

# HP Performance Insight

For the HP-UX, Linux, Solaris, and Windows® operating systems

Software Version: 5.40

---

## Reference Guide

Manufacturing Part Number: J5223-99049

Document Release Date: January 2009

Software Release Date: January 2009



## Legal Notices

### Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

### Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

### Copyright Notices

© Copyright 2009 Hewlett-Packard Development Company, L.P.

This product includes org.apache.commons software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product includes Xerces XML Java Parser software, which is Copyright (c) 1999 The Apache Software Foundation. All rights reserved.

This product includes JDOM XML Java Parser software, which is Copyright (C) 2000-2003 Jason Hunter & Brett McLaughlin. All rights reserved.

This product includes JClass software, which is (c) Copyright 1997, KL GROUP INC. ALL RIGHTS RESERVED.

This product includes J2TablePrinter software, which is © Copyright 2001, Wildcrest Associates (<http://www.wildcrest.com>)

This product includes Xalan XSLT Processor software, which is Copyright (c) 1999 The Apache Software Foundation. All rights reserved.

This product includes EXPAT XML C Processor software, which is Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd and Clark Cooper Copyright (c) 2001, 2002 Expat maintainers.

This product includes Apache SOAP software, which is Copyright (c) 1999 The Apache Software Foundation. All rights reserved.

This product includes O'Reilley Servlet Package software, which is Copyright (C) 2001-2002 by Jason Hunter, jhunter\_AT\_servlets.com. All rights reserved.

This product includes HTTPClient Package software, which is Copyright (C) 1991, 1999 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

This product includes Perl software, which is Copyright 1989-2002, Larry Wall. All rights reserved.

This product includes Skin Look And Feel software, which is Copyright (c) 2000-2002 L2FProd.com. All rights reserved.

This product includes nanoXML software, which is Copyright (C) 2000 Marc De Scheemaecker, All Rights Reserved.

This product includes Sixlegs PNG software, which is Copyright (C) 1998, 1999, 2001 Chris Nokleberg

This product includes cURL & libcURL software, which is Copyright (c) 1996 - 2006, Daniel Stenberg, <daniel@haxx.se>. All rights reserved.

This product includes Quartz - Enterprise Job Scheduler software, which is Copyright 2004-2005 OpenSymphony

This product includes Free DCE software, which is (c) Copyright 1994 OPEN SOFTWARE FOUNDATION, INC., (c) Copyright 1994 HEWLETT-PACKARD COMPANY, (c) Copyright 1994 DIGITAL EQUIPMENT CORPORATION, Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

This product includes DCE Threads software, which is Copyright (C) 1995, 1996 Michael T. Peterson

This product includes Jboss software, which is Copyright 2006 Red Hat, Inc. All rights reserved.

## Trademark Notices

Windows® and Windows Server™ 2003 are U.S. registered trademarks of Microsoft® Corp.

UNIX® is a registered trademark of The Open Group.

Java™ and all Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Oracle® is a registered US trademark of Oracle Corporation, Redwood City, California.

Adobe® is trademark of Adobe Systems Incorporated.

## Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

**<http://h20230.www2.hp.com/selfsolve/manuals>**

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to:

**<http://h20229.www2.hp.com/passport-registration.html>**

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

## Support

Visit the HP Software Support web site at:

**[www.hp.com/go/hpsoftwaresupport](http://www.hp.com/go/hpsoftwaresupport)**

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

**<http://h20229.www2.hp.com/passport-registration.html>**

To find more information about access levels, go to:

**[http://h20230.www2.hp.com/new\\_access\\_levels.jsp](http://h20230.www2.hp.com/new_access_levels.jsp)**



# Contents

1	Introduction	27
	Document Structure	27
	Document Conventions	28
	Command Line Syntax	29
	Functional List of Commands	31
	Node Management	31
	Collectors	32
	Process Control	33
	Table Maintenance Tools and Utilities	34
	Export and Import Tools	35
	Data Processes	36
	Utilities	37
	User Interface	38
	Reporting Tools	38
2	action	41
	Requirements and Restrictions	41
	Syntax	42
	Options	42
	Usage Notes	44
	Installation Issues	44
	General Issues	44
	Collection Issues	45
	Settings	45
	Possible Failures Issues	46
	Example	47
	Messages	47
	General	47

	Status Messages . . . . .	48
<b>3</b>	<b>builder . . . . .</b>	<b>49</b>
	Requirements and Restrictions . . . . .	49
	Syntax . . . . .	50
	Options . . . . .	51
	Usage Notes . . . . .	54
	Examples . . . . .	54
	Error Messages . . . . .	55
<b>4</b>	<b>collection_manager . . . . .</b>	<b>57</b>
	Requirements or Restrictions . . . . .	57
	Syntax . . . . .	58
	Options . . . . .	59
	Naming Convention . . . . .	62
	Usage Notes . . . . .	64
	Modes of Operation . . . . .	64
	Import . . . . .	64
	Modify . . . . .	64
	Export . . . . .	65
	Remove . . . . .	65
	ASCII File . . . . .	65
	Using the collection_manager Command . . . . .	69
	Examples . . . . .	71
	Import Example . . . . .	71
	Modify Example . . . . .	72
	Modify_All Example . . . . .	72
	Export Examples . . . . .	73
	Remove_All Examples . . . . .	73
	Remove Example . . . . .	74
	Error Messages . . . . .	74
<b>5</b>	<b>copypolicy_manager . . . . .</b>	<b>81</b>
	Requirements and Restrictions . . . . .	81
	Syntax . . . . .	81
	Options . . . . .	82



Example .....	82
<b>6 datapipe_manager</b> .....	85
Requirements and Restrictions .....	85
Syntax .....	86
Option Categories .....	87
Options .....	88
Usage Notes .....	95
Examples .....	95
Error Messages .....	97
Command Line Option Errors .....	97
Generic File I/O Errors .....	99
Generic PI Errors .....	100
Generic Database Errors .....	102
Generic Database Connection Errors .....	102
<b>7 db_delete_data</b> .....	103
Requirements and Restrictions .....	103
Syntax .....	104
Options .....	105
Usage Notes .....	108
Examples .....	108
Error Messages .....	109
<b>8 deploytool</b> .....	113
Requirements and Restrictions .....	113
Syntax .....	114
Options .....	114
Usage Notes .....	118
Using the deploytool Command .....	118
Deploy Reports .....	118
Undeploy Reports .....	119
Examples .....	121
Error Messages .....	123
<b>9 dip_manager</b> .....	127

Requirements or Restrictions . . . . .	127
Syntax . . . . .	128
Options . . . . .	129
Naming Convention . . . . .	131
Usage Notes . . . . .	132
Modes of Operation . . . . .	133
Import . . . . .	133
Replace . . . . .	133
Export . . . . .	133
Remove . . . . .	134
ASCII File . . . . .	134
Using the dip_manager Command . . . . .	136
Examples . . . . .	138
Import Example . . . . .	138
Replace Example . . . . .	139
Export Examples . . . . .	140
Export_All Examples . . . . .	140
Remove Example . . . . .	141
Remove_All Examples . . . . .	141
Error Messages . . . . .	142
<b>10 ee_collect . . . . .</b>	<b>147</b>
Requirements and Restrictions . . . . .	147
Syntax . . . . .	148
Format 1 . . . . .	148
Format 2 . . . . .	149
Options . . . . .	150
Usage Notes . . . . .	153
Interval Polling (CBC Mode) . . . . .	153
Direct Polling (STF Mode) . . . . .	153
Log File . . . . .	153
Using the ee_collect Command . . . . .	153
STF Mode . . . . .	154
CBC Mode . . . . .	155
Examples . . . . .	156

Direct Polling of a Datapipe (STF Mode) . . . . .	156
Interval Polling (CBC Mode) . . . . .	156
Error Messages . . . . .	157
General Errors . . . . .	157
TEEL File Statement Errors . . . . .	160
Command Line Option Errors . . . . .	162
Generic File I/O Errors . . . . .	164
<b>11 formdeploytool</b> . . . . .	167
Requirements and Restrictions . . . . .	167
Syntax . . . . .	168
Options . . . . .	169
Usage Notes . . . . .	171
Using the formdeploytool Command . . . . .	171
Deploy Forms . . . . .	171
Undeploy Forms . . . . .	172
Examples . . . . .	173
Error Messages . . . . .	174
<b>12 generate</b> . . . . .	177
Syntax . . . . .	177
Options . . . . .	178
Usage Notes . . . . .	179
Using the generate Command . . . . .	179
Examples . . . . .	180
<b>13 groupctl</b> . . . . .	183
Requirements and Restrictions . . . . .	183
Syntax . . . . .	184
Options . . . . .	185
Usage Notes . . . . .	187
Modes of Operation . . . . .	187
Add . . . . .	188
Modify . . . . .	188
Delete . . . . .	188
Using the groupctl Command . . . . .	188

Add .....	189
Modify.....	189
Delete .....	191
Examples .....	191
Error Messages .....	192
<b>14 groupimport .....</b>	<b>195</b>
Requirements or Restrictions .....	195
Syntax.....	195
Options .....	196
Usage Notes .....	197
Naming Conventions .....	197
File Format .....	197
Example .....	203
<b>15 group_manager .....</b>	<b>205</b>
Requirements or Restrictions .....	205
Syntax.....	206
Options .....	207
Naming Convention.....	213
Usage Notes .....	214
About Groups .....	214
Modes of Operation.....	215
Import.....	215
Export.....	216
Remove.....	217
Definition Files .....	217
Document Type Definition for Group Definitions .....	218
Document Type Definition for Polling Policies.....	219
Using the group_manager Command .....	220
Examples .....	223
Import Examples .....	223
Export Examples.....	224
Remove Examples.....	225
Remove_Policy Examples .....	225
Error Messages .....	226

Syntax Messages.....	226
Value Messages.....	227
<b>16 install.pkg .....</b>	<b>231</b>
Requirements and Restrictions.....	231
Report Pack Directory Structure.....	232
Install.pkg Processing .....	233
Install.pkg Directives.....	233
Install.pkg Syntax Rules .....	233
Document Conventions for Directives .....	234
bin .....	234
Syntax .....	234
Example .....	235
Usage Notes .....	235
database_procedure .....	235
Syntax .....	235
Example .....	235
Usage Notes .....	235
database_table .....	236
Syntax .....	236
Example .....	236
Usage Notes .....	236
database_view.....	237
Syntax .....	237
Example .....	237
Usage Notes .....	237
db_requirement.....	237
Syntax .....	237
Example .....	238
Usage Notes .....	238
dependency .....	238
Syntax .....	238
Example .....	239
Usage Notes .....	239
Legacy Format .....	239
dll.....	240

Syntax .....	240
Example .....	240
Usage Notes .....	240
docdir .....	241
Syntax .....	241
Example .....	241
Usage Notes .....	241
docs .....	242
Syntax .....	242
Example .....	242
Usage Notes .....	242
form .....	242
Syntax .....	243
Example .....	243
Usage Notes .....	243
formdir .....	243
Syntax .....	243
Example .....	244
Usage Notes .....	244
lib .....	244
Syntax .....	244
Example .....	244
Usage Notes .....	245
mibs .....	245
Syntax .....	245
Example .....	245
Usage Notes .....	245
mw_collection_def .....	245
Syntax .....	246
Example .....	246
Usage Notes .....	246
report_dir .....	247
Syntax .....	247
Example .....	247
Usage Notes .....	247
reportlink .....	247

Syntax .....	248
Example .....	248
Usage Notes .....	248
report_pack .....	248
Syntax .....	248
Example .....	249
Usage Notes .....	249
reports .....	249
Syntax .....	249
Example .....	249
Usage Notes .....	250
run_command .....	250
Syntax .....	250
Example .....	250
Usage Notes .....	250
run_sql_script .....	251
Syntax .....	251
Example .....	251
Usage Notes .....	251
scripts .....	251
Syntax .....	251
Example .....	251
Usage Notes .....	252
software_version .....	252
Syntax .....	252
Example .....	253
Usage Notes .....	253
sql. ....	253
Syntax .....	253
Example .....	254
Usage Notes .....	254
tmp. ....	254
Syntax .....	254
Example .....	254
Usage Notes .....	255
trend_timer_def. ....	255

Syntax .....	255
Example .....	255
Usage Notes .....	255
uninstall_cmd .....	256
Syntax .....	256
Example .....	256
Usage Notes .....	256
uninstall_sql .....	256
Syntax .....	256
Example .....	257
Usage Notes .....	257
version .....	257
Syntax .....	257
Example .....	257
Usage Notes .....	257
Sample Layout for the File .....	259
<b>17 indexmaint.</b> .....	261
Requirements and Restrictions .....	261
Syntax .....	261
Options .....	262
Usage Notes .....	264
Table Indexes .....	264
Sybase .....	265
Oracle .....	268
Processing Considerations .....	270
Examples .....	271
<b>18 log_backup</b> .....	273
Syntax .....	273
Options .....	274
Naming Conventions .....	274
Usage Notes .....	275
<b>19 mw_collect</b> .....	277
Requirements and Restrictions .....	277



Syntax . . . . .	277
Option Categories . . . . .	279
Options . . . . .	280
Usage Notes . . . . .	290
Terms . . . . .	290
mw_collect Command . . . . .	291
Configuration File . . . . .	292
Syntax . . . . .	292
Parameters . . . . .	293
By-Variables . . . . .	297
File Locks . . . . .	298
Local Storage of Data . . . . .	298
Interval Polling . . . . .	301
Distributed Polling . . . . .	301
Direct Polling . . . . .	302
Multiple Pollers . . . . .	302
Log File . . . . .	303
Directory Structure . . . . .	304
Examples . . . . .	305
Interval Polling . . . . .	305
Direct Polling . . . . .	305
<b>20 node_manager . . . . .</b>	<b>307</b>
Requirements or Restrictions . . . . .	307
Syntax . . . . .	308
Options . . . . .	308
Usage Notes . . . . .	311
Modes of Operation . . . . .	311
Import . . . . .	311
Delete . . . . .	311
Remove . . . . .	311
Export . . . . .	312
ASCII File . . . . .	312
Using the node_manager Command . . . . .	315
Examples . . . . .	317
Import Examples . . . . .	317

Export Examples.....	318
Remove Examples.....	319
Delete Example.....	319
Error Messages .....	320
<b>21 ovpi_bulk_copy .....</b>	<b>323</b>
Requirements and Restrictions.....	323
Syntax.....	324
Options .....	325
Usage Notes .....	327
Using the ovpi_bulk_copy Command.....	327
Example .....	329
Error Messages .....	330
<b>22 ovpi_run_sql .....</b>	<b>331</b>
Requirements and Restrictions.....	331
Syntax.....	331
Options .....	332
Usage Notes .....	334
Using the ovpi_run_sql Command.....	334
Example .....	335
Error Messages .....	336
<b>23 pa_collect .....</b>	<b>337</b>
Syntax.....	337
Options .....	339
Usage Notes .....	349
<b>24 pa_discovery.....</b>	<b>351</b>
Requirements or Restrictions .....	351
Syntax.....	352
Options .....	352
Usage Notes .....	354
Processing Considerations .....	354
Configuration File Settings .....	354
Syntax .....	355

Parameters .....	356
<b>25 piadmin</b> .....	359
Syntax .....	359
Options .....	360
Usage Notes .....	363
Examples .....	364
<b>26 schedule</b> .....	365
Requirements and Restrictions .....	365
Syntax .....	366
Options .....	367
Usage Notes .....	374
Modes of Operation .....	374
Add .....	374
List .....	374
Remove .....	374
Using the schedule Command .....	374
List .....	375
Remove .....	375
Add .....	375
Examples .....	378
Error Messages .....	380
<b>27 snmpv2dis</b> .....	383
Syntax .....	383
Options .....	384
Usage Notes .....	385
<b>28 tpmaint</b> .....	387
Requirements or Restrictions .....	387
Syntax .....	388
Options .....	389
Usage Notes .....	390
Examples .....	391

<b>29 transform_maint</b>	393
Requirements and Restrictions	393
Syntax	394
Options	395
Usage Notes	396
Using the transform_maint Command	397
Examples	399
Error Messages	400
<b>30 trendcopy</b>	403
Requirements or Restrictions	403
Syntax	405
Option Categories	406
Options	407
Usage Notes	414
Performance Notes	414
Capabilities	415
Keymap Tables	415
Examples	416
Using Various Options to Copy Database Tables	416
Using the Row Filter Options To Copy Data by Date	417
Using the Specific and Debug Options	418
Error Messages	420
General	420
Server Name Error Messages	420
Source Server	421
Destination Server	422
Table Name Error Messages	422
View Error Messages	424
By-variable Error Messages	424
Foreign Key Error Messages	425
Option Error Messages	426
<b>31 trend_discover</b>	429
Requirements or Restrictions	429

IP Discover Syntax . . . . .	430
IP Discover Options . . . . .	431
IP Discover Usage Notes . . . . .	434
How IP Discover Works . . . . .	434
IP Discover View Population . . . . .	435
Specifying Community Strings for an SNMP GET Request . . . . .	435
Community Strings Files . . . . .	436
Example . . . . .	436
SNMP Type Discover Syntax . . . . .	437
SNMP Type Discover Options . . . . .	438
Naming Convention . . . . .	439
Type Discover Usage Notes . . . . .	439
SNMP Type Discover View Population . . . . .	440
Type Definition Files . . . . .	440
Syntax . . . . .	440
SNMP Tests . . . . .	441
Type Definition File Examples . . . . .	443
<b>32 trendexec</b> . . . . .	445
Requirements or Restrictions . . . . .	445
Syntax . . . . .	445
Options . . . . .	446
Example . . . . .	446
<b>33 trend_export</b> . . . . .	447
Syntax . . . . .	447
Overall Process and Functionality . . . . .	448
Controls and Switches . . . . .	449
Input Parameter File . . . . .	452
Examples . . . . .	456
Output Format . . . . .	460
Tab Separated Format: Not Condensed . . . . .	460
Comma Format: Not Condensed . . . . .	461
Tab Format: Condensed . . . . .	462
Comma Format: Condensed . . . . .	462

<b>34 trend_label</b>	463
Syntax	464
Options	465
Usage Notes	467
Populating the dsi_descr Column	467
Locating the Target Row in the Source Data Table	467
Ensuring Property Table/Data Table Compatibility	469
Update Restrictions	469
Extracting Substrings from Column Values	469
Concatenating Column Values	469
Examples	470
<b>35 trendpm</b>	473
Syntax	473
Options	474
pe Option Parameters	476
Procedure Application Type Values	479
<b>36 trend_proc</b>	481
Requirements and Restrictions	481
Syntax	481
Options	482
Usage Notes	483
The trend_proc Input File	483
Input File Definitions	483
Input File Characteristics	484
Processing the Input File	485
Processing with Multiple trend_proc Files	487
Creating a trend_proc File	488
Scheduling a trend_proc File	489
Examples	489
<b>37 trend_sum</b>	493
Requirements or Restrictions	493
Syntax	494
Command-Line Options	494

Input File Keywords . . . . .	495
Option Categories . . . . .	496
Keywords . . . . .	496
Options . . . . .	503
Naming Conventions . . . . .	505
Usage Notes . . . . .	507
Input Files . . . . .	508
By-Variables . . . . .	509
Determine the Last Processed Sample for Hysteresis . . . . .	509
Lag Time . . . . .	510
Reprocessing Data . . . . .	511
Rolling Baseline Table . . . . .	511
NULL Handling . . . . .	513
Troubleshooting . . . . .	513
Date Range Error. . . . .	513
Divide-By-Zero Error. . . . .	514
Statistical Formulas and Variables . . . . .	514
Examples . . . . .	518
<b>38 trendtimer . . . . .</b>	<b>521</b>
Requirements and Restrictions . . . . .	521
Syntax . . . . .	522
Options . . . . .	522
Usage Notes . . . . .	523
The Schedule File . . . . .	523
Schedule File Syntax . . . . .	523
Schedule File Example . . . . .	525
Starting trendtimer . . . . .	526
UNIX Platforms . . . . .	526
Windows Platform . . . . .	527
Stopping trendtimer . . . . .	527
UNIX Platforms . . . . .	527
Windows Platform . . . . .	528
Example . . . . .	528
<b>39 userctl . . . . .</b>	<b>529</b>

Requirements and Restrictions . . . . .	529
Syntax . . . . .	530
Options . . . . .	531
Usage Notes . . . . .	533
Modes of Operation . . . . .	533
Add . . . . .	534
Modify . . . . .	534
Delete . . . . .	534
Using the userctl Command . . . . .	534
Add . . . . .	535
Modify . . . . .	536
Delete . . . . .	537
Examples . . . . .	538
Add . . . . .	538
Modify . . . . .	539
Delete . . . . .	540
Error Messages . . . . .	541
<b>40 userimport . . . . .</b>	<b>543</b>
Requirements or Restrictions . . . . .	543
Syntax . . . . .	544
Options . . . . .	544
Usage Notes . . . . .	545
Naming Conventions . . . . .	545
File Format . . . . .	546
Example . . . . .	549
<b>41 vantage_collect . . . . .</b>	<b>551</b>
Requirements and Restrictions . . . . .	551
Syntax . . . . .	552
Options . . . . .	554
Usage Notes . . . . .	563
Example . . . . .	564
<b>42 viewctl . . . . .</b>	<b>565</b>
Requirements and Restrictions . . . . .	565



Syntax . . . . .	566
Options . . . . .	567
Usage Notes . . . . .	569
Modes of Operation . . . . .	569
Add . . . . .	569
Modify . . . . .	569
Delete . . . . .	569
Using the viewctl Command . . . . .	569
Add . . . . .	570
Modify . . . . .	571
Delete . . . . .	571
Examples . . . . .	572
Error Messages . . . . .	573
<b>43 viewer . . . . .</b>	<b>575</b>
Requirements and Restrictions . . . . .	575
Syntax . . . . .	576
Options . . . . .	577
Usage Notes . . . . .	579
Examples . . . . .	579
<b>Index . . . . .</b>	<b>581</b>



# 1 Introduction

The purpose of this guide is to assist HP Performance Insight (PI) users with data collection, aggregation, management, and reporting of data used for network management.

This reference guide describes the syntax and discusses the use of command-line interface commands. Each command has a separate chapter and appears in alphabetical order by command name. A functional list of the chapters appears at the end of this chapter; see [Functional List of Commands](#) on page 31.

This reference guide also includes an index of the potential error messages in alphabetical order from the various chapters.

## Document Structure

Chapters containing command-line interface commands present the following types of information:

overview	A brief description appears after the command name in the beginning of the chapter that describes its intended use.
Requirements or Restrictions	This section highlights special conditions or limitations that you may need to remember when you use the command.
Syntax	This section specifies the format for the command name and options that appear on the command line.

Option Categories	This section lists related options by class or groups.
Options	This section lists and describes the options for the command and identifies the valid values for the options.
Naming Conventions	This section defines unique file, directory, and extension naming patterns.
Usage Notes	This section provides detailed information concerning the application of the command.
Examples	This section furnishes practical illustrations of the use of the command.
Error Messages	This section lists potential error messages.

## Document Conventions

The following information describes the style and symbol conventions used in this guide.

<b>Helvetica Bold</b>	The <b>Helvetica Bold</b> typeface identifies window titles, menu items, and other items displayed in the GUI. It is also used in procedural steps to indicate the same; for instance:  Select <b>File</b> → <b>Open</b> from the main menu.
<b>Courier Bold</b>	The <b>Courier Bold</b> typeface indicates command line commands, command line options, and code listings used for input such as: <b><code>trend_discover -t</code></b>

<i>Italic</i>	<p><i>Italic</i> text represents command line <i>variables</i> and <i>parameters</i> that are placeholders for values; for example:</p> <p><b>trendcopy -s</b> <i>source_server</i></p> <p>This typeface also identifies <i>document titles</i>.</p>
Courier	<p>The Courier typeface identifies output information from the system such as error messages and other screen messages. It also identifies file names and commands.</p>
Blue	<p>If you are viewing this document as a PDF file, blue text indicates a hypertext link.</p>

## Command Line Syntax

The following information shows the conventions used for command-line interface commands.

Items listed but not contained in brackets ([ ]) are required.

**trendcopy -S** *target\_server*

Note that the **-S** option and the *target\_server* variable are both required.

[ ] Items that appear inside brackets ([ ]) are optional. For example:

**datapipe\_manager** [-T *table\_name*]

Note that the **-T** option and the *table\_name* variable are not required; the brackets are not part of the syntax.

**Bold**

**Bold** items represent commands, options, and keywords that, if used, must be entered as shown. For example:

**trend\_discover -t**

*Italics*

Items that appear in *italics* are parameters that require an actual value. For example, enter an actual table name to replace the *table\_name* parameter.

**datapipe\_manager -T *table\_name***

## { }

When multiple keywords appear in braces ({ }), one of the keywords must be entered on the command line along with the command and option. For example, enter

**datapipe\_manager -p create** as a valid command from the following syntax.

**{  
alter  
create  
delete  
remove  
verifyparms  
}**

**datapipe\_manager -p {  
alter  
create  
delete  
remove  
verifyparms  
}**

# Functional List of Commands

This is a list of the HP Performance Insight (PI) commands grouped by function. For example, the commands that you can use to maintain database tables are in a section called Table Maintenance Tools and Utilities. The list for each section is in alphabetical order. This list contains the following functional groups:

- [Node Management](#) on page 31
- [Collectors](#) on page 32
- [Process Control](#) on page 33
- [Table Maintenance Tools and Utilities](#) on page 34
- [Export and Import Tools](#) on page 35
- [Data Processes](#) on page 36
- [Utilities](#) on page 37
- [User Interface](#) on page 38
- [Reporting Tools](#) on page 38

## Node Management

This section includes the commands that pertain to node management, which are the commands that allow you to manage the nodes on your PI system by discovering them automatically, or by adding, modifying, or deleting them.

**Table 1 Node Management Commands**

Module	Description
<code>node_manager</code>	The <code>node_manager</code> command is a stand-alone utility that enables you to manage nodes, types, and views. It also enables you to manage all relevant SNMP properties for nodes.
<code>pa_discovery</code>	The <code>pa_discovery</code> command is a utility that enables you to discover HP Performance Agent or HP Operations agent on a PI system.
<code>snmpv2dis</code>	The SNMP V2 Discovery Utility ( <code>snmpv2dis</code> ) identifies devices that support the SNMP V2 protocol on a PI system.
<code>trend_discover</code>	The <code>trend_discover</code> command is a utility that enables you to start an automated process to find the nodes on the system, ascertain whether each node is SNMP manageable, identify the type of device, and automatically update tables that control data collection.

## Collectors

This section includes the commands that pertain to collecting data on your PI system. These commands include setting parameters for specific types of devices, maintaining the polling policies for the collection process, or importing data from a flat file.



**Table 2 Collector Commands**

Module	Description
<code>action</code>	You can use the <code>action</code> command to monitor and report end-to-end network performance characteristics on a PI system.
<code>collection_manager</code>	You can use the <code>collection_manager</code> command to add, modify, remove, and export polling policies on a PI system.
<code>dip_manager</code>	You can use the <code>dip_manager</code> command to import, replace, remove, or export directed- instance polling groups on a PI system.
<code>ee_collect</code>	You can use the <code>ee_collect</code> command to import data from a flat file into a datapipe on a PI system.
<code>group_manager</code>	You can use the <code>group_manager</code> command to manage group definitions and polling policies.
<code>mw_collect</code>	You can use the <code>mw_collect</code> command to collect SNMP data from nodes on a PI system.
<code>pa_collect</code>	You can use the <code>pa_collect</code> command to collect data from HP Performance Agent or HP Operations agent on a PI system.
<code>vantage_collect</code>	You can use the <code>vantage_collect</code> command to collect SNMP data from two tables that have a control table and data table relationship on a PI system.

## Process Control

This section includes the commands that pertain to controlling the processing of the data on your PI system.

**Table 3 Process Control Commands**

Module	Description
<code>trendexec</code>	The <code>trendexec</code> program uses the <code>trend_sum</code> program to run the <code>trend_sum</code> procedures listed in the database
<code>trend_proc</code>	The <code>trend_proc</code> command is a utility that enables you to group together multiple interrelated commands on a PI system.
<code>trendtimer</code>	The <code>trendtimer</code> program is the PI scheduler that runs specific commands at scheduled times.

## Table Maintenance Tools and Utilities

This section includes the commands that pertain to maintaining the database tables on your PI system. These commands allow you to create, modify, and remove tables in your database and to maintain those tables by aging out old data or maintaining the indexes for the table.

**Table 4 Table Maintenance Commands**

Module	Description
<code>datapipe_manager</code>	The <code>datapipe_manager</code> command is a utility for creating data and property tables and views in a PI database.
<code>db_delete_data</code>	The <code>db_delete_data</code> command is a utility used to age obsolete data out of the database.
<code>indexmaint</code>	The <code>indexmaint</code> command is a utility used maintain indexes of existing PI tables in the database.

## Export and Import Tools

This section includes the commands that pertain to exporting and importing information such as data, nodes, groups, or polling policies on your PI system.

**Table 5    Export and Import Commands**

Module	Description
<code>collection_manager</code>	You can use the <code>collection_manager</code> command to export and import polling policies on a PI system.
<code>dip_manager</code>	You can use the <code>dip_manager</code> command to import or export directed-instance polling groups on a PI system.
<code>ee_collect</code>	You can use the <code>ee_collect</code> command to import data from a flat file into a datapipe on a PI system.
<code>groupimport</code>	You can use the <code>groupimport</code> command to import multiple user groups with an XML file from the command line on a PI system.
<code>group_manager</code>	You can use the <code>group_manager</code> command to export and import group definitions and polling policies.
<code>node_manager</code>	You can use the <code>node_manager</code> command to export and import nodes on a PI system.
<code>ovpi_bulk_copy</code>	You can use the <code>ovpi_bulk_copy</code> command to bulk load or extract data on a PI system.
<code>trend_export</code>	You can use the <code>trend_export</code> program to export data from the database into a text file on a PI system.
<code>userimport</code>	You can use the <code>userimport</code> command to import multiple user accounts that access the Web Access Server with an XML file from the command line on a PI system.

## Data Processes

This section includes the commands that pertain to the processing of the data on your PI system, such as by copying the data, summarizing it, or managing the stored procedures that manipulate it.

**Table 6    Data Processing Commands**

Module	Description
<a href="#">transform_maint</a>	The <code>transform_maint</code> command is a utility that enables you to perform maintenance tasks related to transformations on a PI system.
<a href="#">trendcopy</a>	The <code>trendcopy</code> command is a utility that enables you to copy data from one PI database to another.
<a href="#">trend_label</a>	The <code>trend_label</code> command is a utility that populates one or more columns in a property table with data from its counterpart in the data table.
<a href="#">trendpm</a>	The <code>trendpm</code> command is a utility that manages raw-to-delta and copy stored procedures on a PI system.
<a href="#">trend_sum</a>	The <code>trend_sum</code> command is an application that manages summarization and aggregation stored procedures on a PI system.

## Utilities

This section includes the commands that pertain to miscellaneous utilities for the PI system.

**Table 7    Utility Commands**

Module	Description
<code>groupctl</code>	You can use the <code>groupctl</code> command to add, delete, or modify a single user group from the command line on a PI system.
<code>groupimport</code>	You can use the <code>groupimport</code> command to add, delete, or modify multiple user groups with an XML file from the command line on a PI system.
<code>log_backup</code>	The <code>log_backup</code> command is a utility that enables you to move a specified file to a new file as a back up. The default backs up the <code>trend.log</code> file to a new file each day.
<code>ovpi_bulk_copy</code>	The <code>ovpi_bulk_copy</code> command is a utility that enables you to bulk load or extract data on a PI system.
<code>ovpi_run_sql</code>	The <code>ovpi_run_sql</code> command is a utility that enables you to run SQL scripts on a PI system.
<code>tpmaint</code>	The <code>tpmaint</code> command is a utility that populates time-period tables, which enable optimized searches of the database.
<code>trend_export</code>	You can use the <code>trend_export</code> program to export data from the database into a text file.

**Table 7    Utility Commands**

Module	Description
<code>userctl</code>	You can use the <code>userctl</code> command to add, delete, or modify a single user account for accessing the Web Access Server from the command line on a PI system.
<code>userimport</code>	You can use the <code>userimport</code> command to add, delete, or modify multiple user accounts that access the Web Access Server with an XML file from the command line on a PI system.
<code>viewctl</code>	You can use the <code>viewctl</code> command to add, delete, or modify a catalog view from the command line on a PI system.

## User Interface

This section includes the commands that pertain to starting the graphical user interface (GUI) clients from the command line on a PI system.

**Table 8    User Interface Commands**

Module	Description
<code>builder</code>	The <code>builder</code> command is the command that you enter to start the Report Builder client application.
<code>piadmin</code>	The <code>piadmin</code> command is the command that you enter to start the Management Console from the command line.
<code>viewer</code>	The <code>viewer</code> command is the command that you enter to start the Report Viewer client application.

## Reporting Tools

This section includes the commands that pertain to the reporting functions on a PI system such as creating, generating, scheduling, deploying, and undeploying reports and forms.

**Table 9     Reporting Tools Commands**

<b>Module</b>	<b>Description</b>
<code>builder</code>	The <code>builder</code> command is the command that you enter to start the Report Builder client application.
<code>deploytool</code>	The <code>deploytool</code> command is a stand-alone utility that enables you to deploy reports or a folder of reports to the Web Access Server so that you can view them from the Web Access Server. You can also use this command to undeploy reports or a folder of reports from the Web Access Server, which removes them from view on the Web Access Server.
<code>formdeploytool</code>	The <code>formdeploytool</code> command is a stand-alone utility that enables you to deploy forms or a folder of forms to the PI Administration Server so that you can view them from the Object Manager. You can also use this command to undeploy forms or a folder of forms from the PI Administration Server, which removes them from view on the Object Manager.
<code>generate</code>	The <code>generate</code> command is the command that you enter to generate the reports for a particular schedule.
<code>install.pkg</code>	The <code>install.pkg</code> file provides Package Manager with the basic information needed to install a report pack.
<code>schedule</code>	The <code>schedule</code> command is a tool that enables you to configure schedules on a PI system.
<code>viewer</code>	The <code>viewer</code> command is the command that you enter to start the Report Viewer client application.





## 2 action

You can use the HP Performance Insight (PI) `action` command, which is the Action poller, to monitor and report end-to-end network performance characteristics on a PI system. In this case, the Action poller performs two directed-instance polls, pre and post, around a central action, which is the collection of the Cisco Ping round trip data between the nodes of each pair.

### Requirements and Restrictions

- This command requires the installation of the `Ping_MIB` report pack.
- To collect data with this command, the collector module for the data table must be **MW**.
- You can invoke the Action Poller from the command line or have `trendtimer` invoke it through an entry in the `trendtimer.sched` file.

# Syntax

The `action` command has the following syntax:

```
action  [-c child_processes]  
        [-d debug_level]  
        [-h]  
        [-i polling_interval]  
        [-p port]  
        [-r retries]  
        [-s round_factor]  
        [-t timeout]  
        [-v]
```

# Options

The `action` command has the following options:

- c**      Use this option to specify the number of concurrent child action poller processes to run. The Action poller runs one child process for each node pair. The processing power of the host determines the value you specify for this option.  
The default is 20.
- d**      Use this option to specify the debug level. Valid values are 0, 1, 2, or 3. The higher the number, the more detailed the debug output.  
The default is 0, which means no debug output. Debug output is written to the standard out. Use this option only with HP Technical Support for testing purposes because of the additional overhead it incurs.
- h**      Use this option to display the command line options (help).

- i** Use this option to specify the polling interval. Each collection group that has the specified polling interval is collected during this invocation of the Action poller.
- p** Use this option to specify the port number for SNMP sessions. The default is 161.
- r** Use this option to specify the number of collection retries to attempt for each SNMP session. The default is -1, which means the default number of tries configured for SNMP.
- s** Use this option to specify the number of seconds for the round factor, which corresponds to the collection interval. The action poller uses this value to round off the collection time (*ta\_period*); for example, if the Action poller kicks off a collection at 3:07, and if you are using the default collection option of 300 seconds (5 minutes), the actual recorded *ta\_period* value for the collection is 3:05. The default is 300.
- t** Use this option to specify the collection timeout, in seconds, for SNMP sessions. The default is -1, which means the default timeout configured for SNMP.
- v** Use this option to display the version stamp for the *action* command. This option is in UPPERCASE.

# Usage Notes

This command is available only when the Ping\_MIB report pack exists on the system. See the documentation with the report pack for more information about using this command.

## Installation Issues

When you use Package Manager to install the Ping\_MIB report pack, the system adds the following entry to the `trendtimer.sched` file:

```
15 - - {DPIPE_HOME}/bin/trend_proc -f {DPIPE_HOME}/scripts/  
Ping_MIB_poll.pro
```

There is an entry in this file to run the Action poller. When the system executes the command every 15 minutes, the entry is the following command:

```
action -i 15
```

To test the initial or new group collection manually, perform the following steps:

- 1 Comment out the Ping\_MIB entry in the `trendtimer.sched` file, if it is not already commented out.

```
#15 - - {DPIPE_HOME}/bin/trend_proc -f {DPIPE_HOME}/scripts/  
Ping_MIB_poll.pro
```

- 2 Enter the following command (if the polling interval is 15):

```
action -i 15
```

- 3 To restore automatic scheduling, remove the comment character from the Action poller entry in the `trendtimer.sched` file.

## General Issues

The file that the system uses to determine if a device is a Cisco Ping device is the `cisco_ping.dis` file. It contains a list of *oids*, which are object ids that identify the device.

If the Cisco Ping agent is running on the device, the poller reads the agent every second looking for completion. It polls for the following two oids:

```
ciscoPingReceivedPackets (1.3.6.1.4.1.9.9.16.1.1.1.10)
ciscoPingCompleted (1.3.6.1.4.1.9.9.16.1.1.1.14)
```

## Collection Issues

There is only one child collector, which is `action_collect`, for `action`.

To collect data with this command, the collector module for the data table must be **MW**.

When there are received packets, the action poller does an SNMP GET to retrieve various oid values such as the min, max, and average round trip times. The min, max, and average round trip times values are rate type data. The oid values follow:

```
ciscoPingPacketSize (1.3.6.1.4.1.9.9.16.1.1.1.5)
ciscoPingSentPackets (1.3.6.1.4.1.9.9.16.1.1.1.9)
ciscoPingReceivedPackets (1.3.6.1.4.1.9.9.16.1.1.1.10)
ciscoPingMinRtt (1.3.6.1.4.1.9.9.16.1.1.1.11)
ciscoPingAvgRtt (1.3.6.1.4.1.9.9.16.1.1.1.12)
ciscoPingMaxRtt (1.3.6.1.4.1.9.9.16.1.1.1.13)
```

## Settings

The following messages identify the settings for the Action poller, which are the Ping Sets. The configuration process sets the Ping Set parameters. See the `Ping_MIB` report pack documentation for more information about setting these parameters.

```
Setting ciscoPingEntryOwner (1.3.6.1.4.1.9.9.16.1.1.1.15) to
owner.
Setting ciscoPingEntryStatus (1.3.6.1.4.1.9.9.16.1.1.1.16) to
4.
Setting ciscoPingProtocol (1.3.6.1.4.1.9.9.16.1.1.1.2) to 1.
Setting ciscoPingAddress (1.3.6.1.4.1.9.9.16.1.1.1.3) to
address.
Setting ciscoPingPacketCount (1.3.6.1.4.1.9.9.16.1.1.1.4) to
5.
Setting ciscoPingPacketSize (1.3.6.1.4.1.9.9.16.1.1.1.5) to
100.
Setting ciscoPingPacketTimeout (1.3.6.1.4.1.9.9.16.1.1.1.6)
to 2000.
Setting ciscoPingDelay (1.3.6.1.4.1.9.9.16.1.1.1.7) to 0.
```

where,

- *owner* is the administrator user name, which is typically **trendadm**.
- *address* is the destination node, which is the ip or name.
- When the `ciscoPingEntryStatus` setting is 4, it means that the system initiates the ping sequence.
- When the `ciscoPingProtocol` setting is 1, it means that the address is an ip address.
- The `ciscoPingPacketSize` value is in bytes.
- The `ciscoPingPacketTimeout` and `ciscoPingDelay` values are in milliseconds.
- When the `ciscoPingDelay` setting is 0, it means that the system should not wait.

All other values are the default values.

## Possible Failures Issues

Some of the usual causes of Ping Set failures include the following reasons.

- The source node is down.
- The source node is not a Cisco Ping Device.
- The Cisco Ping Agent is not running on the device.
- The write community string for Cisco Ping Device is incorrect.
- One or more of the destination nodes are down and ping sequence times out.

## Example

The following command invokes the Action poller to collect the collection table instances from the node pairs defined for all collection groups with a polling interval of 15:

```
action -i 15
```

## Messages

This section describes some of the messages that can occur from `action`. Each message has the following format:

- A brief description about why the message appears. Each new message description starts with a check box.
- The actual message that appears with parameters for any information that may be missing.
- A suggestion about the action to do so that the message does not appear again.

### General

- ☐ The following message appears when the specified oid is missing from the file that determines if a device is a Cisco Ping device.

Not a Cisco Device.

If the device is a Cisco Ping device, verify that the oid for the device is in the `cisco_ping.dis` file.

- ☐ The following message appears when the agent is not running.

SNMP\_TIMEOUT.

Verify that the agent is running on the Cisco device.

## Status Messages

- ❑ The following message appears when the system polls the device, and the ping did not complete.

```
check ping results: ciscoPingCompleted = false
```

The system continues to poll the device while this statement is set to false.

- ❑ The following message appears when the system polls the device, and the ping did complete.

```
check ping results: ciscoPingCompleted = true, packets  
received = number
```

*number* is the number of packets received by the device.

This message does not appear if the timeout message appears.



## 3 builder

You can use the `builder` command to start the Report Builder client application from the command line on a PI system.

### Requirements and Restrictions

- When you connect to a different server with the **-server** option, use the **Browse** option to view the files on the specified server.
- When you use the **-mode remote** option, you must include the **-file** option on the command line at the same time.
- When you use the **-file** option without the **-mode** option on the command line, the system opens the specified file on the local system.
- When you change a parameter with the **-params** option, remember that it is a global option and it applies to every report you open that has the parameter specified.

