. The release number typically ends with a "5" such as 5.5. Non-platform releases must be *explicitly ordered* by customers.

Both platform and non-platform releases can be referred to as *mainline releases* when not discussing distribution or extended support life.

The following matrix provides information on the supported Commercial Systems MPE/iX mainline releases. It lists the currently supported releases and the SPUs they are supported on. The matrix also provides all known factory support termination dates. When a mainline release becomes unsupported, the factory will not provide any support services for that release. Online calls are not accepted and patches are not created; customers are advised to roll to a supported release.

- In general, HP will support the two most recent mainline releases.

- In order to facilitate customers moving to new releases, we will support the oldest release for at least six months after the new release ships. In other words, Release "N-2" will still be supported for at least six months after Release "N" ships.

- The release of an Express Release or a Powerpatch on any particular release does not extend its support life.

Given the rate with which we ship releases, this new strategy implies that any particular release will be supported for at least three years, and in most cases, most releases will be supported for a significantly longer period of time.

This new strategy is effective now. This implies that Release 5.0, which first shipped February 17, 1995, will become obsolete as early as six months after Release 6.0 ships.

Table 7-2   Supported System Release Matrix

Supported Releases	Supported Systems	Support Termination Date
Release 5.0 (C.50.xx)	920,922,925,930,932,935,948,949,950,955,960, 980-100,980-200,980-300,980-400,9x7,9x7LX, 9x7RX,9x7SX,9x8LX,9x8RX,939KS,959KS/100, 959KS/200,959KS/300,959KS/400,969KS/100, 969KS/200,969KS/300,969KS/400,969KS/120, 969KS/220,969KS/320,969KS/420,987/150RX, 987/150SX,987/200RX,987/200SX,990,990DX, 992/100,992/200,992/300,992/400,992/100DX, 992/200DX,992/300DX,992/400DX,991DX, 995/100DX,995/200DX,995/300DX,995/400DX, 995/500DX,995/600DX,995/700DX,995/800DX, 991CX,995/100CX,995/200CX,995/300CX, 995/400CX,995/500CX,995/600CX,995/700CX, 995/800CX	June 1, 1999
Release 5.5 (C.55.xx)	920,922,925,930,932,935,948,949,950,955,960,9x7, 9x7LX,9x7RX,9x7SX,9x8LX,9x8RX,939KS,939KS/0 20, 959KS/100,959KS/200,959KS/300,959KS/400, 969KS/100,969KS/200,969KS/300,969KS/400, 969KS/120,969KS/220,969KS/320,969KS/420, 979KS/100,979KS/200,979KS/300,979KS/400, 980-100,980-200,980-300,980-400,987/150RX, 987/150SX,987/200RX,987/200SX, 989/100,989/200,989/400,989/600,990,990DX, 992/100,992/200,992/300,992/400,992/100DX, 992/200DX,992/300DX,992/400DX,991DX, 995/100DX,995/200DX,995/300DX,995/400DX, 995/500DX,995/600DX,995/700DX,995/800DX, 991CX, 995/100CX,995/200CX,995/300CX, 995/400CX, 995/500CX,995/600CX,995/700CX, 995/800CX 996/80,996/100,996/200,996/300, 996/400,996/500,996/600,996/700,996/800, 997/100,997/200,997/300,997/400,997/500, 997/600,997/800	6 months following next Mainline release after 6.0 (about end of year 2000)

@ Platform Releases are denoted by an @ sign.

8   Catalog of User Documentation

This chapter contains a list of new and recently revised customer manuals for the HP 3000 Computer System. Many were introduced at the time of MPE/iX-Express 3 based on Release 5.5.

For a comprehensive list of available manuals, refer back to the *Communicator 3000 MPE/iX Platform Release 5.5* (30216-90124). If your contract includes Material-Based Services, you will receive both software and manual updates. For additional copies of manual updates, Manual Update Services (MUS) can be ordered.

For detailed history information on a particular manual or manual set, refer to the *MPE/iX Documentation Guide* (32650-90144).

Most of the learning products listed in this section can be individually ordered. See Chapter 6, "Product List—How to Order," for ordering information.

NOTE

If you have purchased a product or currently have support for a product, you will automatically receive the updated manuals for that product.

Customer Manuals

Some MPE/iX manuals are provided in kit format. The kits that contain multiple manuals include a list of the contents.

Table 8-1   **New and Recently Revised MPE/iX Manuals**

Manual Title	Customer Order No.	Latest Edition
COMMUNICATOR 3000 MPE/iX-PowerPatch 5 (PowerPatch Tape C.55.05) Based on Release 5.5 [a]	30216-90257	Jun 1998
HP 3000 MPE/iX System Software Maintenance Manual [b]	30216-90223 R3715	Apr 1997
Read Before Installing	30216-90258	Jun 1998
HP IMAGE/SQL Administration Guide	36385-90001	Aug 1997
TurboIMAGE/XL Database Management System Reference Manual	30391-90001	Aug 1997
ALLBASE/SQL Database Administration Guide	36216-90005	Aug 1997
ALLBASE/SQL Reference Manual	36216-90001	Aug 1997
ALLBASE/SQL Message Manual	36216-90009	Aug 1997
ODBCLink/SE Reference Manual	36217-90406	Jun 1998
Legato NetWork ClientPak for MPE/iX kit:	B5475AA	Dec 1997
Legato NetWorker ClientPak for MPE/iX Installation Guide	B5475-90001	
Legato NetWorker ClientPak for MPE/iX User's Guide	B5475-90002	
Legato NetWorker ClientPak for MPE/iX Release 5.0 Supplement	B5475-90005	Feb 1998
NetBase for MPE Reference Guide	B3933-90001	Jun 1997
NetBase SQL Shadowing Handbook	B3482-90006	Jun 1997
NBSpool for MPE Reference Guide	B3933-90002	Jun 1997
Vista Plus for MPE User's Guide	B3933-90003	Jun 1997
Vista Plus for MPE Administrator's Guide	B3933-90004	Jun 1997

Manual Title	Customer Order No.	Latest Edition
Installation/Update Procedures for SHAREPLEX	B3933-90005	Jun 1997
HP-PB 100VG-AnyLAN Network Adapter Installation and Service	B5425-90001	Aug 1997
HP-PB 100Base-T Network Adapter Installation and Service	B5427-90001	Aug 1997

a. This manual is shipped to all customers who have HP 3000 Support Contracts.

b. This manual is automatically shipped with the PowerPatch.

Index

Index

mainline, 161
non-platform, 161
platform, 161
supported, 161
remote shell, 26
REMSH client, 30
  functionality, 26
  troubleshooting, 30

S

STORE
  with NetWorker Client, 34
support termination date, 161,
    162
supported
  releases, 161
  systems, 162
supported releases, 161
SYM4
  EMC Symmetrix Disk Array, 61
systems
  information, 15
  upgrading from pre-997, 16

T

TCP/IP
  with DTS/TIO, 46
Telnet/iX, 49
termination date
  release support, 161, 162
Transact V enhancements, 115
Transact/iX enhancements, 115
TurboIMAGE/XL
  dynamic detail dataset
    expansion fix, 139
  enhancement summary, 129
  enhancements, 127
TurboSTORE/iX
  7x24, 152
  II, 152
  installing patches, 152
  removing, 153
  with NetWorker Client, 34

U

upgrade to 997, 16
  IO configuration, 16
user documentation, 163

V

VPLUS enhancements, 84

Y

year 2000
  and ALLBASE/SQL, 148

enhancements for MPE/iX, 65
VPLUS enhancements, 84
web page, 64