# Syntax

The `builder` command uses the following syntax:

```
builder [-laf]
          [-h]
          [-help]
          [-u]
          [-p password]
          [-server Application server host]
          [-port port]
          [-file reportname]
          [-debug ]
          [ -log logfile name]
          [-mode (local | remote)]
          [-directory directory path]
          [-params]
          [-protocol ( http | https )]
```

# Options

The `builder` command has the following options:

- |                   |   |
|-------------------|---|
| <b>-debug</b>     | Use this option to enable diagnostic messages, which are an extra level of detail included in the log file.<br>Valid values are: <ul style="list-style-type: none"><li>• <b>true</b>: enables diagnostic messages</li><li>• <b>false</b>: does not enable diagnostic messages</li></ul> The default is <b>false</b> .   |
| <b>-file</b>      | Use this option to specify the name of the report you want to open automatically when you run <code>builder</code> . You can use the absolute or relative path with the name of the report file. For a remote file, you need to give the remote location in reference to its deployed location.   |
| <b>-log</b>       | Use this option to specify the name of the log file to open. Include the path for the name. The name should have a slash as the first character in the name because the system adds the prefix <code>PI</code> to the specified name.<br>The default log file is <code>builder.log</code> .   |
| <b>-mode</b>      | Use this option to specify the location for the file you want to access. There are two values; they are: <ul style="list-style-type: none"><li>• <b>local</b>: when the file is on the local system.</li><li>• <b>remote</b>: when the file is on the Web Access Server.</li></ul> The default is <b>local</b> .<br>Use the <b>-file</b> option to specify the name and location of the file. You must use the <b>-file</b> option with this option to open a remote file automatically when you run <code>builder</code> . |
| <b>-p</b>         | This option specifies the password for the login process.<br>If you do not use this option with the <b>-u</b> option, the system prompts for the username and password.   |
| <b>-directory</b> | This option specifies the directory that contains the reports to open.  |

**-params**

Use this option to specify the report parameters to change report defaults at run time. A parameter has the following format: *parameter=value*. This is a global option; it applies to every report you open that has the parameter specified.

When you specify more than one parameter, separate the parameters with a comma (,). When a parameter value contains a space, enclose all the parameters in one set of quotation marks. The following example shows multiple parameters with one parameter that has a space in the value.

```
-params "INTERFACE=92,CUSTOMER=All Telco"
```

When a parameter value contains a character that is special to the command interface (shell) such as a comma, precede the character with a backward slash (\), for example:

```
-params "INTERFACE=92,CUSTOMER=Telco\,North"
```

See the *HP Performance Insight Guide to Building and Viewing Reports* for details about how to create and view the parameters associated with a report using Report Builder, and for details about how to view and modify the parameters associated with a report using the Web Access Server.

**-protocol**

Use this option to specify the communication protocol. Valid values are http or https. The default value is **http**.

- port** Use this option to specify the port number of the Web Access Server that you want to access from the Report Builder client application.
- The default for this option is the port number supplied during the PI installation, which is port number 80, in most cases.
- server** Use this option to specify the host name of the Web Access Server that you want to access from the Report Builder client application. If you want to access a system in a different domain, you need to specify the full domain name for the host name.
- The default for this option is the server host name supplied during the PI installation.
- u** This option specifies the username for the login process.
- If you do not use this option with the **-p** option, the system prompts for the username and password.

# Usage Notes

You can use Report Builder to build customized reports based on data from the PI database, modify existing reports, and deploy reports to the Web Access Server. See the *HP Performance Insight Guide to Building and Viewing Reports* for more information about using the Report Builder client application.

If you want to browse the reports on a remote system when you run `builder`, you can use the `-server` option alone. However, if you want to open a specific report file automatically when you run `builder`, you can add the `-mode remote` option with the `-file` option.

## Examples

The following examples illustrate some uses of the `builder` command.

### Example 1

If you want to use the Report Builder client application to access reports on a different system such as `powder2`, and you want to bypass the Login dialog box, enter the following command.

```
builder -server powder2 -u trendadm -p trendadm -port 80
```

### Example 2

If you want to use the Report Builder client application to access reports on a different system such as `testsrvr1` with a different log file in the `OVPI/log` directory that has the name, `test_build1.log`, enter the following command.

```
builder -server testsrvr1.abc.xyz.com -log /log/  
test_build1.log
```

# Error Messages

This section describes some of the messages that can occur from builder. Each message has the following format:

- A brief description about why the message appears. Each new message description starts with a check box.
- The actual message that appears with parameters for any information that may be missing.
- A suggestion about the action to do so that the message does not appear again.

- ❑ If the following error message appears, the specified option is missing a value.

Error in arguments.

A value is required for argument *option* : *desc*.

Verify that the specified option has the correct value.

- ❑ If the following error message appears, the specified option is not a valid option.

Error in arguments *option* is not valid for this program.

Verify the spelling of the *option* on the command line. Check [Syntax](#) on page 50 for the list of valid options.

- ❑ If the following error message appears, the system could not connect to the specified server.

Failed to connect to *server\_name*.

Verify the spelling of the *server\_name* on the command line. Note that you may need to use full domain name for the *server\_name*.





## 4 collection\_manager

The `collection_manager` command is a stand-alone utility that enables you to manage polling policies on a PI system. It enables you to add, modify, remove, and export polling policies.

You can also add, modify, and remove polling policies from the Management Console. See the *HP Performance Insight Administration Guide* for more information.

### Requirements or Restrictions

- All required files must have the following parameters in the file: *policy\_name*, *table\_name*, and *group*.
- An error occurs if more than one of the following options appears on the command line at the same time: **-import**, **-export**, **-modify**, **-modify\_all**, **-remove**, or **-remove\_all**.
- An error occurs if the **-file** option does not appear with one of the following options on the command line at the same time: **-import**, **-modify\_all**, **-remove\_all**.

# Syntax

The `collection_manager` command uses the following syntax:

```
collection_manager  [-database db_name]  
                    [-debug debug_level]  
                    [-descr description]  
                    [ {  
                        -export  
                        -import  
                        -modify policy_name  
                        -modify_all  
                        -remove policy_name  
                        -remove_all  
                    } ]  
                    [-file file_name]  
                    [-group group_category.group_name]  
                    [-help]  
                    [-interval num_minutes]  
                    [-pollfrom host_name]  
                    [ {  
                        -v  
                        -version  
                    } ]
```

# Options

The `collection_manager` command has the following options:

- database**      This option identifies the database where the changes occur. The database must appear in the list of available database servers. See the *HP Performance Insight Administration Guide* for more information about adding database servers to the list using the Web Access Server. The default is the database identified as the default in the database server list.
- debug**          Use this option to set the debug output level. The higher the number, the more detailed the information. Debug output writes to standard output. Use this option only for testing in coordination with Technical Support due to the additional overhead it places on `collection_manager`. The default is no debug output.
- descr**          Use this option to modify a polling policy description directly from the command line. If you enter more than one word, you must enclose the description in double quotation marks ("").  
  
You must use the **-modify** option with this option.  
  
This option provides the same type of information as the `desc` parameter in the ASCII file.
- export**          Use this option to generate a file containing the polling policies in the current collection catalog. Use the **-file** option to specify the output file name; otherwise, `collection_manager` writes the data to standard output. If the specified output file already exists, the system overwrites the file. See [ASCII File](#) on page 65 for the format of the file. See [Export](#) on page 65 for more information.  
  
This option cannot appear on the command line when the **-import**, **-modify**, **-modify\_all**, **-remove**, or **-remove\_all** option appears on the command line.

<b>-file</b>	<p>This option identifies the file name, which is the text file that contains the information about the polling policies to import, modify, or remove. If the file is not in the current working directory, you must specify the fully qualified path to the file. See <a href="#">ASCII File</a> on page 65 for details on setting up this file.</p> <p>This is a required option when the <b>-import</b>, <b>-modify_all</b>, or <b>-remove_all</b> option appears on the command line.</p>
<b>-group</b>	<p>Use this option to modify the name of the group assigned to the polling policy from the command line. It must be in the form <i>group_category.group_name</i>. See <a href="#">Naming Convention</a> on page 62 for a more detailed description of this name.</p> <p>You must use the <b>-modify</b> option with this option.</p> <p>This option provides the same type of information as the <i>group</i> parameter in the ASCII file.</p>
<b>-help</b>	<p>This option is the help option, which displays the command-line syntax for the <code>collection_manager</code> command.</p>
<b>-import</b>	<p>Use this option to import polling policies. It requires the <b>-file</b> option to identify the file that contains the list of polling policies to import.</p> <p>This option cannot appear on the command line when the <b>-export</b>, <b>-modify</b>, <b>-modify_all</b>, <b>-remove</b>, or <b>-remove_all</b> option appears on the command line.</p>
<b>-interval</b>	<p>Use this option to modify the polling interval directly from the command line. The polling interval is the number of minutes between polling requests.</p> <p>See the <i>poll_interval</i> parameter in the ASCII file on <a href="#">ASCII File</a> on page 65 for a list of the valid values.</p> <p>You must use the <b>-modify</b> option with this option.</p>

- modify** Use this option to modify an existing polling policy. You must specify the *policy\_name* on the command line following this option.
- This option must appear on the same line as one or more of the following options: **-descr**, **-group**, **-interval**, and **-pollfrom**. See [Modify](#) on page 64 for more information.
- This option cannot appear on the command line when the **-export**, **-import**, **-modify\_all**, **-remove**, or **-remove\_all** option appears on the command line.
- modify\_all** Use this option to modify the polling policies specified with the **-file** option. See [ASCII File](#) on page 65 for a description of the associated file. See [Modify](#) on page 64 for more information.
- This option cannot appear on the command line when the **-export**, **-import**, **-modify**, **-remove**, or **-remove\_all** option appears on the command line.
- pollfrom** Use this option to change the name of the polling station for an existing polling policy directly from the command line.
- You must use the **-modify** option with this option.
- This option provides the same type of information as the *poll\_from* parameter in the ASCII file.
- remove** Use this option to remove a polling policy from the collection catalog. You must specify the *policy\_name* on the command line following this option. See [Remove](#) on page 65 for more information.
- This option cannot appear on the command line when the **-export**, **-import**, **-modify**, **-modify\_all**, or **-remove\_all** option appears on the command line.

- remove\_all**      Use this option to remove all the polling policies listed in the associated file from the collection catalog. It requires the **-file** option to identify the file that contains the list of polling policies to remove. See [Remove](#) on page 65 for more information.  
This option cannot appear on the command line when the **-export**, **-import**, **-modify**, **-modify\_all**, or **-remove** option appears on the command line.
- v**                Use this option to display the current version of the `collection_manager` utility.  
This option is in UPPERCASE.
- version**        Use this option to display the current version of the `collection_manager` utility.

## Naming Convention

The **-group** option or the *group* parameter must use the following format, *group\_category.group\_name*.

The *group\_name* portion of this parameter is the name of the group that contains the list of objects to collect. It appears in the Select Group to Poll From pull-down list in the Edit Polling Policy or Create Collection dialog in Polling Policy Manager.

The *group\_category* portion of this parameter identifies the kind of group for the corresponding *group\_name*. If the group is a **type** list, then the *group\_category* is **type**. Similarly, if the group is a **view** list, then the *group\_category* is **view**. If the group is a single node group, then the *group\_category* is **node**. Otherwise, the *group\_category* is the same as the property table name.

The following table shows the typical association for the *group\_category* value to the corresponding kind of group in the Collect Data From field in the Edit Polling Policy or Create Collection dialog in Polling Policy Manager.

**Table 10 Typical Values for the Group Category Parameter**

Group Category Value	Value in Collect Data from Field
<b>type</b>	All Nodes of the Same Type. A Combination of Type and View.
<b>view</b>	All Nodes in Same View
<b>node</b>	A Single Node
<i>property_table_name</i>	Specific Instances. Custom Groups

See the *HP Performance Insight Administration Guide* for more information about Polling Policy Manager.

# Usage Notes

This section describes the available modes, the parameters for the associated ASCII file, and how to use the `collection_manager` command.

Every group belongs to a group category. If the referenced group is not a type, view, or node, then the category is the name of the property table that contains the objects defined in the group.

## Modes of Operation

The `collection_manager` command has four modes of operation: import, modify, export, and remove.

### Import

The *import* mode provides the ability to define multiple polling policies. It requires a file that contains at least three parameters in each record to define each polling policy; they are *policy\_name*, *table\_name*, and *group*. See [ASCII File](#) on page 65 for more information about the file.

### Modify

The *modify* mode provides the ability to change specific settings in existing polling policies. There are two options available; they are **-modify** and **-modify\_all**.

- The **-modify** option enables you to modify a single polling policy in the system by entering the parameters on the command line. You must include at least one of the following options in any combination on the command line at the same time: **-descr**, **-group**, **-interval**, and **-pollfrom**. When you use this option, `collection_manager` changes only the specified settings in the polling policy.
- The **-modify\_all** option enables you to modify one or more polling policies in the system with the parameters from a file. You must include the **-file** option on the command line. You can change any combination of the following parameters in the file for each polling policy definition: *poll\_interval*, *poll\_from*, *group*, and *desc*; however, if you do not specify a modifiable parameter, `collection_manager` replaces the missing value



with the default. Note, however, that each polling policy in the file must have the following three parameters: *policy\_name*, *table\_name*, and *group*; and a value or a placeholder for every parameter. See [ASCII File](#) on page 65 for more information about the file.

You can use the **-database** option with either option to specify the server that contains the polling policies.

## Export

The *export* mode provides the ability to create a file containing the existing polling policies in the catalog.

## Remove

The *remove* mode provides the ability to remove polling policies from the catalog. There are two options available; they are **-remove** and **-remove\_all**.

- The **-remove** option enables you to remove a single polling policy from the system by using the command line.
- The **-remove\_all** option enables you to remove one or more polling policies from the system by using a file. You must include the **-file** option on the command line. In this mode, *collection\_manager* requires the *policy\_name*, *table\_name*, and *group* parameters in the associated file; however, it uses only the *policy\_name* parameter and ignores all the other parameters. Note that each polling policy in the file must have a value or a placeholder for every parameter. See [ASCII File](#) on page 65 for more information about the file.

You can use the **-database** option with either option to specify the server that contains the polling policies.

## ASCII File

Three of the four modes of operation require an ASCII file. All of these modes require at least three parameters: *policy\_name*, *table\_name*, and *group*; but each record in the file must have a value or a placeholder for every parameter in the record. The ASCII file that contains the polling policies must be in the following format:

*policy\_name, table\_name, poll\_interval, datapipe\_name, poll\_from,  
user\_name, server\_name, group, group\_server, desc*



If any of the required parameters contain invalid values for a particular option, an error message occurs and the process may terminate at that point. In some cases, the system skips the record and checks the rest of the file for additional errors. If an error occurs, verify that all records contain valid values before you resubmit the file. Verify that there are no blank lines; otherwise, an error message occurs and the process terminates at that point.

Use the tab delimiter as a placeholder for any parameter that is missing; otherwise, the system skips the record, displays an error message, and stops processing at that point.

The descriptions for the parameters in the file follow:

<i>policy_name</i>	<p>This parameter specifies the name of the polling policy. The length of the <i>policy_name</i> can be up to 30 characters.</p> <p>It appears in the Policy Name field in the polling policy.</p> <p>This is a required parameter for the <b>-import</b>, <b>-modify_all</b>, and <b>-remove_all</b> options.</p>
<i>table_name</i>	<p>This parameter specifies the SQL name of the table for the collected data. The table must already exist in the database. This parameter is the SQL name that corresponds to the Alias table name that appears in the Data to Poll For field in the polling policy. Use Table Manager to view the corresponding SQL names and the Alias names for tables. If you need to create a collectable table, see <a href="#">datapipe_manager</a> on page 85 or see the <i>HP Performance Insight TEEL Reference Guide</i>.</p> <p>This is a required parameter for the <b>-import</b>, <b>-modify_all</b>, and <b>-remove_all</b> options.</p>
<i>poll_interval</i>	<p>This parameter specifies the polling interval in minutes. The polling interval is the length of time between polling requests. Valid values are:</p> <ul style="list-style-type: none"><li>• 0: off</li><li>• 5: 5 minutes</li><li>• 10: 10 minutes</li><li>• 15: 15 minutes</li><li>• 20: 20 minutes</li><li>• 60: 1 hour</li><li>• 1440: 1 day</li></ul> <p>The default value is 0.</p> <p>This parameter must contain a valid value when it is in the file; otherwise, an error message occurs. If you do not enter a value for this parameter, <i>collection_manager</i> uses the default.</p> <p>It appears in the Poll Interval field in the polling policy.</p> <p>You can change this attribute in an existing polling policy with the <b>-modify</b> and <b>-interval</b> options on the command line.</p>

<i>datapipe_name</i>	<p>This parameter specifies the datapipe that contains the data. It must exist on the server specified in the <i>poll_from</i> parameter.</p> <p>The default value is <code>dpipe_snmp</code>.</p> <p>It appears in the Use Datapipe field in the polling policy.</p>
<i>poll_from</i>	<p>This parameter specifies the server that contains the data. The datapipe specified in the <i>datapipe_name</i> parameter must exist on this server. If you do not enter a value for this parameter, <code>collection_manager</code> uses the default.</p> <p>The default value is the name of the local host server.</p> <p>It appears in the Polling Assigned to field in the polling policy.</p> <p>You can change this attribute in an existing polling policy with the <b>-modify</b> and <b>-pollfrom</b> options on the command line.</p>
<i>user_name</i>	<p>This parameter specifies the owner of the polling policy. Currently, the value is always <b>trendadm</b>, which is the default value. The system ignores any other value at this time.</p> <p>It appears in the User field in the polling policy.</p>
<i>server_name</i>	<p>This parameter specifies the target server name, which is the database server that contains the collected data.</p> <p>The default value is the value flagged as the default in the PI connections file, unless this is overridden by the <b>-database</b> command line option.</p> <p>It appears in the Server field in the polling policy.</p>

<i>group</i>	<p>This parameter specifies the type of group, which is the <i>group_category</i>, along with the associated <i>group_name</i> for the polling policy. It must already exist in the group catalog. See <a href="#">Naming Convention</a> on page 62 for a more detailed description of this name.</p> <p>It must be in the form <i>group_category.group_name</i>.</p> <p>This is a required parameter for the <b>-import</b>, <b>-modify_all</b>, and <b>-remove_all</b> options.</p> <p>You can change this attribute in an existing polling policy with the <b>-modify</b> and <b>-group</b> options on the command line.</p>
<i>group_server</i>	<p>Currently, this parameter is the same as the <i>server_name</i> parameter.</p>
<i>desc</i>	<p>This parameter provides a description for the polling policy. If you do not enter a value for this parameter, <i>collection_manager</i> uses the default.</p> <p>The default for this parameter is NULL.</p> <p>You can change this attribute in an existing polling policy with the <b>-modify</b> and <b>-descr</b> options on the command line.</p> <p>It appears in the Description field in the polling policy.</p>

## Using the *collection\_manager* Command

This section shows some formats of the command for the various modes.

- If you enter the *collection\_manager* command without any options, the system displays an error message followed by the help information.
- To display the syntax and options for this command, enter the following:

```
collection_manager -help
```

- To display the version information for this command, enter one of the following commands:

```
collection_manager -V
```

or

```
collection_manager -version
```

- To import one or more polling policies, enter the following command:

**collection\_manager -import -file *file\_name***

*file\_name* is the name of the ASCII file that contains the list of nodes with their corresponding attributes to import.

- To modify a single polling policy, enter the following command:

**collection\_manager -modify *policy\_name* -interval *num\_minutes* -descr *description* -group *group\_category.group\_name* -pollfrom *host\_name***

*policy\_name* is the name of the polling policy to modify.

*num\_minutes* is the number of minutes between polling requests.

*description* is the modified description of the polling policy.

*group\_category* is the kind of group for the corresponding *group\_name*.

*group\_name* is the name of the group that contains the list of nodes to poll.

*host\_name* is the name of the server that contains the data.

Note that you can use any combination of these options with the **-modify** option.

- To modify one or more polling policies, enter the following command:

**collection\_manager -modify\_all -file *file\_name***

*file\_name* is the name of the ASCII file that contains the list of polling policies with their corresponding settings to modify.

- To remove a single polling policy, enter the following command:

**collection\_manager -remove *policy\_name***

*policy\_name* is the name of the polling policy to remove.

- To remove one or more polling policies, enter the following command:

**collection\_manager -remove\_all -file *file\_name***

*file\_name* is the name of the ASCII file that contains the list of polling policy names to remove.

- To export a list of nodes to a file, enter the following command:  
**collection\_manager -export -file *file\_name***  
*file\_name* is the name of the ASCII file that contains the list of existing polling policies on the system.
- To export a list of nodes to the screen, enter the following command:  
**collection\_manager -export**

## Examples

This section has examples that show each mode of `collection_manager`. They have the following characteristics.

- All examples see the sample file used in the import example.
- These examples do not contain any blank lines, and any spacing that appears between lines is for readability.
- Any information that appears about the **-file** option for one example also applies to other examples, such as the need for a qualified path.

### Import Example

To import a list of polling policies from a file named `policy_in.txt`, enter the following command:

```
collection_manager -import -file policy_in.txt
```

The following is an example of the contents for the input file `policy_in.txt`:

```
test_policy_1,test_data_tbl,20,test_dp,,trendadm,,
view.view_ex_lst,,Test polling policy 1.
test_policy_2,testtime_,0,testtime,,trendadm,,
type.type_ex_lst,,Test polling policy 2.
test_policy_3,test_data_1_tbl,15,dpipe_snmp,,trendadm,,
view.vw_lst2,,Test polling policy 3.
test_policy_4,test_data_1_tbl,1440,dpipe_snmp,,trendadm,,
type.type_ex_lst,,Test polling policy 4.
```

Note that the file does not contain any blank lines, the spacing here is for readability.

## Modify Example

To change a particular setting in an existing polling policy, use the **-modify** option with the corresponding option for the setting.

### Example 1

If you want to change the polling interval for `test_policy_3` from 15 minutes to an hour, enter the following command:

```
collection_manager -modify test_policy_3 -interval 60
```

### Example 2

If you want to change the group of nodes for `test_policy_2` from `type_ex_lst` to `tp_lst1` and change the polling interval from Off to 15 minutes, enter the following command:

```
collection_manager -modify test_policy_2 -group  
type.tp_lst1 -interval 15
```

## Modify\_All Example

Another way to make the changes described above at the same time is to use an ASCII file containing the polling policy definitions. The file should contain the following entries:

```
test_policy_2,testtime_,15,testtime,,trendadm,,type.tp_lst1,,  
Test polling policy 2.  
test_policy_3,test_data_1_tbl,60,dpipe_snmp,,trendadm,,  
view.vw_lst2,,Test polling policy 3.
```

Note that every parameter must have a valid value or placeholder in the file. Furthermore, the following three parameters must be in the file with valid values: *policy\_name*, *table\_name*, and *group*. If you use a placeholder for a parameter, `collection_manager` uses the default. In this example, the *poll\_from*, *server\_name*, and *group\_server* parameters use the defaults. The name of the file in this example is `policy_mod.txt`. To make these changes, enter the following on the command line:



```
collection_manager -modify_all -file policy_mod.txt
```

## Export Examples

### Example 1

To generate a file containing all the polling policy definitions that are in the collection catalog, you can specify an ASCII file to store the definitions. For example, if you want to export the current list of polling policies to a file called `policy_export.txt`, enter the following command:

```
collection_manager -export -file policy_export.txt
```

### Example 2

If you do not specify a file, `collection_manager` exports the polling policy definitions to standard output. The polling policy definitions appear on the screen when you enter the following command:

```
collection_manager -export
```

### Example 3

If you want to export the polling policies from a different server named **bear** to a file named `policy_out.txt`, enter the following command:

```
collection_manager -export -file policy_out.txt  
-database bear
```

## Remove\_All Examples

### Example 1

If you want to remove multiple polling policies, you can use an ASCII file to specify the list. To remove the policies in the file, `policy_mod.txt`, enter the following:

```
collection_manager -remove_all -file policy_mod.txt
```

## Example 2

Note that you need to fully qualify the path for the file if it is in a different directory. For example, if the file were in the lists directory from the D drive on a Windows machine, you would enter the following command:

```
collection_manager -remove_all -file  
d:\lists\policy_mod.txt
```

## Remove Example

If you want to remove a single polling policy such as test\_policy\_2, enter the following command:

```
collection_manager -remove test_policy_2
```

## Error Messages

This section describes some of the messages that can occur from collection\_manager. Each message has the following format:

- A brief description about why the message appears. Each new message description starts with a check box.
  - The actual message that appears with parameters for any information that may be missing.
  - A suggestion about the action to do so that the message does not appear again.
- ☐ If the following error message appears, there is a command-line syntax error. This means that a required mode option is missing.

```
A mode of operation is required.  
Exiting program with code 1.
```

Verify that the command has only one of the following options on the command line at the same time: **-import**, **-export**, **-modify**, **-modify\_all**, **-remove**, or **-remove\_all**.

- ☐ If the following message appears, the value for the **-interval** option on the command line is not a numeric character.

Collection interval must be a number.  
Exiting program with code 1.

Verify that the value for the **-interval** option on the command line is an integer number and that it does not contain a letter or other characters such as a dot.

- ❑ If the following message appears, the specified *policy\_name* that is either with the **-modify** option or in the associated file does not exist.

Collection policy\_name does not exist.

Verify that the spelling of the policy name is correct or that the polling policy exists; you can use Polling Policy Manager to view the existing list of polling policies. The *policy\_name* may be in either of the following locations: with the **-modify** option on the command line or in the first position of each record in the associated file.

- ❑ If the following error message appears, the database specified with the **-database** option on the command line does not exist.

Connection URL not found.  
Exiting program with code 3.

Verify the spelling of the database name. If the spelling is correct, you can add the database using the Web Access Server.

- ❑ If the following message appears, the host name for the **-pollfrom** option on the command line does not appear in the list of Datapipe Installations for the specified datapipe in the policy.

Could not find installed datapipe for host\_name.

Verify the following items:

- The spelling of the host name is correct.
- The host name exists.
- The host name is in the list of Datapipe Installations for the specified datapipe.

You can use Polling Policy Manager to check the list of Datapipe Installations. From the **Edit** menu, select **Datapipe Installations**. The **Edit Datapipe Installations** window appears. Select the specified datapipe from the **Select Datapipe** field. Verify the host name appears in the **Hostname** column of the **Current Datapipe Installations** field. At this point you can add or modify the **Datapipe Installation**, if necessary.

- ❑ If the following message appears, the value for the *poll\_interval* parameter in the associated file is not a numeric character.

Could not convert string to integer on line *n* - interval invalid.

Exiting program with code 4.

Verify that the value for the *poll\_interval* parameter in the associated file is an integer number and that it does not contain a letter or other characters such as a dot.

- ❑ If the following message appears, the datapipe name for the *datapipe\_name* parameter on the specified line in the associated file does not exist for the *table\_name* parameter on the same line.

Datapipe *datapipe\_name* on line *n* does not exist for table *table\_name*.

Verify that the spelling of the datapipe name or the table's SQL name is correct, or that the datapipe exists. You can use Polling Policy Manager to check which datapipe exists for a selected table. From the Edit Polling Policy window, the alias name of the table appears in the Data to Poll For field and the datapipe name appears in the Use Datapipe field. Note that you must use the SQL name for the table in the ASCII file and not the alias name; you can use Table Manager to view the list of tables for the corresponding names.

- ❑ If the following message appears, the value for the **-debug** option on the command line is not a numeric character.

Debug level must be a number.

Exiting program with code 1.

Verify that the value for the **-debug** option on the command line is an integer number and that it does not contain a letter or other characters such as a dot.

- ❑ If the following message appears, the value for the **-debug** option on the command line is not valid.

Debug level must be between 0 and 3.

Exiting program with code 1.

Verify that the value for the **-debug** option on the command line is an integer number and is one of the following numbers: 0, 1, 2, or 3.

- ❑ If the following error message appears, the specified record in the file has a syntax error.

Exception: Wrong number of delimiters on line n.  
Exiting program with code 4.

Check the specified record in the file and fix the error. Verify that the record has either a value or a placeholder for all parameters or that the line is not blank.

- ❑ If the following error message appears, the file name specified on the command line in the **-file** *file\_name* option does not exist.

File *file\_name* does not exist.  
Exiting program with code 2.

Verify the spelling of the file name or that the file exists in the specified location. You may have to supply a fully qualified path with the file name.

- ❑ If the following message appears, the value for the **-group** option on the command line does not exist.

Group *group\_name* does not exist.

Verify the following items:

- The spelling of the *group\_category* or the *group\_name* portion of the **-group** option value is correct.
- The *group\_name* portion of the **-group** option value exists for the type of group in the *group\_category* portion.
- The *group\_category* portion of the **-group** option value has the correct name.

You can use Polling Policy Manager to view the existing list of polling groups. From the **Edit** menu, select **Polling Groups**. The Edit Polling Groups window appears. The *group\_category* portion corresponds to the Select Kind of Group field. The *group\_name* portion corresponds to the Select Group field. You can also view the existing list of polling groups from the Edit Polling Policy window. See [Naming Convention](#) on page 62 for more information.

- ❑ If the following message appears, the *group* parameter on the specified line in the associated file does not exist.

Group *group\_name* on line n does not exist.

Verify the following items:

- The spelling of the *group\_category* or the *group\_name* portion of the *group* parameter is correct.

- The *group\_name* portion of the *group* parameter exists for the type of group in the *group\_category* portion.
- The *group\_category* portion of the *group* parameter has the correct name.

You can use Polling Policy Manager to view the existing list of polling groups. From the **Edit** menu, select **Polling Groups**. The Edit Polling Groups window appears. The *group\_category* portion corresponds to the Select Kind of Group field. The *group\_name* portion corresponds to the Select Group field. You can also view the existing list of polling groups from the Edit Polling Policy window. See [Naming Convention](#) on page 62 for more information.

- ❑ If the following message appears, the value for the **-interval** option on the command line is not a valid value for a polling interval.

```
Invalid collection interval.
Exiting program with code 1.
```

Verify that the value for the **-interval** option on the command line is a valid value for a polling interval. See the description of the *poll\_interval* parameter on [page 67](#) for the valid values.

- ❑ If the following message appears, the value for the *poll\_interval* parameter on the specified line in the associated file is not a valid value for a polling interval.

```
Invalid poll interval on line n.
Exiting program with code 4.
```

Verify that the value for the *poll\_interval* parameter on the specified line in the associated file is a valid value for a polling interval. See the description of the *poll\_interval* parameter on [page 67](#) for the valid values.

- ❑ If the following error message appears, the **-file** option is missing on the command line for an option that requires a file.

```
No file specified.
Exiting program with code 1.
```

Verify that the **-file** option is on the command line with one of the following options at the same time: **-import**, **-modify\_all**, **-remove\_all**.

- ❑ If the following error message appears, there is a command-line syntax error. This means that there are multiple mode options on the same command line.

Only one mode of operation is allowed.  
Exiting program with code 1.

Verify that the command has only one of the following options on the command line at the same time: **-import**, **-export**, **-modify**, **-modify\_all**, **-remove**, or **-remove\_all**.

- ❑ If the following message appears, the host name for the *poll\_from* parameter on the specified line in the associated file does not appear in the list of Datapipe Installations for the *datapipe\_name* parameter on the same line.

Poll from host\_name on line n not installed on datapipe  
datapipe\_name.

Verify the following items:

- The spelling of the host name is correct in the file.
- The host name exists.
- The host name is in the list of Datapipe Installations for the specified datapipe.

You can use Polling Policy Manager to check the list of Datapipe Installations. From the **Edit** menu, select **Datapipe Installations**. The Edit Datapipe Installations window appears. Select the specified datapipe from the Select Datapipe field. Verify the host name appears in the Hostname column of the Current Datapipe Installations field. At this point you can add or modify the Datapipe Installation, if necessary.

- ❑ If the following message appears, the table name for the *table\_name* parameter on the specified line in the associated file does not exist.

Table table\_name on line n does not exist.

Verify that the spelling of the table's SQL name is correct or that the table exists. You can use Table Manager to view the existing list of tables. Make sure that you use the table name from the SQL Name column in your ASCII file.

- ❑ If the following error message appears, the command is missing at least one setting to modify in the specified polling policy.

The **-modify** option requires at least one of the following:  
**-descr**, **-group**, **-pollfrom**, or **-interval**.  
Exiting program with code 1.

Verify that at least one of the following options is on the command line with the **-modify** option: **-descr**, **-group**, **-interval**, and **-pollfrom**.



## 5 cospolicy\_manager

You can use the HP Performance Insight (PI) `cospolicy_manager` command to create copy policies on the Central Server. The copy policies specify the tables that you want to copy, the type of copy and if the copy policy is enabled. This command imports the copy policies from a comma separated text file.

### Requirements and Restrictions

You can use the `cospolicy_manager` command to create copy policies only on the Central Server. You cannot use this command to create copy policies on Satellite or Standalone Server.

### Syntax

The `cospolicy_manager` command has the following syntax:

```
cospolicy_manage -import  
r  
  
    -file fileName  
    [- d debugLevel  
    [-v version  
    [-h help
```

Where, *fileName* is the name of the comma separated text file containing the information to create the copy policies. The format of this text file is as follows:

```
name_of_copypolicy,description_of_copypolicy,source_tablename  
,destination_tablename,enable_flag
```

There should be one entry for each table that is to be trendcopied from the Satellite Server to Central Server.

## Options

The `copypolicy_manager` command has the following options:

- import** This option is required. Use this option to import the copy policies available in the text file.
- file** This option is required. Use this option to specify the text file containing the copy policy descriptions. There should be one entry for each table that is to be trendcopied from the Satellite Server to Central Server.
- d** Use this option to specify the debug level. Valid values are 0, 1, and 2.  
The default is 0, which means no debug output. Debug output is written to the standard out. Use this option only with HP Technical Support for testing purposes because of the additional overhead it incurs.
- v** Use this option to specify the version of PI.
- h** Use this option to display the command line options (help).

## Example

The following command is used to import copy policies for InterfaceReporting Report pack:

```
copypolicy_manager -import -file copypolicy_import.txt
```

where the comma separated `copypolicy_import.txt` file has the following contents:

CopyPolicy\_InterfaceResource, CopyPolicy\_InterfaceResource, property and data, SHIRDevPorts, SHIRDevPorts



---

## 6 datapipe\_manager

You can use the `datapipe_manager` command to create data and property tables and views in a PI database. You can also use the `datapipe_manager` to create archive tables and data retention profiles.

See the *HP Performance Insight TEEL Reference Guide* for more information.

### Requirements and Restrictions

- You can create a data or property table using a TEEL file or a template.
- You can create a view using a TEEL file only.
- The **alter** mode of the **-p** option is only available from the command line.

# Syntax

A parameter less `datapipe_manager` command displays the command syntax.

```
datapipe_manager [-a name]  
                  [-A]  
                  [-b]  
  
                  [-c { as-is  
                        on  
                        off } ]  
  
                  [-C [n],sqlname,type,size,alias[,default=value  
                  ,null={ yes  
                          no } ,foreign_key(table.dsi_key_id)  
                  ,desc=desc]]  
                  [-d debug_level]  
                  [-e]  
                  [-h]  
                  [-L alias_name]  
                  [-n hostname]  
  
                  [-p { alter  
                        create  
                        delete  
                        remove  
                        verifyparms } ]  
  
                  [-P]  
  
                  [-r { copy  
                        r2d } ]  
  
                  [-s source_db_name]  
                  [-S target_db_name]  
                  [-t source_table_name]  
                  [-T target_table_name]  
                  [-v]  
                  [-x]
```

# Option Categories

The following table shows the `datapipe_manager` options that apply for the various modes of the `-p` option.

**Table 11    `datapipe_manager` Option Categories**

alter	create		delete	remove	verifyparms
	TEEL File	Template			
-a	-a	-t	-S	-a	-a
-T	-c	-T	-b	-b	
-C	-n	-c	-T	-n	
-d	-S	-s	-x	-x	
	-r	-S	-d	-d	
	-P	-r			
	-e	-L			
	-d	-n			
		-P			
		-e			
		-d			

# Options

The `datapipe_manager` utility has the following options.

- a** Specifies the TEEL file *name* in **create** mode, or the datapipe *name* in **remove** mode. The TEEL file must have the `.teel` extension. PI looks for the file in the current directory; otherwise, the file name must be fully qualified.

You can use this option for **create**, **remove**, or **verifyparms** modes.
- A** When you use this option, `datapipe_manager` creates an archive table for a given data table from the command line. For more information about using this option, see [Example 5](#) on page 96.
- b** Indicates that `datapipe_manager` does not automatically remove a property table when the last data table that references it is removed. This option overrides the default, which is to automatically remove the property table.
- c** Sets the compatibility mode. The values are **as-is**, **on**, and **off**.

The value **as-is** specifies that the table is the same as the input. For template-based creation, the system uses an exact copy of the input table's columns. For TEEL-based creation, the system creates only those columns specified in the TEEL file.

The value **on** specifies that compatibility mode is on, which creates the table with the legacy header/footer columns. Note that if there are column names with well-known names for the header/footer columns in the TEEL file, the system ignores those columns in this mode.

The value 'off' specifies that compatibility mode is off, which creates the table with a minimal set of management columns.

The default is 'on' for TEEL-based creation.

The default is 'as-is' for template-based creation.

This option is only valid in **create** mode.



- C** Describes column information. This option is only valid in **alter** mode.
- *n*: This parameter can be a non-negative integer that specifies the column that contains the data in the data table, an OID, or a macro prefixed with a # character. For a raw-SNMP table, a valid OID string is required. A raw-SNMP table cannot have two columns specified with the same OID.
  - *sqlname*: Name of the column used to generate the dictionary entries within the PI database. The size of the name depends on the naming convention specified in the database engine, and is a maximum of 30 characters. This is a positional parameter.
  - *type*: Specifies the type of data in the column. This is a positional parameter. For more information, see the *HP Performance Insight TEEL Reference Guide*.
  - *size*: Corresponding length for the type of data specified in the *type* parameter. Only the **char\_string**, **snmp\_char\_string**, **hex\_string**, **snmp\_hex\_string**, and **numeric** types require a size. The size for the **numeric** type is a precision value. This is a positional parameter.

- *alias*: Additional name for the column that may define the contents more clearly. The size for the name may be a maximum of 255 characters. When specifying an *alias*, all missing parameters in this statement require placeholders. This is a positional parameter. For more information, see the *HP Performance Insight TEEL Reference Guide*.
  - **default**: Specifies the default *value* for the data in the column.
  - **null**: Specifies that the column can be NULL. The values are **yes** and **no**. The default is yes, which allows NULL. If the **default** option is specified, then **null=no**.
  - **foreign\_key**: Identifies a unique attribute from a pre-existing property table to include as an element in the datapipe. Specify the name of the parent *table* that contains the data with **dsi\_key\_id** as the column name.
  - **desc**: Use this parameter to add a description for the column. When you add the description, enclose it in double (“) quotation marks, and omit any spaces around the equal sign. If you omit this parameter, the value is NULL.

This option is in UPPERCASE.

- d** Set a debug output level. Values of **1**, **2**, or **3** are valid. The higher the number, the more detailed the information. The default is no debug output. Debug output writes to standard out. Only use this option for testing in coordination with Technical Support due to the additional overhead it places on `datapipe_manager`.
- e** Indicates that `datapipe_manager` creates the table without datapipe registry entries. This option overrides the default behavior, which is to create the table with datapipe registry entries.
- h** Displays help information. This option overrides all other options on the command line.
- L** Specifies the alias name for the new table when using template-based creation.

This option is in UPPERCASE.

- n** For **create** mode, specifies the name of the PI collector for datapipe registration. By default, this is the local host.
- For **remove** mode, specifies the name of the installed datapipe. If the *hostname* parameter is missing, the local hostname is the name of the installed datapipe.
- This option is only valid in **create** or **remove** mode.
- p** Specifies the execution mode. The values are **alter**, **create**, **delete**, **remove**, and **verifyparms**. Note that **register** is a legacy parameter that is equivalent to **create**.
- **alter**: Modifies a registered table that already exists in the PI dictionary by adding a column to it. Both data and property tables can have columns appended.
- Use the **-C** option to specify the column information. The *datapipe\_manager* utility verifies that the column does not already exist, and then invokes the appropriate command to update the tables in the PI dictionary. Use the Appendix 5 option to enter the table name from the command line. This mode is only available from the command line.
- If you attempt to add non-nullable columns, *datapipe\_manager* adds those columns as nullable columns instead; it does not add by-variable columns.
- Note that after you alter a collectible table, you need to run the *mw\_collect* command with the **-b** option once to make the change across dependent objects. The modification to the table does not have any effect on the collection until this command is run.

- **create:** Creates any type of table in PI as long as the PI database knows the type. There are two methods to create a data or property table, with a TEEL file or an input table name as a template. The input table can exist on either a local or remote database. There is only one method to create a view, and that is with a TEEL file.

The TEEL file method validates the TEEL file by checking for any syntax errors as in `verifyparms` mode, generates any settings not specified in the file, and adds the table to the PI dictionary. This method requires the use of the **-a** option.

The input template table method uses the command line to specify the source template table name with the **-t** option and the target table name with the **-T** option. It then validates that the target table does not already exist, generates the new target table definition from the source template table, and adds the table to the PI dictionary.

Note that if the table already exists, `datapipe_manager` goes into **alter** mode. For example, if the existing table has eight columns and you specify a ten-column table, `datapipe_manager` adds two columns to the existing table.

- **delete:** Deletes both data and property tables. `datapipe_manager` attempts to clean up the dictionary entries associated with the specified table before dropping it from the database. A foreign key relationship may prevent certain tables from being dropped or removed from the system.

Enter this mode from the command line with the **-T** option to specify the table name. If the table named with the **-T** option is a property table, `datapipe_manager` verifies that the property table does not have any data tables mapped to it before dropping it.

- **remove:** Suspends the distribution of a datapipe on a collection station, but does not actually remove the table from the database. Use either the **delete** mode from this option or the Drop Table(s) command from Table Manager to remove the table from the database. This mode requires the use of the **-a** option.
- **verifyparms:** Checks the syntax for all the statements in the TEEL file. This mode requires the use of the **-a** option. This is the default mode when the **-p** parameter is missing.

- P** When you use this option, `datapipe_manager` creates the property table in **as-is** mode. This means that `datapipe_manager` creates the table using only the property columns in the TEEL file or from the existing property table in template mode, and it does not generate any columns automatically such as `dsi_target_name`, `dsi_table_key`, and `dsi_descr`. It verifies that the following columns exist in the input set with their corresponding attributes and at least one object or collection by-variable:

Column Name	Data Type	NULL Attribute	Default
<code>dsi_key_id</code>	numeric(10)	not NULL	
<code>dsi_status</code>	tinyint	not NULL	<b>2</b>
<code>dsi_status_time</code>	unix_time	not NULL	current time
<code>dsi_bv_state</code>			

This option is in UPPERCASE.

- r** Specifies which database objects to generate for a raw table. The values are **copy** and **r2d**.
- **copy**: Generates the required database objects for processing, which are the rate table, two upload tables, and the **copy** stored procedure. The **copy** procedure copies data from the upload table to the rate table and filters the data so that there is no duplicate data.
  - **r2d**: Generates the required database objects for raw-to-delta processing, which are the rate table, two upload tables, a last keys table, and the **raw-to-delta** stored procedure.
- s** Specifies the source database name. This database can exist on either a local or remote system. The default for the database name is the name of the database specified as default in the `systems.xml` file. This option is only valid in **create** mode.
- S** Specifies the target database name. This database can exist on either a local or remote system. By default, the database is on the local system, and the database name is the name of the database specified as default in the `systems.xml` file.
- This option is in UPPERCASE.

- t** Specifies the source SQL or alias table name. This table can exist on either a local or remote database.
- T** Specifies the target SQL or alias table name. By default, the target table name is the same as the source table name.  
This option is in UPPERCASE.
- v** Displays version information for `datapipe_manager`. This option is not valid with any other options.  
Note that this option is in UPPERCASE.
- x** Indicates that the system does not remove the table if a dependent collection policy exists. This option overrides the default behavior, which is that the system automatically removes any dependent collection policies when it deletes a data table.

## Usage Notes

The `datapipeline_manager` program is a utility that creates tables based on input from a TEEL file or a template. A *template* is an existing table. It also creates views based on input from a TEEL file only. See the *HP Performance Insight TEEL Reference Guide* for more information about TEEL file statements.

The **alter** mode of the **-p** option is only available from the command line. You can add one column at a time to a data or property table. The **alter** mode does not modify existing columns.

If you need to add multiple columns to a data or property table, you can use a TEEL file that specifies multiple column definitions for the table, and then use the **create** mode of the **-p** option to add the missing columns to the table. `datapipeline_manager` compares the specified table to the definition and adds the missing columns to the table.

## Examples

### Example 1

To have `datapipeline_manager` create and register a datapipeline from a TEEL file named `test.teel` in compatibility mode, use the following command:

```
datapipeline_manager -a test.teel -p create
```

### Example 2

To have `datapipeline_manager` create and register a datapipeline from a TEEL file named `test.teel` in non-compatibility mode, use the following command:

```
datapipeline_manager -a test.teel -p create -c off
```

### Example 3

To have `datapipeline_manager` use as-is mode to create a new raw table named **newtable** with the same definition as an existing table named **oldtable** and then set it up for raw-to-delta processing, use the following command:

```
datapipeline_manager -p create -t oldtable -T newtable -r r2d
```

#### Example 4

To have `datapipe_manager` modify an existing property table named **oldtable** by adding a new text column that is 10 characters long and named **newcol**, use the following command:

```
datapipe_manager -p alter -T oldtable -C  
,newcol,char_string,10
```

#### Example 5

To have `datapipe_manager` create an archive table for a given data table from the command line, use the following command:

```
datapipe_manager -p create -t source_tablename -A -s  
source_databasename
```

**-s** is optional. If the source database name is not specified, the default database is used.

The **-p** option (create mode) creates an archive table when the ArchiveTableName directive is specified in the TEEL file for archiving purpose.

The **-p** option (delete mode) deletes an existing archive table if the associated data table is deleted.



# Error Messages

This section describes some of the messages that can occur from `datapipe_manager`. Each message has the following format:

- A brief description about why the message appears. Each new message description starts with a check box.
- The actual message that appears with parameters for any information that may be missing.
- A suggestion about the action to do so that the message does not appear again.

## Command Line Option Errors

- ❑ If the following error message appears, the specified option requires double quotation marks to enclose its corresponding values.

Arguments for option *option* must be enclosed in double quotes.

Verify that there is a double quote after the option and before the first value and another double quote after the last value for the specified option.

- ❑ If the following error message appears, the specified option has an incorrect value.

Argument *value* is invalid for option *option*.

Verify that the value is valid and is in the appropriate format for the specified option.

- ❑ If the following error message appears, the specified option has an incorrect value.

Argument *value* is not defined for the option *option*.

Verify that the value is valid and is in the appropriate format for the specified option.

- ❑ If the following error message appears, the specified value is no longer valid for the specified option. The value may have been valid in a previous version of the software.

Argument *value* is obsolete for option *option*.

Verify the valid values available for the specified option in the current release of the software.

- ❑ If the following error message appears, the specified file does not exist.

File *file\_name* referenced by option *option* does not exist.

Verify that the specified file name has the correct path on the command line. Remember to check the spelling of each member of the path and the file name.

- ❑ If the following error message appears, the specified command requires the specified option.

Missing required option *option* for command *command\_name*.

Verify that the *xxx* option is on the command line.

- ❑ If the following error message appears, the specified option does not require a value for the specified command.

Option *option* does not take an argument for command *command\_name*.

Verify that the specified option does not have a value after it on the command line.

- ❑ If the following error message appears, the specified option is not valid for the specified command.

Option *option* is not defined for command *command\_name*.

Verify that the specified option is not on the command line for the specified command.

- ❑ If the following error message appears, the specified option is no longer available for the *xxx* command.

Option *option* is obsolete.

Verify that the specified option is not on the command line.

- ❑ If the following error message appears, the specified option is missing the corresponding value for the specified command.

Option *option* requires an argument for command *command\_name*.

Verify that the specified option has its corresponding value and is in the appropriate format on the command line.

- ❑ If the following error message appears, the specified option requires a valid value.

Option *option* requires valid argument.

Verify that the specified option has its corresponding value and is in the appropriate format on the command line.

- ❑ If the following error message appears, the specified options are mutually exclusive.

The option *option1* cannot be used with option *option2*.

Verify that only one of the specified options is on the command line.

- ❑ If the following error message appears, the system is unable to create the specified file from the specified option.

Unable to create file *file\_name* referenced by option *option*.

Verify that the specified file has write permission and the correct path for the file on the command line. Remember to check the spelling of each member of the path and the file name.

## Generic File I/O Errors

- ❑ If the following error message appears, the file does not have the appropriate permissions to access the file.

File does not have read access *file\_name*.

Verify the following:

- The spelling of the file name is correct.
- The file exists in the specified location. You may have to supply a fully qualified path with the file name.
- The file has the proper permissions to access it.

- ❑ If the following error message appears, the file name specified on the command line does not exist.

File *file\_name* does not exist.

Verify that the specified file name has the correct path and it is in the appropriate location. Remember to check the spelling of each member of the path and the file name.

- ❑ If the following error message appears, the system cannot read the specified file.

I/O error in reading file *file\_name*.

Verify that the specified file has read access and that the user has permission to read it.

- ❑ If the following error message appears, the system cannot write to the specified file.

I/O error in writing to file *file\_name*.

Verify that the specified file has write access and that the user has permission to write to it.

- ❑ If the following error message appears, the specified file has invalid statements in it.

Incorrect syntax in file *file\_name*.

Verify that the specified file has correct format and that the statements have the correct format.

## Generic PI Errors

- ❑ If the following error message appears, the specified database is not available.

Failed to connect with: *database\_name*.

Verify the following:

- The spelling of the database name is correct, as specified in the *systems.xml* file with the `<Name>` tag.
- The database exists, and is running.
- The database allows connections.

- ❑ If the following error message appears, the specified environment variable, which is required, is not set.

Required Environment variable not set: *env\_variable\_name*.

Verify that the specified environment variable is set correctly.

Some of the environment variables that PI uses are the following:

COLLECT_HOME	The directory where the PI collector modules store their working files, which includes the cache files. The default is DPIPE_HOME/collect.
DPIPE_HOME	The directory where PI resides
DPIPE_TMP	The directory that contains the temporary files for PI. The default is DPIPE_HOME/tmp.
TREND_LOG	The directory that contains the log files for PI. The default is DPIPE_HOME/log.

- ❑ If the following error message appears, the system could not process the specified XML file.

Failed parsing XML. Input=**xml\_file\_name**.

Verify that the name of the XML file is correct and that statements have the correct format.

- ❑ If the following error message appears, the same command string is currently running.

The previous instance is already running. The process will not be executed: *command\_string*.

Verify that the specified command string is correct. If it is, verify that the same process is not running and then rerun it.

## Generic Database Errors

- ❑ If the following error message appears, the specified table is not in the database.

SQL table `"table_name"` does not exist.

Verify the spelling of the SQL name for the table is correct or that the table exists in the database. You can use Table Manager to view the existing list of tables.

- ❑ If the following error message appears, the specified table is not in the database.

Alias table `"table_name"` does not exist.

Verify the spelling of the Alias name for the table is correct or that the table exists in the database. You can use Table Manager to view the existing list of tables.

## Generic Database Connection Errors

- ❑ If the following error message appears, the system could not connect to the specified database. The message should provide additional information.

The connection to the database `"database_name"` could not be established: *reason*.

Verify that the database name is correct and that it exists. If you need to add another database, use the Web Access Server.

- ❑ If the following error message appears, the system could not connect to the default database specified in the `systems.xml` file.

The connection to the default database could not be established.

Verify that the default database entry in the `systems.xml` file is correct.

## 7 db\_delete\_data

The `db_delete_data` command archives data from a data table if archive is set to ON for the data table. The `db_delete_data` command moves data from a data table to its archive table after the specified retention time for that table is over. The command deletes the data from an archive table after the specified archive retention time for that archive table is over.

If archive is set to OFF for the data table, the data is not archived. Data is deleted from the data table after the specified retention time for that table is over.

### Requirements and Restrictions

- If you omit the `-f`, `-s`, and `-t` options, all data tables are aged, which is the default.
- The `-f`, `-s`, and `-t` options are mutually exclusive.
- If you use the `-u` option, you must include the `-U` option on the command line at the same time.

# Syntax

The `db_delete_data` command uses the following syntax:

```
db_delete_data    [-c 1]
                   [-d debug_level]
                   [-f table_category]
                   [-h]
                   [-i aging_days]
                   [-m ta_period]
                   [-q day_of_week]
                   [-s sqlname]
                   [-S]
                   [-t alias_name]
                   [-u update_statistics]
                   [-U day_of_week]
                   [-v]
```



# Options

The `db_delete_data` command has the following options:

- c**            Use this option to set the number of deletions allowed to run concurrently.

This option can have value between 1 to 10. `db_delete_data` can spawn a maximum of 10 threads.

The default is 1; that is, `db_delete_data` processes each table one at a time. At this time, if the value for this option is greater than 1, `db_delete_data` still processes the tables one at a time.
- d**            Use this option to set the debug output level. Values of 0, 1, 2, or 3 are valid. The higher the number, the more detailed the information. Debug output is written to standard output.

The default is 0, which means no debug output.
- f**            Use this option to delete data only in tables whose source is the value of this option. Valid values for *table\_category* appear in the Category list under Data Tables in the Table Manager display.

This option is mutually exclusive with the **-s** and **-t** options.

If you omit the **-f**, **-s**, and **-t** options, tables from all sources are aged, which is the default.
- h**            Use this option to display the command format help.
- i**            Use this option to override the default aging value in the database each time it appears on the command line for `db_delete_data`. The *aging\_days* parameter for this option is the number of days that PI retains the data in a table. The `db_delete_data` program deletes data that has been in the table for one day more than the number specified.

The default is to use the aging value set for the table in the database. Use Table Manager to view the default aging value for the table. See the *HP Performance Insight Administration Guide* for details.

- m** Use this option to perform key ID-based deletions depending on the value of this option. The only value currently available for this option is **ta\_period**, which deletes data by using the time period only.  
The default is **ta\_period**.
- q** Use this option to specify the day of the week to rebuild the index.  
Valid values are:
- **Su**: Sunday
  - **Mo**: Monday
  - **Tu**: Tuesday
  - **We**: Wednesday
  - **Th**: Thursday
  - **Fr**: Friday
  - **Sa**: Saturday
  - **All**: Every day
- This option invokes the `indexmaint` utility before running `db_delete_data`.
- s** Use this option to age data only in the table specified. Enter the name of the table as shown in the SQL Name column of the Table Manager display. See the *HP Performance Insight Administration Guide* for details.  
This option is mutually exclusive with the **-t** and **-f** options.  
If you omit the **-f**, **-s**, and **-t** options, all data tables are aged, which is the default.
- S** Use this option to specify the database server name. The database can exist on either a local or remote system. By default, the database is on the local system, and the database name is the default name specified in the `systems.xml` file. This option is in UPPERCASE.

- t** Use this option to age data only in the table specified. Enter the name of the table as shown in the Alias Name column of the Table Manager display. See the *HP Performance Insight Administration Guide* for details.
- This option is mutually exclusive with the **-f** and **-s** options.
- If you omit the **-f**, **-s**, and **-t** options, all data tables are aged, which is the default.
- u** Invokes the `indexmaint` utility to update the index statistics page.
- Valid values are:
- **1**: Before running `db_delete_data`.
  - **2**: After running `db_delete_data`.
  - **3**: Before and after running `db_delete_data`.
- On Oracle databases, the default is 2.
- On Sybase databases, the default is 3.
- If you use this option, you must also use the **-U** option to specify which day to run the update.
- U** Use this option to specify the day of the week to update the database statistics page. Valid values are:
- **Su**: Sunday
  - **Mo**: Monday
  - **Tu**: Tuesday
  - **We**: Wednesday
  - **Th**: Thursday
  - **Fr**: Friday
  - **Sa**: Saturday
  - **All**: Every day
- If you specify this option, you must also specify the **-u** option.
- The default is All.
- v** Use this option to display the version stamp for `db_delete_data`.

# Usage Notes

The `db_delete_data` utility deletes data from a table when that data has been stored longer than the retention period specified for the table in the database. You can find the data retention period for various tables in the Table Manager display. See the *HP Performance Insight Administration Guide* for details.

It executes automatically according to the schedule specified in the `trendtimer.sched` file. You can also run `db_delete_data` at any time from the command line.

## Examples

The following examples illustrate some uses of the `db_delete_data` command.

### Example 1

The following command deletes data from each PI database table one at a time according to the aging criteria specified for each table in the database.

```
db_delete_data
```

### Example 2

The following command runs update statistics on Sunday after `db_delete_data` completes.

```
db_delete_data -u 2 -U Su
```

### Example 3

The following command ages the data in the `mib-II-ifEntry` table, deleting data according to the aging value set for this table in the database.

```
db_delete_data -t mib-II-ifEntry
```

#### Example 4

The following command invokes the **indexmaint** utility every Wednesday to rebuild the indexes for the tables, and then deletes the data from each PI database table.

```
db_delete_data -q We
```

## Error Messages

This section describes some of the messages that can occur from `db_delete_data`. Each message has the following format:

- A brief description about why the message appears. Each new message description starts with a check box.
- The actual message that appears with parameters for any information that may be missing.
- A suggestion about the action to do so that the message does not appear again.

- ❑ If the following error message appears, there was an error with the corresponding **indexmaint** command.

```
indexmaint utility program returned non-zero exit status.
```

See the error messages associated with the corresponding **indexmaint** command.

- ❑ If the following error message appears, the specified program generated the specified error.

```
Executing program program_name generated error: msg.
```

Verify that the specified program has the appropriate information.

- ❑ If the following error message appears, the **-u** and **-U** options are not on the command line at the same time.

Option **-u** must be used with option **-U**.

Verify that the **-u** and **-U** options are on the command line at the same time.

- ❑ If the following error message appears, the specified SQL command generated an error in the middleware.

Executing the SQL command "*command*" generated middleware error "*msg*".

See the following messages; otherwise, call Technical Support.

- ❑ If the following error message appears, there was a database I/O exception.

`db_delete_data: "msg".`

See the message in quotation marks to determine the action to take.

- ❑ If the following error message appears, the system generated a database error.

Received non-deadlock database error.

See the preceding messages to determine the actions to take.

- ❑ If the following error message appears, there was an error from the middleware while closing the database connection.

Failure on closing database connection generated the middleware error "*msg*".

See the error messages associated with the corresponding middleware message.

- ❑ If the following error message appears, the specified table is missing the `ta_period` column.

Column `ta_period` not found in table "*table\_name*".

Verify that the table name is correct and that it has a `ta_period` column.

- ❑ If the following error message appears, the name of the stored procedure is longer than 30 characters.

Stored procedure "*proc\_name*" is longer than 30 bytes.

Verify that the name of the stored procedure is correct and that the length of the name is 30 characters or less.

- ❑ If the following error message appears, the stored procedure failed.

One or more threads returned with failure.

See the preceding messages to determine the actions to take.

- ❑ If the following error message appears, the value for the specified option is incorrect.

`"value" is a bad argument for the option "option".`

Verify that the specified option has the correct value. See [Syntax](#) on page 104 for the format of the option and [Options](#) on page 105 for the details about the option.

- ❑ If the following error message appears, the value for the specified option is no longer supported.

`The argument "value" of the option "option" is no longer supported.`

Verify that the specified option has the current value. See [Syntax](#) on page 104 for the format of the option and [Options](#) on page 105 for the details about the option.

- ❑ If the following error message appears, the maximum number of threads exceeded the specified number.

`Maximum number of threads can not exceed number.`

Verify that the value for the `-c` option does not exceed the specified number.





## 8 deploytool

The `deploytool` command is a stand-alone utility that enables you to deploy reports or a folder of reports to the Web Access Server so that you can view them from the Web Access Server on a PI system. You can also use this command to undeploy reports or a folder of reports from the Web Access Server, which removes them from view on the Web Access Server.

If you want to deploy or undeploy reports to the Web Access Server using the GUI tools, you can use the Deployment Wizard or the Web Access Server. See the *HP Performance Insight Guide to Building and Viewing Reports* for more information.

### Requirements and Restrictions

- If you deploy or undeploy a report when your Web browser is open, you must click the **Refresh** button on the browser to see the result of the action.
- Either the `-d` option or the `-r` option is a required option on the command line; however, both of them cannot be on the command line at the same time.

# Syntax

The `deploytool` command uses the following syntax:

```
deploytool  -c { -deploy  
                  -undeploy }  
              [-d dir_path  
              [-b]  
              [-g group_name  
              -h hostname  
              [-i rpt_desc  
              [-l deploy_loc  
              [-n rpt_display_name  
              -p port_num  
              -P password  
              [-r rpt_name_path  
              -s  
              -t deploy_type  
              -U username
```

# Options

The `deploytool` command has the following options:

- c** This option specifies the task to perform. Valid values are:
- **deploy**
  - **undeploy**
- This is a required option.
- d** This option specifies the directory that contains the reports to deploy or undeploy. When you use this option, the task applies to the contents of the directory. This means that when you deploy a directory, the system deploys the entire contents of the directory; or, when you undeploy a directory, the system undeploys the entire contents of the directory.
- When you use this option, the **-r** option should not be on the same command line.
- [-b]** This option requires PI Web Access Server document root. By default this value is *"/*. This option is required when document root is changed to other value
- g** This option specifies the name of the group of users to which you want to deploy the report.
- For more information about how to create groups, see the *userimport* and *groupimport* utilities or see the *HP Performance Insight Administration Guide*.
- h** This option specifies the Web Access Server host name where you want to deploy or undeploy the report.
- This is a required option.
- i** This option specifies the description for the report. Use double quotation marks to enclose the text for the description.
- The description is optional; it appears in the Description field on the Reports page in the Catalog folder.
- l** This option specifies the deployment location that is relative to the system or users folder.
- If you want to deploy reports to the top-level directory, use a forward slash (*/*) as the value for the deployment location.
- n** This option specifies the name for the report. Use double quotation marks to enclose the name if it contains spaces.
- When you use the Web Access Server, this name appears as the report name in the list of deployed reports in the Catalog folder.

- p** This option specifies the port number for the Web Access Server specified with the **-h** option.  
You must enter this option even though the default for this option is the port number supplied during the PI installation, which is port number **80**, in most cases.  
This is a required option.
- P** This option specifies the corresponding password for the username that accesses the Web Access Server.  
This option is in UPPERCASE.  
This is a required option.
- s** This option specifies that SSL is used as the communications protocol.

- r** This option specifies the name of a single report you want to deploy or undeploy.  
In deploy mode, this value can contain the absolute or relative path that is the current location of the report. If this value does not include a path, then the report is in the current working directory.  
In undeploy mode, this value must be the name of the report only; this value should not include the path. The system uses the path specified in the **-l** option.  
When you use this option, the **-d** option should not be on the same command line.
- t** This option specifies the folder where the deployed reports must be.  
Valid values:
- user
  - system
  - group
- All users can view the reports in the system folder.  
The user folder only shows those reports and folders that the user can access. An administrative user can view all user reports.  
This is a required option.
- U** This option specifies the username of the user that accesses the Web Access Server. The deployed reports appear in the folder for this user name.  
This option is in UPPERCASE.  
This is a required option.

# Usage Notes

To view a report, you must first publish it by deploying it to a server for viewing. You can use this tool to deploy reports to the Web Access Server. There are two main folders that contain the deployed reports on the Web Access Server; they are the System and Users folders in the Catalog.

You must have Administrator privileges to deploy reports to the System folder; however, any user can deploy reports to their folder in the Users folder. For more information about deploying reports to the System folder, see the *HP Performance Insight Administration Guide*.

When you want to remove a report from the Web Access Server so that you can no longer view it, you undeploy it. You can use this tool to undeploy reports from the Web Access Server. You must have Administrator privileges to undeploy reports from the System folder and to undeploy Report Packs.

## Using the deploytool Command

This section shows some formats of the `deploytool` command.

If you enter the `deploytool` command without any options, the system displays an error message followed by the help information.

### Deploy Reports

- To deploy a single report, enter the following command:

```
deploytool -U username -P password -h host_name -p port_num  
-c deploy -r rpt_name -t deploy_type -l deploy_loc -i "rpt_desc"  
-n "rpt_display_name"
```

where,

*username* is the user that accesses the Web Access Server.

*password* is the password for the corresponding user name.

*host\_name* is the name of the host for the Web Access Server.

*port\_num* is the port number for the Web Access Server.

*rpt\_name* is the name of the report with the absolute or relative path.

*deploy\_type* is the deployment type.

*deploy\_loc* is the deployment location.

*rpt\_desc* is the description of the report.

*rpt\_display\_name* is the name of the report that appears in the catalog.

- To deploy multiple reports in the same directory, enter the following command:

```
deploytool -U username -P password -h host_name -p port_num  
-c deploy -t deploy_type -l deploy_loc -d dir_path
```

where,

*username* is the user that accesses the Web Access Server.

*password* is the password for the corresponding user name.

*host\_name* is the name of the host for the Web Access Server.

*port\_num* is the port number for the Web Access Server.

*deploy\_type* is the deployment type.

*deploy\_loc* is the deployment location.

*dir\_path* is the directory that contains the reports.

If your configuration has two Web Access Server systems, you would first install the report packs on the Central Server using Package Manager and deploy the reports to one of your Web Access Server systems. You would then run the `deploytool` command on the Central Server to deploy reports to your second Web Access Server.

The following example shows how the `trendadm` user deploys the entire directory of reports from the PI installation directory (`C:\OVPI` on a Windows system) to the `system` folder on the second Web Access Server system called `northeast`:

```
deploytool -U trendadm -P trendadm -h northeast -p 80  
-c deploy -l / -d .\reports
```

## Undeploy Reports

- To undeploy a single report, enter the following command:

```
deploytool -U username -P password -h host_name -p port_num  
-c undeploy -r rpt_name -t deploy_type -l deploy_loc
```

Where,

*username* is the user that has permission to undeploy the report.

*password* is the password for the corresponding user name.

*host\_name* is the name of the host for the Web Access Server.

*port\_num* is the port number for the Web Access Server.

*rpt\_name* is the name of the report with the absolute or relative path.

*deploy\_type* is the deployment type.

*deploy\_loc* is the deployment location.

- To undeploy an entire directory, which removes the directory and its contents, enter the following command:

```
deploytool -U username -P password -h host_name -p port_num  
-c undeploy -t deploy_type -d dir_path -l deploy_loc
```

Where,

*username* is the user that has permission to undeploy the reports.

*password* is the password for the corresponding user name.

*host\_name* is the name of the host for the Web Access Server.

*port\_num* is the port number for the Web Access Server.

*deploy\_type* is the deployment type.

*dir\_path* is the directory that contains the reports.

*deploy\_loc* is the deployment location.



# Examples

The following examples illustrate some uses of the `deploytool` command that an Administrator, such as the **trendadm** user, can enter.

## Example 1: Deploy a Report to the Web Access Server (UNIX)

To deploy a report with the name **execsum.rep** from the current working directory on the **cartman** host to the **testreports/Lan** directory in the **system** folder and then display it with a name of **Executive Summary** and a description, you can use the following command.

```
deploytool -h cartman -p 80 -U trendadm -P trendadm
-c deploy -r execsum.rep -t system -l testreports/Lan/
-i "This report gives an overview of your lan." -n
"Executive Summary"
```

## Example 2: Undeploy a Report from the Web Access Server (UNIX)

To undeploy a report with the name **execsum.rep** from the **testreports/Lan** directory in the **system** folder on the **cartman** host, you can use the following command.

```
deploytool -h cartman -p 80 -U trendadm -P trendadm
-c undeploy
-r execsum.rep -t system -l testreports/Lan/
```

## Example 3: Deploy a Directory of Reports to a Different User

To deploy all the reports in the **d:\ovpi\reports\Interface\_Reporting\Admin** directory on the **powder2** host to the **user1\testreports** directory in the **user** folder, you can use the following command.

```
deploytool -h powder2 -p 80 -U user1 -P test1 -c deploy
-d d:\ovpi\reports\Interface_Reporting\Admin -t user
-l testreports
```

#### Example 4: Deploy a Directory of Reports (UNIX)

To deploy all the reports in the `/user/reports/test` directory on the **cartman** host to the `trendadm/testreports` directory in the user folder, you can use the following command.

```
deploytool -h cartman -p 80 -U trendadm -P trendadm  
-c deploy  
-d /user/reports/test -t user -l testreports
```

#### Example 4: Undeploy a Directory of Reports

To undeploy all the reports in the `user/trendadm/testreports` directory on the **cartman** host, you can use the following command.

```
deploytool -h cartman -p 80 -U trendadm -P trendadm  
-c undeploy  
-t user -d testreports -l /
```

# Error Messages

This section describes some of the messages that can occur from **deploytool**. Each message has the following format:

- A brief description about why the message appears. Each new message description starts with a check box.
- The actual message that appears with parameters for any information that may be missing.
- A suggestion about the action to do so that the message does not appear again.

- If the following message appears, the user is unauthorized to deploy the specified directory.

Directory deployment failed: Unauthorized.

Verify the following:

- The user (**-U**) and password (**-P**) values are correct.
- The specified user has the appropriate privileges to deploy the directory.

- If the following message appears, the user is unauthorized to undeploy the specified directory.

Directory undeployment failed: Unauthorized.

Verify the following:

- The user (**-U**) and password (**-P**) values are correct.
- The specified user has the appropriate privileges to undeploy the directory.

- If the following error message appears, there is a command-line syntax error.

Error processing command line: A value is required for argument *option: description*.

Verify that every option on the command line has the appropriate value. The description specifies what information is missing.

- ❑ If the following error message appears, there is a command-line syntax error.

Error processing command line: *option* is not valid for this program.

Remove the invalid option from the command line. Check [Syntax](#) on page 114 for the list of valid options.

- ❑ If the following error message appears, there is a command-line syntax error.

Error processing command line: Option *option* must be specified.

Verify that all required options are on the command line: **-c, -d or -r, -h, -p, -P, -t, -U**.

- ❑ If the following message appears, the type of task to perform is incorrect.

Incorrect value for -c argument. Please specify "deploy" or "undeploy".

Verify that the value for the **-c** option is **deploy** or **undeploy**.

- ❑ If the following message appears, the location for the deployed reports is incorrect.

Invalid deployment type specified. Please specify "system" or "user".

Verify that the value for the **-t** option is **system** or **user**.

- ❑ If the following message appears, the directory name is incorrect.

Invalid directory specified: *directory\_name*.

Verify that the directory name and path are correct for the **-d** option.

- ❑ If the following message appears, the report name is incorrect.

Invalid report specified: *report\_name*.

Verify that the name and the path of the report are correct for the **-r** option.

- ❑ If the following message appears, an input or output error occurred while reading the report or sending it to the Web Access Server.

I/O error occurred.

Verify that the file exists and that it is readable.

- ❑ If the following error message appears, there is a command-line syntax error.

`option option must be specified.`

Verify that required options for a specific task are on the command line, for example `-l` to deploy a report.

- ❑ If the following message appears, the user is unauthorized to deploy the specified report.

`Report deployment failed: Unauthorized.`

Verify the following:

- The user (`-U`) and password (`-P`) values are correct.
- The specified user has the appropriate privileges to deploy the report.

- ❑ If the following message appears, the user is unauthorized to undeploy the specified report.

`Report undeployment failed: Unauthorized.`

Verify the following:

- The user (`-U`) and password (`-P`) values are correct.
- The specified user has the appropriate privileges to undeploy the report.

- ❑ If the following message appears, the report or directory name is missing on the command line.

`Report or directory name must be specified.`

Verify that the option for the report name (`-r`) or the directory name (`-d`) is on the command line. One of these options must appear on the command line.

- ❑ If the following message appears, there was an unusual error.

`Unknown error occurred.`

Verify that the values for all options are correct. Check the host name, especially if the following message also appears:

`SEVERE: Unknown host. host_name.`



## 9 dip\_manager

The `dip_manager` command is a stand-alone utility that enables you to manage directed instance polling groups on a PI system. It enables you to import, replace, remove, or export directed instance polling groups.

### Requirements or Restrictions

- All required files must have the following statements in the file: **GroupCategory** before **GroupName**.
- An error occurs if more than one of the following options appears on the command line at the same time: **-import**, **-export**, **-export\_all**, **-remove**, **-remove\_all**, or **-replace**.
- An error occurs if the **-file** option does not appear with one of the following options on the command line at the same time: **-export\_all**, **-import**, **-remove\_all**, or **-replace**.

# Syntax

The `dip_manager` command uses the following syntax:

```
dip_manager  [-database db_name]  
               [-debug debug_level]  
               [ { -export grp_category.grp_name  
                  -export_all  
                  -import  
                  -remove grp_category.grp_name  
                  -remove_all  
                  -replace } ]  
               [-file file_name]  
               [-help]  
               [ { -v  
                  -version } ]
```



# Options

The `dip_manager` command has the following options:

- database** This option identifies the database where the changes occur. The database must appear in the list of available database servers. See the *HP Performance Insight Administration Guide* for more information about adding database servers to the list using the Web Access Server. The default is the database identified as the default in the database server list.
- debug** Use this option to set the debug output level. The higher the number, the more detailed the information. Debug output writes to standard output. Use this option only for testing in coordination with HP Technical Support due to the additional overhead it places on `dip_manager`. The default is no debug output.
- export** Use this option to generate a file containing the directed instance polling group specified on the command line. The name of the group must be in the format of *grp\_category.grp\_name*; see [Naming Convention](#) on page 131. Use the **-file** option to specify the output file name; otherwise, `dip_manager` writes the data to standard output. See [Export Examples](#) on page 140 for more information.  
  
This option cannot appear on the command line when the **-export\_all**, **-import**, **-remove**, **-remove\_all**, or **-replace** option appears on the command line.
- export\_all** Use this option to generate a file containing all the directed instance polling group definitions. It requires the **-file** option to specify the output file name. See [Export Examples](#) on page 140 for more information.  
  
This option cannot appear on the command line when the **-export**, **-import**, **-remove**, **-remove\_all**, or **-replace** option appears on the command line.

- file** This option identifies the file name, which is the text file that contains the directed instance polling group definitions to import, export, remove, or replace. If the file is not in the current working directory, you must specify the fully qualified path to the file. See [ASCII File](#) on page 134 for details on setting up this file.
- This is a required option when the **-import**, **-export\_all**, **-remove\_all**, or **-replace** option appears on the command line.
- help** This option is the help option, which displays the command-line options for the `dip_manager` command.
- import** Use this option to import directed instance polling group definitions. It requires the **-file** option to identify the file that contains the list of directed instance polling groups to import. See [Import Example](#) on page 138 for more information.
- This option cannot appear on the command line when the **-export**, **-export\_all**, **-remove**, **-remove\_all**, or **-replace** option appears on the command line.
- remove** Use this option to remove a directed instance polling group. You must specify the directed instance polling group on the command line following this option. The name of the group must be in the format of *grp\_category.grp\_name*; see [Naming Convention](#) on page 131. See [Remove Example](#) on page 141 for more information.
- This option cannot appear on the command line when the **-export**, **-export\_all**, **-import**, **-remove\_all**, or **-replace** option appears on the command line.

- remove\_all** Use this option to remove multiple directed instance polling groups. It requires the **-file** option to identify the file that contains the list of directed instance polling groups to remove. See [Remove Example](#) on page 141 for more information.  
This option cannot appear on the command line when the **-export**, **-export\_all**, **-import**, **-remove**, or **-replace** option appears on the command line.
- replace** Use this option to replace the contents of specified directed instance polling groups. It requires the **-file** option to identify the ASCII file that contains the replacement directed instance polling groups. See [Replace Example](#) on page 139 for more information.  
This option cannot appear on the command line when the **-export**, **-export\_all**, **-import**, **-remove**, or **-remove\_all** option appears on the command line.
- v** Use this option to display the current version of the
- version** `dip_manager` utility. You can use either an uppercase **-V** or the lowercase, spelled-out form **-version**.

## Naming Convention

If you want to use the command line to specify a group to be removed or exported, use the following format to identify the group:

*grp\_category.grp\_name*.

The *grp\_name* portion of this parameter is the name of the group that contains the list of nodes or instances to poll. It appears in the Select Group to Poll From pull-down list in Polling Policy Manager. It also appears in the Select Group field in the Edit Polling Groups window for a specific category.

The *grp\_category* portion of this parameter identifies the kind of group for the corresponding *grp\_name*. If the group is a **type** list, then the *grp\_category* is **type**. Similarly, if the group is a **view** list, then the *grp\_category* is **view**. If the group is a single node group, then the *grp\_category* is **node**. Otherwise, the *grp\_category* is the same as the property table name.

The following table shows the typical association for the *grp\_category* value to the corresponding kind of group in the Collect Data From field in Polling Policy Manager.

**Table 12 Typical Values for the Group Category Parameter**

Group Category Value	Value in Collect Data From Field
<b>type</b>	All Nodes of the Same Type A Combination of Type and View
<b>view</b>	All Nodes in Same View
<b>node</b>	A Single Node
<i>property_table_name</i>	Specific Instances Custom Groups

See the *HP Performance Insight Administration Guide* for more information about Polling Policy Manager.

## Usage Notes

The `dip_manager` utility applies only to enumerated list groups. If you have a rule-based group, you can convert it to an enumerated list group and then use it with `dip_manager`. To convert the group, use the Edit a Specific Instance Group window from Polling Policy Manager to select specific instances; select the **Save Selected Instances Only** box, and then click **OK**.

You can view the contents of the groups that you import or replace from Polling Policy Manager; see the “Managing Polling Groups” section in the *HP Performance Insight Administration Guide* for the instructions.

Every group belongs to a group category. If the referenced group is not a **type** or **view**, then the category is the name of the property table that contains the objects defined in the group.

The rest of this section describes the available modes, the parameters for the associated ASCII file, and how to use the `dip_manager` command.

## Modes of Operation

The `dip_manager` command has four modes of operation: import, replace, export, and remove. All modes apply only to directed instance polling groups that are enumerated type groups. If you use a rule-based group, `dip_manager` displays an error message.

### Import

The *import* mode enables you to add directed instance polling groups to the system. You must use the **-file** option to specify the name of the file that contains the group definitions to be imported. See [ASCII File](#) on page 134 for more information about the file. If a group in the file already exists, `dip_manager` adds the entries from the file to the existing group.

### Replace

The *replace* mode enables you to replace the members of existing directed instance polling groups. You must use the **-file** option to specify the name of the file that contains the existing groups with their new group member objects. See [ASCII File](#) on page 134 for more information about the file.

In this mode, `dip_manager` deletes the member objects from the specified group and then adds the objects from the corresponding file for that group. If the *grp\_category* parameter does not exist, `dip_manager` displays an error message. If the *grp\_name* parameter does not exist, `dip_manager` creates the group.

### Export

The *export* mode enables you to create a file containing directed instance polling group definitions from enumerated type groups. There are two ways to export directed instance polling group definitions:

- Create an export file containing just one directed instance polling group definition by entering the **-export** option followed by the group identifier on the command line. The group identifier must be in the format of *grp\_category.grp\_name*; see [Naming Convention](#) on page 131 for an explanation of the group identifier.

- Create an export file containing all the directed instance polling groups from enumerated type groups on the system by entering the **-export\_all** option on the command line. You must also use the **-file** option and specify the output file name on the command line. Only the enumerated type groups appear in the file.

## Remove

The *remove* mode enables you to remove directed instance polling groups from the system that are enumerated type lists. There are two ways to remove directed instance polling group definitions:

- Remove groups one at a time by entering the **-remove** option followed by the group identifier on the command line. The group identifier must be in the format of *grp\_category.grp\_name*; see [Naming Convention](#) on page 131 for an explanation of the group identifier.
- Remove multiple groups by using the **-remove\_all** option. You must include the **-file** option. With this option, *dip\_manager* uses only the *grp\_category* and *grp\_name* attributes in the associated file. See [ASCII File](#) for more information about the file.

## ASCII File

The ASCII file that contains the directed instance polling groups must be in the following format:

```
GroupCategory = grp_category
GroupName = grp_name
Object = object_by-variable1, object_by-variable 2,...
```

The descriptions for the parameters in the ASCII file follow:

<i>grp_category</i>	<p>This parameter specifies the directed instance polling group category. The values may be <b>type</b>, <b>view</b>, or the property table name.</p> <p>It is a required parameter for all input files.</p>
<i>grp_name</i>	<p>This parameter specifies the directed instance polling group name. The length of the <i>grp_name</i> can be up to 30 characters. This group must be an enumerated type group.</p> <p>It is a required parameter for all input files.</p>
<i>object_by-variable</i>	<p>This parameter specifies the contents of the directed instance polling group by identifying each object. It corresponds to the object by-variables associated with the property table. If the property table has three object by-variables, then each <b>Object</b> entry must have three values separated by commas. If the statement does not contain the proper number of object by-variables, <i>dip_manager</i> displays an error message and skips the statement.</p> <p>Many tables use the two default object by-variables, which are <i>dsi_target_name</i> and <i>dsi_table_key</i>.</p> <p>For example, if the group consists of routers, each object is a router, and the object by-variables identify each router.</p> <p>It is a required parameter when you use the <b>-import</b> or <b>-replace</b> option.</p>

The following syntax rules apply to the file:

- The spelling of the statements GroupCategory, GroupName, and Object, must match exactly; they are case sensitive.
- There may be multiple group definitions in the file.
- The GroupCategory statement must appear before the GroupName statement in the file for each group.
- There may be multiple Object statements for each GroupCategory / GroupName pair.

- The number of object by-variables must match the number of by-variables for the corresponding property table.
- Commas separate the by-variables on the Object statement.
- The statements may contain spaces.
- The file may contain blank lines.



If any of the statements in the file contain syntax errors, an error message occurs and the process terminates at that point. The system does not check the rest of the file for additional errors.

If valid GroupCategory and GroupName statements appear in the file, `dip_manager` imports or replaces the group even if the corresponding Object statements are not valid. The group does not have any entries.

If there are invalid Object statements in the file for remove mode, `dip_manager` ignores them and removes the groups; it stops processing if there is a syntax error.

## Using the `dip_manager` Command

This section shows some formats of the command for the various modes.

- If you enter the `dip_manager` command without any options, the system displays an error message followed by the help information.
- To display the syntax and options for this command, enter:

```
dip_manager -help
```

- To display the version information for this command, enter one of the following commands:

```
dip_manager -V
```

or

```
dip_manager -version
```

- To import directed instance polling group definitions, enter the following command:

```
dip_manager -import -file file_name
```

where, *file\_name* is the name of the ASCII file that contains the list of directed instance polling group definitions to import.



- To replace one or more directed instance polling group definitions, enter the following command:

**dip\_manager -replace -file *file\_name***

where, *file\_name* is the name of the ASCII file that identifies the directed instance polling group definitions to change or add.

- To remove multiple directed instance polling group definitions, enter the following command:

**dip\_manager -remove\_all -file *file\_name***

where, *file\_name* is the name of the ASCII file that identifies the directed instance polling group definitions to remove.

- To remove a single directed instance polling group from the system, enter the following command:

**dip\_manager -remove *grp\_category.grp\_name***

where, *grp\_category* is the kind of group for the corresponding *grp\_name*.

*grp\_name* is the name of the group that contains the list of instances to poll.

- To export a single directed instance polling group definition to the screen, enter the following command:

**dip\_manager -export *grp\_category.grp\_name***

where, *grp\_category* is the kind of group for the corresponding *grp\_name*.

*grp\_name* is the name of the group that contains the list of instances to poll.

- To export a single directed instance polling group definition to a file, enter the following command:

**dip\_manager -export *grp\_category.grp\_name* -file *file\_name***

where, *grp\_category* is the kind of group for the corresponding *grp\_name*.

*grp\_name* is the name of the group that contains the list of instances to poll.

*file\_name* is the name of the ASCII file that contains directed instance polling group definition.

- To export all directed instance polling group definitions to a file, enter the following command:

```
dip_manager -export_all -file file_name
```

where, *file\_name* is the name of the ASCII file that contains the directed instance polling group definition.

## Examples

This section has examples that show each mode of `dip_manager`. They have the following characteristics:

- All examples see the sample file used in the import example.
- Any information that appears about the **-file** option for one example also applies to other examples, such as the need for a qualified path.

### Import Example

To import a list of directed instance polling groups from a file named `dip_grp_in.txt`, enter the following command:

```
dip_manager -import -file dip_grp_in.txt
```

The following is an example of the contents for the input file, `dip_grp_in.txt`:

```
GroupCategory = test_prop_tbl
GroupName = test_dip_group_a
Object = test_node_1,abc
Object = test_node_2,xyz
Object = tst_a,20
Object = tst_b,25
```

```
GroupCategory = test_prop_tbl
GroupName = test_dip_group_b
Object = test_node_3,def
Object = test_node_4,ghi
Object = tst_c,20
Object = tst_d,25
```

You can see the results from this command by using the Polling Groups option from Polling Policy Manager; from the Edit Polling Groups window, do the following:

- Select **Specific Instances** from the Select Kind of Group field.
- Select the desired group name from the Select Group field.
- Click **Edit**. The Edit a Specific Instance Group window shows the group name, property table name, and the list of instances.

## Replace Example

To change the contents of existing directed instance polling groups, use the following command:

```
dip_manager -replace -file dip_grp_rep.txt
```

The following is an example of the contents for the input file, `dip_grp_rep.txt`:

```
GroupCategory = test_prop_tbl  
GroupName = test_dip_group_c  
Object = test_node_5,abc  
Object = test_node_6,def
```

```
GroupCategory = test_prop_tbl  
GroupName = test_dip_group_b  
Object = test_node_7,abc  
Object = test_node_8,xyz
```

In this example, `dip_manager` deletes the current contents of `test_dip_group_b` and replace it with the contents from this file; then it adds the group `test_dip_group_c`, since it did not already exist.

## Export Examples

If you want to export the contents of a single directed instance polling group, such as `test_dip_group_b`, to a file named `dip1.txt`, enter the following command:

```
dip_manager -export test_prop_tbl.test_dip_group_b  
-file dip1.txt
```

If you want to export the contents of a single directed instance polling group, such as `test_dip_group_b`, to the screen, enter the following command:

```
dip_manager -export test_prop_tbl.test_dip_group_b
```

## Export\_All Examples

To generate a file containing all the directed instance polling group definitions that are in the catalog, you can specify an ASCII file to store the definitions. For example, if you want to export the current list of directed instance polling groups to a file called `dip_grp_out.txt`, enter the following command:

```
dip_manager -export_all -file dip_grp_out.txt
```

If you do not specify a file, `dip_manager` exports the directed instance polling group definitions to standard output. The directed instance polling group definitions appear on the screen when you enter the following command:

```
dip_manager -export_all
```

If you want to export the directed instance polling groups from a different server named **bear** to a file named `dip_grp_bear.txt`, enter the following command:

```
dip_manager -export_all -file dip_grp_bear.txt  
-database bear
```

## Remove Example

If you want to remove a single directed instance polling group such as `test_dip_group_c`, enter the following command:

```
dip_manager -remove test_prop_tbl.test_dip_group_c
```

## Remove\_All Examples

If you want to remove multiple directed instance polling groups, you can use an ASCII file to specify the list. To remove the directed instance polling groups in the file `dip_grp_rem.txt`, enter the following:

```
dip_manager -remove_all -file dip_grp_rem.txt
```

Note that you need to fully qualify the path for the file if it is in a different directory. For example, if the file were in the `lists` directory from the D drive on a Windows machine, you would enter the following command:

```
dip_manager -remove_all -file d:\lists\dip_grp_rem.txt
```

# Error Messages

This section describes some of the messages that can occur from `dip_manager`. Each message has the following format:

- A brief description about why the message appears. Each new message description starts with a check box.
  - The actual message that appears with parameters for any information that may be missing.
  - A suggestion about the action to do so that the message does not appear again.
- ❑ If the following error message appears, there is a command-line syntax error. This means that a required mode option is missing.

```
A mode of operation is required.  
Exiting program with code 1.
```

Verify that the command has only one of the following options on the command line at the same time: **-import**, **-export**, **-export\_all**, **-remove**, **-remove\_all**, or **replace**.

- ❑ If the following error message appears, the database specified with the **-database** option on the command line does not exist.

```
Connection URL not found.  
Exiting program with code 3.
```

Verify the spelling of the database name. If the spelling is correct, you can add the database using the Web Access Server.

- ❑ If the following message appears, the value for the **-debug** option on the command line is not a numeric character.

```
Debug level must be a number.  
Exiting program with code 1.
```

Verify that the value for the **-debug** option on the command line is an integer number and that it does not contain a letter or other characters such as a dot.

- ❑ If the following message appears, the value for the **-debug** option on the command line is not valid.

```
Debug level must be between 0 and 3.  
Exiting program with code 1.
```

Verify that the value for the **-debug** option on the command line is an integer number and is one of the following numbers: 0, 1, 2, or 3.

- ❑ If the following error message appears, the file name specified on the command line in the **-file** *file\_name* option does not exist.

```
File file_name does not exist.  
Exiting program with code 2.
```

Verify the spelling of the file name or that the file exists in the specified location. You may have to supply a fully qualified path with the file name.

- ❑ If the following message appears, the *grp\_name* parameter on the command line with the **-remove** option is not an enumerated list. It could be a rule-based group.

```
Group grp_name must be an enumerated list.
```

If the group is a rule-based group, you can convert it to an enumerated list. See [Usage Notes](#) on page 132.

- ❑ If the following message appears, the *grp\_name* parameter on the command line with the **-export** option does not exist or is not an enumerated list.

```
Group grp_name not found or not an enumerated instance group.  
Exiting program with code 7.
```

Verify the spelling of the group name or that the group exists for the corresponding *grp\_category*. If the group exists, it may be a rule-based group; if it is, you can convert it to an enumerated list. See [Usage Notes](#) on page 132.

- ❑ If the following message appears, the entry for the **-export** or **-remove** option did not have the format *grp\_category.grp\_name* specified on the command line.

```
Group name must be specified category.group.  
Exiting program with code 1.
```

Verify the entry is in the format *grp\_category.grp\_name* and resubmit.

- ❑ If the following message appears, the GroupCategory statement on the specified line is missing in the associated file.

```
Missing category name - line n1.  
Exiting program with code 4.
```

Verify the `GroupCategory` statement is in the associated file. It must appear before the `GroupName` statement in the file for each group.

- ❑ If the following error message appears, the **-file** option is missing on the command line for an option that requires a file.

```
No file specified.  
Exiting program with code 1.
```

Verify that the **-file** option is on the command line with one of the following options at the same time: **-import**, **export\_all**, **-remove\_all**, or **replace**.

- ❑ If the following message appears, the number of entries on the specified line in the associated file does not match the number of object by-variables for the specified property table.

```
Object by-variable mismatch on line n1 - n2 expected.
```

Verify that each line in the associated file has the correct number of object by-variables on it separated by commas. If you need to verify the number of object by-variables for the property table, use Table Manager to view the columns in the property table. For more information about Table Manager, see the *HP Performance Insight Administration Guide*.

- ❑ If the following error message appears, there is a command-line syntax error. This means that there are multiple mode options on the same command line.

```
Only one mode of operation is allowed.  
Exiting program with code 1.
```

Verify that the command has only one of the following options on the command line at the same time: **-import**, **-export**, **-export\_all**, **-remove**, **-remove\_all**, or **replace**.

- ❑ If the following error message appears, there is a syntax error on the specified line in the file.

```
Syntax error - line n.  
Exiting program with code 4.
```

Verify the spelling of the statements in the file and that they have the correct format.

- ❑ If the following message appears, the specified *grp\_name* does not exist.

```
The group grp_name does not exist.
```



The *grp\_name* may be in either of the following locations: with the **-remove** option on the command line or with the GroupName statement in the associated file. In some cases, the *grp\_category* name may be incorrect on the command line or in the GroupCategory statement in the associated file. Verify the following items:

- The spelling of the *grp\_name* parameter is correct.
- The *grp\_name* exists for the type of group in the *grp\_category*.
- The spelling of the *grp\_category* parameter is correct and it has the appropriate designation.

You can use Polling Policy Manager to view the existing list of directed instance polling groups. From the Edit menu, select **Polling Groups**. The Edit Polling Groups window appears. The *grp\_category* corresponds to the Select Kind of Group field. The *grp\_name* corresponds to the Select Group field. You can also view the existing list of directed instance polling groups from the Edit Polling Policy window. See [Naming Convention](#) on page 131 for more information.

- If the following message appears, the *grp\_category* portion of the GroupCategory statement in the associated file is not a valid category or property table name.

The group category *grp\_category* was not found.

Verify the spelling of the *grp\_category* name. If this name is a property table, verify the property table exists. For more information, see the *HP Performance Insight Administration Guide*.

- If the following error message appears, there is a value missing after the **-export** option.

Value after -export expected.  
Exiting program with code 1.

Enter a value after the **-export** command on the command line.

- If the following error message appears, there is a value missing after the **-remove** option.

Value after -remove expected.  
Exiting program with code 1.

Enter a value after the **-remove** command on the command line.



## 10 ee\_collect

You can use the `ee_collect` command to import data from a flat file into a datapipe on a PI system. It uses the instructions from a TEEL file associated with it to import the data. See the *HP Performance Insight TEEL Reference Guide* for more information.

### Requirements and Restrictions

- The corresponding TEEL file must contain the import mapping and processing instructions.
- The TEEL file must contain the **DataPipe** statement, which specifies the registered datapipe.
- The TEEL file must contain the **TrendTableName** statement, the **PropertyTableName** statement, or both.
- The Single TEEL File (STF) mode requires the **-a** option.
- The Catalog-Based Collection (CBC) mode requires the **-i** option.
- The system ignores any of the following legacy options that appear on the command line: **-e**, **-p**, **-y**, **-z**, **-Z**.

# Syntax

A parameter less `ee_collect` command has the following syntax.

## Format 1

Format 1 executes by interval, which is Catalog-Based Collection Mode. In this format, `ee_collect` obtains the instructions from the TEEL file identified in the associated polling policy.

```
ee_collect  [-b]
              [-c max_process]
              [-d debug_level]
              [-h]
              [-H alt_poller_name]
              -i interval

              [-m {n
                   all}]
              [-mw option_list]
              [-n]
              [-N]
              [-R enable recovery mode for bcp_gateway]
              [-v]
              [-w percentage]
```

## Format 2

Format 2 executes by TEEL file name for the datapipe, which is Single TEEL File Mode.

```
ee_collect  -a TEEL_file_name  
              [-b]  
              [-c max_process]  
              [-d debug_level]  
              [-h]  
              [-m  $\left\{ \begin{smallmatrix} n \\ \text{all} \end{smallmatrix} \right\}$  ]  
              [-mw option_list]  
              [-N]  
              [-s source_data_file]  
              [-S source_directory]  
              [-u disposition[,target_directory,timestamp]]  
              [-v]  
              [-w percentage]
```

# Options

The `ee_collect` command has the following options:

- a**      Use this option to specify the name of the TEEL file that contains the instructions to process.  
This is a required option for Single TEEL File (STF) mode.
- b**      This option enables the `bcp_gateway` program to back up the collected data if the data upload process fails. If this option is specified with the `ee_collect` command, `bcp_gateway` program backs up the failed batch of collected data for reviewing the failure line later. Note that the `bcp_gateway` program is responsible for uploading collected data into PI database.
- c**      Use this option to specify the maximum number of child-collection processes that `ee_collect` spawns at the same time. When `ee_collect` runs, it starts child processes that actually do the collection. It spawns a process for each specified file.  
The valid values range from 1 - 25.  
The system default is **5**, which means that `ee_collect` spawns up to 5 child-collection processes at the same time.
- d**      Use this option to set the debug output level. The higher the number, the more detailed the information. Debug writes the output to standard output. Only use this option for testing in coordination with HP Technical Support due to the additional overhead it places on `ee_collect`.  
This option is for `ee_collect` only; to turn on debugging for underlying processes, use the **-mw** option.  
The valid values are 1, 2, or 3.  
The default is no debug output, which is **0**.
- h**      This option is the help option, which displays the command-line syntax for the `ee_collect` command. This option overrides all other options on the command line.

- H** Use this option to specify an alternate poller name. When you run `ee_collect` in distributed mode, with the **-n** option, the poller compares the local hostname to the Poll From field in the polling policy. When you use the **-H** option, `ee_collect` compares the Poll From field in the polling policy to the alternate poller name. See [Distributed Polling](#) on page 301 for more information. You must include the **-n** option with this option on the command line to get the desired results.

This option is in UPPERCASE.

This option is available only in Catalog-Based Connection (CBC) mode.
- i** Use this option to specify the Collection ID that `ee_collect` uses to run entries in the polling policy, which have this value in their Interval field. Note that the configuration of `trendtimer` determines the frequency for running `ee_collect`; however, for consistency, if a collection request has a collection ID of **5**, the system runs the command every 5 minutes.

This is a required option for Catalog-Based Connection (CBC) mode.
- m** Use this option to specify the maximum number of rows that `ee_collect` processes for each file in the datapipe. This option overrides the MaxRows statement in the TEEL file.

The system default is **a11**.
- mw** Use this option to pass options to `mw_collect`. Enclose the list of options in double quotation marks. Separate each option from the next option with a space.

The following `mw_collect` options are invalid in this option:

**-a, -c, -f, -H, -i, -n, -t, -w**
- n** Use this option to enable distributed polling. If this option is used, `ee_collect` only executes the collection request if the Poll From field in the polling policy for this collection request matches the hostname of the machine on which `ee_collect` is running. If you omit this option, `ee_collect` executes all polling requests whose interval matches the value of the **-i** option, regardless of the hostname specified to do the polling in the polling instructions.

This option is available only in Catalog-Based Connection (CBC) mode.

- N** Use this option to import data only for managed objects that already exist in the target database. When you use this option, `ee_collect` only imports the managed objects that already exist in the associated property table, which are the existing keys.  
This option is in UPPERCASE.
- R** Use this option to enable the `bcp_gateway` to run in recovery mode. The `bcp_gateway` process uploads the collected data into the PI database. This option is in UPPERCASE.
- s** Use this option to specify a single, source data file. This option overrides the `SourceFile` or `SourceDirectory` statements in the TEEL file. You must supply the complete path with the file name.  
This option is available only in Single TEEL File (STF) mode.
- S** Use this option to specify the source data directory. This option overrides the `SourceFile` or `SourceDirectory` statements in the TEEL file. You must supply the complete path for the directory name. You can use wildcard characters with this option. The system processes all files in this specified directory.  
This option is available only in Single TEEL File (STF) mode.  
This option is in UPPERCASE.
- u** Use this option to specify the source file disposition. The values are **keep**, **copy**, **delete**, or **move**. The **copy** and **move** values require a path for the directory location to store the source file; the filename remains the same. Note that there should not be a space after the comma (,) and before the directory path. This option overrides the `SourceDisposition` statement in the TEEL file.  
The system default is **keep**.  
This option is available only in Single TEEL File (STF) mode.
- V** Use this option to display the current version of the `ee_collect` utility. This option overrides all other options on the command line, except the help option.  
This option is in UPPERCASE.
- w** Use this option to stop the collection of data when the database-used size reaches the specified percentage. The collection routine checks the `dbstats` tables to estimate the current state of the database-full size and determines if it should write to the database. The default is **90** for 90%.



# Usage Notes

If the table does not exist before `ee_collect` tries to import the data, `ee_collect` creates the table based on the instructions in the import TEEL file in non-compatibility mode, which has `-c off` on the `datapipe_manager` command line.

## Interval Polling (CBC Mode)

When you invoke `ee_collect` in the Catalog-Based Collection (CBC) mode with the `-i` option, which is Format 1, it reads the polling control table in the database for the list of instructions whose intervals match the value of the `-i` option on the `ee_collect` command line. Each entry in the list specifies which polled device group and datapipe to collect.

Use Polling Policy Manager to enter the polling policy information for the datapipe.

## Direct Polling (STF Mode)

When you invoke `ee_collect` in the Single TEEL File (STF) mode with the `-a` option, which is Format 2, it imports the data for the datapipe specified in the TEEL file. The specified TEEL file is the value of the `-a` option, and it contains the instructions to import the data.

## Log File

The log entries from `ee_collect` are in the file `$TREND_LOG/trend.log`.

## Using the `ee_collect` Command

This section shows some formats of the command in the various modes. Note that some of the options override the corresponding statements in the associated TEEL file. Some of the options can be on the command line in either mode, so the formats with those options appear in each mode; you can combine those options in any combination that meets your requirements.

- If you enter the `ee_collect` command without any options, the system displays the help information.
- To display the syntax and options for this command, enter:  
**`ee_collect -h`**
- To display the version information for this command, enter:  
**`ee_collect -V`**

## STF Mode

- To specify a specific TEEL file, enter the following command:  
**`ee_collect -a file_name`**  
Where, *file\_name* is the name of the TEEL file that contains the instructions to import the data into the datapipe.
- To copy the source files to another directory, enter the following command:  
**`ee_collect -a file_name -u directory_name`**  
Where, *file\_name* is the name of the TEEL file that contains the instructions to import the data into the datapipe.  
*directory\_name* is the location for the copies of the source files.
- To import the source files from a directory that is different than the directory or file specified in the TEEL file, enter the following command:  
**`ee_collect -a file_name -S directory_name`**  
Where, *file\_name* is the name of the TEEL file that contains the instructions to import the data into the datapipe.  
*directory\_name* is the name of the directory that contains the source files.
- To import a source file from a file that is different than the directory or file specified in the TEEL file, enter the following command:  
**`ee_collect -a file_name -s src_file_name`**  
Where, *file\_name* is the name of the TEEL file that contains the instructions to import the data into the datapipe.  
*src\_file\_name* is the name of the source file to import with the complete path.

- To import data only for managed objects that already exist in the target database and process only some rows of data rather all the rows, enter the following command:

```
ee_collect -a file_name -N -m number
```

Where, *file\_name* is the name of the TEEL file that contains the instructions to import the data into the datapipe.

*number* is the number of rows to process.

## CBC Mode

- To specify an interval, enter the following command:

```
ee_collect -i interval
```

Where, *interval* is the collection id that matches the interval in the polling policy.

- To enable distributed polling, enter the following command:

```
ee_collect -i interval -n
```

Where, *interval* is the collection id that matches the interval in the polling policy.

- To import data from a specific poller, enter the following command:

```
ee_collect -i interval -n -H alt_poller_name
```

Where, *interval* is the collection id that matches the interval in the polling policy.

*alt\_poller\_name* is the name of the poller in the Poll From field in the polling policy.

- To import data only for managed objects that already exist in the target database and process only some rows of data rather all the rows, enter the following command:

```
ee_collect -i interval -N -m number
```

Where, *interval* is the collection id that matches the interval in the polling policy.

*number* is the number of rows to process.

# Examples

This section shows examples using the various modes of `ee_collect`.

## Direct Polling of a Datapipe (STF Mode)

To have `ee_collect` poll a node defined in a TEEL file named `test`, use the command:

```
ee_collect -a test
```

Note that `test.teel` is in the `$DPIPE_HOME/lib` directory.

## Interval Polling (CBC Mode)

### Example 1

To have `ee_collect` run all collection requests with a collection id of 10, which shows as a 10-minute interval in the polling policy, enter the following on the command line:

```
ee_collect -i 10
```

Note that the interval polling entries for `ee_collect` are in the `trendtimer` schedule file `$DPIPE_HOME/lib/trendtimer.sched`. The system invokes `ee_collect` according to that schedule.

### Example 2

To have `ee_collect` run all collection requests with a collection id of 10 for a particular hostname, which is the hostname where `ee_collect` is running, enter the following on the command line:

```
ee_collect -i 10 -n
```

# Error Messages

This section describes some of the messages that can occur from **ee\_collect**. Each message has the following format:

- A brief description about why the message appears. Each new message description starts with a check box.
- The actual message that appears with parameters for any information that may be missing.
- A suggestion about the action to do so that the message does not appear again.

## General Errors

- ❑ If the following error message appears, the specified data table is not in the PI dictionary.

Data table `"table_name"` not found in PI dictionary.

Verify that the data table exists in the PI dictionary. You can use Table Manager to view the current list of data tables in the system.

- ❑ If the following error message appears, `ee_collect` was unable to convert the TEEL file for a property only import.

Failed to convert property only TEEL file `"file_name"`.

Note that `ee_collect` converts TEEL files used for importing from the 4.6 version to the 5.3 version of TEEL. See the messages that follow for additional information.

- ❑ If the following error message appears, `ee_collect` was unable to convert the TEEL file for data import.

Failed to convert TEEL file `"file_name"`.

Note that `ee_collect` converts TEEL files used for importing from the 4.6 version to the 5.3 version of TEEL. See the messages that follow for additional information.

- ❑ If the following error message appears, the system failed to create the specified control file for `ee_collect`. Typically, this message appears with other messages that provide additional information.

Failed to create dpipe\_file control file : *file\_name*.

See any messages that follow for additional information; otherwise, contact HP Technical Support.

- ❑ If the following error message appears, the system failed to create the specified `mw_collect` configuration file. Typically, this message appears with other messages that provide additional information.

Failed to create mw config file : *file\_name*.

See any messages that follow for additional information; otherwise, contact HP Technical Support.

- ❑ If the following error message appears, the system failed to run the `mw_collect` command. Typically, this message appears with other messages that provide additional information.

Failed to execute `mw_collect`.

Verify that all variables are set and connections are in place.

- ❑ If the following error message appears, the corresponding `ee_collect` command with the `-i` option failed. Typically, this message appears with other messages that provide additional information.

Failed to get TEEL file from dictionary for datapipe : *datapipe\_name* and poll\_from : *server\_name*.

Verify the specified datapipe has a registration entry in the dictionary for the TEEL file on the specified server. The server name is from the *poll\_from* parameter in the corresponding collection policy. If the `-H` option is on the command line, then the server name is the alternate poller name specified with that option. See any messages that follow for additional information.

- ❑ If the following error message appears, the corresponding `ee_collect` command with the `-a` option failed. Typically, this message appears with other messages that provide additional information.

Failed to perform TEEL base collection.

Verify that the file name is correct for the `-a` option and that all corresponding options with their appropriate values appear on the command line. See any messages that follow for additional information.

- ❑ If the following error message appears, the system was unable to validate the rule file for a specific datapipe and host name.

Failed to validate rule file for a given datapipe and hostname.

Verify that the name of the rule file is correct, and that it is in the installed datapipe dictionary. Typically, this message appears with other messages that provide additional information.

- ❑ If the following error message appears, there is no collection policy defined for this interval.

No collections defined for the interval : *value*.

Verify that a collection policy exists for the interval specified with the **-i** option, and that the table in the collection policy uses the `dsi_ee` collector.

- ❑ If the following error message appears, the rule file, which is the TEEL file name specified in the **-a** option, is different than the rule file registered in the PI dictionary for a specific datapipe and host name.

Rule file mismatch for datapipe : *datapipe\_name* and host\_name : *host\_name*.

Verify the following:

- The TEEL file name specified in the **-a** option is correct for the specified datapipe and host name.
  - The corresponding rule file appears in the installed datapipe dictionary.
  - The specified datapipe does not have another TEEL (rule) file registered for it; if it does, use another TEEL file or datapipe name.
- ❑ If the following error message appears, the specification for the source directory name or source file name is missing either in the TEEL file or on the command line.

Source data files must be specified either in TEEL file or through command line option.

Verify that the `SourceDirectory` or the `SourceFile` statement is in the TEEL file, or that the **-s** or **-S** option is on the command line with the appropriate value.

## TEEL File Statement Errors

- ❑ If the following error message appears, the value for the CollectorModule statement is incorrect.

Collector module must be specified as "dsi\_ee" in TEEL file *file\_name*.

Verify that the value for the CollectorModule statement in the TEEL file is dsi\_ee when you use ee\_collect to import the data.

- ❑ If the following error message appears, the datapipe name in the collection policy is different than the datapipe name in the TEEL file.

DataPipe mismatch between TEEL file and collection policy : *datapipe\_name*.

Verify that the name of the datapipe is correct in the DataPipe statement in the TEEL file; otherwise, create a corresponding collection policy for the datapipe.

- ❑ If the following error message appears, the data table name in the collection policy is different than the data table name in the TEEL file.

Data table mismatch between TEEL file and collection policy : *table\_name*.

Verify that the name of the data table is correct in the TrendTableName statement in the TEEL file; otherwise, create a corresponding collection policy for the specified data table.

- ❑ If the following error message appears, the system failed to run the PostProcessor statement in the TEEL file. Typically, this message appears with other messages that provide additional information.

Failed to execute post-processor *post\_proc\_name*.

Verify that the command syntax for the PostProcessor statement in the TEEL file is correct, and that it includes all the parameters necessary to run the command. Make sure that the command is the one you want to process after the system processes the TEEL file. See any messages that follow for additional information.

- ❑ If the following error message appears, the system failed to run the PreProcessor statement in the TEEL file. Typically, this message appears with other messages that provide additional information.

Failed to execute pre-processor *pre\_proc\_name*.



Verify that the command syntax for the PreProcessor statement in the TEEL file is correct, and that it includes all the parameters necessary to run the command. Make sure that the command is the one you want to process before the system processes the TEEL file. See any messages that follow for additional information.

- ❑ If the following error message appears, there are new and legacy import statements in the TEEL file.

New and legacy import statements can not be used together.  
Replace legacy statements with new statements.

Verify that new and legacy import statements, such as ImportData and CoreColumn, do not appear in the TEEL file at the same time. The TEEL file should not contain any of the following statements:

- CoreColumn.xxx statements
- DataColumn statements with an offset
- PropertyColumn statements with an offset

- ❑ If the following error message appears, the *path* specified in the SourceDirectory statement or the *filename* specified in the SourceFile statement in the TEEL file is incorrect.

Source file or directory *file\_name* not found.

Verify that the spelling of the file or directory name is correct, or that the file or directory name exists.

- ❑ If the following error message appears, a required statement is missing in the specified TEEL file.

Required directive *statement* is missing in TEEL file  
*file\_name*.

Verify that the following statements are in the TEEL file to import the data.

- DataPipe
- TrendTableName
- PropertyTableName (for property table imports)
- ImportData

## Command Line Option Errors

- ❑ If the following error message appears, the specified option requires double quotation marks to enclose its corresponding values.

Arguments for option *option* must be enclosed in double quotes.

Verify that there is a double quote after the option and before the first value and another double quote after the last value for the specified option.

- ❑ If the following error message appears, the specified option has an incorrect value.

Argument *value* is invalid for option *option*.

Verify that the value is valid and is in the appropriate format for the specified option.

- ❑ If the following error message appears, the command line has an option specified in the **-mw** option that `ee_collect` already passes to `mw_collect`.

Argument "*value*" is not allowed with "-mw" option.

Verify that the **-mw** option does not contain any of the following options to pass to `mw_collect`.

**-a, -c, -f, -H, -i, -n, -t, -w**

- ❑ If the following error message appears, the specified option has an incorrect value.

Argument *value* is not defined for the option *option*.

Verify that the value is valid and is in the appropriate format for the specified option.

- ❑ If the following error message appears, the specified value is no longer valid for the specified option. The value may have been valid in a previous version of the software.

Argument *value* is obsolete for option *option*.

Verify the valid values available for the specified option in the current release of the software.

- ❑ If the following error message appears, the specified file does not exist.

File *file\_name* referenced by option *option* does not exist.

Verify that the specified file name has the correct path on the command line. Remember to check the spelling of each member of the path and the file name.

- ❑ If the following error message appears, the specified command requires the specified option.

Missing required option *option* for command *command\_name*.

Verify that the *xxx* option is on the command line.

- ❑ If the following error message appears, the specified option does not require a value for the specified command.

Option *option* does not take an argument for command *command\_name*.

Verify that the specified option does not have a value after it on the command line.

- ❑ If the following error message appears, the specified option is not valid for the specified command.

Option *option* is not defined for command *command\_name*.

Verify that the specified option is not on the command line for the specified command.

- ❑ If the following error message appears, the specified option is no longer available for the *xxx* command.

Option *option* is obsolete.

Verify that the specified option is not on the command line.

- ❑ If the following error message appears, the specified option is missing the corresponding value for the specified command.

Option *option* requires an argument for command *command\_name*.

Verify that the specified option has its corresponding value and is in the appropriate format on the command line.

- ❑ If the following error message appears, the specified option requires a valid value.

Option *option* requires valid argument.

Verify that the specified option has its corresponding value and is in the appropriate format on the command line.

- ❑ If the following error message appears, the specified options are mutually exclusive.

The option *option1* can not be used with option *option2*.

Verify that only one of the specified options is on the command line.

- ❑ If the following error message appears, the system is unable to create the specified file from the specified option.

Unable to create file *file\_name* referenced by option *option*.

Verify that the specified file has write permission and the correct path for the file on the command line. Remember to check the spelling of each member of the path and the file name.

## Generic File I/O Errors

- ❑ If the following error message appears, the file does not have the appropriate permissions to access the file.

File does not have read access *file\_name*.

Verify the following:

- The spelling of the file name is correct.
- The file exists in the specified location. You may have to supply a fully qualified path with the file name.
- The file has the proper permissions to access it.

- ❑ If the following error message appears, the file name specified on the command line does not exist.

File *file\_name* does not exist.

Verify that the specified file name has the correct path and it is in the appropriate location. Remember to check the spelling of each member of the path and the file name.

- ❑ If the following error message appears, the system cannot read the specified file.

`I/O error in reading file file_name.`

Verify that the specified file has read access and that the user has permission to read it.

- ❑ If the following error message appears, the system cannot write to the specified file.

`I/O error in writing to file file_name.`

Verify that the specified file has write access and that the user has permission to write to it.

- ❑ If the following error message appears, the specified file has invalid statements in it.

`Incorrect syntax in file file_name.`

Verify that the specified file has correct format and that the statements have the correct format.

- ❑ If the following error message appears, `ee_collect` was unable to locate the source files.

`No source data file(s) found. ee_collect Exiting.`

Verify that the `SourceDirectory (-s)` or `SourceFile (-s)` statements have the correct path or file name specified in the TEEL file (or on the command line); otherwise, verify that the specified files exist in the specified location.

- ❑ If the following error message appears, `ee_collect` was unable to locate the target directory name specified in the `SourceDisposition` statement in the TEEL file or the `-u` option on the command line.

`Target archive directory "directory_name" not found.`

Verify that the `SourceDisposition` statement in the TEEL file or `-u` option on the command line have the correct path specified; otherwise, verify that the specified path exists.



# 11 formdeploytool

The `formdeploytool` command is a stand-alone utility that enables you to deploy forms or a folder of forms to the PI Administration Server so that you can view them from the Object Manager on a PI system. You can also use this command to undeploy forms or a folder of forms from the PI Administration Server, which removes them from view on the Object Manager.

If you want to deploy or undeploy forms to the PI Administration Server using the GUI tools, you can use the Form Deployment Wizard or Package Manager. See the *HP Performance Insight Guide to Building and Viewing Reports* or the *HP Performance Insight Administration Guide* for more information.

## Requirements and Restrictions

- If you deploy or undeploy a form when the management console is open, you must click the **Refresh** button and, in some cases, select the appropriate device to see the result of the action.
- Either the `-d` option or the `-r` option is a required option on the command line; however, both of them cannot be on the command line at the same time.

# Syntax

The `formdeploytool` command uses the following syntax:

```
formdeploytool  -c { -deploy  
                  -undeploy }  
                  [ -d dir_path ]  
                  -h hostname  
                  [ -i form_desc ]  
                  [ -l deploy_loc ]  
                  [ -n display_name ]  
                  -p port_num  
                  -P password  
                  [ -r form_name_path ]  
                  [ -S ]  
                  -U username
```



# Options

The `formdeploytool` command has the following options:

- c** This option specifies the task to perform.  
Valid values are:
  - **deploy**
  - **undeploy**This is a required option.
- d** This option specifies the directory that contains the forms to deploy or undeploy. When you use this option, the task applies to the contents of the directory. This means that when you deploy a directory, the system deploys the entire contents of the directory; or, when you undeploy a directory, the system undeploys the entire contents of the directory.  
  
When you use this option, the **-r** option should not be on the same command line.
- h** This option specifies the host name where you want to deploy or undeploy the form.  
  
This is a required option.
- i** This option specifies the description for the form. Use double quotation marks to enclose the text for the description.  
  
The description is optional; it appears in the description field that shows when the list of tasks shows the Details view.
- l** This option specifies the deployment location that is relative to the Admin folder.  
  
If you want to deploy forms to the top-level directory, use a forward slash (/) as the value for the deployment location.
- n** This option specifies the name for the form in deploy mode. Use double quotation marks to enclose the name if it contains spaces.  
  
When you use the Object Manager, this name appears as the form name in the list of tasks.

- p** This option specifies the port number for the host name specified with the **-h** option.

You must enter this option even though the default for this option is the port number supplied during the PI installation, which is port number **80**, in most cases.

This is a required option.
- P** This option specifies the corresponding password for the username that has permission to deploy or undeploy the form.

This option is in UPPERCASE.

This is a required option.
- r** This option specifies the name of a single form you want to deploy or undeploy. Use the actual name as it appears in the directory.

In deploy mode, this value can contain the absolute or relative path that is the current location of the form. If this value does not include a path, then the form is in the current working directory.

In undeploy mode, this value must be the name of the form only; this value should not include the path. In this case, the system uses the path specified in the **-l** option.

When you use this option, the **-d** option should not be on the same command line.
- s** This option specifies that SSL is used as the communications protocol.
- U** This option specifies the username of the user that has permission to deploy or undeploy the form.

This option is in UPPERCASE.

This is a required option.

# Usage Notes

To view a form, you must first publish it by deploying it to a server for viewing. You can use this tool to deploy forms to the PI Administration Server.

You must have Administrator privileges to deploy forms to the server. For more information about forms, see the *HP Performance Insight Guide to Building and Viewing Reports*.

When you want to remove a form from the Object Manager so that you can no longer view it, you undeploy it. You can use this tool to undeploy forms from the PI Administration Server. You must have Administrator privileges to undeploy forms from the server.

## Using the formdeploytool Command

This section shows some formats of the `formdeploytool` command.

If you enter the `formdeploytool` command without any options, the system displays an error message followed by the help information.

### Deploy Forms

- To deploy a single form, enter the following command.

```
formdeploytool -U username -P password -h host_name  
-p port_num -c deploy -r form_name -l deploy_loc -i "form_desc"  
-n "display_name"
```

Where, *username* is the user that has permission to deploy the form.

*password* is the password for the corresponding user name.

*host\_name* is the name of the host where the action occurs.

*port\_num* is the port number of the host where the action occurs.

*form\_name* is the source name of the form with the absolute or relative path.

*deploy\_loc* is the target deployment location.

*form\_desc* is the description of the form.

*display\_name* is the name of the form that appears in the Object Manager.

- To deploy multiple forms in the same directory, enter the following command.

```
formdeploytool -U username -P password -h host_name  
-p port_num -c deploy -l deploy_loc -d dir_path
```

Where, *username* is the user that has permission to deploy the form.

*password* is the password for the corresponding user name.

*host\_name* is the name of the host where the action occurs.

*port\_num* is the port number of the host where the action occurs.

*deploy\_loc* is the target deployment location.

*dir\_path* is the directory that contains the deployed forms.

## Undeploy Forms

- To undeploy a single form, enter the following command.

```
formdeploytool -U username -P password -h host_name  
-p port_num -c undeploy -r form_name -l deploy_loc
```

Where, *username* is the user that has permission to undeploy the form.

*password* is the password for the corresponding user name.

*host\_name* is the name of the host where the action occurs.

*port\_num* is the port number of the host where the action occurs.

*form\_name* is the actual name of the form in the deployed directory.

*deploy\_loc* is the source location that contains the deployed form.

- To undeploy an entire directory, which removes the directory and its contents, enter the following command:

```
formdeploytool -U username -P password -h host_name  
-p port_num -c undeploy -d dir_path
```

Where, *username* is the user that has permission to undeploy the forms.

*password* is the password for the corresponding user name.

*host\_name* is the name of the host where the action occurs.

*port\_num* is the port number of the host where the action occurs.

*dir\_path* is the directory that contains the deployed forms.

# Examples

The following examples illustrate some uses of the `formdeploytool` command that an Administrator, such as the `trendadm` user, can enter.

## Example 1: Deploy a Form to the Object Manager

To deploy a form with the name `update_node.frep` from the current working directory on the **cartman** host to the `testforms/Lan` directory and then display it with a name of Update Node and a description, you can use the following command.

```
formdeploytool -h cartman -p 80 -U trendadm -P trendadm  
-c deploy -r update_node.frep -l testforms/Lan/ -i "This  
form enables you to update a node." -n "Update Node"
```

## Example 2: Undeploy a Form from the Web Access Server (UNIX)

To undeploy a form with the name `update_node.frep` from the `testforms/Lan` directory on the **cartman** host, you can use the following command.

```
formdeploytool -h cartman -p 80 -U trendadm -P trendadm  
-c undeploy -r update_node.frep -l testforms/Lan/
```

## Example 3: Deploy a Directory of Forms to a Different User

To deploy all the forms in the `d:\ovpi\forms\Interface_Reporting\Admin` directory on the **powder2** host to the `user1\testforms` directory, you can use the following command.

```
formdeploytool -h powder2 -p 80 -U user1 -P test1  
-c deploy -d d:\ovpi\forms\Interface_Reporting\Admin  
-l testforms
```

## Example 4: Deploy a Directory of Forms (UNIX)

To deploy all the forms in the `/user/forms/test` directory on the **cartman** host to the `trendadm/testforms` directory, you can use the following command.

```
formdeploytool -h cartman -p 80 -U trendadm -P trendadm  
-c deploy -d /user/forms/test -l testforms
```

### Example 5: Undeploy a Directory of Forms

To undeploy all the forms in the `user/trendadm/testforms` directory on the **cartman** host, you can use the following command.

```
formdeploytool -h cartman -p 80 -U trendadm -P trendadm  
-c undeploy -d testforms
```

## Error Messages

This section describes some of the messages that can occur from `formdeploytool`. Each message has the following format:

- A brief description about why the message appears. Each new message description starts with a check box.
- The actual message that appears with parameters for any information that may be missing.
- A suggestion about the action to do so that the message does not appear again.

- ❑ If the following message appears, the user is unauthorized to deploy the specified directory.

Directory deployment failed.

Verify the following:

- The user (**-U**) and password (**-P**) values are correct.
- The specified user has the appropriate privileges to deploy the directory.

- ❑ If the following message appears, the user is unauthorized to undeploy the specified directory.

Directory undeployment failed.

Verify the following:

- The user (**-U**) and password (**-P**) values are correct.
- The specified user has the appropriate privileges to undeploy the directory.

- ❑ If the following error message appears, there is a command-line syntax error.

Error processing command line: A value is required for argument *option* : *description*.

Verify that every option on the command line has the appropriate value. The description specifies what information is missing.

- ❑ If the following error message appears, there is a command-line syntax error.

Error processing command line: *option* is not valid for this program.

Remove the invalid option from the command line. Check [Syntax](#) on page 168 for the list of valid options.

- ❑ If the following error message appears, there is a command-line syntax error.

Error processing command line: Option *option* must be specified.

Verify that all required options are on the command line: **-c**, **-d** or **-r**, **-h**, **-p**, **-P**, **-U**.

- ❑ If the following message appears, there is an error on the command line.

Form deployment failed.

Verify that the value for each option has the correct value, such as the values for the user (**-U**) and password (**-P**) options, or that the **-r** option has the appropriate type of file specified.

- ❑ If the following message appears, the form or directory name is missing on the command line.

Form or directory name must be specified.

Verify that the option for the form name (**-r**) or the directory name (**-d**) is on the command line. One of these options must appear on the command line.

- ❑ If the following message appears, there is an error on the command line.

Form undeployment failed.

Verify that the value for each option has the correct value, such as the value for the **-r** option is the actual name of the form as it appears in the directory and not the name specified for the **-n** option.

- ❑ If the following message appears, the type of task to perform is incorrect.

Incorrect value for -c argument. Please specify "deploy" or "undeploy".

Verify that the value for the **-c** option is **deploy** or **undeploy**.

- ❑ If the following message appears, the directory name is incorrect.

Invalid directory specified: *directory\_name*.

Verify that the directory name and path are correct for the **-d** option.

- ❑ If the following message appears, the form name is incorrect.

Invalid form specified: *form\_name*.

Verify that the name and the path of the form are correct for the **-r** option.

- ❑ If the following error message appears, there is a command-line syntax error.

*option*option must be specified.

Verify that required options for a specific task are on the command line, for example **-l** to deploy a form.

- ❑ If the following message appears, there was an incorrect host name.

SEVERE: Unknown host. *host\_name*.

Verify that the spelling of the host name is correct, and that it exists in the database.

- ❑ If the following message appears, the user is unauthorized to deploy or undeploy the specified form.

Unauthorized.

Verify the following:

- The user (**-U**) and password (**-P**) values are correct.
- The specified user has the appropriate privileges to deploy or undeploy the form.



## 12 generate

You can use the `generate` command to generate the reports for a particular schedule on a PI system.

### Syntax

The `generate` command uses the following syntax:

```
generate [ -event uid_num ]  
          -host hostname  
          [-debug on | off]  
          -log log_filename  
          -pass report_password  
          -port port_num  
          [ -schedule schedule_name ]  
          -user report_username
```

# Options

The `generate` command has the following options:

- event**      Use this option to specify which report you want to run when it is from a schedule that includes multiple report entries.  
The value for this option must be the UID of the desired report.  
If you do not know the report's UID, see the discussion of the **-list** option in [schedule](#) on page 365.
- debug**      Displays the debug information on the console. The default option is "off".
- host**      Use this option to specify the name of the Web Access Server where the schedule resides to generate the reports.  
This is a required option.
- log**      Use this option to specify the name of the log file with its full path. This file logs the schedule generation information when the system generates the reports.  
This is a required option.
- pass**      Use this option to specify the corresponding password to the PI username required to access the Web Access Server.  
This is a required option.
- port**      Use this option to specify the Web Access Server port where the schedule resides to generate the reports.  
The default value is 80.  
This is a required option.
- schedule**   Use this option to specify the name of the schedule you want to generate. If the schedule's name is longer than one word, enclose it in quotation marks, for example:  
**-schedule="Executive Summary"**.
- user**      Use this option to specify the PI username required to access the Web Access Server.  
This is a required option.

# Usage Notes

You can use this command to generate reports on an as-needed basis. For example, if the scheduled reports did not run due to unexpected situations, such as a server was down, you could run the report from the Web Access Server or you could use the `generate` command to run the scheduled report from the command line.

When you run the `generate` command, the system checks the schedule for reports that need to be run, based on their triggers. For more information about schedules and defining report triggers, see [schedule](#) on page 365.

If you use the **-query** option with the `schedule` command, it is not necessary to use the `generate` command. The **-query** option causes the generator to automatically check the schedule to determine if any reports are due to be run. Similarly, if you schedule a report using the Web Access Server, you do not need to use the `generate` command.

## Using the generate Command

This section shows some formats of the `generate` command.

- If you enter the `generate` command without any options, the system displays the help information. Use the following format.

**generate**

- To generate a report, enter the following command:

```
generate -user username -pass password -host host_name  
-port port_num -schedule sched_name -log log_name  
-event report_uid
```

Where, *username* is the user name for the report.

*password* is the password for the corresponding user name.

*host\_name* is the name of the host for the Web Access Server.

*port\_num* is the port number for the Web Access Server.

*sched\_name* is the name of the schedule that contains the added entry.

*log\_name* is the name of the log file with its full path.

*report\_uid* is the identification number of the event to run.

# Examples

The following examples illustrate some uses of the `generate` command.

## Example 1

To generate all the reports in the schedule named `system`, use the following command.

```
generate -user generation -pass generation -host rover  
-port 80 -schedule system -log /tmp/generate.log
```

The schedule generation information is logged in the `generate.log` file, which has a path of `/tmp`.

## Example 2

To generate one report with the UID of `trendadm-1079052721695` in a schedule called `test_sched_1` that has multiple reports, use the following command.

```
generate -user trendadm -pass trendadm -host powder2  
-port 80 -schedule test_sched_1 -log /tmp/test_gen_1.log  
-event trendadm-1079052721695
```

Note that the schedule has three reports. You can use the `schedule` command with the `-list` option to locate the UID for the report to generate.

```
schedule -host powder2 -port 80 -user trendadm -pass  
trendadm -schedule test_sched_1 -list
```

```
trendadm-1079052857539|EventSummary.rep|Thresholds  
trendadm-1079052818507|RecentEvents.rep|Thresholds  
trendadm-1079052721695|System Performance|Admin
```

### Example 3:

This example shows how one user can generate a schedule for another user. To generate a schedule called `schedule01` in the folder of `ncanfield`, use the following command.

```
generate -user generation -pass generation -host rover  
-port 80 -schedule ncanfield\schedule01 -log  
\tmp\generate.log
```

Note that you can do this only if the **-user** value has administrator permissions.



## 13 groupctl

You can use the `groupctl` command to add, delete, or modify a single user group on a PI system. A user group is a collection or subset of user accounts that have access to the Web Access Server. You can use this command to perform the same function as the `groupimport` command ([page 195](#)), which is a bulk utility that performs the same actions as this utility except that it uses an XML formatted file for large numbers of user groups.

If you want to manage a user group from the Web Access Server with the GUI tools, see the *HP Performance Insight Administration Guide* for more information.

### Requirements and Restrictions

- You must be an administrative user to use this command.
- Each time you invoke this utility, you must enter the required options: **-group**, **-host**, **-mode**, **-port**, **-pwd**, and **-user**.
- Use the **-keepusers** option to keep existing users in the group, regardless of the type of change.

# Syntax

The `groupctl` command uses the following syntax:

```
groupctl [ -constraint column_name:operator:value ]  
          -group groupname  
          [ -help ]  
          -host hostname  
          [ -keepusers ]  
          [ -member entry_name1 [ , entry_name2, ... , entry_nameN ] ]  
          -mode type  
          -port number  
          -pwd adm_pwd  
          [ -protocol http|https ]  
          -user adm_user  
          [ -verbose ]  
          [ -version ]
```



# Options

The `groupctl` command has the following options:

**-constraint** Use this option to specify the constraint, which is the filter for the user group, to add or remove. This constraint applies to every user in the group.

The format for the entry is *column\_name:operator:value* where, *column\_name* is the name of the column to filter on in every query and *operator* is the boolean operator to apply. Valid operators are:

- `=` equal
- `<` less than
- `<=` less than or equal
- `>` greater than
- `>=` greater than or equal
- `<>` not equal
- `like`
- `not like`
- `in`
- `not in`

*value* is the value of the filter for the specified column. If the SQL type of the column is CHAR or VARCHAR, enclose the value in single quotation marks.

For example, if you want to only display the data for the records that have a customer id value of 10, use the following constraint.

```
cust_id=:10
```

The `groupctl` command accepts multiple instances of this option. If there are multiple instances of this option, the system joins the filters together.

**-group** Use this option to specify the name of the user group to act upon.

This is a required option.

**-help** Use this option to display the syntax for the command.

<b>-host</b>	<p>Use this option to specify the Web Access Server hostname where the transaction occurs.</p> <p>This is a required option.</p>
<b>-keepusers</b>	<p>Use this option to keep the existing users in a user group when you add new users to the same group.</p> <p>By default, the system removes existing users from a user group when you add new users to it.</p>
<b>-member</b>	<p>Use this option to specify the username to add or remove from the user group. You can enter multiple members with a comma-separated list of usernames for this option. Do not use any spaces in the list.</p>
<b>-mode</b>	<p>This option specifies the type of transaction to perform that affects the entire user group. For example, if you specify <b>delete</b> for this option, the system removes the user group from the catalog. Valid entries are:</p> <ul style="list-style-type: none"> <li>• <b>add</b>: Use this mode to add a new user group.</li> <li>• <b>delete</b>: Use this mode to delete an entire user group.</li> <li>• <b>modify</b>: Use this mode to modify an existing user group by adding members and filters.</li> </ul> <p>This is a required option.</p>
<b>-port</b>	<p>Use this option to specify the Web Access Server port number where the transaction occurs.</p> <p>You must enter this option even though the default for this option is the port number supplied during the PI installation, which is port number 80, in most cases.</p> <p>This is a required option.</p>
<b>-pwd</b>	<p>Use this option to specify the corresponding password for the username that has authorization to make the specified changes.</p> <p>This is a required option.</p>
<b>-protocol</b>	<p>This option specifies that https is used as the communications protocol. If this option is not specified, http is used.</p>

<b>-user</b>	Use this option to specify the username that has authorization to make the specified changes. This username must have administrative privileges. This is a required option.
<b>-verbose</b>	Use this option to turn on verbose messaging.
<b>-version</b>	Use this option to display the current version of <code>groupctl</code> .

## Usage Notes

The purpose of this command is to manage a single user group. You can control access to the Web Access Server by combining a set of user accounts that have a common interest into a user group. Note that a user group can contain other user groups.

For example, you can create a user group called *Thunderbolt* for all users in a company called Thunderbolt, Inc. This is useful if you are a service provider who wants to limit the data that certain companies can view.

In another case, you can create a group called *All*, which contains some user accounts and four groups called *North*, *East*, *South*, and *West*. The constraint placed on the users in the *North*, *East*, *South*, and *West* groups lets those users view only the data for their specific region. The users in the *All* group can view all the data that the users in the *North*, *East*, *South*, and *West* groups can view because the top-level group inherits the constraints of the groups below it in the hierarchy.

When you create groups, try to create effective constraints for the groups to limit the number of groups you might have in your system. For example, instead of using `interface` as a constraint and having 1000 or more groups for each interface, use `cust-id` as a constraint instead. A large number of groups can affect the efficiency and size of the query sent to the database.

## Modes of Operation

The `groupctl` command has three modes of operation: add, modify, and delete.

## Add

The *add* mode provides the ability to add a user or a filter to a user group that is on the Web Access Server.

## Modify

The *modify* mode provides the ability to change the members or the filter of the user group that is on the Web Access Server.

## Delete

The *delete* mode provides the ability to remove a user or a filter from a user group that is on the Web Access Server.

## Using the groupctl Command

This section shows some formats of the command for the various modes. There is a minimum of six required options for the `groupctl` command. Each mode shows the command with the required options along with the other options for the particular task; however, only the definitions for the new options appears for each subsequent command. The definitions for the required options appear below.

- All `groupctl` commands must have all the following options for each task:

```
groupctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -group groupname -mode type
```

where, *host\_name* is the name of the host for the Web Access Server.

*port\_num* is the port number for the Web Access Server.

*adm\_user* is the administrative user name that has authorization to make the specified changes.

*adm\_pwd* is the corresponding password for the administrative user that has authorization to make the specified changes.

*groupname* is the name of the user group to add, modify, or delete.

*type* is the type of action to perform, such as add, modify, or delete.

- If you enter the `groupctl` command without any options, the system displays the help information. Use the following format.

```
groupctl
```

- If you want to display the version for the `groupctl` command, enter the following command.

```
groupctl -version
```

## Add

The following formats show various options for adding user groups. Note that you can combine the additional options in any manner that meets your requirements.

- To add a user group without any members in it, enter the following command.

```
groupctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -group groupname -mode add
```

- To add a user group with a member in it, enter the following command.

```
groupctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -group groupname -mode add -member entry_name
```

*entry\_name* is the user or group name to add to the group.

- To add a user group with a filter, enter the following command.

```
groupctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -group groupname -mode add  
-constraint column_name:operator:value
```

Where, *column\_name* is the name of the column for the filter.

*operator* is the operation the system performs for the filter.

*value* is the value for the filter.

## Modify

- To modify an existing user group that does not have a filter by adding a member to it, enter the following command.

```
groupctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -group groupname -mode modify -keepusers  
-member entry_name
```

Where, *entry\_name* is the user or user group name to add to the user group.

- To modify an existing user group by adding a filter to it or changing the filter for it, enter the following command.

```
groupctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -group groupname -mode modify -keepusers  
-constraint column_name:operator:value
```

Where, *column\_name* is the name of the column for the filter.

*operator* is the operation the system performs for the filter.

*value* is the value for the filter.

- To modify an existing user group that does have a filter by adding a member to it, enter the following command.

```
groupctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -group groupname -mode modify -keepusers  
-constraint column_name:operator:value -member entry_name
```

Where, *entry\_name* is the user or user group name to add to the user group.

*column\_name* is the name of the column for the filter.

*operator* is the operation the system performs for the filter.

*value* is the value for the filter.

- To remove a filter from an existing user group and keep the members of that group, enter the following command.

```
groupctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -group groupname -mode modify -keepusers
```

- To replace the members of an existing user group that does not have a filter, enter the following command.

```
groupctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -group groupname -mode modify  
-member entry_name1, entry_name2, ... , entry_nameN
```

Where, *entry\_name* is the user or user group name to add to the user group. In this case, you can enter any combination of user or group names.

Note that if you want to remove one or more members from a user group, you have to modify the group and supply the desired list of members with the **-member** option.

## Delete

To delete a user group, enter the following command:

```
groupctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -group groupname -mode delete
```

## Examples

The following examples illustrate some uses of the `groupctl` command that an Administrator, such as the `trendadm` user, can enter.

### Example 1: Add a Group without Members

To add a group with the name **group1** on the **powder2** host without any members, you can use the following command.

```
groupctl -host powder2 -port 80 -user trendadm -pwd  
trendadm -group group1 -mode add
```

### Example 2 : Add a Group with Members

To add a group with the name **group2** on the **powder2** host with two members **user1** and **user2**, you can use the following command.

```
groupctl -host powder2 -port 80 -user trendadm -pwd  
trendadm -group group2 -mode add -member user1,user2
```

### Example 3: Add a Filter to an Existing Group

To add a filter that requires the **cust\_id** column to have the value 10 to an existing group that has the name **group1** on the **powder2** host without removing the existing members, you can use the following command.

```
groupctl -host powder2 -port 80 -user trendadm -pwd  
trendadm -group group1 -mode modify -keepusers  
-constraint cust_id=:10
```

#### Example 4: Replace Members in an Existing Group

To replace members in an existing group that has the name **group1** on the **powder2** host with members such as **user1** and **user2**, you can use the following command.

```
groupctl -host powder2 -port 80 -user trendadm -pwd  
trendadm -group group1 -mode modify -member user1,user3
```

#### Example 5: Delete a Group

To delete a group with the name **group2** from the **powder2** host, you can use the following command.

```
groupctl -host powder2 -port 80 -user trendadm -pwd  
trendadm -group group2 -mode delete
```

## Error Messages

This section describes some of the messages that can occur from `groupctl`. Each message has the following format:

- A brief description about why the message appears. Each new message description starts with a check box.
  - The actual message that appears with parameters for any information that may be missing.
  - A suggestion about the action to do so that the message does not appear again.
- ❑ If the following message appears, the user is unauthorized to create, modify, or delete a user group.

Unauthorized.

Verify the following:

- The user (**-user**) and password (**-pwd**) values are correct.
- The specified user has administrator privileges.







# 14 groupimport

You can use the `groupimport` command to add, modify, or delete Web Access Server groups on a PI system. This utility is an extension of the Web Access Server Group Accounts feature under the Administration link on the Management Console.

## Requirements or Restrictions

- There are five required command-line options that you must enter each time you invoke the utility.
- Any file to be imported must be in the Extensible Markup Language (XML) interchange format specified in this chapter.

## Syntax

The `groupimport` command uses the following syntax:

```
groupimport    -c communications protocol  
                -f group_XML_file_name  
                -h application_server_name  
                [-help]  
                -p application_server_port_number  
                -P administrator_password  
                -U administrator_username
```

# Options

The `groupimport` commands have the following options:

- c**        The communications protocol that you want to use (http or https).  
             If this option is not specified, http is used by default.
- f**        Name of the text file containing Web Access Server group  
             information.
- h**        Name of the Web Access Server.
- help**     Display the command-line options for the `groupimport` command.
- p**        Port number of the Web Access Server.
- P**        Password of the PI administrator.  
             This option is in UPPERCASE.
- U**        Username of the PI administrator.  
             This option is in UPPERCASE.

# Usage Notes

This section describes the naming convention and file format for the text file that contains the XML tag sets when you use the `groupimport` command.

## Naming Conventions

The `groupimport` text file parameters must follow the naming conventions listed below:

- 1 The tag set parameter formats can be all alphabetic characters (upper and lower case), all numeric symbols, and all special characters except those listed in step 2 below.
- 2 DO NOT use any of the following characters in the `groupimport` text file:
  - single quotation mark ( ' )
  - double quotation mark ( " )
  - ampersand ( & )
  - comma ( , )
  - space (blank)

## File Format

The `groupimport` text file uses XML tag sets to define the data to be imported into the Web Access Server. The following figure shows all XML tag sets available for use in the `groupimport` text files. [Table 13](#) on page 199 describes all XML tag sets available for use in the `groupimport` text files.

**Figure 1 groupimport Text File Format**

```
<?xml version="1.0" encoding="UTF-8"?>
<Groups>
  <Group>
    <Action>action</Action>
    <Name>group_name</Name>
    <ChildGroups>
      <Child>child_group_name</Child>
    </ChildGroups>
    <Users>
      <User>user_in_group_name</User>
    </Users>
    <FilterConstraints>
      <Constraint>
        <LeftHandSide>filter_on_column</LeftHandSide>
        <Operator>mathematic_operator</Operator>
        <RightHandSide>filter_value</RightHandSide>
      </Constraint>
    </FilterConstraints>
  </Group>
</Groups>
```



All the XML tag sets include the angle brackets (< >) as part of the tag.

**Table 13 groupimport File - XML Tag Definitions**

XML TAG SETS	DESCRIPTION
<code>&lt;?xml version="1.0"  encoding="UTF-8"? &gt;</code>	<p>Enables the parser to validate the XML format. It must appear as the first line in the file.</p>
<code>&lt;Groups&gt; &lt;/Groups&gt;</code>	<p><b>&lt;Groups&gt;</b> is the opening tag for the <b>&lt;Groups&gt;&lt;/Groups&gt;</b> tag set.</p> <p>There can be multiple <b>&lt;Group&gt;&lt;/Group&gt;</b> tag sets inside the <b>&lt;Groups&gt;&lt;/Groups&gt;</b> tag set.</p> <p>All <b>&lt;Group&gt;&lt;/Group&gt;</b> tag sets are contained within the <b>&lt;Groups&gt;&lt;/Groups&gt;</b> tag set for each group being defined.</p> <p>This tag set is required.</p>
<code>&lt;Action&gt; &lt;/Action&gt;</code>	<p>Procedure tag set that defines the operation to perform. If a group exists and the <b>Add</b> action is specified, the request is ignored.</p> <p>Valid values = <b>Add, Modify, Delete</b></p> <p>Default = <b>Add</b></p>
<code>&lt;Name&gt; &lt;/Name&gt;</code>	<p>Defines group name.</p> <p>This tag set is required.</p>
<code>&lt;ChildGroups&gt; &lt;/ChildGroups&gt;</code>	<p><b>&lt;ChildGroups&gt;</b> is the opening tag for the <b>&lt;ChildGroups&gt;&lt;/ChildGroups&gt;</b> tag set.</p> <p>There can be multiple <b>&lt;Child&gt;&lt;/Child&gt;</b> tag sets inside the <b>&lt;ChildGroups&gt;&lt;/ChildGroups&gt;</b> tag set.</p> <p>All <b>&lt;Child&gt;&lt;/Child&gt;</b> tag sets are contained within the <b>&lt;ChildGroups&gt;&lt;/ChildGroups&gt;</b> tag set for each sub-group being defined.</p> <p>This tag set is not required if there are no sub-groups.</p>

**Table 13 groupimport File - XML Tag Definitions (cont'd)**

XML TAG SETS	DESCRIPTION
<b>&lt;Child&gt;</b> <b>&lt;/Child&gt;</b>	<p><b>&lt;Child&gt;</b> is the opening tag for the <b>&lt;Child&gt;</b> <b>&lt;/Child&gt;</b> tag set. This tag set specifies the sub-group name.</p> <p>There can be multiple <b>&lt;Child&gt;</b><b>&lt;/Child&gt;</b> tag sets inside the <b>&lt;ChildGroups&gt;</b> <b>&lt;/ChildGroups&gt;</b> tag set.</p> <p>All <b>&lt;Child&gt;</b><b>&lt;/Child&gt;</b> tag sets are contained within the <b>&lt;ChildGroups&gt;</b> <b>&lt;/ChildGroups&gt;</b> tag set for each sub-group being defined.</p> <p>This tag set is not required if there are no sub-groups.</p>
<b>&lt;Users&gt;</b> <b>&lt;/Users&gt;</b>	<p><b>&lt;Users&gt;</b> is the opening tag for the <b>&lt;Users&gt;</b> <b>&lt;/Users&gt;</b> tag set.</p> <p>There can be multiple <b>&lt;User&gt;</b><b>&lt;/User&gt;</b> tag sets inside the <b>&lt;Users&gt;</b><b>&lt;/Users&gt;</b> tag set.</p> <p>All <b>&lt;User&gt;</b><b>&lt;/User&gt;</b> tag sets are contained within the <b>&lt;Users&gt;</b><b>&lt;/Users&gt;</b> tag set for each user assigned to the group.</p> <p>This tag set is not required if there are no users assigned to the group being defined.</p>



**Table 13 groupimport File - XML Tag Definitions (cont'd)**

XML TAG SETS	DESCRIPTION
<b>&lt;User&gt;</b> <b>&lt;/User&gt;</b>	<p><b>&lt;User&gt;</b> is the opening tag for the <b>&lt;User&gt;</b> <b>&lt;/User&gt;</b> tag set. This tag set specifies one user's name.</p> <p>There can be multiple <b>&lt;User&gt;</b><b>&lt;/User&gt;</b> tag sets inside the <b>&lt;Users&gt;</b><b>&lt;/Users&gt;</b> tag set.</p> <p>All <b>&lt;User&gt;</b><b>&lt;/User&gt;</b> tag sets are contained within the <b>&lt;Users&gt;</b><b>&lt;/Users&gt;</b> tag set for each user assigned to the group.</p> <p>This tag set is not required if there are no users assigned to the group being defined.</p>
<b>&lt;FilterConstraints&gt;</b> <b>&lt;/FilterConstraints&gt;</b>	<p><b>&lt;FilterConstraints&gt;</b> is the opening tag for the <b>&lt;FilterConstraints&gt;</b> <b>&lt;/FilterConstraints&gt;</b> tag set.</p> <p>There can be multiple <b>&lt;Constraint&gt;</b> tag sets inside the <b>&lt;FilterConstraints&gt;</b> tag set.</p> <p>All <b>&lt;Constraint&gt;</b><b>&lt;/Constraint&gt;</b> tag sets are contained within the <b>&lt;FilterConstraints&gt;</b> tag set for each constraint being defined.</p> <p>This tag set is not required if there are no constraints.</p>

**Table 13 groupimport File - XML Tag Definitions (cont'd)**

XML TAG SETS	DESCRIPTION
<b>&lt;Constraint&gt;</b> <b>&lt;/Constraint&gt;</b>	<p><b>&lt;Constraint&gt;</b> is the opening tag for the <b>&lt;Constraint&gt;&lt;/Constraint&gt;</b> tag set. This tag set defines the filters for the group.</p> <p>Each <b>Constraint</b> tag contains three internal tags that define the filter.</p> <ul style="list-style-type: none"> <li>• <b>&lt;LeftHandSide&gt;</b> defines the column to filter on.</li> <li>• <b>&lt;Operator&gt;</b> defines the boolean operator to apply. The valid operators are: <ul style="list-style-type: none"> <li>• <b>=, &lt;, &lt;=, &gt;, &gt;=, &lt;&gt;, like, not like</b></li> </ul> </li> <li>• <b>&lt;RightHandSide&gt;</b> defines the value of the filter. If the SQL type of the column is a CHAR or VARCHAR, enclose the value in single quotation marks.</li> </ul> <p>If there are multiple <b>Constraint</b> tags for a single group, the filters are joined together with AND.</p> <p>This tag set is not required if there are no filters.</p>

## Example

The sample `groupimport` file shown in the following figure, when called from the command line by the command listed below, adds one group:

```
groupimport -h app_server -p app_server_port -U admin_username  
-P admin_password -f full_path_and_filename
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<Groups>  
  <Group>  
    <Action>Add</Action>  
    <Name>Technical Services</Name>  
    <Users>  
      <User>wbentley</User>  
      <User>jsmith</User>  
    </Users>  
  </Group>  
</Groups>
```



# 15 group\_manager

The `group_manager` utility is designed to manage group definitions and polling policies. Note that you can use both `collection_manager` and `group_manager` to define polling policies.

If you want more information about groups, see [About Groups](#) on page 214.

## Requirements or Restrictions

- An error occurs if the **-infile** option does not appear with the **-import** option.
- An error occurs if more than one of the following options appears on the command line at the same time: **-import**, **-export**, **-export\_all**, **-export\_policy**, **-export\_policy\_all**, **-remove**, or **-remove\_policy**.
- The default database for all database options is the default database identified in the `systems.xml` file.

# Syntax

The `group_manager` command uses the following syntax:

```
group_manager      [ -backup ]  
                    [ -database db_name ]  
                    [ -datadb datadb_server ]  
                    [ -debug debug_level ]  
  
                    [ {  
                        -export category.group  
                        -export_all  
                        -export_policy policy_name  
                        -export_policy_all  
                        -import  
                        -remove category.group  
                        -remove_policy policy_name  
                    } ]  
  
                    [ -force ]  
                    [ -groupdb groupdb_server ]  
                    [ -help ]  
                    [ -infile file_name ]  
                    [ -outfile file_name ]  
                    [ -pollfrom host_name ]  
                    [ -use_default ]  
                    [ -user user_name ]  
  
                    [ {  
                        -v  
                        -version  
                    } ]
```

# Options

The `group_manager` command has the following options:

- backup** Use this option with the **-remove** or **-remove\_policy** option to back up a group definition or polling policy before PI removes it. If the group is a derived group, and you use the **-force** option, `group_manager` backs up all dependent groups before removing them.
- PI writes the backup file for the group to the `$DPIPE_HOME/lib/groups` directory; the format of the file name is `TG_group_category.xml`. It writes the backup file for the polling policy to the `$DPIPE_HOME/lib/collection_defs` directory; the format of the file name is `PP_policy_name.xml`.
- database** This option identifies the database where the changes occur. The database must appear in the list of available database servers. See the *HP Performance Insight Administration Guide* for more information about adding database servers to the list using the Web Access Server.
- The default is the database identified as the default in the database server list.
- datadb** Use this option with the **-import** option to specify the database where PI stores collected data. This is the data database; see [data database](#) on page 291 for the definition. The value associated with this option overrides the corresponding value, if any, in the input file that defines the polling policy.
- The corresponding value in `collection_manager` is the `server_name` parameter in the ASCII file. The corresponding value in Polling Policy Manager is in the Server column on the database information page.
- You can use this option only when you are importing a polling policy. The `datadb_server` must exist in the list of PI databases before the import.

<b>-debug</b>	Use this option to set the debug output level. The higher the number, the more detailed the information.
or	
<b>-d</b>	Debug output writes to standard output. Use this option only for testing in coordination with Technical Support due to the additional overhead it places on <code>group_manager</code> . The default is no debug output.
<b>-export</b>	Use this option to generate a file containing the group definition specified on the command line. You must specify the group in the format <i>category.group</i> ; see <a href="#">Naming Convention</a> on page 213 for a description of these parameters. You can use the <b>-outfile</b> option to specify the output file name and storage location; otherwise, <code>group_manager</code> writes the data to a self-generated file in the <code>\$DPIPE_HOME/lib/groups</code> directory.  You can use this option only when you are exporting group definitions; it does not include polling policies. This option cannot appear on the command line when the <b>-export_all</b> , <b>-import</b> , <b>-remove</b> , or <b>-remove_policy</b> option appears on the command line.
<b>-export_all</b>	Use this option to generate files containing all the group definitions. You can use the <b>-outfile</b> option to specify the storage location; otherwise, <code>group_manager</code> writes the data to self-generated files in the <code>\$DPIPE_HOME/lib/groups</code> directory. It generates one file for each group, and it names each file in this form: <code>TG_group_category.xml</code> ; see <a href="#">Naming Convention</a> on page 213 for a description of these parameters.  You can use this option only when you are exporting group definitions; it does not include polling policies. This option cannot appear on the command line when the <b>-export</b> , <b>-import</b> , <b>-remove</b> , or <b>-remove_policy</b> option appears on the command line.



- export\_policy** Use this option to generate a file containing the polling policy specified on the command line. You can use the **-outfile** option to specify the output file name and storage location; otherwise, `group_manager` writes the data to a self-generated file in the `$DPIPE_HOME/lib/collection_defs` directory. You can use this option only when you are exporting polling policies; it does not include group definitions. This option cannot appear on the command line when the **-export**, **-export\_all**, **-export\_policy\_all**, **-import**, **-remove**, or **-remove\_policy** option appears on the command line.
- export\_policy\_all** Use this option to generate files containing all the polling policy definitions. You can use the **-outfile** option to specify the storage location; otherwise, `group_manager` writes the data to self-generated files in the `$DPIPE_HOME/lib/collection_defs` directory. It generates one file for each group, and it names each file in this form: `PP_PollingPolicyName.xml`. You can use this option only when you are exporting polling policies; it does not include group definitions. This option cannot appear on the command line when the **-export**, **-export\_all**, **-export\_policy**, **-import**, **-remove**, or **-remove\_policy** option appears on the command line.
- force** Use this option with the **-remove** option to force the removal of all groups that depend on the group you are removing.

<b>-groupdb</b>	<p>Use this option with the <b>-import</b> option to specify the database that contains the list of nodes to be polled. This is the topology database; see <a href="#">topology database</a> on page 291 for the definition. The value associated with this option overrides the corresponding value, if any, in the polling policy.</p> <p>The corresponding value in <code>collection_manager</code> is the <code>group_server</code> parameter in the ASCII file.</p> <p>You can use this option only when you are importing a polling policy. The <code>groupdb_server</code> must exist in the list of PI databases before the import.</p>
<b>-help</b>	<p>This option is the help option, which displays the command-line options for the <code>group_manager</code> command.</p>
<b>-import</b>	<p>Use this option to import one group or polling policy at a time. It requires the <b>-infile</b> option to identify the file that contains a group definition or polling policy to import.</p> <p>This option cannot appear on the command line when the <b>-export</b>, <b>-export_all</b>, <b>-remove</b>, or <b>-remove_policy</b> option appears on the command line.</p>
<b>-infile</b>	<p>This option identifies the XML file that contains the information about a polling policy or group to import. If the file is not in the current working directory, you must specify the fully qualified path to the file. See <a href="#">Definition Files</a> on page 217 for details on setting up this file.</p> <p>This is a required option when the <b>-import</b> option appears on the command line.</p>
<b>-outfile</b>	<p>This option identifies the name of the file or directory where PI stores the exported group definition. You should use the directory name with the <b>-export_all</b> option.</p> <p>If this file already exists, <code>group_manager</code> overwrites it.</p> <p>You can use this option with the <b>-export</b> and <b>-export_all</b> options.</p>

<b>-pollfrom</b>	<p>Use this option with the <b>-import</b> option to specify the server that contains the data for the polling policy. The value associated with this option overrides the corresponding value, if any, in the polling policy.</p> <p>The corresponding value in <code>collection_manager</code> is the <code>poll_from</code> parameter in the ASCII file or the <b>-pollfrom</b> option on the command line. The corresponding value in Polling Policy Manager is the Polling Assigned to field when you create or edit a polling policy.</p> <p>You can use this option only when you are importing a polling policy. The <code>host_name</code> must exist before the import.</p>
<b>-remove</b>	<p>Use this option to remove a group definition. You must identify the group in the format <code>category.group</code> on the command line following this option; see <a href="#">Naming Convention</a> on page 213 for a description of these parameters.</p> <p>If you want to remove all groups that depend on this removed group, include the <b>-force</b> option on the command line.</p> <p>This option cannot appear on the command line when the <b>-export</b>, <b>-export_all</b>, <b>-import</b>, or <b>-remove_policy</b> option appears on the command line.</p>
<b>-remove_policy</b>	<p>Use this option to remove a polling policy from the collection catalog.</p> <p>This option cannot appear on the command line when the <b>-export</b>, <b>-export_all</b>, <b>-import</b>, or <b>-remove</b> option appears on the command line.</p>

<b>-use_default</b>	<p>When you use this option to import polling policies, <code>group_manager</code> uses the default values for the <code>&lt;DataDb&gt;</code>, <code>&lt;TopDb&gt;</code>, and <code>&lt;PollFrom&gt;</code> tags in the polling policy definition file. The default values for these tags are from the default database identified in the <code>systems.xml</code> file.</p> <p>In the <code>export_policy</code> modes, the definition files do not contain the <code>&lt;DataDb&gt;</code>, <code>&lt;TopDb&gt;</code>, and <code>&lt;PollFrom&gt;</code> values, when you use this option.</p>
<b>-user</b>	<p>Use this option to specify the owner of the imported group or polling policy. The value associated with this option overrides the corresponding value, if any, in the group definition or polling policy.</p> <p>The corresponding value in <code>collection_manager</code> is the <code>user_name</code> parameter in the ASCII file. The corresponding value in Polling Policy Manager is in the User column on the database information page.</p> <p>The default is <b>trendadm</b>.</p>
<b>-V</b> or <b>-version</b>	<p>Use this option to display the current version of the <code>group_manager</code> utility.</p> <p>You can use either an uppercase <b>-V</b> or the lowercase, spelled-out form (<b>-version</b>).</p>

# Naming Convention

The `group_manager` utility uses XML files for the import and export modes. The import files should have `.xml` as the suffix. The export files have `.xml` as the suffix for each file unless you specify an existing file name that has a different suffix. The following rules apply for export mode.

- If you specify a file name with `.xml` as the suffix, `group_manager` creates a file by that name.
- If you specify a file for output that already exists, `group_manager` overwrites that file.
- If you do not use `.xml` as the suffix and the file does not already exist, `group_manager` treats the entire name as a directory name. Within that directory, it creates one or more files that use the following naming convention: `TG_group_category.xml`. The descriptions of the *group* and *category* parameters follow.

The *group* portion of the name is the name of the group that contains the list of objects to collect. It may appear in the **Select Group to Poll From** pull-down list in Polling Policy Manager.

The *category* portion of the name identifies the kind of group for the corresponding *group*. If the group is a **type** list, then the *category* is **type**. Similarly, if the group is a view list, then the *category* is **view**. If the group is a single node group, then the *category* is **node**. Otherwise, the *category* is the same as the property table name.

The following table shows the typical association for the *category* value to the corresponding kind of group that may appear in the Collect Data From field in Polling Policy Manager.

**Table 14 Typical Values for the Group Category Parameter**

Category Value	Value in Collect Data From Field
<b>type</b>	All Nodes of the Same Type A Combination of Type and View

**Table 14 Typical Values for the Group Category Parameter**

Category Value	Value in Collect Data From Field
<b>view</b>	All Nodes in Same View
<b>node</b>	A Single Node
<i>property_table_name</i>	Specific Instances Custom Groups

These group and category definitions apply to the format of the group identity used with the **-export** or **-remove** option, which is *category.group*.

See the *HP Performance Insight Administration Guide* for more information about Polling Policy Manager.

## Usage Notes

This section provides a brief description of groups in general and the available modes of operation. It also explains how to create an XML file to import a polling policy or a group and how to run the `group_manager` command.

### About Groups

A PI group represents a set of managed objects. A managed object can be a variety of things, such as a router, a customer, or a location. Each managed object is persisted in a PI database as a row in a property table. A property table is a set of managed objects that are similar to one another.

You can use `group_manager` to define and manage three types of groups:

Derived	A group that contains other, logically related groups. Currently, in PI a derived group can contain a maximum of two other groups.
Enumerated-list	A group that contains a list of distinct objects.
Rule-based	A group whose membership is determined by a set of associated rules. The membership may change at various points in time with each evaluation of the rules. Currently, in PI a rule-based group can evaluate a maximum combination of two rules.

Every group combines with a group category to create a unique identity for the group. A *group category* is a set of objects that corresponds to an existing property table. A group is a subset of the objects in the group category that could contain any number of objects in the set. The membership in the group depends on the type of group: derived, enumerated list, or rule-based.

Most group categories use the name of the property table for identification. However, there are three well-known group categories: type, view, and node. They pertain to the property table that contains the entire list of nodes. Any group associated with these group categories is an enumerated-list type of group.

## Modes of Operation

The `group_manager` command has three modes of operation: import, export, and remove.

### Import

The *import* mode enables you to import a group definition or polling policy using an XML file. See [Definition Files](#) on page 217 for more information about the file.

You must use the **-infile** option with the **-import** option. You may use the following options when you import a group definition or a polling policy.

- database** Specifies the server where PI stores the imported group definition or polling policy.
- user** Specifies the owner of the imported group or polling policy.

You may use these options only when you import a polling policy.

- datadb** Specifies the database where PI stores collected polling data.
- groupdb** Specifies the database that contains the list of nodes to be polled.
- pollfrom** Identifies the computer that performs the polling specified in a polling policy.
- use\_default** Assigns the default values to the fields associated with the **-datadb**, **-groupdb**, and **-pollfrom** options from the default database identified in the `systems.xml` file.

## Export

The *export* mode provides the ability to create one or more files containing existing group definitions or polling policies. Four options are available:

**-export**, **-export\_all**, **-export\_policy**, and **-export\_policy\_all**.

You can use the following options to export group definitions:

- The **-export** option enables you to export a single group definition to a file by specifying the group on the command line.
- The **-export\_all** option enables you to export all existing group definitions to individual files in a directory that you can specify.

You can use the following options to export polling policies:

- The **-export\_policy** option enables you to export a single polling policy to a file by specifying the polling policy on the command line.
- The **-export\_policy\_all** option enables you to export all existing polling policies to individual files in a directory that you can specify.



You may use this option when you export a polling policy.

**-use\_default** Excludes the values from the fields associated with the **-datadb**, **-groupdb**, and **-pollfrom** options in the polling policy.

You may use the following option when you export a group definition or a polling policy.

**-database** Specifies the server that contains the group definitions or polling policies.

## Remove

The *remove* mode enables you to delete group definitions and polling policies. Two options are available: **-remove** and **-remove\_policy**.

- The **-remove** option enables you to remove a group definition from the system.
- The **-remove\_policy** option enables you to remove a polling policy from the system.

You may use any of the following options in remove mode.

**-backup** Backs up the group definition or polling policy before removing it from the system.

**-database** Specifies the server that has the group definition or polling policy that PI removes.

You may only use the following option when you remove a group definition.

**-force** Removes all groups that depend on the group designated for removal. This option is for group definitions only.

## Definition Files

To import a group definition or polling policy with *group\_manager*, you create an XML file containing a Document Type Definition (DTD) that defines the group or polling policy. When *group\_manager* exports a group definition or polling policy, it exports the DTD to the output file. For guidance in programming in XML, see your XML documentation.

## Document Type Definition for Group Definitions

The following DTD is the set of rules that `group_manager` uses to create groups.

```
<!?xml version='1.0'?>
<!-- DTD DEFINITION -->
<!DOCTYPE GROUP [
<!-- document type definition for group definitions -->
<!-- A group definition document contains element GROUP -->
<!-- that defines the group -->
<!ELEMENT GROUP ( GroupName, GroupCategory, description?,
GroupOwner?, GroupType )>
<!ELEMENT GroupName (#PCDATA)>
<!ELEMENT GroupCategory (#PCDATA)>
<!ELEMENT GroupOwner (#PCDATA)?>
<!ELEMENT description (#PCDATA)?>

<!-- groups can be any of the following types -->
<!ELEMENT GroupType (derived|enumerated|rule)>
<!-- Definition of various group types -->
<!-- Definition for the Derived type -->
<!-- Currently a derived group can have only 2 groups; -->
<!-- one of them must be based on ksi_managed_node -->
<!ELEMENT derived (MemberGroup, MemberGroup)>
<!ELEMENT <MemberGroup (GROUP) >
<!ATTLIST MemberGroup logop (enter|and|or|not) #REQUIRED>

<!--***** end of Derived definition *****
-->

<!-- Definition for the Enumerated List type -->
<!-- Note: For now, we assume that the data will not be
loaded;
<!-- that is, the MBR tables will stay empty -->
<!ELEMENT enumerated EMPTY >

<!--***** end of Enumerated definition ***** -->

<!-- Definition for the Rule type -->
<!ELEMENT rule (column|(column, column))>
<!ELEMENT column EMPTY>
```

```

<!ATTLIST column name CDATA #REQUIRED>
<!ATTLIST column op (eq|lt|le|gt|ge|ne|like|not_like)
#REQUIRED>
<!ATTLIST column logop (enter|and|or|not) #REQUIRED>
<!ATTLIST column value CDATA #REQUIRED>

]

<!-- END OF THE DTD -->

```

## Document Type Definition for Polling Policies

The following DTD is the set of rules that **group\_manager** uses to create polling policies.

```

<?xml version='1.0'?>
<!-- DTD DEFINITION -->
<!DOCTYPE PollDefinition [
<!ELEMENT PollDefinition (Name, Interval, Table,
PollForGroup, PollFrom?, TopDB?, DataDB?, description?) >
<!ATTLIST PollDefinition datapipe CDATA #REQUIRED>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Interval (#PCDATA)>

<!-- Note: Table name must be SQL name, that is, real table
-->
<!ELEMENT Table (#PCDATA)>

<!ELEMENT PollForGroup EMPTY>
<!ATTLIST PollForGroup groupName CDATA #REQUIRED>
<!ATTLIST PollForGroup groupCategory CDATA #REQUIRED>
<!ELEMENT PollFrom (#PCDATA)>
<!ELEMENT TopDb (#PCDATA)>
<!ELEMENT DataDb (#PCDATA)>
<!ELEMENT description (#PCDATA)>

]>
<!-- END OF THE DTD -->

```

## Using the group\_manager Command

This section shows some formats of the command for the various modes.

- If you enter the `group_manager` command without any options, the system displays an error message followed by the help information.
- To display the syntax and options for this command, enter:

```
group_manager -help
```

- To display the version information for this command, enter:

```
group_manager -V
```

or

```
group_manager -version
```

- To import a group definition or a polling policy, enter the following command:

```
group_manager -import -infile file_name
```

Where, *file\_name* is the name of the file that contains the group definition or polling policy to import. If you do not specify a path to the file, `group_manager` expects to find the file in the current working directory.

- To export a group definition to a file with a system-generated name in the `$DPIPE_HOME/lib/groups` directory, enter the following command:

```
group_manager -export category.group
```

Where, *category* identifies the kind of group to export for the corresponding *group* name. See [Naming Convention](#) on page 213 for a description of this parameter.

*group* is the name of the group definition. See [Naming Convention](#) on page 213 for a description of this parameter.

The format of the file name is `TG_group_category.xml`. See [Naming Convention](#) on page 213 for more information.

- To export a group definition to a specified file or directory, enter the following command:

```
group_manager -export category.group -outfile file_name
```

Where, *category* identifies the kind of group to export for the corresponding *group* name. See [Naming Convention](#) on page 213 for a description of this parameter.

*group* is the name of the group definition. See [Naming Convention](#) on page 213 for a description of this parameter.

*file\_name* is the name of the file or directory that contains the exported group definition. If this is a file name, it must already exist. If this is a directory name, the format of the file name is `TG_group_category.xml`. See [Naming Convention](#) on page 213 for more information.

- To export all group definitions to individual files in the `$DPIPE_HOME/lib/groups` directory, enter the following command:

```
group_manager -export_all
```

The format of each file name in the directory is `TG_group_category.xml`. See [Naming Convention](#) on page 213 for more information.

- To export all group definitions to individual files in a specific directory, enter the following command:

```
group_manager -export_all -outfile dir_name
```

Where, *dir\_name* is the name of the directory that contains the exported group definitions. If this name already exists as a file, `group_manager` overwrites this file with each group definition and consequently end up with only the last group definition.

The format of each file name in the directory is `TG_group_category.xml`. See [Naming Convention](#) on page 213 for more information.

- To export a polling policy to a file with a system-generated name in the `$DPIPE_HOME/lib/collection_defs` directory, enter the following command:

```
group_manager -export_policy policy_name
```

Where, *policy\_name* identifies the polling policy to export.

The format of the file name is `PP_policy_name.xml`.

- To export all polling policies to individual files in the `$DPIPE_HOME/lib/collection_defs` directory, enter the following command:

```
group_manager -export_policy_ all
```

The format of each file name in the directory is `PP_policy_name.xml`.

- To remove a group, enter the following command:

```
group_manager -remove category.group
```

Where, *category* identifies the kind of group to remove from the system for the corresponding *group* name. See [Naming Convention](#) on page 213 for a description of this parameter.

*group* is the name of the group definition. See [Naming Convention](#) on page 213 for a description of this parameter.

- To remove a polling policy, enter the following command:

```
group_manager -remove_policy policy_name
```

Where, *policy\_name* is the name of the polling policy that `group_manager` removes from the system.

- To back up a polling policy before removing it, enter the following command:

```
group_manager -remove_policy policy_name -backup
```

Where, *policy\_name* is the name of the polling policy that `group_manager` removes from the system.

The backup file is in the `$DPIPE_HOME/lib/collection_defs` directory. The format of the file name is `PP_policy_name.xml`.

# Examples

This section has examples for each mode of `group_manager`.



Whenever you enter a file name you must specify the complete path; otherwise, `group_manager` refers to the current directory.

## Import Examples

### Example 1

To import group definition from a file named `groups_in.xml`, enter the following command:

```
group_manager -import -infile groups_in.xml
```

The following is an example of the contents of `groups_in.xml`.

```
<?xml version="1.0" ?>
<!-- Group Definition -->

<!-- this is enumerated group -->

GROUP>
  <GroupName> tp_lst1 </GroupName>
  <GroupCategory> type </GroupCategory>
  <GroupType>
    <enumerated/>
  </GroupType>
</GROUP>
```

### Example 2

To import a polling policy from a file named `policy_in.xml`, enter the following command:

```
group_manager -import -infile policy_in.xml
```

The following is an example of the contents of `policy_in.xml`.

```
<!-- Polling Policy Definition -->

<!-- this is PollingPolicy -->
```

```

<PollDefinition datapipe ="testtime">
  <Name> test_policy_gm </Name>
  <Interval> 20 </Interval>
  <Table> testtime_ </Table>
  <PollForGroup  groupName="tp_1st1" groupCategory="type"/
>
  <PollFrom> powder2 </PollFrom>
  <TopDb> POWDER2 </TopDb>
  <DataDb> POWDER2 </DataDb>
  <description> This is a test policy. </description>
</PollDefinition>

```

## Export Examples

### Example 1

To generate a file named `export_group_1.xml`, containing a single group definition, **super.group1**, enter the following command:

```
group_manager -export super.group1 -outfile
export_group_1.xml
```

If you do not specify a full directory path for the storage of the file, `group_manager` saves the file in `$DPIPE_HOME/lib/groups` directory.

### Example 2

To generate one export file for each group in the system and place the files in a directory named `export_groups`, enter the following command:

```
group_manager -export_all -outfile export_groups
```

The `group_manager` utility assigns a unique name to each file, using this format: `TG_group_category.xml`. See [Naming Convention](#) on page 213 for a description of the parameters in the name. For example, if a group with a **type** category has the name `Frame_Relay_Interfaces`, then the name for that exported file is `export_groups/TG_Frame_Relay_Interfaces_type.xml`.



## Remove Examples

### Example 1

If you want to remove a group definition named **super.group2** from the system, enter the following command:

```
group_manager -remove super.group2
```

### Example 2

If **super.group2** is a member of a derived group, and you want to remove it and all its dependent groups, enter the following command:

```
group_manager -remove super.group2 -force
```

### Example 3

If you want to create backups of **super.group2** and its dependent groups before removing them from the system, enter the following command:

```
group_manager -remove super.group2 -force -backup
```

In this case, `group_manager` creates one backup file for each group definition in the `$DPIPE_HOME/lib/groups` directory; each file name has the format `TG_group_category.xml`. See [Naming Convention](#) on page 213 for a description of the parameters in the name.

## Remove\_Policy Examples

### Example 1

If you want to remove a single polling policy named `test_policy_2`, enter the following command:

```
group_manager -remove_policy test_policy_2
```

### Example 2

You can back up the polling policy before removing it with the following command:

```
group_manager -remove_policy test_policy_2 -backup
```

In this case, `group_manager` creates a backup file for the polling policy in the `$DPIPE_HOME/lib/collection_defs` directory. The format of the backup file name is `PP_policy_name.xml`. In this example, the backup file name is `$DPIPE_HOME/lib/collection_defs/PP_test_policy_2.xml`.

## Error Messages

This section describes some of the messages that can occur from `group_manager`. Each message has the following format:

- A brief description about why the message appears. Each new message description starts with a check box.
- The actual message that appears with parameters for any information that may be missing.
- A suggestion about the action to do so that the message does not appear again.

## Syntax Messages

- ❑ If the following error message appears, there is a command-line syntax error. This means that a required mode option is missing.

```
unknown operation 0
```

Verify that the command line has one of the following options after the **group\_manager** command: **-import**, **-export**, **-export\_all**, **-remove**, or **-remove\_policy**.

- ❑ If the following error message appears, the **-import** option is on the command line and the **-infile** option is missing.

```
Object definition file name must be specified.
```

Verify that the **-infile** option is on the command line with the **-import** option.

- ❑ If the following error message appears, there is an option with incorrect syntax.

command line error:[argument *value* is invalid].

Verify that the spelling and the syntax for each option on the command line is correct. For example, verify that you use **-infile** or **-outfile** instead of **-file** with the appropriate option.

- ❑ If the following error message appears, an option is missing its corresponding value.

command line error:[argument *option* missing value].

Verify that any of the following options on the command line has the appropriate corresponding value: **-database**, **-datadb**, **-export**, **-groupdb**, **-infile**, **-outfile**, **-pollfrom**, **-remove**, **-remove\_policy**, **-user**. See [Syntax](#) on page 206.

- ❑ If the following error message appears, the input file name specified in **-infile** option does not exist.

input file *file\_name* is not a file.

Verify that the **-infile** option is on the command line with the **-import** option.

## Value Messages

- ❑ If the following error message appears, the file name specified on the command line in the **-infile** *file\_name* option does not exist.

Input file *file\_name* doesn't exist.

Verify the spelling of the file name or that the file exists in the specified location. You may have to supply a fully qualified path with the file name.

- ❑ If the following error message appears, the specified value on the command line does not exist in the specified table in the database, and the error comes from the specified stored procedure.

*value* *value* in the table *table\_name* does not exist.  
(in stored procedure *stored\_procedure\_name*), ErrorCode:  
*number*

Verify that the option has the correct value on the command line. This message may also occur if the corresponding polling policy is in the table that contains the collection instructions, incorrectly.

- ❑ If the following error message appears, the database specified with the **-database** option on the command line does not exist.

```
Failed to connect to database_name database.  
Connection URL not found.
```

Verify the spelling of the database name. If the spelling is correct, you can add the database using the Web Access Server.

- ❑ If the following error message appears, the specified group name is not in the database when you use the **-export** or **-remove** option.

```
Failed processing TREND object.  
[failed getting group definition from DB for group  
name=group_name group category=category].  
The group group_name does not exist.
```

Verify the spelling of the group name and the category name. This message may also occur if you enter an incorrect policy name in the format *xxx.yyy* for the **-remove\_policy** option.

- ❑ If the following error message appears, the policy name specified with the **-remove\_policy** option does not exist.

```
Failed processing TREND object.  
[failed exporting polling policy [policy_name]].  
The collection policy_name does not exist.
```

Verify the spelling of the policy name. You can verify that the polling policy exists by checking Polling Policy Manager or `collection_manager`.

- ❑ If the following error message appears, a value on the command line is incorrect that prevented `group_manager` from creating the polling policy.

```
Failed processing TREND object.  
[failed creating polling policy (trndbexp) pollPolicy:  
[policy_name] datapipe {datapipe_name} dataDB {datadb_value}  
topDB {groupdb_value} table {table_name} interval  
{interval_value} group {group: category=(category_name)  
name=(group_name)} pollFrom {pollfrom_value} user  
{user_value} ].
```

Verify the values for the options on the command line.

- If the following error message appears, the values in the XML file may be incorrect.

```
unable to create TrendObject.  
pollPolicy: [policy_name] datapipe {datapipe_name} dataDB  
{datadb_value} topDB {groupdb_value} table {table_name}  
interval {interval_value} group {group:  
category=(category_name) name=(group_name)} pollFrom  
{pollfrom_value} user {user_value}.  
specified group doesn't exist
```

Verify that the values in the XML file are correct. In particular, check the values for the group category and group name. One of these values may not exist or the combination of these values is incorrect.



## 16 install.pkg

The `install.pkg` file provides Package Manager with the basic information needed to install a report pack. For more information about using Package Manager to install report packs, see the *HP Performance Insight Administration Guide*.

This chapter explains the following:

- The report pack directory structure.
- How Package Manager processes the `install.pkg` file.
- The directives that the `install.pkg` file uses.

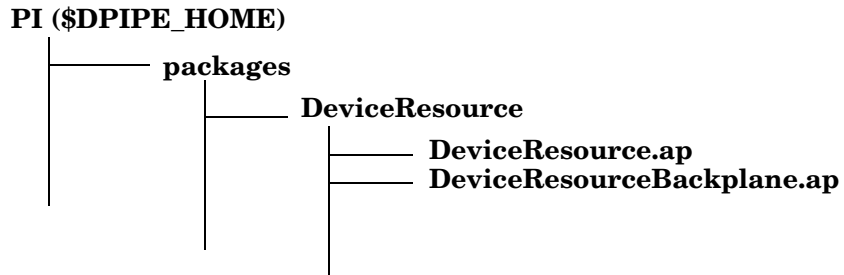
### Requirements and Restrictions

- The file name must be `install.pkg`.
- Do not use spaces between the directive and the value.
- The **version** directive must appear with the **report\_pack** directive.

# Report Pack Directory Structure

The PI installation program gives each report pack its own folder in the `$DPIPE_HOME/packages` directory. `$DPIPE_HOME` is the directory into which you installed PI. The `packages` directory contains a folder for each report pack and separate folders for the report pack's demo and upgrade versions. Each report pack's folder name must have the extension `.ap`. For example, the name of the folder for the DeviceResource report pack is `DeviceResource.ap`.

The following figure shows the directory structure for the DeviceResource report pack.





# Install.pkg Processing

When you use Package Manager to install selected report packs, Package Manager performs the following:

- Processes the `install.pkg` file
- Deploys the reports to the selected Web Access Server

After Package Manager has installed all of the selected reports, it runs Type Discovery only if you had selected that option during the install wizard setup.

## Install.pkg Directives

The following sections describe the `install.pkg` directives and include examples of their use.

Package Manager processes the directives in the order they appear in the `install.pkg` file.

- The `report_pack` and `version` directives should always be the first entries in the file.
- Any `dependency`, `software_version`, and `db_requirement` directives should follow the `report_pack` and `version` directives.
- If directive B depends on directive A, remember to put directive A before directive B in the file.

## Install.pkg Syntax Rules

The following rules apply to all `install.pkg` files.

- The comment character is `#`. Any line of an `install.pkg` file with this character in the first position on the line is a comment.
- The `install.pkg` file does not ignore spaces on either side of the colon. Do not put any spaces between the directive and its value.
- All `install.pkg` directives use the following format:

*directive:value*

## Document Conventions for Directives

This section lists the conventions used in this chapter for defining each `install.pkg` directive.

- The description for each directive follows, with each directive starting on a new page in alphabetical order.
- Each directive has a syntax description that uses the following rules:
  - **Bold** items in Courier represent keywords that, if used, must be entered as shown.
  - *Italic* items represent parameters for which the user assigns a value.
  - If items appear in braces (`{}`), one of the items must be selected.
  - Items that appear in brackets (`[ ]`) are optional.
- If an item is positional, it must appear in the order shown. If a positional item is missing, it must have a placeholder to mark its position. The placeholder is typically a comma (`,`). For example, if a directive has 3 positional parameters and the second parameter is missing, the format for the directive is the following:

**directive1=param1,,param3**

If directive has all of the parameters, the format for the directive is the following:

**directive1=param1,param2,param3**

## bin

This directive instructs Package Manager to copy the given file from the directory path that contains the `install.pkg` file to the `$DPIPE_HOME/bin` directory.

### Syntax

**bin:***file\_name*

*file\_name* is the name of the file to copy and register.

## Example

To copy a Perl script, `myperl.pl`, to the `bin` directory, add the following directive to the `install.pkg` file:

```
bin:myperl.pl
```

## Usage Notes

Package Manager registers the file as a component of the report pack with a type of TREND module.

When the report pack is uninstalled, Package Manager removes the file from the `$DPIPE_HOME/bin` directory.

The `install.pkg` file may have multiple **bin** directives in it.

## database\_procedure

This directive instructs Package Manager to register the specified procedure.

## Syntax

```
database_procedure:procedure_name
```

*procedure\_name* is the SQL name of the procedure to register in the database.

## Example

If a procedure, `myproc`, is to be used by the report pack, add the following directive to the `install.pkg` file:

```
database_procedure:myproc
```

## Usage Notes

Package Manager registers the specified procedure as a “TREND database procedure”.

Note that this directive instructs Package Manager to make a registry entry only. The system actually creates the table when Package Manager executes the **run\_command** directive with the `datapipe_manager` command. For more information, see [run\\_command](#) on page 250.

When the report pack is uninstalled, Package Manager removes the procedure by executing `datapipe_manager` in **delete** mode.

The `install.pkg` file may have multiple `database_procedure` directives in it.

## database\_table

This directive instructs Package Manager to register the specified table.

### Syntax

**database\_table:***table\_name*

*table\_name* is the SQL name of the table to register in the database.

### Example

If a table, `mycollection`, is to be used by the report pack, add the following directive to the `install.pkg` file:

```
database_table:mycollection
```

### Usage Notes

Package Manager registers the specified table as a TREND database table.

Note that this directive instructs Package Manager to make a registry entry only. The system actually creates the table when Package Manager executes the **run\_command** directive with the `datapipe_manager` command. For more information, see [run\\_command](#) on page 250.

When the report pack is uninstalled, Package Manager removes the table by executing `datapipe_manager` in **delete** mode.

The `install.pkg` file may have multiple `database_table` directives in it.

## database\_view

This directive instructs Package Manager to register the specified view.

### Syntax

**database\_view:***view\_name*

*view\_name* is the SQL name of the view to register in the database.

### Example

If a view, myview, is to be used by the report pack, add the following directive to the `install.pkg` file:

**database\_view:myview**

### Usage Notes

Package Manager registers the specified view as a “TREND database view”.

Note that this directive instructs Package Manager to make a registry entry only. The system actually creates the table when Package Manager executes the `run_command` directive with the `datapipe_manager` command. For more information, see [run\\_command](#) on page 250.

When the report pack is uninstalled, Package Manager removes the table by executing `datapipe_manager` in **delete** mode.

The `install.pkg` file may have multiple **database\_view** directives in it.

## db\_requirement

This directive specifies the type of database the report pack requires.

### Syntax

**db\_requirement:***db\_type*

*db\_type* is **Sybase**, **Oracle**, or **All**.

## Example

If the report pack requires an Oracle database to run, add the following directive to the `install.pkg` file:

```
db_requirement:Oracle
```

## Usage Notes

The **All** option indicates that the report pack can only run on an Oracle or Sybase database.

The **Oracle** option indicates that the report pack can only run on an Oracle database.

The **Sybase** option indicates that the report pack can only run on a Sybase database.

The check for the current database and the directive values occur in the Report Pack Selection Wizard window of the Package Manager application. If there is a discrepancy between the directive value of the report pack and the current database, Package Manager does not include that report pack in the selection list for possible installation.

## dependency

This directive specifies the name and version of a report pack that the current report pack depends on.

## Syntax

```
dependency:rptpack_name, version
```

Where, *rptpack\_name* is the name of the report pack required for dependency.

*version* is the version of the report pack required for dependency. The format is either *major.minor* or *major* where *major* is an integer value that represents a major release and *minor* is an integer value that represents a minor release within a major release.

## Example

If the report pack defined in the `install.pkg` file is dependent upon another report pack, `Interface_Reporting`, version 2.0, add the following directive to the `install.pkg` file.

```
dependency:Interface_Reporting,2.0
```

## Usage Notes

You can use an asterisk (\*) as the *major* or *minor* value for the *version* parameter. When used as a *major* value, its value is **1**. When used as a *minor* value, its value is **0**.

For example, the version for the following directive is 1.0.

```
dependency:myReportPack,*.*
```

In this example, the current report pack depends on a report pack named `myReportPack` that has a version of 1.0.

The report pack specified in this directive must be installed before you use the directive, so that Package Manager can install the current report pack, which is specified in the **report\_pack** directive. Package Manager pre-selects the dependencies and install them first, if they are not already installed. Furthermore, when Package Manager installs the current report pack, it records the dependency in the report pack registry. See [report\\_pack](#) on page 248 for more information about specifying the current report pack.

When you select a report pack to uninstall, Package Manager pre-selects any report pack that is dependent on the selected report pack.

The `install.pkg` file can have multiple dependency directives in it.

## Legacy Format

The system supports the old format of this directive, which is as follows:

```
dependency:rptpack_name
```

*rptpack\_name* is the name of the report pack required for dependency.

When you use this format of the directive, a version directive must follow it. See the directive [version](#) on page 257 for details. Do not include the version directive if you are using the other format for this directive.

For example, if the report pack defined in the `install.pkg` file is dependent upon another report pack, `myLanWan` at version 1.5, add the following directives to the `install.pkg` file.

```
dependency:myLanWan
```

```
version:1.5
```



Note that the `install.pkg` file cannot include both formats of the directive, which is with (new) and without (old) the report pack version. Use only one format for all **dependency** directives in each `install.pkg` file.

## dll

This directive instructs Package Manager to copy the given file from the directory path that contains the `install.pkg` file to the `$DPIPE_HOME/dll` directory.

### Syntax

```
dll:file_name
```

*file\_name* is the name of the `dll` file to copy and register.

### Example

If the report pack requires a `dll` file, `new.dll`, add the following directive to the `install.pkg` file:

```
dll:new.dll
```

### Usage Notes

Package Manager registers the file as a component of the report pack with a type of `TREND dll`.

When the report pack is uninstalled, Package Manager removes the file from the `$DPIPE_HOME/dll` directory.

The `install.pkg` file may have multiple **dll** directives in it.



## docdir

This directive instructs Package Manager to copy the documents from the specified directory, which is in the `$DPIPE_HOME` directory, to a newly created directory for the report pack in the `$DPIPE_HOME/docs` directory.

### Syntax

**docdir:***doc\_path*

*doc\_path* is the location where the source documents are stored.

### Example

If you want to add the contents of a directory named `$DPIPE_HOME/current_files` that contains the user documents for the `Interface_Reporting` report pack to the `$DPIPE_HOME/docs` directory, add the following directives to the `install.pkg` file:

**report\_pack:Interface\_Reporting**

**docdir:current\_files**

In this example, Package Manager creates the directory, `$DPIPE_HOME/docs/Interface_Reporting`, and then copies the contents of the directory `$DPIPE_HOME/current_files/*` to it.

### Usage Notes

Package Manager prepends `$DPIPE_HOME/` to the directory path specified in the `docdir` directive to locate the source documents. It then uses the value from the `report_pack` directive and creates a new directory for it under the `$DPIPE_HOME/docs` directory. See [report\\_pack](#) on page 248 for more information.

Package Manager registers the directory as a component of the report pack with a type of TREND documentation directory.

When a report pack is uninstalled, Package Manager removes the corresponding directory from the `$DPIPE_HOME/docs` directory.

The `install.pkg` file can have multiple `docdir` directives in it. Note, however, that if there are files in the source directories with the same name and path, the files in the last directive overwrites the files from the previous directives.

## docs

This directive instructs Package Manager to copy the given file from the directory path that contains the `install.pkg` file to the `$DPIPE_HOME/docs` directory.

### Syntax

**docs:***doc\_name*

*doc\_name* is the name of the document file to copy and register.

### Example

If a user document, `User_Guide.doc`, is part of the report pack, add the following directive to the `install.pkg` file:

**docs:User\_Guide.doc**

### Usage Notes

Package Manager registers the file as a component of the report pack with a type of TREND documentation.

When a report pack is uninstalled, Package Manager removes the corresponding files from the `$DPIPE_HOME/docs` directory.

The `install.pkg` file may have multiple `docs` directives in it.

## form

This directive instructs Package Manager to deploy the given file from the directory path that contains the `install.pkg` file to the Web Access Server located on the localhost system.

## Syntax

**form:***file\_name*

*file\_name* is the name of the form file to deploy and register.

## Example

If the package contains a form file, `myform.frep`, add the following directive to the `install.pkg` file:

**form:myform.frep**

## Usage Notes

This directive applies only when the user of the report pack selects the Deploy Reports option while installing the report pack with Package Manager.

Package Manager registers the file as a component of the report pack with a type of TREND form.

When the report pack is uninstalled, Package Manager undeploys the file from the Web Access Server located on the localhost system. This occurs only when the user of the report pack selects the Undeploy Reports option while uninstalling the report pack with Package Manager.

The `install.pkg` file may have multiple form directives in it.

## formdir

This directive instructs Package Manager to deploy the forms from the specified directory to the Web Access Server located on the localhost system.

## Syntax

**formdir:***forms\_dir*

*forms\_dir* is the location where the source forms are stored and the name of the newly created directory in the `$DPIPE_HOME/forms/deploy/admin` directory.

## Example

If you want to add the `Interface_Reporting_Forms` directory that contains the forms to deploy to the `$DPIPE_HOME/forms/deploy/admin` directory, add the following directive to the `install.pkg` file:

```
formdir:Interface_Reporting_Forms
```

## Usage Notes

This directive applies only when the user of the report pack selects the Deploy Reports option while installing the report pack with Package Manager.

Package Manager registers the directory as a component of the report pack with a type of TREND form directory.

When the report pack is uninstalled, Package Manager undeploys the directory from the Web Access Server located on the localhost system. This occurs only when the user of the report pack selects the **Undeploy Reports** option while uninstalling the report pack with Package Manager.

## lib

This directive instructs Package Manager to copy the given file from the directory path that contains the `install.pkg` file to the `$DPIPE_HOME/lib` directory.

## Syntax

```
lib:file_name
```

*file\_name* is the name of the library file to copy and register.

## Example

If the package contains a TEEL file, `my_table.teel`, add the following directive to the `install.pkg` file:

```
lib:my_table.teel
```

## Usage Notes

Package Manager registers the file as a component of the report pack with a type of TREND library file.

When the report pack is uninstalled, Package Manager removes the file from the `$DPIPE_HOME/lib` directory.

The `install.pkg` file may have multiple **lib** directives in it.

## mibs

This directive instructs Package Manager to copy the given file from the directory path that contains the `install.pkg` file to the `$DPIPE_HOME/mibs` directory.

## Syntax

**mibs:***MIB\_file\_name*

*MIB\_file\_name* is the name of the MIB file to copy and register.

## Example

If the report pack contains a MIB file, `myMib.mib`, add the following directive to the `install.pkg` file:

**mibs:myMib.mib**

## Usage Notes

Package Manager registers the file as a component of the report pack with a type of TREND mibs.

When the report pack is uninstalled, Package Manager removes the file from the `$DPIPE_HOME/mibs` directory.

The `install.pkg` file may have multiple **mib** directives in it.

## mw\_collection\_def

This directive instructs Package Manager to install a collection policy.

## Syntax

```
mw_collection_def:mib_path=table_category,poll_interval=frequency,table_name=collection_table,[collection_name=collection_name,]  
read_comm=public,username=trendadm,device_type=type_list_name,  
hostname=LOCAL
```

Where, *table\_category* should match the category for the table being collected.  
*frequency* is 5, 10, 15, 20, 60, or 1440.

*collection\_table* is the alias name of the table. Note that this table must already exist.

*collection\_name* is the name of the collection policy. This is an optional parameter; if you do not enter it, the default collection policy name is used.

*type\_list\_name* is the name of the enumerated group for the category of type.

## Example

If the package requires a collection definition for a table called mycollection, with a type list called frame\_relay at an interval of 15 minutes and a collection policy name of **theCollection**, add the following directive to the `install.pkg` file:

```
mw_collection_def:mib_path=Frame,poll_interval=15,table_name=mycollection,read_comm=public,username=trendadm,device_type=  
frame_relay,hostname=LOCAL,collection_name=theCollection
```

## Usage Notes

Package Manager registers the component to be of type TREND collector definition.

The default collection name contains a timestamp.

Note that when the package is uninstalled, no specific action is taken for this component.

## report\_dir

This directive instructs Package Manager to create a target directory for the reports and to copy those reports from the packages directory.

### Syntax

**report\_dir:rpt\_dir\_name**

*rpt\_dir\_name* is the name of the target reports directory to create.

### Example

The following example instructs Package Manager to create the directory, `$DPIPE_HOME/reports/my_reports`. It then copies the contents of the `my_reports` sub-directory in the directory path that contains the `install.pkg` file to the new directory, `$DPIPE_HOME/reports/my_reports`.

**report\_dir:my\_reports**

### Usage Notes

This directive instructs Package Manager to create a directory under `$DPIPE_HOME/reports` with the specified report directory name. In addition, Package Manager copies the contents of that sub-directory in the directory path that contains the `install.pkg` file to the newly created directory.

Package Manager creates a registry entry for this directive.

When the report pack is uninstalled, Package Manager removes the subdirectory and its contents from `$DPIPE_HOME/reports`.

This directive has the same function as `reports` directive. See [reports](#) on page 249.

## reportlink

This directive instructs Package Manager to deploy the given file from the directory path that contains the `install.pkg` file to the selected Web Access Server.

## Syntax

**reportlink:***file\_name*

*file\_name* is the name of the tlg file to copy and register.

## Example

If the report pack has links in it in a file named `my_link_grp.tlg`, add the following directive to the `install.pkg` file:

**reportlink:my\_link\_grp.tlg**

## Usage Notes

Package Manager registers the file as a component of the report pack with a type of TREND report link.

This directive applies only when the user of the report pack selects the Deploy Reports option while installing the report pack with Package Manager.

When the report pack is uninstalled, Package Manager undeploys the file from the selected Web Access Server. This occurs only when the user of the report pack selects the Undeploy Reports option while uninstalling the report pack with Package Manager.

The `install.pkg` file may have multiple **reportlink** directives in it.

You can view links in this file from the Management Console, through Catalog Manager.

## report\_pack

This directive specifies the name of the report pack.

## Syntax

**report\_pack:***rptpack\_name*

*rptpack\_name* is the name of the report pack defined in the `install.pkg` file. This name can be up to 50 characters long.



## Example

The report pack is in the `$DPIPE_HOME/packages/my_report_pack.ap` directory for the 2.1 version, add the following directives to the `install.pkg` file:

```
report_pack:my_report_pack  
version:2.1
```

## Usage Notes

This name must match the name of the directory containing the report pack contents (without the `.ap` extension). This name is the primary identification for the registered report pack.

A version directive must follow a `report_pack` directive. See the directive [version](#) on page 257 for details.

The `install.pkg` file may have only one `report_pack` directive in it.

## reports

This directive instructs Package Manager to create a target directory for the reports and to copy those reports from the packages directory.

## Syntax

```
reports:rpt_dir_name
```

*rpt\_dir\_name* is the name of the target reports directory to create.

## Example

The following example instructs Package Manager to create the directory, `$DPIPE_HOME/reports/my_reports`. It then copies the contents of the `my_reports` sub-directory in the directory path that contains the `install.pkg` file to the new directory, `$DPIPE_HOME/reports/my_reports`.

```
reports:my_reports
```

## Usage Notes

This directive instructs Package Manager to create a directory under `$DPIPE_HOME/reports` with the specified report directory name. In addition, Package Manager copies the contents of that sub-directory in the directory path that contains the `install.pkg` file to the newly created directory.

Package Manager creates a registry entry for this directive.

When the report pack is uninstalled, Package Manager removes the subdirectory and its contents from `$DPIPE_HOME/reports`.

This directive has the same function as `report_dir` directive. See [report\\_dir](#) on page 247.

## run\_command

This directive instructs Package Manager to run the specified command during installation.

### Syntax

**run\_command:***command*

*command* is the command to run during installation.

### Example

Suppose a TEEL file, `mycollection.teel`, contained the table definition for the **mycollection** table. Add the following directive to create the table:

```
run_command:{DPIPE_HOME}/bin/datapipe_manager -p create  
-a {DPIPE_HOME}/packages/my_report_pack.ap/  
mycollection.teel
```

## Usage Notes

Specify the complete path for the *command* along with all required parameters for it.

There is no registry entry for the **run\_command** directive.

The `install.pkg` file may have multiple **run\_command** directives in it.

## run\_sql\_script

This directive instructs Package Manager to run the specified SQL script during installation.

### Syntax

**run\_sql\_script:***script\_name*

*script\_name* is the name of the SQL script to run during installation.

### Example

Suppose the package requires an SQL script, `create_custom_proc.sql`, that contains a procedure definition. Add the following directive to the `install.pkg` file:

```
run_sql_script:create_custom_proc.sql
```

### Usage Notes

There is no registry entry for the `run_sql_script` directive.

The `install.pkg` file may have multiple `run_sql_script` directives in it.

## scripts

This directive instructs Package Manager to copy the given file from directory path that contains the `install.pkg` file to the `$DPIPE_HOME/scripts` directory.

### Syntax

**scripts:***file\_name*

*file\_name* is the name of the file to copy and register.

### Example

If the report pack uses a summary file, `utilization.sum`, add the following directive to the `install.pkg` file:

**scripts:utilization.sum**

## Usage Notes

Package Manager registers the file as a component of the report pack with one of the following types:

File Suffix	Component Type
.dis	<b>TRENDdiscover file</b>
.pro	<b>TRENDproc file</b>
.rep or .srep	<b>TREND qss file</b>
.sql	<b>SQL script</b>
.sum	<b>TRENDsum file</b>
all others	<b>TREND Other</b>

When the report pack is uninstalled, Package Manager removes the component from the \$DPIPE\_HOME/scripts directory.

The `install.pkg` file may have multiple scripts directives in it.

## software\_version

This directive specifies the version string required for the corresponding PI software that supports the report pack.

### Syntax

**software\_version:***version\_string*

*version\_string* has the following format: *major.minor.point.patch*.

*major* is an integer that represents the major release value.

*minor* is an integer that represents the minor release value.

*point* is an integer that represents the point release value.

*patch* is an integer that represents the patch release value.

## Example

If the corresponding software required to use this report pack is at version 5.4, add the following directive to the `install.pkg` file:

```
software_version:5.40.0.0
```

## Usage Notes

If the software version of the PI software is greater than or equal to the software version specified in this directive, then Package Manager enables the installation of the report pack.

For example, if the PI software version is 5.4.0.3 and the value of the `software_version` directive is 5.4.0.4, then Package Manager does not allow the installation of this report pack; however, if the PI software version is 5.4.0.4 and the value of the `software_version` directive is 5.4.0.3, then Package Manager enables the installation of this report pack.

Note that the `version.prp` file, which is located in the `DPIPE_HOME/data` directory, contains the software version of the PI software. PI compares the `version_string` value to the software version listed in this file.

Package Manager registers the file as a component of the report pack with a type of TREND software version.

Note that you cannot use an asterisk (\*) in place of any value. You must enter an integer for each value.

## sql

This directive instructs Package Manager to copy the given file from the appropriate directory for the installed database in the package directory to the `$DPIPE_HOME/scripts/db` directory.

`db` is the database where PI is connected.

## Syntax

```
sql:file_name
```

*file\_name* is the name of the file to copy and register.

## Example

If the report pack uses an SQL script file, `update.sql`, add the following directive to the `install.pkg` file:

```
sql:update.sql
```

## Usage Notes

The package directory contains a directory for each type of database that PI supports. These directories contain the appropriate SQL scripts for the specified databases. Package Manager locates the specified file from the appropriate directory and copy it to the `$DPIPE_HOME/scripts/db` directory. For example, if PI is installed on an Oracle database and you are installing a package with the name ABC, Package Manager locates the file in the `$DPIPE_HOME/packages/ABC/ABC.ap/Oracle` directory and copy it to the `$DPIPE_HOME/scripts/Oracle` directory. If the `install.pkg` has the **sql** directive as specified in the example section, Package Manager copies the file `$DPIPE_HOME/packages/ABC/ABC.ap/Oracle/update.sql` as `$DPIPE_HOME/scripts/Oracle/update.sql`.

Package Manager registers the file as a component of the report pack with a type of PI SQL file.

The `install.pkg` file may have multiple sql directives in it.

## tmp

This directive instructs Package Manager to copy the given file from the directory path that contains the `install.pkg` file to the `$DPIPE_HOME/tmp` directory.

## Syntax

```
tmp:file_name
```

*file\_name* is the name of the file to copy and register.

## Example

If the report pack uses the file, `mylog.txt`, add the following directive to the `install.pkg` file:

**tmp:mylog.txt**

## Usage Notes

Package Manager registers the file as a component of the report pack with a type of TREND temporary file.

When the package is uninstalled, Package Manager removes the file from the `$DPIPE_HOME/tmp` directory.

The `install.pkg` file may have multiple `tmp` directives in it.

## trend\_timer\_def

This directive contains the necessary information for Package Manager to insert an entry in the `trendtimer.sched` file.

## Syntax

**trend\_timer\_def:time\_control - - command**

*time\_control* is when the corresponding *command* runs.

*command* is the command to run at the specified time.

See [trendtimer](#) on page 521 for details on the allowed time control options.

## Example

The report pack requires that a procedure file, `myfile.pro`, run at 4:00 a.m. daily; add the following directive to the `install.pkg` file:

```
trend_timer_def:24:00+4:00 - - {DPIPE_HOME}/bin/  
trend_proc -f {DPIPE_HOME}/scripts/myfile.pro
```

## Usage Notes

Package Manager registers the entry as a component of type TRENDtimer definition.

Remember to fully qualify the command.

When the report pack is uninstalled, Package Manager removes the entry from the `trendtimer.sched` file.

The `install.pkg` file may have multiple `trend_timer_def` directives in it.

## uninstall\_cmd

This directive specifies a command that Package Manager runs only when it removes the report pack.

### Syntax

**uninstall\_cmd:***command*

*command* is the command to run during the uninstall process.

### Example

If the report pack removes a directory upon uninstall, add the following directive to the `install.pkg` file:

```
uninstall_cmd:rmdir /tmp/junk
```

### Usage Notes

Package Manager registers this component to the report pack with a type of TREND uninstall run command.

The `install.pkg` file may have multiple `uninstall_cmd` directives in it.

## uninstall\_sql

This directive specifies an SQL script that Package Manager runs only when it removes the report pack.

### Syntax

**uninstall\_sql:***script\_name*

*script\_name* is the name of the SQL script to run during the uninstall process.



## Example

If the report pack runs a script, `drop_custom_procs.sql`, that drops some custom procedures, add the following directive to the `install.pkg` file:

```
uninstall_sql:drop_custom_procs.sql
```

## Usage Notes

The SQL script to be run must exist in the `$DPIPE_HOME/scripts` directory.

Package Manager registers this component to the report pack with a type of TREND uninstall run sql script.

The `install.pkg` file may have multiple `uninstall_sql` directives in it.

## version

This directive specifies the version string of a report pack.

## Syntax

```
version:version_string
```

*version\_string* is the version information for the corresponding report pack identified with this directive. The length of the string can be up to 20 characters.

## Example

If `my_report_pack` is at version 2.1 and is dependent upon the report pack `myLanWan` at version 1.5, add the following directives to the `install.pkg` file:

```
report_pack:my_report_pack  
version:2.1  
dependency:myLanWan  
version:1.5
```

## Usage Notes

This directive provides the version information for the following reasons:

- To specify the version of the current report pack during installation.
- To specify the required version for a dependency report pack.

The version directive must follow the `report_pack` directive or the dependency directive. The combination of the *version\_string* and the *rptpack\_name* in sequential directives uniquely identify the report pack.

# Sample Layout for the File

The following example is a sample layout for the `install.pkg` file. It just shows the possible order for the directives in the file. It does not use any existing file names. The comments appear for clarification.

```
## Datapipe Name and Version  ##
report_pack:xyz
version:1.0

## Datapipe Name Dependencies ##
software_version:1.0.0.0
dependency:abc_datapipe,1.0

## Reports to be installed in {DPIPE_HOME}/reports directory
##
report_dir:xyz

## Create db tables during install. These tables will be
removed during uninstall. ##
database_table:abc_view
database_table:xyz
database_table:Kxyz_property

## Files to install in DPIPE_HOME/lib directory ##
lib:abc_prop.teel

## Files to install in DPIPE_HOME/mibs directory ##
mibs:xyz_abc.mib

## Files to install in DPIPE_HOME/scripts directory ##
scripts:xyz_process.pro
scripts:xyz.sum
scripts:xyz.sql
lib:xyz.teel

## Install the Package ##
run_command:{DPIPE_HOME}/bin/trend_proc
-f {DPIPE_HOME}/packages/xyz/xyz.ap/install.pro
```

```
run_command:{DPIPE_HOME}/bin/datapipe_manager -p register
-a {DPIPE_HOME}/packages/xyz/xyz.ap/table_def.teel

## Clean up procedures for uninstall ##
uninstall_sql:xyz_remove_components.sql

## Data Collection Setup ##
mw_collection_def:mib_path=xyz,poll_interval=15,
table_name=xyzcollect,read_comm=public,username=trendadm,
device_type=abc_datapipe,hostname=LOCAL

## Update trendtimer.sched file. ##
trend_timer_def:1:00+20 - - {DPIPE_HOME}/bin/trend_proc
-f {DPIPE_HOME}/scripts/abc.pro
trend_timer_def:24:00+3:00 - - {DPIPE_HOME}/bin/trend_proc
-f {DPIPE_HOME}/scripts/xyz.pro
```

# 17 indexmaint

You can use the `indexmaint` command to maintain indexes of existing data tables on a PI system.

## Requirements and Restrictions

- When you use the `-g` option, you must include the `-t` option on the command line.
- An error occurs if the `-K` option appears on the command line with the `-t` option.
- An error occurs if the following option appear on the command line at the same time: `-c`, `-f`, and `-l`.

## Syntax

The `indexmaint` command uses the following syntax:

```
indexmaint  [ -c ]  
            [-D]  
            [ -d debug_level ]  
            [ -e index_name ]  
            [ -f ]  
            [ -g { -unique  
                  -nonunique } ,index_name,col1 [ ,col2, ... ,colN ] ]
```

```
[ -h ]  
[ -K ]  
[ -l ]  
[ -n ]  
[ -r ]  
[ -S database_server_name ]  
[ -t table_name ]  
[ -v ]
```

## Options

The `indexmaint` command has the following options:

- c** This option is available for an Oracle database only.  
Use this option to coalesce indexes instead of rebuilding them.  
An error occurs if this option is on the same command line as the **-f** and **-l** options.
- D** Use this option to maintain data tables only. An error occurs if this option is on the same command line as the **-t** option.
- d** Use this option to specify the debug output level. Use the value **1** to see the debug output. Use this option only for testing in coordination with HP Technical Support due to the additional overhead it places on `indexmaint`.  
The default is no debug output.  
Debug output writes to standard output.
- e** Use this option to specify the name of the index to drop.

- f** On Sybase, use this option to unconditionally drop and recreate all non-primary indexes according to the guidelines in [Table 15](#) on page 265. You can update the primary indexes by running the Sybase command `update statistics`.  
On Oracle, use this option to rebuild all indexes and their associated partitions if any of the following conditions exist:
- The index is not an index-organized table (IOT).
  - The index is an IOT and it is not a primary index.
- The system analyzes all of the indexes statistics according to the guidelines in [Table 16](#) on page 268.  
An error occurs if this option is on the same command line as the **-c** option.
- g** Use this option to create an index for one or more columns. This option requires the following information:
- Specify the unique qualities of the index; valid values are **unique** or **nonunique**.
  - Enter the name of the index, which can be up to 30 characters.
  - Specify one or more columns where the system creates this index.
- h** Use this option to display all command line options (help).
- k** Use this option to maintain property and keymap tables only.  
An error occurs if this option is on the same command line as the **-t** option.  
This option is in UPPERCASE.
- l** Use this option to list existing indexes and write the results to the standard output.  
An error occurs if this option is on the same command line as the **-c** option.  
This option is lowercase “L.”
- n** Use this option to bypass the verification process for the required indexes and their associated index columns on each table.  
This option overrides the default, which is to verify.  
You should use this option only when you use the **-g** option on the same command line.

- r** This option is available for Sybase databases only.  
Use this option to prevent the creation of clustered indexes using the sorted-data option.  
This option overrides the default, which creates the clustered indexes using the sorted-data option.  
If the creation fails with a data out of order message, the default applies.  
When the clustered indexed columns contain sorted data, this option causes `indexmaint` to run faster.
- s** Use this option to specify the database server name.  
This option is in UPPERCASE.
- t** Use this option to maintain the index for the specified table only.  
An error occurs if this option is on the same command line as the **-k** option.
- v** Use this option to display the version number.  
This option is in UPPERCASE.

## Usage Notes

You can use the `indexmaint` command to rebuild and rename indexes of existing data tables in the table dictionary on a PI system to improve the efficiency of raw-to-delta and other processes. All actions appear in the `trend.log` file.

## Table Indexes

In order for a PI system to operate efficiently, all user data tables must have properly defined and maintained indexes. The `indexmaint` utility checks that the correct indexes exist on a table and then it creates the indexes, as needed. When you run `indexmaint` without any options, it checks all tables in the PI database, as follows. It locates the data tables, checks the type for each table, verifies the indexes, and then creates the appropriate indexes, if necessary.



## Sybase

If the appropriate index exists, `indexmaint` runs the Sybase command, `update statistics`, on the index. The following table provides a list of the indexes for each PI table type.

**Table 15 Table Indexes — Sybase**

Table Type	Index Name	Index Type	Indexed columns (in order)	Notes
archive event raw	ind1	nonclustered	ta_period dsi_key_id_	
archive raw	ind2	nonclustered	dsi_key_id_ ta_period	
baseline forecast rank rate summary	cuind	unique, clustered	ta_period dsi_key_id_	
baseline forecast rank rate summary	uind1	unique, nonclustered	dsi_key_id_ ta_period	
trendit	cuind	unique clustered	ta_period dsi_key_id_ dsi_agg_type	
trendit	uind1	unique, nonclustered	dsi_key_id_ ta_period dsi_agg_type	
property	pk_ prop_ tbl_name	unique, clustered	dsi_key_id	
property	uind1	unique, nonclustered	object_by_var_list	

**Table 15 Table Indexes — Sybase**

<b>Table Type</b>	<b>Index Name</b>	<b>Index Type</b>	<b>Indexed columns (in order)</b>	<b>Notes</b>
property	uind2	unique, nonclustered	<i>object_by_var_list _reverse</i>	
property	uind3	unique, nonclustered	<i>collection_by_ var_list dsi_bv_state</i>	This index exists only if the collection by-variables are different from the object by-variables and if the dsi_bv_state column exists.
property	uind4	unique, nonclustered	<i>collection_by_ var_list_reverse dsi_bv_state</i>	This index exists only if the collection by-variables are different from the object by-variables and if the dsi_bv_state column exists.
keymap	pk_ keymap_ tbl_name	unique, clustered	<i>local_key_id foreign_server_id foreign_key_id</i>	
keymap	uind1	unique, nonclustered	<i>foreign_server_id local_key_id foreign_key_id</i>	
keymap	uind2	unique, nonclustered	<i>foreign_key_id foreign_server_id</i>	

**Table 15 Table Indexes — Sybase**

<b>Table Type</b>	<b>Index Name</b>	<b>Index Type</b>	<b>Indexed columns (in order)</b>	<b>Notes</b>
lkeys	cuind	unique, clustered	dsi_key_id_ ta_period	
lkeys	uind1	unique, nonclustered	ta_period dsi_key_id_	
lkeys	uind2	unique, nonclustered	ta_period dsi_key_id_ ta_period	

*collection\_by\_var\_list* is the list of collection by-variables in their original order.

*collection\_by\_var\_list\_reverse* is the list of collection by-variables in reverse order.

*foreign\_key\_id* is the dsi\_key\_id\_ value on the remote server.

*foreign\_server\_id* is the identification for the remote server.

*local\_key\_id* is the dsi\_key\_id\_ value on the local server.

*object\_by\_var\_list* is the list of object by-variables in their original order.

*object\_by\_var\_list\_reverse* is the list of object by-variables in reverse order.

*pk\_keymap\_tbl\_name* is the name of the keymap table with pk\_ as a prefix.

*pk\_prop\_tbl\_name* is the name of the keymap table with pk\_ as a prefix.

## Oracle

If the appropriate index exists, `indexmaint` runs the Oracle command, `Analyze Table`, on the index. The following table provides a list of the indexes for each PI table type.

**Table 16 Table Indexes — Oracle**

Table Type	Index Name	Index Type	Indexed columns (in order)	Notes
archive event raw	<i>tbl_name_I1</i>	nonunique, normal	ta_period dsi_key_id_	
archive raw	<i>tbl_name_I2</i>	nonunique, normal	dsi_key_id_ ta_period	
baseline forecast rank rate summary	<i>pk_tbl_name</i>	IOT, unique, primary	ta_period dsi_key_id_	
baseline forecast rank rate summary	<i>tbl_name_I1</i>	unique, normal	dsi_key_id_ ta_period	
trendit	<i>pk_tbl_name</i>	IOT, unique, primary	ta_period dsi_key_id_ dsi_agg_type	
trendit	<i>tbl_name_I1</i>	normal	dsi_key_id_ ta_period dsi_agg_type	
property	<i>pk_prop_tbl_name</i>	IOT, unique, primary	dsi_key_id	
property	<i>prop_tbl_name_I1</i>	unique, normal	<i>object_by_var_list</i>	

**Table 16 Table Indexes — Oracle**

<b>Table Type</b>	<b>Index Name</b>	<b>Index Type</b>	<b>Indexed columns (in order)</b>	<b>Notes</b>
property	<i>prop_ tbl_name_I2</i>	unique, normal	<i>object_by_var_list _reverse</i>	
property	<i>prop_ tbl_name_I3</i>	unique, normal	<i>collection_by_ var_list dsi_bv_state</i>	This index exists only if the collection by-variables are different from the object by-variables and if the dsi_bv_state column exists.
property	<i>prop_ tbl_name_I4</i>	unique, normal	<i>collection_by_ var_list_reverse dsi_bv_state</i>	This index exists only if the collection by-variables are different from the object by-variables and if the dsi_bv_state column exists.
keymap	<i>pk_keymap_ tbl_name</i>	unique, normal	<i>local_key_id foreign_server_id foreign_key_id</i>	
keymap	<i>keymap_ tbl_name_I1</i>	unique, normal	<i>foreign_server_id local_key_id foreign_key_id</i>	
keymap	<i>keymap_ tbl_name_I2</i>	unique, normal	<i>foreign_server_id foreign_key_id</i>	
lkeys	<i>pk_tbl_name</i>	IOT, unique, primary	<i>dsi_key_id_ ta_period</i>	
lkeys	<i>tbl_name_I1</i>	unique, normal	<i>ta_period dsi_key_id_</i>	
lkeys	<i>tbl_name_I2</i>	unique, normal	<i>ta_period_old dsi_key_id_ ta_period</i>	

*collection\_by\_var\_list* is the list of collection by-variables in their original order.

*collection\_by\_var\_list\_reverse* is the list of collection by-variables in reverse order.

*foreign\_key\_id* is the *dsi\_key\_id* value on the remote server.

*foreign\_server\_id* is the identification for the remote server.

*local\_key\_id* is the *dsi\_key\_id* value on the local server.

*object\_by\_var\_list* is the list of object by-variables in their original order.

*object\_by\_var\_list\_reverse* is the list of object by-variables in reverse order.

*keymap\_tbl\_name* is the name of the keymap table.

*prop\_tbl\_name* is the name of the property table.

*tbl\_name* is the name of the associated table.

## Processing Considerations

The `indexmaint` utility runs against either the entire system without command line options or against a single table with the `-t` option. For each table processed, it verifies that the correct indexes exist with the correct names. If an index is correct but does not have the correct name, the utility renames it according to the guidelines in [Table 15](#) on page 265. If the utility finds an index by one of the names in [Table 15](#) on page 265, but does not match the correct index profile, it drops the index and recreates it according to the guidelines. When you run the utility in default mode without options, it runs one of the following commands on each table:

- On a Sybase system, it runs the `update statistics` command.
- On an Oracle system, it runs the `Analyze Table` command.

The `indexmaint` utility may take a considerable amount of time to build these indexes on large tables. After you run the `indexmaint` utility for the entire system the first time, you may choose to place a once-a-day run entry in `trendtimer.sched` that executes `indexmaint` in default mode. This once-a-day entry handles any new tables introduced into the system that do not contain the proper indexes. You can add the entry shown below to the `trendtimer.sched` file to run `indexmaint` once a day in default mode.

```
23:00 - - ${DPIPE_HOME}/bin/indexmaint
```

Note that `indexmaint` only creates and drops indexes on data tables, if necessary. It does not create and drop indexes on property tables; it only updates the statistics on them.

# Examples

## Example 1

This command line entry checks all user data tables in the database for correct indexes and creates any missing indexes.

```
indexmaint
```

## Example 2

This entry rebuilds all the indexes for all of the user data tables in the PI system.

```
indexmaint -f
```

## Example 3

This command checks all user tables for the current index settings.

```
indexmaint -l
```

## Example 4

This command checks and possibly fixes only the user data table called *mytable*.

```
indexmaint -t mytable
```

## Example 5

This command displays the help information for the command. The output follows the command.

```
indexmaint -h
```





# 18 log\_backup

Backs up a specified file by moving the file to a new file.

## Syntax

The `log_backup` command uses the following syntax:

```
log_backup  [ -d debug_level ]  
              [ -f source_file_name ]  
              [ -n dir_name ]  
              [ -u ]  
              [ -v ]
```

# Options

The `log_backup` command has the following options:

- d** Set a debug output level. Values of **0**, **1**, **2**, or **3** are valid. The higher the number, the more detailed the information. The default is **0**, which means no debug output. Debug output is written to standard output.
- f** Specify the complete path for the source file name.  
The default input file name is `$TREND_LOG/trend.log`; if that file does not exist, `log_backup` looks for `$DPIPE_HOME/log/trend.log`.
- n** Specify a new directory to store the file.  
The default directory name is `$TREND_LOG/backup`.
- u** This option is the help option, which displays the command-line options for the `log_backup` command.
- v** Displays the version stamp for `log_backup`.

## Naming Conventions

The default naming convention for the output files uses the complete spelling of the day of the week appended to the source file name with a dot. The default output file name has the following format.

`$TREND_LOG/backup/file_name.prev_day`

*file\_name* is the name of the source file specified with **-f**.

*prev\_day* is the name of the previous day of the week, such as `sunday` or `wednesday`.

### Back-Up Files Created

The seven default back-up files for the `trend.log` file created by this command are listed below.

```
$TREND_LOG/backup/trend.log.monday  
$TREND_LOG/backup/trend.log.tuesday  
$TREND_LOG/backup/trend.log.wednesday  
$TREND_LOG/backup/trend.log.thursday  
$TREND_LOG/backup/trend.log.friday  
$TREND_LOG/backup/trend.log.saturday  
$TREND_LOG/backup/trend.log.sunday
```

Similarly, `log_backup` creates the following back-up files for specified files such as `audit.log` and `metrics.log`.

```
$TREND_LOG/backup/audit.log.monday  
...  
$TREND_LOG/backup/audit.log.sunday  
  
$TREND_LOG/backup/metrics.log.monday  
...  
$TREND_LOG/backup/metrics.log.sunday
```

## Usage Notes

Entries to run this utility are stored automatically in the `trendtimer.sched` file during the PI installation procedure.

This utility creates a backup file by moving the source file to another file that has the same name appended with a suffix for the previous day of the week. If you specify a new directory name, it stores the file in the same manner in the specified directory.



# 19 mw\_collect

You can use the `mw_collect` command to poll nodes for SNMP data on a PI system.

## Requirements and Restrictions

To collect data with this command, the collector module for the data table must be **MW**.

## Syntax

The `mw_collect` command uses the following syntax:

```
mw_collect  [ -a ]  
              [ -A ]  
              [ -b ]  
              [ -c max_processes ]  
              [ -C wait_time ]  
              [ -d debug_level ]  
              [ -D thread_wait_time ]  
              [ -e ]  
              [ -E clock_error_value ]  
              [ -f config_file ]  
              [ -F min_disk_pct ]
```

```

[ -g ]
[ -G debug_level_pm ]
[ -h hostname ]
[ -H alternate_poller_name ]
-i interval
[ -I check_index ]
[ -k ]
[ -K suppress_spikes ]
[ -L ]
[ -M minimum_filter ]
[ -n ]
[ -N retry_interval ]
[ -o timeout ]
[ -p max_entries_per_pdu ]
[ -P snmp_port ]
[ -q log_info_level ]
[ -Q name ]
[ -r retries ]
[ -R min_rows ]
[ -s round_factor ]
[ -S snmp_version ]
[ -t table_name ]
[ -u ]
[ -V ]
[ -w high_water_mark ]
[ -W high_water_mark_log ]
[ -X ]
[ -Y ]
[ -Y delta_time ]
[ -z child_debug_level ]
[ -Z debug_log_bcpgateway ]

```



The following options are currently available in this version of PI for compatibility with previous versions of PI.

- **-j**: To direct the poller to use the SNMP version as defined in the database. It uses the SNMP version associated with each node.
- **-T**: To direct the poller to not perform fast time conversion.

## Option Categories

The following command line options are available for the parent poller.

**Table 17 mw\_collect Option Categories**

Category	Options
Typical	<b>-a, -b, -c, -C, -D, -e, -F, -g, -h, -H, -i, -k, -n, -o, -p, -P, -Q, -r, -R, -s, -S, -t</b>
Miscellaneous	<b>-d, -f, -L, -q, -u, -V, -w, -W, -Z, -y, -z</b>
Raw-to-Rate	<b>-A, -E, -G, -I, -K, -M, -N, -X, -Y</b>

# Options

The `mw_collect` command has the following options:

- a** Directs the child collectors of `mw_collect` to output the collected data in ASCII format. You may want to do this for debugging purposes.
- A** Enables archiving of raw data. The default is no archiving. When you use this option, it turns on archiving. The archive function stores the collected data in a raw data table. This option is in UPPERCASE. This option is equivalent to the `@bArchive=1` parameter in `trendpm`.
- b** Forces regeneration of definition files and worktable. The poller analyzes the structure of the table to be collected and creates definitions about how to load data into database for the `bcp_gateway` process. This is a one-time process most of the time. However, when the poller realizes that the structures of the data table or related property tables have changed, it regenerates the `bcp_gateway` process definitions and worktable. The **-b** option forces the regeneration of these definition files, and cleans up any internal database objects used during data loading. See the discussion for the **-g** option listed lower in this table for more information. See the discussion for the **-k** option listed lower in this table for more information.
- c** Specifies the number of collection processes to run concurrently. When `mw_collect` runs, it starts child processes that actually do the collections. The default is 5. You can reduce SNMP collection cycles by increasing this number.



- C** Specifies the number of minutes that each child process can run. The system kills the child process if it runs longer than the specified number of minutes.

The default number of minutes is **30** minutes for SNMP child collectors (`dpipe_snmp`).

The default number of minutes is **3** minutes for PA or Operations agent child collectors (`dpipe_pa`).

This option is in UPPERCASE.
- d** Sets the debug output level for the parent instance of `mw_collect`. Values of **0**, **1**, **2**, or **3** are valid. The higher the number, the more detailed the information.

The default is **0**, which means no debug output. Debug output is written to standard out. You should use this option only for testing in coordination with HP Technical Support due to the additional overhead it places on `mw_collect`.

The **-z** and **-Z** options set the debug level for the child processes of `mw_collect`. The **-d**, **-z**, and **-Z** options can be used together to control the debug level of parent and child processes independently.
- D** This option specifies how many seconds the parent thread should wait for signals from the collector and `bcp_gateway` threads.

It is normal to get `thread_wait` timeout messages in the log file when waiting for signals from the `bcp_gateway` or `trendpm` thread due to potential long running jobs when writing to the database. The program exits when a `thread_timeout` occurs, unless the `thread_timeout` occurred when all collector jobs are finished and only `bcp_gateway` or `trendpm` threads remained. In this case, the timeout only appears in the `trend.log` file.

For SNMP Collection, the default is the same value as the **-C** timeout value, which is **1800** seconds. For example, if the **-C** value is **2** (2 minutes), then the **-D** value defaults to **120** (120 seconds).

For PA or Operations agent collection, the default value is **1800** seconds and it is not related with **-C** option

This option is in UPPERCASE.

- e Turns on GETBULK when using SNMP-V2.  
When the poller uses SNMP-V2, you can opt to use the GETBULK request instead of GETNEXT when getting SNMP data.
  
- E Sets the percentage level for valid data. The value is the percentage of difference between the delta values of two Received Timestamps and two System Uptimes. These statistics come from two consecutive raw data samples.  
For example, if r1 and s1 are the Received Timestamp and System Uptime for the first sample, and r2 and s2 are the Received Timestamp and System Uptime for the second sample, then the calculation for the value is  $((r2 - r1) - (s2 - s1)) * 100 / (r2 - r1)$ . During processing, if the calculated value for the samples exceeds the value set by this option then the samples are rejected.  
The default value is **10**.  
This option is in UPPERCASE.  
This option is equivalent to the **@zerror** parameter in `trendpm`.
  
- f Specifies the name of the configuration file that contains additional parameters. See [Configuration File](#) on page 292 for descriptions of the parameters.
  
- F Specifies the minimum disk space (as percent of total disk space) that must exist on the disk where the `$COLLECT_HOME` directory resides. If less space exists on the directory, `mw_collect` does not run.  
The default is **5**, which represents 5 percent.  
This option is in UPPERCASE.

- g** Use 3.5.x compatibility. That is, use the same method for inserting data into the table as version 3.5.x and earlier.

The **-g** option loads data into archive (raw) tables directly. The `bcp_gateway` definition file contains the setting for the **-g** option. This means that if you already invoked `mw_collect` without the **-g** option and you need the **-g** option, then use the **-b** option to regenerate the definition file. Similarly, if you already invoked `mw_collect` with the **-g** option and you do not need the **-g** option, then use the **-b** option to regenerate the definition file. Otherwise, `mw_collect` uses the same setting for the **-g** option currently in effect when you originally generated the definition file.
- G** Sets the debug output level for the `trendpm` process of `mw_collect`. Values of **0**, **1**, **2**, or **3** are valid. The higher the number, the more detailed the information.

The default is **0**, which means no debug output.

Debug output is written to standard out. You should only use this option for testing in coordination with HP Technical Support due to the additional overhead it places on `mw_collect`.

This option is in UPPERCASE.

This option is equivalent to the `@debug_level` parameter in `trendpm`.
- h** Names the host to be polled, which must be an SNMP-manageable device. This name is saved in the database along with the data returned from the node, therefore, to be consistent with other data, use the same node name format used when entering nodes in the database.
- H** Specifies an alternate poller name. When you run `mw_collect` in distributed mode, with the **-n** option, the poller compares the local hostname to the Poll From field in the polling policy. When you use the **-H** option, `mw_collect` compares the Poll From field in the polling policy to the alternate poller name.

This option is in UPPERCASE.

See [Distributed Polling](#) on page 301 for more information.

- i** Is the Collection ID. `mw_collect` executes the entries in the polling policy that have this value in their interval field. How frequently `mw_collect` is actually run depends on the configuration of `trendtimer`, but the idea is to be consistent so that a collection request with a collection ID of **5** is run every 5 minutes. See [File Locks](#) on page 298 for additional information.

This option is required.
- I** Specifies whether to use existing indices on the upload table or to drop existing indices and then recreate them. The value **1** means that the existing indices on the upload table are used. The value **0** means that the existing indices are dropped and then recreated. If the value is **1** and the proper indices are missing then the raw-to-delta process fails.

The default is **0**.

This option is in UPPERCASE.

This option is equivalent to the `@bCheck_index` parameter in `trendpm`.
- k** Populates the property tables (but not the data tables) for the devices you are polling.

The `bcp_gateway` definition file contains the setting for the **-k** option. This means that if you already invoked `mw_collect` without the **-k** option and you need the **-k** option, then use the **-b** option to regenerate the definition file. Similarly, if you already invoked `mw_collect` with the **-k** option and you do not need the **-k** option, then use the **-b** option to regenerate the definition file. Otherwise, `mw_collect` uses the same setting for the **-k** option currently in effect when you originally generated the definition file.

- K** Specifies whether to reject samples if there are spikes. A spike is defined when a counter is manually reset, the difference of two consecutive samples from a counter exceeds the spike threshold, and the second sample is less than the first sample. The value of the spike threshold is  $2^{31}$  for 32-bit counters or  $2^{51}$  for 64-bit counters. Remember if the difference of the samples is negative; account for the rollover of the counter by adding  $2^{32}$  for 32-bit counters or  $2^{52}$  for 64-bit counters.
- Valid values for *suppress\_spikes* are:
- **1**: Rejects samples if a spike occurs.
  - **0**: Does not reject samples. The default is **0**.
- This option is in UPPERCASE.
- This option is equivalent to the **@bSuppress\_spike** parameter in `trendpm`.
- L** Specifies that the collected data be stored locally instead of in the PI database. When you use this option, `mw_collect` uses any previously saved collection definitions, and does not use the database.
- This option is in UPPERCASE.
- M** Sets the *minimum\_filter* value. The procedure rejects the sample if the delta value of a counter falls below this value.
- The default value is **-1**, which means to accept the entire sample.
- This option is in UPPERCASE.
- This option is equivalent to the **@line\_suppress\_value** parameter in `trendpm`.
- n** Enables distributed polling. If this option is used, `mw_collect` executes the collection request only if the Poll From field in the polling policy for this collection request matches the hostname of the machine on which `mw_collect` is running. If you omit this option, `mw_collect` executes all polling requests whose interval matches the value of the **-i** option, regardless of the hostname specified to do the polling in the polling instructions.
- You can set an alternate hostname to poll with the **-H** option. See [Distributed Polling](#) on page 301 for more information.

- N**      Sets the *retry\_interval*, which is the number of seconds the procedure must wait to acquire a lock on an upload table.  
The default value is **10**, which is 10 seconds.  
This option is in UPPERCASE.  
This option is equivalent to the **@retry\_interval** parameter in *trendpm*.
  
- o**      Number of seconds *mw\_collect* is to wait for a response after sending an SNMP request.  
The default is **1** second. (SNMP timeout.)
  
- p**      The number of SNMP variables to include in the varbind list in the GETNEXT PDU (Protocol Data Unit) request. It is possible to generate a GETNEXT request that yields a response that is too long to transmit.  
The default value is **20**.
  
- P**      This option enables the collection of SNMP data from the specified port rather than the default SNMP port of **161**.  
This option is in UPPERCASE.
  
- q**      Sets the log information level for the parent instance of *mw\_collect*.  
Values of **0**, **1**, **2**, or **3** are valid. The higher the number, the more detailed the information. The default is **0**, which means no log information for the parent instance of *mw\_collect*.  
If the *\$TREND\_LOG* environment variable is set, then the log is written in the directory specified by the *\$TREND\_LOG* environment to a log file named *parent\_collect\_dbg.log*; the path and file name is *\$TREND\_LOG/parent\_collect\_dbg.log*.  
If the *\$TREND\_LOG* environment variable is not set, then the log is written to the same log file name in the *\$DPIPE\_HOME/log* directory; the path and file name is *\$DPIPE\_HOME/log/parent\_collect\_dbg.log*.  
You can modify the log file name using the **-Q** option.

- Q** Specifies the *name* for the ROOT of the trace log file. You can use this option to change the name of the log file, as follows:  
*path/name\_pid\_ppid\_polltime\_dbg.log*  
*path* is \$TREND\_LOG if the environment variable is set; otherwise, it is \$DPIPE\_HOME/log.  
*name* is the name of the file that you supply.  
*pid* is the process id number for this process.  
*ppid* is the process id number of the parent process.  
*polltime* is the number of seconds since 01/01/1970 00:00.  
 When you use this option, set the **-q** option to a value greater than 0.  
 This option is in UPPERCASE.
- r** Specifies the *retries*, which is the number of times `mw_collect` resends an SNMP request before assuming the target node is not going to respond.  
 The default value is **5**. (SNMP retries)
- R** Sets the minimum number of rows to collect before starting the loading of data into database. The number of rows is an approximation, since there is an assumption that each row is 500 bytes.  
 The default value is **1000** rows, which means that the loading of the file into the database starts when the size is 500,000 bytes (500 \* 1000).  
 This option is in UPPERCASE.
- s** This option rounds off the collection time (*ta\_period*). If the `mw_collect` parent kicks off a collection at 3:07, and if you are using the default collection option of **300** seconds (5 minutes), the actual *ta\_period* value for the collection is recorded as 3:05.
- S** Specifies the SNMP version. This option is in UPPERCASE.  
 Valid values are :
- **1** for SNMP V1
  - **2** for SNMP V2C.
- This option overrides the values read from the database.

- t** Specifies the *table\_name*, which is the name of the MIB table that you want to collect. The data table must already exist in the database.
- u** Displays the command line formats for `mw_collect`.
- v** Displays the version stamp for `mw_collect`.  
This option is in UPPERCASE.
- w** Specifies the *high\_water\_mark*. The high water mark stops collection of data when the database-used size reaches the specified percentage.  
Valid values are **1 - 100**.  
The default parameter is **90** for 90%.
- W** Specifies the *high\_water\_mark\_log*. The high water mark stops collection of data when the log-used size reaches the specified percentage. Valid values are **1 - 100**.  
The default parameter is **90** for 90%.  
This option is in UPPERCASE.
- x** This option turns off `trendpm` capability.  
This option is in UPPERCASE.
- y** This option disables the calculation of `total count` for `bcp_gateway` metrics.
- Y** Specifies which clock to use to calculate Delta Time.  
Valid values are:
  - **1**: Directs the procedure to use System Uptime to calculate Delta Time.
  - **0**: Directs the procedure to use the `received_ts` column for the calculation.
 The default is **0**.  
This option is in UPPERCASE.  
This option is equivalent to the **@bDelta\_time** parameter in `trendpm`.



- z** Sets the debug level for child collector processes.  
Values of **0**, **1**, **2**, or **3** are valid. The higher the number, the more detailed the information.  
The default is **0**, which means no debug output.  
Debug output is written to standard out. You should only use this option for testing in coordination with HP Technical Support due to the additional overhead it places on `mw_collect`.
- Z** This option specifies whether to turn on debugging or logging or both for the `bcp_gateway` process. When the logging option is turned on, the information is written to the log file. The name of the log file varies depending on which environment variable is set.  
If the `$TREND_LOG` environment variable is set, then the log is written in the directory specified by the `$TREND_LOG` environment to a log file named `bcp_gateway_dbg.log`; the path and file name is `$TREND_LOG/bcp_gateway_dbg.log`.  
If the `$TREND_LOG` environment variable is not set, then the log is written to the same log file name in the `$DPIPE_HOME/log` directory; the path and file name is `$DPIPE_HOME/log/bcp_gateway_dbg.log`.  
Valid values for this option are:
- **1**: Turn on logging.
  - **2**: Turn on debugging.
  - **3**: Turn on both logging and debugging.
- This option is in UPPERCASE.



If any logging option is set to **on** and you are running `trendtimer.sched`, add an entry to `trendtimer.sched` to back up log files. To do that, add the following command to `trendtimer.sched`:

**DPIPE\_HOME}/bin/log\_backup -f *input\_file\_name***

# Usage Notes

`mw_collect` is the PI SNMP polling application for generic MIB tables. Its simplest use is to collect a specific table once from a single node, and store the information in the PI database. Fully utilized, `mw_collect` follows user-defined instructions for the following:

- To poll a dynamic list of nodes anywhere on the network for data at regular intervals.
- To store the results on a local or remote system depending on who requests the information.
- Provide the ability to run without the database in cache mode.

The `mw_collect` command invokes the `mw_collect` parent poller. The `mw_collect` parent poller, in turn, invokes multiple child pollers. Each child poller collects a single node/table combination. The `-c` option specifies the number of child pollers that run concurrently.

There are extra parameters that you can set in `mw_collect` that supplement the command-line options for `mw_collect`. The parameters are in a configuration file. You can specify the name of this file with the `-f` option on the command-line. See [Configuration File](#) on page 292 for descriptions of the parameters and the file.

To collect data with this command, the setting for the **CollectorModule** statement in the TEEL definition file for the data table must be **MW**.

## Terms

This section defines the common terms that apply to `mw_collect`.

parent poller	The main part of the SNMP poller. It is responsible for determining what to collect, starting child processes that perform collections and load data into database, and providing all the <i>how-to rules</i> to these processes.
child poller	A child process that is spawned by the parent poller. It is responsible for the actual SNMP collection. A child poller deals with only one agent (node) and one table at a time.

bcp_gateway	A child process that is spawned by the parent poller. It is responsible for loading collected data into the database. For more information on bcp_gateway see <a href="#">page 300</a> .
data database	A database server where the poller stores collected data, and from where the poller reads table definitions.
topology database	A database server from which the poller reads all the node information; that is, views, types, and nodes.
caching poller	A feature that enables the poller to run without a database. See <a href="#">Directory Structure</a> on page 304.
instance of poller or Collector ID	A collection process for a given interval. The instance of the poller is identified by the interval. See <a href="#">File Locks</a> on page 298 for more information.

## mw\_collect Command

- The `mw_collect` command invokes the `mw_collect` parent poller, which controls the collection process. The parent poller is responsible for the following:
  - It determines what information to collect and from which device. It reads the *what to collect* information from the database and records this data in local files. If access to the database is not available the parent poller reads the local files instead.
  - It has the ability to start multiple children.
  - It starts and controls child processes that collect data and load it into the database.
  - It supplies rules on how to process the data to child processes.
  - It starts the raw-to-rate process on the collected data.

Each child poller collects a single node and table combination.

- A child poller does not have access to the database.
- A child poller is responsible for actual collection of data, but the parent must tell the child what methods to use.

When the child poller finishes, the data that is collected is appended to the latest existing holding file for loading into the PI database. The latest holding file is loaded into the database when the file is at least 500,000 bytes or every child poller has finished. See [Directory Structure](#) on page 304 for a discussion of the directories in which these holding files are stored before they are loaded into the PI database.

`mw_collect` calls the following procedures for each table and destination database pair:

- The `bcp_gateway` procedure updates property and data tables in the PI database.
- The `trendpm` procedure generates rate data in the PI database after `mw_collect` collects all the data and stores it in the database for a particular table.

## Configuration File

The configuration file for `mw_collect` contains additional parameters that you can set for `mw_collect` to process. You only need to set these parameters in special circumstances such as at the direction of HP Technical Support. The `-f` option specifies the name of the file. See [Syntax](#) on page 277 for the rules about the file and [Parameters](#) on page 293 for the descriptions of the parameters in the file.

Note that the configuration file requires only the parameters that you want to set.

## Syntax

The following rules apply to the `mw_collect` configuration file.

- The parameters can be in any case.
- The values may be case sensitive.
- The parameters can appear in any order.
- The format is *parameter = value*.

This list of parameters shows the syntax for them in alphabetical order. They appear in mixed case in this list for readability.

**BackupFailedBatch** = 1  
**BatchSize** = *num\_rows*  
**CategoryName** = *name*  
**CollectDefName** = *file\_name*  
**CollectHome** = *directory\_name*  
**DataDB** = *server\_name*  
**DontLock** = 1  
**DSQUERY** = *value*  
**DumpBcpG** = 1  
**Element** = *name* [, *option ...*]  
**Feeder** = *value*  
**GatewayWatchdog** = *num\_seconds*  
**GroupName** = *name*  
**IgnoreSnmpDB** = 1  
**KeepData** = 1  
**max\_tran\_row\_cnt** = *num\_rows*  
**Method** = *value*  
**NoNewObjects** = 1  
**PollTime** = *num\_seconds*  
**TopDB** = *server\_name*  
**TrendpmRetryCount** = *num\_retries*  
**TrendpmRetryInterval** = *num\_seconds*

## Parameters

This list of parameters provides the descriptions for them. You can use any case. They appear in this list in mixed case for readability.

**Table 18 mw\_collect Configuration File Parameters**

Parameter	Description
<b>BackupFailedBatch</b>	Enables the <code>bcp_gateway</code> process to back up the failed batch of collected data for reviewing the failure line in the batch.  Note that you must run the collection in ASCII mode using <b>-a</b> option of <code>mw_collect</code> command for this parameter to work.
<b>BatchSize</b>	Specifies the size of the batch for <code>bcp_gateway</code> to process, which is the number of rows.
<b>CategoryName</b>	Specifies the category of the group to be polled. If you set this parameter, you must also set the <b>GroupName</b> parameter.
<b>CollectDefName</b>	Specifies the name of the <code>CollectDef</code> file that contains the cached collection definitions.
<b>CollectHome</b>	overrides the value in the <code>\$COLLECT_HOME</code> environment variable for this process. See <a href="#">Directory Structure</a> on page 304 for more information.
<b>DataDB</b>	Specifies the server name for the Data database. See <a href="#">Terms</a> on page 290 for more information.
<b>DontLock</b>	Specifies that <code>mw_collect</code> not lock the files in the <code>\$COLLECT_HOME</code> directory. See <a href="#">File Locks</a> on page 298 for more information.  The default is <b>0</b> , which is to lock the files. Use <b>1</b> to set, which is to not lock the files.
<b>DSQUERY</b>	overrides the setting of the default database.

**Table 18 mw\_collect Configuration File Parameters**

Parameter	Description
<b>DumpBcpG</b>	<p>Specifies that <code>mw_collect</code> write the BCP rules to standard output without processing, which means that it does not load the data into the database.</p> <p>The default is <b>0</b>. Use <b>1</b> to set.</p>
<b>Element</b>	<p>Specifies an element that <code>mw_collect</code> passes to a child processor. The child processor determines the format for this statement; for example, you can use the following format to pass a host name to <code>dpipe_snmp</code>:</p> <p><i>element = name, community, version, retry, timeout</i></p> <p>The configuration file may have multiple <b>element</b> parameters in it.</p>
<b>Feeder</b>	<p>Specifies the name of the Feeder, which is the method that defines how the data is collected. Values are:</p> <ul style="list-style-type: none"> <li>• EE: <code>ee_collect</code></li> <li>• HST: <code>vantage_collect</code></li> <li>• MW: <code>mw_collect</code></li> <li>• SR: <code>pa_collect</code></li> </ul>
<b>GatewayWatchdog</b>	<p>Specifies the timeout of the gateway watchdog in seconds. Use <b>0</b> for no timeout.</p> <p>This parameter applies only to Windows systems.</p> <p>The default is <b>900</b> seconds.</p>
<b>GroupName</b>	<p>Specifies the name of the group to be polled. If you set this parameter, you must also set the <b>CategoryName</b> parameter.</p>

**Table 18 mw\_collect Configuration File Parameters**

Parameter	Description
<b>IgnoreSnmpDB</b>	<p>Specifies that <code>mw_collect</code> ignore the SNMP version value from the database.</p> <p>You can use this parameter to turn off the default, which is equivalent to the <code>-j</code> option in earlier versions, when you set the value to <b>1</b>.</p> <p>The default is <b>0</b>. Use <b>1</b> to ignore.</p>
<b>KeepData</b>	<p>Specifies that <code>bcp_gateway</code> keep the input data files instead of deleting them by default.</p> <p>The default is <b>0</b>. Use <b>1</b> to keep the files.</p>
<b>max_tran_row_cnt</b>	<p>Passes this value, which is the <b>@max_tran_row_cnt</b> parameter, to <code>trendpm</code>. See <b>@max_tran_row_cnt</b> on <a href="#">page 478</a> for more information.</p> <p>HP strongly recommends that you use the default value. If this value is too low, it may affect performance; if this value is too high, it may affect concurrency.</p>
<b>Method</b>	<p>Relates to the <code>ByVarInfo</code> <b>TEEL</b> statement; it specifies how to process the data. Valid values are:</p> <ul style="list-style-type: none"> <li>• <b>MTD_HST</b>: Specifies that the poller collects data from the intersection of the objects between the <i>property_table</i> value in the <b>ByVarInfo</b> statement and the collection group.</li> <li>• <b>MTD_HST2</b>: Specifies that the poller collects data from the intersection of the objects between the <i>property_table</i> value in the <b>ByVarInfo</b> statement and the collection group along with the rest of the objects in the collection group.</li> </ul>



**Table 18 mw\_collect Configuration File Parameters**

Parameter	Description
<b>NoNewObjects</b>	Specifies that <code>mw_collect</code> only load the data, and not insert any new managed objects into the corresponding property table. The default is <b>0</b> . Use <b>1</b> to set.
<b>PollTime</b>	Passes the start time of the polling cycle value to a child process. This value is the number of seconds since midnight of January 1, 1970.
<b>TopDB</b>	Specifies the server name for the Topology database. See <a href="#">Terms</a> on page 290 for more information.
<b>TrendpmRetryCount</b>	Passes this value, which is the <b>@retry_count</b> parameter, to <code>trendpm</code> . See <code>@retry_count</code> on <a href="#">page 478</a> for more information.
<b>TrendpmRetryInterval</b>	Passes this value, which is the <b>@retry_interval</b> parameter, to <code>trendpm</code> . The value for the <b>-N</b> option on the <code>mw_collect</code> command line overrides this setting during processing. For more information, see the <b>-N</b> option on <a href="#">page 278</a> or <code>@retry_interval</code> on <a href="#">page 478</a> .

## By-Variables

In the current PI release, the by-variable replaces the current unique key, which is the combination of the `target_name` and `table_key` columns. A combination of by-variable columns in the property table uniquely defines an object. For more information, see [By-Variables](#) on page 509.

## File Locks

`mw_collect` enables only one instance of collection for a particular Collection ID to run at the same time. Furthermore, only one `bcp_gateway` process runs at a time for a particular Collection ID, `table_name`, and data database combination.

When the parent poller `mw_collect` runs, it attempts to place an exclusive lock on the running file in the `$COLLECT_HOME/feeder/collection_id` directory. If the lock fails, the poller assumes that the previous instance of the poller is still running and exits. The parent poller releases the lock on the running file when every collector child finishes. Note that the lock releases before the poller finishes loading all the data into the database so that the polling continues even though the data loading portion is not finished.

Similarly, when `mw_collect` runs a `bcp_gateway` process, it tries to place a lock on `$COLLECT_HOME/feeder/collection_id/tablename_dbname.running`. If the lock fails, the parent poller assumes that the previous job is still running and does not attempt to start the `bcp_gateway` process. The data is stored locally.

## Local Storage of Data

`mw_collect` provides the ability to perform collection without a database. It saves the collection definition locally.

After the poller connects to the database, it refreshes the database with the saved information. Note that the poller always uses information from the database when it is available, and does not then rely on locally saved information.

The following table summarizes this feature:

Process	Failure Point	Result Cache Features
Parent Poller	Failed to connect at startup.	Use cached information for collection. Store data locally.
Parent Poller	During initialization, lost connection.	Use cached information (in most cases). Store data locally.
Parent Poller	Determined during initialization that database is full or transaction log is full.	Store data locally.

Process	Failure Point	Result Cache Features
bcp_gateway	Failed to connect to a given database.	Returns code to calling parent thread. Parent does not attempt to load data into this database during this cycle. Data is stored locally.
bcp_gateway	Lost connection to database, database is full, or transaction log is full.	<p>Removes successfully loaded data from the input file; deletes entire file when operating in ASCII mode (-a).</p> <p>Returns code to calling parent thread. (see above)</p> <p>Depending on when the database connection was lost or the database became full during the bcp_gateway process, the removal of successfully loaded data may not be accurate. This may cause insertion of duplicate data into raw data, which fails (if there are indexes) and generate error messages in trend.log.</p> <p>bcp_gateway ignores this error message and continues loading new data.</p>
bcp_gateway	Failed loading data for other reason than above.	<p>If you enabled the BackupFailedBatch process, bcp_gateway creates a backup of the failed batch file on the filesystem with a .failed extension.</p> <p>This backup process works only for ASCII batch files.</p>

## Interval Polling

`mw_collect` is mainly used to poll for SNMP data at regular intervals. When invoked, `mw_collect` reads the polling control table in the database for the list of instructions whose intervals match the value of the `-i` option on the `mw_collect` command line. Each entry in the list specifies a MIB table to be collected and the device group, specific device, or specific instance from a device to be polled for the table.

This polling policy information is entered automatically when you install a report pack. You can also enter or modify polling policy information through the Polling Policy Manager. See the *HP Performance Insight Administration Guide* for information about using Package Manager to install a report pack and for information about specifying your own polling policies.

`mw_collect` queries the database for the list of fields in the table to be collected. Then, for each node to be polled, it obtains the node SNMP Read Community string from the database and polls the node for those fields. After all target devices have been polled, the returned data is written to the database and `mw_collect` exits.

## Distributed Polling

To distribute the polling load to the most efficient locations, you can install `mw_collect` on any system in your network.

In the simplest case, all polling instructions, node information, and data tables are on a single system and all polling is done from that system. It is also possible to maintain polling instructions on one system; store target device names, their community strings, and polling View and Type lists on a second system; run polling from a third system; store the returned data on a fourth system; and run reports against the returned data from a fifth system. This is the most extensive example of distributing the application. However, any combination from the simplest to the most extensive is possible and can be configured according to your requirements.

The `trendtimer` program invokes `mw_collect`. By default, `mw_collect` uses the default database entry in the `systems.xml` file to determine the database to be queried for polling instructions. Since any system can access a central database from anywhere on the network to get its list of polling instructions,

you can define the polling policy for an entire network from one central location. As an alternative, you can define polling locally for each region within that region.

You can access information about the nodes to poll, including their community strings and their membership in polling View and Type lists, in Polling Policy Manager. See the *HP Performance Insight Administration Guide* for more information.

The polling instructions include the name of the system that is intended to do the polling.

- When `mw_collect` is run with the `-n` option, it checks this field to see if the hostname matches the name of the system on which it is running. If they do not match, `mw_collect` does not start the poll.
- The `-H` option is run with the `-n` option to specify an alternate poller name. When `-H` option is used, then `mw_collect` compares the Poll From field in the polling policy to the *alternate\_poller\_name*.
- Without the `-n` option, `mw_collect` ignores the hostname field and runs the poll regardless of any system name specified there.

The group to which the user who defined a polling request belongs also specifies the name of the database to which the collected data is to be written. After the polls are completed, `mw_collect` loads the collected data into that database.

## Direct Polling

You can use `mw_collect` to immediately collect particular MIB table values from a particular node on demand by executing `mw_collect` with the `-t` and `-h` options from the command line.

## Multiple Pollers

If you have multiple pollers, it is recommended that you edit the `trendtimer.sched` file and add the `-X` option to all `mw_collect` commands; this prevents the pollers from running the raw to delta processing invoked by the `trendpm` command. If you do not do this, you could see performance issues.

You should edit the file for all of the pollers *except* one. For example, if you have a satellite system polling (that is, PI's Performance Manager component is installed on this system) with three remote pollers, edit the file for all of the pollers, leaving the satellite system's file as is. If the satellite system is not polling, then edit the file for two of the pollers leaving one as is.

To turn off raw to delta processing:

- 1 Edit the `trendtimer.sched` file.
- 2 Add the `-X` option to all of the `mw_collect` lines in the file.

For example, if the lines in your file read as follows:

```
5 - - {DPIPE_HOME}/bin/mw_collect -n -i 5 -K 1
10 - - {DPIPE_HOME}/bin/mw_collect -n -i 10 -K 1
15 - - {DPIPE_HOME}/bin/mw_collect -n -i 15 -K 1
20 - - {DPIPE_HOME}/bin/mw_collect -n -i 20 -K 1
60 - - {DPIPE_HOME}/bin/mw_collect -n -i 60 -K 1
24:00+1:00 - - {DPIPE_HOME}/bin/mw_collect -n -i 1440 -K 1
```

Edit the lines to look like the following:

```
5 - - {DPIPE_HOME}/bin/mw_collect -n -i 5 -K 1 -X
10 - - {DPIPE_HOME}/bin/mw_collect -n -i 10 -K 1 -X
15 - - {DPIPE_HOME}/bin/mw_collect -n -i 15 -K 1 -X
20 - - {DPIPE_HOME}/bin/mw_collect -n -i 20 -K 1 -X
60 - - {DPIPE_HOME}/bin/mw_collect -n -i 60 -K 1 -X
24:00+1:00 - - {DPIPE_HOME}/bin/mw_collect -n -i 1440 -K 1 -X
```

- 3 Save the file.

## Log File

`mw_collect` makes log entries in the file `trend.log`. This file is in either the `$TREND_LOG` directory, if the `$TREND_LOG` environment variable is set, or the `$DPIPE_HOME/log` directory if the `$TREND_LOG` environment variable is not set.

## Directory Structure

Temporary files, data files, and cached polling policy files that `mw_collect` uses are stored under the following directory:

**`$COLLECT_HOME`**/*feeder\_name*/*collection\_id*

where:

**`$COLLECT_HOME`** Is the directory that the `$COLLECT_HOME` environment variable points to (typically, `$DPIPE_HOME/collect`).

*feeder\_name* Always has the value `MW` for instances of `mw_collect`.

*collection\_id* Identifies the collection ID, which is a number corresponding to the value of the `-i` option on the `mw_collect` command.

One of the elements under this directory is the `BCP` directory, which is where data that is ready to be loaded into the database is stored. In this case, the format for this directory is as follows:

**`$COLLECT_HOME`**/*feeder\_name*/*collection\_id*/`BCP`



# Examples

The following examples show the format of the `mw_collect` command that you can use for the different methods of polling.

## Interval Polling

### Example 1

To have `mw_collect` start all collection requests with an interval id of **10**, invoke `mw_collect` with the command line:

```
mw_collect -i 10
```

### Example 2

To have `mw_collect` start all collection requests with an interval id of **5** on the machine on which `mw_collect` is installed, putting a maximum of 20 MIB variables in each packet, trying to poll the node a maximum of 3 times, and waiting 2 seconds between polling attempts, invoke `mw_collect` with the command line:

```
mw_collect -n -i 5 -p 20 -o 2 -r 3
```

Interval polling entries for `mw_collect` are in the following file:

```
$DPIPE_HOME/lib/trendtimer.sched
```

## Direct Polling

To have `mw_collect` poll a node called `foo.HP.com` for the MIB-II `ifEntry` table, use the command:

```
mw_collect -h foo.HP.com -t mib-II_ifEntry -i 59
```

In this example, `mw_collect` uses the `-h` and `-t` options to collect the data, and uses the `-i` option, which is the Collection ID, for the file names; see [Directory Structure](#) on page 304. In this case, `mw_collect` does not read from the database polling policy that corresponds to interval **59**.



## 20 node\_manager

The `node_manager` command is a stand-alone utility that enables you to manage nodes, types, and views. It also enables you to manage all relevant SNMP properties for nodes.

There is another method to add nodes to a PI system; you can import nodes from a file or define nodes using Polling Policy Manager. See the *HP Performance Insight Administration Guide* for more information.

### Requirements or Restrictions

- An error occurs if more than one of the following options appears on the command line at the same time: **-import**, **-export**, **-delete**, **-remove**, **-insert\_only**.
- An error occurs if the **-file** option does not appear with one of the following options on the command line at the same time: **-import**, **-delete**, **-remove**.

# Syntax

The `node_manager` command uses the following syntax:

```
node_manager  [ -database db_name ]  
                [ -debug debug_level ]  
                [ { -delete  
                  { -export  
                  { -import  
                  { -remove } ]  
                [ -file file_name ]  
                [ -help ]  
                [ -insert_only ]  
                [ { -v  
                  { -version } ]
```

# Options

The `node_manager` command has the following options:

**-database** This option identifies the database where the changes occur. The database must appear in the list of available database servers. See the Web Access Server in the *HP Performance Insight Administration Guide* for more information about adding database servers to the list. The default is the database identified as the default in the database server list.

<b>-debug</b>	<p>Use this option to set the debug output level. Values of <b>1</b>, <b>2</b>, or <b>3</b> are valid. The higher the number, the more detailed the information. Debug output writes to standard output. Only use this option for testing in coordination with HP Technical Support due to the additional overhead it places on <code>node_manager</code>.</p> <p>The default is no debug output.</p>
<b>-delete</b>	<p>Use this option to delete one or more nodes from the system. It requires the <b>-file</b> option to identify the file that contains the list of nodes to delete. This option ignores all other parameters in the file except the <code>node_name</code>. See <a href="#">Delete</a> on page 311 for more information.</p> <p>This option cannot appear on the command line when the <b>-import</b>, <b>-export</b>, or <b>-remove</b> option appears on the command line.</p>
<b>-export</b>	<p>Use this option to generate a file containing the nodes in the current node catalog. The output file also contains any type or view assignments for the nodes. Use the <b>-file</b> option to specify the output file name; otherwise, <code>node_manager</code> writes the data to standard output. If the specified output file already exists, the system overwrites the file. See <a href="#">ASCII File</a> on page 312 for the format of the file. See <a href="#">Export</a> on page 312 for more information.</p> <p>This option cannot appear on the command line when the <b>-import</b>, <b>-delete</b>, or <b>-remove</b> option appears on the command line.</p>
<b>-file</b>	<p>This option identifies the file name, which is the text file that contains the information about the nodes, types, and views to import, delete, or remove. If the file is not in the current working directory, you must specify the fully qualified path to the file. See <a href="#">ASCII File</a> on page 312 for details on setting up this file.</p> <p>This is a required option when the <b>-import</b>, <b>-delete</b>, or <b>-remove</b> option appears on the command line.</p>
<b>-help</b>	<p>This option is the help option, which displays the command-line options for the <code>node_manager</code> command.</p>

- import** Use this option to import nodes, assign the corresponding SNMP attributes, and populate any type or view lists. It requires the **-file** option to identify the file that contains the list of nodes to import. See [Import](#) on page 311 for more information.
- This option cannot appear on the command line when the **-export**, **-delete**, or **-remove** option appears on the command line.
- insert\_only** Use this option with the **-import** option to add nodes that do not already exist. If any of the nodes already exist, *node\_manager* skips them and not update them.
- remove** Use this option to remove one or more nodes from one or more type or view lists. It requires the **-file** option to identify the file that contains the list of nodes with the corresponding type or view lists to remove. This option ignores all other parameters in the file except the *node\_name*, *type\_name*, and *view\_name*. See [Remove](#) on page 311 for more information.
- This option cannot appear on the command line when the **-import**, **-delete**, or **-export** option appears on the command line.
- v** Use this option to display the current version of **node\_manager**.
- This option is in UPPERCASE.
- version** Use this option to display the current version of **node\_manager**.

# Usage Notes

This section provides a brief description of the different modes available, the parameters for the associated ASCII file, and using the `node_manager` command.

## Modes of Operation

The `node_manager` command has four modes of operation: import, delete, remove, and export.

### Import

The *import* mode provides the ability to add nodes to the system. It also provides the ability to assign the SNMP attributes to the node as well as add the node to specified type and view lists.

### Delete

The *delete* mode provides the ability to delete all the nodes in the associated file from the system. When `node_manager` deletes a node from the system, it deletes the node from all enumerated group list memberships. It does not immediately delete any node that has dependencies; it marks those nodes for deletion at a later time. A node has a dependency when a property table has a record that references that node. When there are no more records that reference that node, `node_manager` deletes that node the next time it runs in delete mode.

In this mode, `node_manager` uses only the *node\_name* in the associated file and ignores all other attributes. Note, however, that each node in the file must have a value or a placeholder for every attribute associated with that node. See [ASCII File](#) on page 312 for more information about the file.

### Remove

The *remove* mode provides the ability to remove the nodes in the associated file from the specified type and view lists. Since a node can be a member of multiple type and view lists, the node may appear multiple times in the file.

In this mode, `node_manager` uses the `node_name`, `type_name`, and `view_name` parameters in the associated file and ignores all other attributes. Note, however, that each node in the file must have a value or a placeholder for every attribute associated with that node. See [ASCII File](#) on page 312 for more information about the file.

## Export

The *export* mode provides the ability to create a file of the existing nodes on the system.

## ASCII File

Three of the four modes of operation require an ASCII file. Some of the modes require only a subset of the parameters, but each record in the file must have a valid value or a placeholder for every parameter in the record. The ASCII file that contains a list of nodes must be in the following format:

*node\_name, read\_community, write\_community, type\_name, view\_name, snmp\_v1\_flag, snmp\_v2\_flag, node\_status, desc, snmp\_profile, port\_num, num\_retries, timeout\_sec, num\_oids, get\_bulk\_size*

Use the tab delimiter as placeholder for any parameter that is missing; otherwise, the system skips the record. The descriptions for the parameters in the file follow:

<i>node_name</i>	<p>This parameter specifies the name of the node or the IP address for the node.</p> <p>This parameter must appear in the file for the <b>-import</b>, <b>-delete</b>, or <b>-remove</b> mode.</p>
<i>read_community</i>	<p>This parameter specifies the read community string for the node. If the profile for the read and write community string pair does not exist, the system creates it by concatenating the specified read and write community strings from this file. You can view the list of profiles from the Edit Community String Profiles window in Polling Policy Manager.</p> <p>The default value is <b>public</b>.</p>



<i>write_community</i>	<p>This parameter specifies the write community string for the node. If the profile for the read and write community string pair does not exist, the system creates it by concatenating the specified read and write community strings from this file. You can view the list of profiles from the Edit Community String Profiles window in Polling Policy Manager.</p> <p>The default value is <b>private</b>.</p>
<i>type_name</i>	<p>This parameter specifies the name of the type list that contains the list of nodes to poll. If you specify this field in the file, you must specify the group name. In <b>-import</b> mode, if this name does not exist, <b>node_manager</b> creates a type list with the specified name. This is the same name as the Group Name on the Edit Type Group window from Polling Policy Manager.</p> <p>If this parameter is blank, <b>node_manager</b> does not assign this node to a type list group or remove this node from a type list group.</p>
<i>view_name</i>	<p>This parameter specifies the name of the view list that contains the list of nodes to poll. If you specify this field in the file, you must specify the group name. In <b>-import</b> mode, if this name does not exist, <b>node_manager</b> creates a view list with the specified name. This is the same name as the Group Name on the Edit View Group window from Polling Policy Manager.</p> <p>If this parameter is blank, <b>node_manager</b> does not assign this node to a view list group or remove this node from a view list group.</p>
<i>snmp_v1_flag</i>	<p>This parameter specifies that the node supports the SNMP protocol when the value is <b>1</b>.</p> <p>The valid values are <b>0</b> and <b>1</b>.</p> <p>The default value is <b>1</b>.</p>

<i>snmp_v2_flag</i>	<p>This parameter specifies that the node supports the SNMPv2 protocol when the value is <b>1</b>.</p> <p>The valid values are <b>0</b> and <b>1</b>.</p> <p>The default value is <b>0</b>.</p> <p><b>Note:</b> If are unsure if a node supports SNMPv2, leave this parameter blank.</p>
<i>node_status</i>	<p>This parameter indicates the status of the node upon import.</p> <p>The valid values are <b>0</b> for inactive and <b>1</b> for active.</p> <p>The default value is <b>1</b>.</p>
<i>desc</i>	<p>This parameter provides a description for the node. If this parameter is blank, the value is blank, which appears as a space in the export file. The length can be up to 255 characters.</p>
<i>snmp_profile</i>	<p>This parameter specifies the name of the SNMP profile assigned to the node. If this profile exists, <b>node_manager</b> assigns the profile to the node and ignore the following parameters; otherwise, <b>node_manager</b> creates the profile using the following parameters and then assign the profile to the node.</p> <p>When this parameter is blank, the value is <b>default</b>, and <b>node_manager</b> ignores the following parameters.</p>
<i>port_num</i>	<p>This parameter specifies the SNMP port number for the new SNMP profile.</p> <p>The default value is <b>161</b>.</p>
<i>num_retries</i>	<p>This parameter specifies the number of retries for the new SNMP profile.</p> <p>The default value is <b>5</b>.</p>
<i>timeout_sec</i>	<p>This parameter specifies the number of milliseconds allowed for timing out on an SNMP request.</p> <p>The default value is <b>1000</b>.</p>

<i>num_oids</i>	This parameter specifies the number of OIDs to include in a PDU for the new SNMP profile. The default value is <b>20</b> .
<i>get_bulk_size</i>	This parameter specifies the size to use for a get bulk SNMPv2 request. The default value is <b>50</b> .



If any of the parameters contain invalid values, an error message occurs and the process terminates at that point. The system does not check the rest of the file for additional errors. If an error occurs, verify that all records contain valid values before you resubmit the file; otherwise, an error occurs again. Verify that there are no blank lines; otherwise, an error message occurs.

## Using the `node_manager` Command

This section shows some formats of the command for the various modes.

- If you enter the `node_manager` command without any options, the system displays an error message followed by the help information.
- To display the syntax and options for this command, enter:  
**`node_manager -help`**
- To display the version information for this command, enter:  
**`node_manager -V`** or **`node_manager -version`**
- To import a list of nodes, enter the following command:

**`node_manager -import -file file_name`**

*file\_name* is the name of the ASCII file that contains the list of nodes with their corresponding attributes to import.

- To import a list of nodes and only add those nodes that do not already exist, enter the following command:

**`node_manager -import -insert_only -file file_name`**

*file\_name* is the name of the ASCII file that contains the list of nodes with their corresponding attributes to import.

- To delete a list of nodes, enter the following command:

**node\_manager -delete -file *file\_name***

*file\_name* is the name of the ASCII file that contains the list of nodes to delete.

- To remove a list of nodes from type and view lists, enter the following command:

**node\_manager -remove -file *file\_name***

*file\_name* is the name of the ASCII file that contains the list of nodes with their corresponding type and view list names to remove.

- To export a list of nodes to the screen, enter the following command:

**node\_manager -export**

- To export a list of nodes to a file, enter the following command:

**node\_manager -export -file *file\_name***

*file\_name* is the name of the ASCII file that contains the list of existing nodes on the system.

# Examples

This section has examples that show each mode of `node_manager`. They have the following characteristics:

- All examples see the sample file used in the import example.
- Any information that appears about the **-file** option for one example also applies to other examples, such as the need for a qualified path.

## Import Examples

### Example 1

To import a list of nodes from a file named `node_in.txt`, enter the following command:

```
node_manager -import -file node_in.txt
```

The following is an example of the contents for the input file, `node_in.txt`:

```
tst_a,,,,,,,,,
tst_b,,,tp_lst1,vw_lst2,,,dsc2,,,,,
tst_c,public,private,,,1,0,1,dsc3,smp_1,161,10,500,20,50
tst_d,public,private,,,1,0,1, ,default,161,15,100,20,50
tst_e,public,private,tp_lst1,,1,0,1,dsc5,default,,,,,
tst_f,public,private,,vw_lst2,1,0,1, ,default,,,,,
tst_g,,,,,,,,,

```

### Example 2

To add nodes that do not already exist to the node list on the database named **powder2** from a list of nodes in a file named `n_in.txt` on the database named **bear**, which is on a different system, enter the following command:

```
node_manager -import -file n_in.txt -database bear  
-insert_only
```

## Export Examples

### Example 1

If you want to export a list of nodes to a file named `node_current.txt`, enter the following command:

```
node_manager -export -file node_current.txt
```

The following is an example of the contents of the output file, `node_current.txt`:

```
tst_b,public,private,tp_lst1,,1,0,1,dsc2,default,161,5,1000,20,50
tst_e,public,private,tp_lst1,,1,0,1,dsc5,default,161,5,1000,20,50
tst_b,public,private,,vw_lst2,1,0,1,dsc2,default,161,5,1000,20,50
tst_f,public,private,,vw_lst2,1,0,1,
,default,161,5,1000,20,50
tst_a,public,private,,,1,0,1, ,default,161,5,1000,20,50
tst_c,public,private,,,1,0,1,dsc3,smp_1,161,10,500,20,50
tst_d,public,private,,,1,0,1, ,default,161,5,1000,20,50
tst_g,public,private,,,1,0,1, ,default,161,5,1000,20,50
```

Note that the node `tst_b` appears in the list twice because it has both type and view lists associated with it; and node `tst_c` has different SNMP properties because it has a different SNMP profile, but `tst_d` does not because it uses the default SNMP profile.

### Example 2

To export a list of nodes that are on the database named `bear` to a file named `node_bear.txt`, which is on the system entering the command and has the database named `powder2`, enter the following command:

```
node_manager -export -file node_bear.txt -database bear
```

## Remove Examples

### Example 1

If you want to remove a list of nodes using a file named `node_rem.txt`, enter the following command:

```
node_manager -remove -file node_rem.txt
```

An example of the contents of the input file, `node_rem.txt`, are the following:

```
tst_e,public,private,tp_lst1,,1,0,1,,default,,,,,  
tst_f,public,private,,vw_lst2,1,0,1,,default,,,,,
```

### Example 2

Note that you need to fully qualify the path for the file if it is in a different directory. For example, if the file were in the **lists** directory from the **D** drive on a Windows machine, you would enter the following command:

```
node_manager -remove -file d:\lists\node_rem.txt
```

## Delete Example

To delete a list of nodes using a file named `node_out.txt`, enter the following command:

```
node_manager -delete -file node_out.txt
```

An example of the contents of the input file, `node_out.txt`, are the following:

```
tst_b,,,tp_lst1,vw_lst2,,,,dsc2,,,,,  
tst_c,public,private,,,1,0,1,dsc3,smp_1,161,10,500,20,50  
tst_d,public,private,,,1,0,1,,default,,,,,  
tst_g,,,,,,,,,,,,,
```

# Error Messages

This section describes some of the messages that can occur from `node_manager`. Each message has the following format:

- A brief description about why the message appears. Each new message description starts with a check box.
  - The actual message that appears with parameters for any information that may be missing.
  - A suggestion about the action to do so that the message does not appear again.
- If the following error message appears, there is a command-line syntax error. This means that required options are missing or there are multiple mode options on the same line.

```
A mode of operation is required.  
Exiting program with code 1.
```

Verify that the command specifies the mode option or that the following mode options do not appear on the command line at the same time:

**-import, -export, -delete, -remove.**

- If the following error message appears, the database specified with the **-database** option on the command line does not exist.

```
Connection URL not found.  
Exiting program with code 3.
```

Verify the spelling of the database name. If the spelling is correct, you can add the database using the Web Access Server.

- If the following error message appears, the specified record in the file has a syntax error.

```
Exception: Wrong number of delimiters on line n.  
Exiting program with code 4.
```

Check the specified record in the file and fix the error. Verify that the record has either a value or a placeholder for all 15 parameters or that the line is not blank.



- ❑ If the following error message appears, the file name specified on the command line in the **-file** *file\_name* option does not exist.

File *file\_name* does not exist.  
Exiting program with code 2.

Verify the spelling of the file name or that the file exists in the specified location. You may have to supply a fully qualified path with the file name.

- ❑ If the following informational message appears, the record for the specified *node\_name* in the file is missing a type or view group list name. The system is skipping the record.

Node *node\_name* does not have a type or view specified - skipping.

Check the specified record in the file and fix the error. Verify that the record has either an existing type or view group list name.

- ❑ If the following informational message appears, a record in the file is providing an incorrect type group list name. The system is skipping the record.

The type group *type\_name* does not exist - skipping.

Check the record in the file for the specified *type\_name*. Enter the correct type group list name for the record.

- ❑ If the following informational message appears, a record in the file is providing an incorrect view group list name. The system is skipping the record.

The view group *view\_name* does not exist - skipping.

Check the record in the file for the specified *view\_name*. Enter the correct view group list name for the record.



## 21 ovpi\_bulk\_copy

You can use the `ovpi_bulk_copy` command to bulk load or extract data on a PI system.

### Requirements and Restrictions

- An error occurs if more than one of the following options appears on the command line at the same time: **-infile**, **-outfile**.
- An error occurs if the **-table** option does not appear on the command line.

# Syntax

The `ovpi_bulk_copy` command uses the following syntax:

```
ovpi_bulk_copy    [ -bcp "option_list" ]  
                  [ -database server_name ]  
                  [ -debug debug_level ]  
                  [ -ftc "char" ]  
                  [ -help ]  
                  [ -keepctl ]  
                  { -infile } file_name  
                  { -outfile }  
                  [ -logfile result_file ]  
                  [ -sqlldr "option_list" ]  
                  -table table_name  
                  [ -timefile timefile_name ]  
                  [ -v ]  
                  [ -version ]
```

# Options

The `ovpi_bulk_copy` command has the following options:

- bcp** This option specifies the command line options for the Sybase `bcp` command. See the Sybase documentation for the valid options.  
You must use double quotation marks to enclose the options.  
This option may appear on the command line with the **-sqlldr** option.
- database** This option specifies the name of the database server that imports or exports data.  
The default for this option is the default database server.
- debug** Use this option to set the debug output level. The higher the number, the more detailed the information. Debug output writes to standard out. Use this option only for testing in coordination with HP Technical Support due to the additional overhead it places on `ovpi_bulk_copy`.  
The default is no debug output.
- ftc** This option specifies the field delimiter, which is the character that separates the fields, in the input or output file. This option enables you to identify the character in the file so that it is the same on all database servers.  
You must use double quotation marks to enclose the character.  
The default for this option is “|” tab delimiter.
- help** This option is the help option, which displays the command-line syntax for the `ovpi_bulk_copy` command.  
This option overrides all other options on the command line.

- keepctl** Use this option to preserve the Oracle Loader control files under `$DPIPE_HOME/tmp` directory. You can use this option only if you are using the **-infile** option with `ovpi_bulk_copy` command. **-keepctl** option is not available if you are using **-outfile** option with `ovpi_bulk_copy` command.  
Note that this option is available only for Oracle databases.
- infile** This option specifies the name of the file that contains the data to be loaded into the specified table. This file name may contain a relative or fully qualified path; it may also contain the substitution keyword, `{DEVENDOR}`. See [Usage Notes](#) on page 327 for more information.  
This option or the **-outfile** option is a required option.
- logfile** This option specifies the name of the file that contains any output generated from the `bcp` or `sqlldr` command. If this name is not a fully qualified path, the file is in the current working directory. If you do not specify a name, the results of the output are in standard out.
- outfile** This option specifies the name of the file that contains the data from the specified table. This file name may contain a relative or fully qualified path; it may also contain the substitution keyword, `{DEVENDOR}`. See [Usage Notes](#) on page 327 for more information.  
This option or the **-infile** option is a required option.
- sqlldr** This option specifies the command line options for the Oracle `sqlldr` command. See the Oracle documentation for the valid options.  
You must use double quotation marks to enclose the options.  
This option may appear on the command line with the **-bcp** option.
- table** This option specifies the name of the table to import or export.

- timefile**      Use this option to upload data collected within a specific period. This option is mandatory for Oracle databases. The `timefile` contains information about the start time and the end time of the data uploaded in PI database.  
                  For example, if you want to upload the data collected from Jan 11 2008 till Feb 11 2008, specify the start time and the end time in the `timefile` using the following format:  
                  Format: "**start time | end time |**"  
                  Example: **DD-MM-YYYY HH24 | DD-MM-YYYY HH24 |**
- v**              Use this option to display the current version of the `ovpi_bulk_copy` utility. This option overrides all other options on the command line, except the help option.  
                  This option is in UPPERCASE.
- version**      Use this option to display the current version of the `ovpi_bulk_copy` utility. This option overrides all other options on the command line, except the help option.

## Usage Notes

This command dynamically determines whether to run the Sybase `bcp` command or the Oracle `sqlldr` command, depending on the database type.

You can use the substitution keyword, **{DBVENDOR}**, in a path when the database type is part of the path. For example, if you want to import data from a file with the name `load_data_1` in the `system1/ Sybase/load` directory on a Sybase database or in the `system1/Oracle/load` directory on an Oracle database, you can use the following path and file name with the **-infile** option:

```
-infile system1/{DBVENDOR}/load/load_data_1
```

## Using the `ovpi_bulk_copy` Command

This section shows some formats of the command for the various tasks.

- If you enter the `ovpi_bulk_copy` command without any options, the system displays the help information.
- To display the syntax and options for this command, enter the following command.

```
ovpi_bulk_copy -help
```

- To display the version information for this command, enter the following command.

```
ovpi_bulk_copy -V
```

- To import a specific data file on the default database into a specific table and send the output to standard out, enter the following command.

```
ovpi_bulk_copy -infile file_name -table table_name
```

*file\_name* is the path and file name of the data file to import.

*table\_name* is the name of the table that contains the data.

- To import a specific data file on a specific database into a specific table and send the output to a specific file, enter the following command.

```
ovpi_bulk_copy -infile file_name -table table_name  
-database server_name -logfile result_file_name
```

Where, *file\_name* is the path and file name of the data file to import.

*table\_name* is the name of the table that contains the data.

*server\_name* is the name of database server where the table resides.

*result\_file\_name* is the name of the file that contains the output from the program that loads the data.

- To export data from a specific table on a specific database into a specific data file and send the output to a specific file, enter the following command.

```
ovpi_bulk_copy -outfile file_name -table table_name  
-database server_name -logfile result_file_name
```

Where, *file\_name* is the path and file name of the data file to export.

*table\_name* is the name of the table that contains the data.

*server\_name* is the name of database server where the table resides.

*result\_file\_name* is the name of the file that contains the output from the program that loads the data.



- To export data from a specific table using a specific set of options with the `sqlldr` command on an Oracle system and the `bcp` command on a Sybase system and send the output to standard out, enter the following command.

```
ovpi_bulk_copy -outfile file_name -table table_name  
-sqlldr "option_list" -bcp "option_list"
```

Where, *file\_name* is the path and file name of the data file to export.

*table\_name* is the name of the table that contains the data.

*option\_list* is the list of options separated by spaces that the `sqlplus` or `isql` utility use.

- To export data from a specific table using a specific set of options with the `sqlldr` command on an Oracle system and the `bcp` command on a Sybase system with both files using the same delimiter and send the output to standard out, enter the following command.

```
ovpi_bulk_copy -outfile file_name -table table_name  
-sqlldr "option_list" -bcp "option_list" -ftc char
```

*file\_name* is the path and file name of the data file to export.

*table\_name* is the name of the table that contains the data.

*option\_list* is the list of options separated by spaces that the `sqlplus` or `isql` utility use.

*char* is the character that separates the fields in the output file.

## Example

The following example illustrates a use of the `ovpi_bulk_copy` tool.

### Example: Copy Data to an Oracle Database Server

To copy the contents of the file `c:/tmp/Oracle/data` to a table called `dep` on an Oracle database server called **caspianc**, use the following command:

```
ovpi_bulk_copy -infile c:/tmp/{DBVENDOR}/data -table dep  
-database caspianc -ftc "\","
```

# Error Messages

This section describes some of the messages that can occur from `ovpi_bulk_copy`. Each message has the following format:

- A brief description about why the message appears. Each new message description starts with a check box.
  - The actual message that appears with parameters for any information that may be missing.
  - A suggestion about the action to do so that the message does not appear again.
- ❑ If the following error message appears, the **-table** option is missing on the command line.

Missing the required '-table' option.

Verify the **-table** option with the name of the table is on the command line.

- ❑ If the following error message appears, the specified *server\_name* is missing in the `systems.xml` file.

The database server *server\_name* was not found in the `systems.xml` file.

Verify the spelling of the *server\_name* specified in the **-database** option on the command line. Check the System Manager on the Web Access Server to determine if the specified *server\_name* is in the `systems.xml` file. See the *HP Performance Insight Administration Guide* for more information about System Manager.

- ❑ If the following error message appears, the default database server name is missing in the `systems.xml` file.

The default database server was not found in the `systems.xml` file.

Check the System Manager on the Web Access Server to determine if the default server name is in the `systems.xml` file. See the *HP Performance Insight Administration Guide* for more information about System Manager.

## 22 ovpi\_run\_sql

You can use the `ovpi_run_sql` command to run SQL scripts on an PI system.

### Requirements and Restrictions

An error occurs if the **-sqlscript** option does not appear on the command line.

### Syntax

The `ovpi_run_sql` command uses the following syntax:

```
ovpi_run_sql      [ -database server_name ]  
                  [ -debug debug_level ]  
                  [ -help ]  
                  [ -isql "option_list" ]  
                  [ -logfile result_file ]  
                  [ -sqlplus "option_list" ]  
                  -sqlscript sql_script_name  
                  [ -v ]  
                  [ -version ]
```

# Options

The `ovpi_run_sql` command has the following options:

- database** This option specifies the name of the database server that runs the SQL script.  
The default for this option is the default database server.
- debug** Use this option to set the debug output level. The higher the number, the more detailed the information. Debug output writes to standard out. Use this option only for testing in coordination with HP Technical Support due to the additional overhead it places on `ovpi_run_sql`.  
The default is no debug output.
- help** This option is the help option, which displays the command-line syntax for the `ovpi_run_sql` command. This option overrides all other options on the command line.
- isql** This option specifies the command line options for the Sybase `isql` command. See the Sybase documentation for the valid options.  
You must use double quotation marks to enclose the options.  
This option may appear on the command line with the **-sqlplus** option.
- logfile** This option specifies the name of the file that contains any output generated from the specified script. If this name is not a fully qualified path, the file is in the current working directory.  
If you do not specify a name, the results of the output is in standard out.
- sqlplus** This option specifies the command line options for the Oracle `sqlplus` command. See the Oracle documentation for the valid options.  
You must use double quotation marks to enclose the options.  
This option may appear on the command line with the **-isql** option.

- sqlscript** This option specifies the name of the SQL script to run. This name may contain a relative or fully qualified path; it may also contain the substitution keyword, **{DBVENDOR}**. See [Usage Notes](#) on page 334 for more information.  
This is a required option.
- v** Use this option to display the current version of the `ovpi_run_sql` utility. This option overrides all other options on the command line, except the **-help** option.  
This option is in UPPERCASE.
- version** Use this option to display the current version of the `ovpi_run_sql` utility. This option overrides all other options on the command line, except the **-help** option.

# Usage Notes

This command dynamically determines whether to run the Sybase `isql` command or the Oracle `sqlplus` command depending on the installed database type.

You can use the substitution keyword, `{DBVENDOR}`, in a path when the database type is part of the path. For example, if you want to run an SQL script with the name `sql_script_task_1` in the `system1/ Sybase/task` directory on a Sybase database or in the `system1/Oracle/task` directory on an Oracle database, you can use the following path and file name with the `-sqlscript` option:

```
-sqlscript system1/{DBVENDOR}/task/sql_script_task_1
```

## Using the `ovpi_run_sql` Command

This section shows some formats of the command for the various tasks.

- If you enter the `ovpi_run_sql` command without any options, the system displays the help information.
- To display the syntax and options for this command, enter the following command.

```
ovpi_run_sql -help
```

- To display the version information for this command, enter the following command.

```
ovpi_run_sql -V
```

- To run a script on the default database and send the output to standard out, enter the following command.

```
ovpi_run_sql -sqlscript sql_script_name
```

*sql\_script\_name* is the path and file name of the SQL script to run.

- To run a script on a specific database and send the output to a specific file, enter the following command.

```
ovpi_run_sql -sqlscript sql_script_name -database server_name  
-logfile result_file_name
```

Where, *sql\_script\_name* is the path and file name of the SQL script to run.

*server\_name* is the name of database server where the script runs.

*result\_file\_name* is the name of the file that contains the output from running the specified script.

- To run a script on a specific database while using a specific set of options with the `sqlplus` command on an Oracle database and send the output to a specific file, enter the following command.

```
ovpi_run_sql -sqlscript sql_script_name -database server_name  
-sqlplus "option_list" -logfile result_file_name
```

Where, *sql\_script\_name* is the path and file name of the SQL script to run.

*server\_name* is the name of database server where the script runs.

*option\_list* is the list of options separated by spaces that the `sqlplus` utility uses.

*result\_file\_name* is the name of the file that contains the output from running the specified script.

- To run a script using a specific set of options with the `sqlplus` command on an Oracle database and the `isql` command on a Sybase database and send the output to standard out, enter the following command.

```
ovpi_run_sql -sqlscript sql_script_name -sqlplus "option_list"  
-isql "option_list"
```

*sql\_script\_name* is the path and file name of the SQL script to run.

*option\_list* is the list of options separated by spaces that the `sqlplus` or `isql` utility uses.

## Example

To run a script with the name `sql_script_task_1` in the `system1/ Sybase/ task` directory on a Sybase database and in the `system1/Oracle/task` directory on an Oracle database on the default database, enter the following command.

```
ovpi_run_sql -sqlscript system1/{DBVENDOR}/task/  
sql_script_task_1
```

# Error Messages

This section describes some of the messages that can occur from `ovpi_run_sql`. Each message has the following format:

- A brief description about why the message appears. Each new message description starts with a check box.
  - The actual message that appears with parameters for any information that may be missing.
  - A suggestion about the action to do so that the message does not appear again.
- ❑ If the following error message appears, the default database server name is missing in the `systems.xml` file.

The default database server was not found in the `systems.xml` file.

Check the System Manager on the Web Access Server to determine if the default server name is in the `systems.xml` file. See the *HP Performance Insight Administration Guide* for more information about System Manager.

- ❑ If the following error message appears, the specified `server_name` is missing in the `systems.xml` file.

The database server `server_name` was not found in the `systems.xml` file.

Verify the spelling of the `server_name` specified in the **-database** option on the command line. Check the System Manager on the Web Access Server to determine if the specified `server_name` is in the `systems.xml` file. See the *HP Performance Insight Administration Guide* for more information about System Manager.

- ❑ If the following error message appears, the **-sqlscript** option is missing on the command line.

Missing the required `'-sqlscript'` option.

Verify the **-sqlscript** option with the name of the SQL script is on the command line.



## 23 pa\_collect

You can use the PI `pa_collect` command to collect data from HP Performance Agent (PA) or HP Operations agent on a PI system.

### Syntax

The `pa_collect` command uses the following syntax:

```
pa_collect    [ -d debug_level ]  
               [ -E sum_level ]  
               -i interval  
               [ -K end_date ]  
               [ -q log_info_level ]  
               [ -Q name ]  
               [ -u ]  
               [ -v ]  
               [ -e ]
```

In addition, the `pa_collect` command can use the following `mw_collect` options, because the `pa_collect` command is a wrapper around the `mw_collect` command:

```
[ -a ]  
[ -A ]  
[ -b ]  
[ -c max_processes ]
```

[ **-C** *wait\_time* ]  
 [ **-D** *thread\_wait\_time* ]  
 [ **-e** ]  
 [ **-f** *config\_file* ]  
 [ **-F** *min\_disk\_pct* ]  
 [ **-g** ]  
 [ **-G** *debug\_level\_pm* ]  
 [ **-h** *hostname* ]  
 [ **-H** *alternate\_poller\_name* ]  
 [ **-I** *check\_index* ]  
 [ **-k** ]  
 [ **-L** ]  
 [ **-M** *minimum\_filter* ]  
 [ **-n** ]  
 [ **-N** *retry\_interval* ]  
 [ **-o** *timeout* ]  
 [ **-p** *max\_entries\_per\_pdu* ]  
 [ **-P** *snmp\_port* ]  
 [ **-r** *retries* ]  
 [ **-R** *min\_rows* ]  
 [ **-s** *round\_factor* ]  
 [ **-S** *snmp\_version* ]  
 [ **-t** *table\_name* ]  
 [ **-u** ]  
 [ **-w** *high\_water\_mark* ]  
 [ **-W** *high\_water\_mark\_log* ]  
 [ **-X** ]  
 [ **-Y** ]  
 [ **-Y** *delta\_time* ]  
 [ **-z** *child\_debug\_level* ]  
 [ **-Z** *debug\_log\_bcpgateway* ]



The `pa_collect` command processes its options first and then passes the results and any other `mw_collect` options that appear on its command line to `mw_collect` to complete the processing.

For more information about the `mw_collect` command, see [mw\\_collect](#) on page 277.

## Options

The `pa_collect` command can use the following options:

- a** Directs the child collectors of `pa_collect` to output the collected data in ASCII format. You may want to do this for debugging purposes.
- A** Enables archiving of raw data. The default is no archiving. When you use this option, it turns on archiving. The archive function stores the collected data in a raw data table. This option is in UPPERCASE. This option is equivalent to the **@bArchive=1** parameter in `trendpm`.
- b** Forces regeneration of definition files and worktable. The poller analyzes the structure of the table to be collected and creates definitions about how to load data into database for the `bcp_gateway` process. This is a one-time process most of the time. However, when the poller realizes that the structures of the data table or related property tables have changed, it regenerates the `bcp_gateway` process definitions and worktable. The **-b** option forces the regeneration of these definition files, and cleans up any internal database objects used during data loading. See the discussion for the **-g** option lower in this table for more information. See the discussion for the **-k** option lower in this table for more information.

- c** Specifies the number of collection processes to run concurrently. When `pa_collect` runs, it starts child processes that actually do the collections.  
The default is **5**.  
You can reduce SNMP collection cycles by increasing this number.
- C** Specifies the number of minutes that each child process can run. The system kills the child process if it runs longer than the specified number of minutes.  
The default is **3** minutes.  
This option is in UPPERCASE.
- d** Sets the debug output level for the parent instance of `pa_collect`. Values of **0**, **1**, **2**, or **3** are valid. The higher the number, the more detailed the information.  
The default is **0**, which means no debug output. Debug output is written to standard out. You should only use this option for testing in coordination with HP Technical Support due to the additional overhead it places on `pa_collect`.
- D** This option specifies how many seconds the parent thread should wait for signals from the collector and `bcp_gateway` threads.  
It is normal to get `thread_wait` timeout messages in the log file when waiting for signals from the `bcp_gateway` or `trendpm` thread due to potential long running jobs when writing to the database. The program exits when a `thread_timeout` occurs, unless the `thread_timeout` occurred when all collector jobs are finished and only `bcp_gateway` or `trendpm` threads remained. In this case, the timeout only appears in the `trend.log` file.  
The default is the same value as the **-C** timeout value, which is **1800** seconds. For example, if the **-C** value is **2** (2 minutes), then the **-D** value defaults to **120** (120 seconds).  
This option is in UPPERCASE.

**-e** This option enables the `bcp_gateway` program to back up the collected data if the data upload process fails. If this option is specified with the `pa_collect` command, `bcp_gateway` program backs up the failed batch of collected data for reviewing the failure line later. Note that the `bcp_gateway` program is responsible for uploading collected data into PI database.

**-E** Sets the summarization level for the data in minutes. Use this value to modify the granularity of the data rather than changing the polling interval. For example, if **i 60** sets the frequency of the polling to an hour and you want to gather the data at 15-minute intervals instead, you can add **-E 15** to the command line to modify the granularity, as follows:

```
pa_collect -i 60 -E 15
```

Note that when you increase the granularity for collecting the data, it does increase the collection time because you are collecting more data.

The valid values are the number of minutes and can be the following:

- **0**: non-summarized data
- **5**: 5-minute summarized data
- **15**: 15-minute summarized data
- **30**: 30-minute summarized data
- **60**: 1-hour summarized data
- **180**: 3-hour summarized data
- **360**: 6-hour summarized data
- **720**: 12-hour summarized data
- **1440**: 1-day summarized data
- **10080**: 1-week summarized data

The default value is **60**.

This option is in UPPERCASE.

**-f** Specifies the name of the configuration file that contains additional parameters. See [Configuration File](#) on page 292 for descriptions of the parameters.

- F** Specifies the minimum disk space (as percent of total disk space) that must exist on the disk where the `$COLLECT_HOME` directory resides. If less space exists on the directory, `pa_collect` does not run.

The default is **5**, which represents 5 percent.

This option is in UPPERCASE.
- g** Use 3.5.x compatibility. That is, use the same method for inserting data into the table as version 3.5.x and earlier.

The **-g** option loads data into archive (raw) tables directly. The `bcp_gateway` definition file contains the setting for the **-g** option. This means that if you already invoked `pa_collect` without the **-g** option and you need the **-g** option, then use the **-b** option to regenerate the definition file. Similarly, if you already invoked `pa_collect` with the **-g** option and you do not need the **-g** option, then use the **-b** option to regenerate the definition file. Otherwise, `pa_collect` uses the same setting for the **-g** option currently in effect when you originally generated the definition file.
- G** Sets the debug output level for the `trendpm` process of `pa_collect`. Values of **0**, **1**, **2**, or **3** are valid. The higher the number, the more detailed the information.

The default is **0**, which means no debug output.

Debug output is written to standard out. You should only use this option for testing in coordination with HP Technical Support due to the additional overhead it places on `pa_collect`.

This option is in UPPERCASE.

This option is equivalent to the `@debug_level` parameter in `trendpm`.
- h** Names the host to be polled, which must be an SNMP-manageable device. This name is saved in the database along with the data returned from the node, therefore, to be consistent with other data, use the same node name format used when entering nodes in the database.

- H** Specifies an alternate poller name. When you run `pa_collect` in distributed mode, with the **-n** option, the poller compares the local hostname to the Poll From field in the polling policy. When you use the **-H** option, `pa_collect` compares the Poll From field in the polling policy to the alternate poller name.

This option is in UPPERCASE.

See [Distributed Polling](#) on page 301 for more information.
- i** Is the Collection ID. `pa_collect` executes the entries in the polling policy that have this value in their **Interval** field. How frequently `pa_collect` is actually run depends on the configuration of `trendtimer`, but the idea is to be consistent so that a collection request with a collection ID of **5** is run every 5 minutes. See [File Locks](#) on page 298 for additional information.

This option is required.
- I** Specifies whether to use existing indices on the upload table or to drop existing indices and then recreate them. The value **1** means that the existing indices on the upload table are used. The value **0** means that the existing indices are dropped and then recreated. If the value is **1** and the proper indices are missing then the raw-to-delta process fails.

The default is **0**.

This option is in UPPERCASE.

This option is equivalent to the `@bCheck_index` parameter in `trendpm`.
- k** Populates the property tables (but not the data tables) for the devices you are polling.

The `bcp_gateway` definition file contains the setting for the **-k** option. This means that if you already invoked `pa_collect` without the **-k** option and you need the **-k** option, then use the **-b** option to regenerate the definition file. Similarly, if you already invoked `pa_collect` with the **-k** option and you do not need the **-k** option, then use the **-b** option to regenerate the definition file. Otherwise, `pa_collect` uses the same setting for the **-k** option currently in effect when you originally generated the definition file.

- K** The `dpipe_pa` child collector uses this value as the start time for collecting data from the agent. The format of the date is `yyyymmdd`. Where, `yyyy` is a 4-digit year, `mm` is a 2-digit month, and `dd` is a 2-digit day.
- This option is in UPPERCASE.
- This option can be used to collect older data from Performance Agents. You must take special care for space before using this option when you are using Oracle with partition, as the backend database. History data must be stored in rate tables which have partitions for each hour. As a result when we use this option to collect older data, certain number of partitions gets created based on the 'start time' specified for `-K` and considerable amount of space is utilized for partition creation.
- L** Specifies that the collected data be stored locally instead of in the PI database. When you use this option, `pa_collect` uses any previously saved collection definitions, and does not use the database.
- This option is in UPPERCASE.
- M** Sets the *minimum\_filter* value. The procedure rejects the sample if the delta value of a counter falls below this value. The default value is `-1`, which means to accept the entire sample.
- This option is in UPPERCASE.
- This option is equivalent to the `@line_suppress_value` parameter in `trendpm`.
- n** Enables distributed polling. If this option is used, `mw_collect` executes the collection request only if the Poll From field in the polling policy for this collection request matches the hostname of the machine on which `pa_collect` is running. If you omit this option, `pa_collect` executes all polling requests whose interval matches the value of the `-i` option, regardless of the hostname specified to do the polling in the polling instructions. You can set an alternate hostname to poll with the `-H` option. See [Distributed Polling](#) on page 301 for more information.



- N** Sets the *retry\_interval*, which is the number of seconds the procedure must wait to acquire a lock on an upload table. The default value is **10**, which is 10 seconds. This option is in UPPERCASE. This option is equivalent to the **@retry\_interval** parameter in trendpm.
- o** Number of seconds pa\_collect is to wait for a response after sending an SNMP request. The default is **1** second. (SNMP timeout.)
- p** The number of SNMP variables to include in the varbind list in the GETNEXT PDU (Protocol Data Unit) request. It is possible to generate a GETNEXT request that yields a response that is too long to transmit. The default value is **20**.
- P** This option enables the collection of SNMP data from the specified port rather than the default SNMP port of **161**. This option is in UPPERCASE.
- q** Sets the log information level for the parent instance of pa\_collect. Values of **0**, **1**, **2**, or **3** are valid. The higher the number, the more detailed the information. The default is **0**, which means no log information for the parent instance of pa\_collect. If the \$TREND\_LOG environment variable is set, then the log is written in the directory specified by the \$TREND\_LOG environment to a log file named pa\_collect\_dbg.log; so the path and file name is \$TREND\_LOG/pa\_collect\_dbg.log. If the \$TREND\_LOG environment variable is not set, then the log is written to the same log file name in the \$DPIPE\_HOME/log directory; so the path and file name is \$DPIPE\_HOME/log/pa\_collect\_dbg.log. You can modify the log file name using the **-Q** option.

- Q** Specifies the *name* for the ROOT of the trace log file. You can use this option to change the name of the log file, as follows:
- path/name\_dbg.log*
- path* is \$TREND\_LOG if the environment variable is set; otherwise, it is \$DPIPE\_HOME\log. *name* is the name of the file that you supply.
- When you use this option, you must set the **-q** option to a value greater than 0.
- This option is in UPPERCASE.
- r** Specifies the *retries*, which is the number of times pa\_collect resends an SNMP request before assuming the target node is not going to respond.
- The default value is 5. (SNMP retries)
- R** Sets the minimum number of rows to collect before starting the loading of data into database. The number of rows is an approximation, since there is an assumption that each row is 500 bytes.
- The default value is 1000 rows, which means that the loading of the file into the database starts when the size is 500,000 bytes (500 \* 1000).
- This option is in UPPERCASE.
- s** This option rounds off the collection time (ta\_period). If the pa\_collect parent kicks off a collection at 3:07, and if you are using the default collection option of 300 seconds (5 minutes), the actual ta\_period value for the collection is recorded as 3:05.
- S** Specifies the SNMP version. This option is in UPPERCASE.
- Valid values are:
- 1 for SNMP V1
  - 2 for SNMP V2C.
- This option overrides the values read from the database.
- t** Specifies the *table\_name*, which is the name of the MIB table that you want to collect. The data table must already exist in the database.

- u** Displays the command line formats for `pa_collect`.
- v** Displays the version stamp for `pa_collect`.  
This option is in UPPERCASE.
- w** Specifies the *high\_water\_mark*. The high water mark stops collection of data when the database-used size reaches the specified percentage.  
Valid values are **1 - 100**.  
The default parameter is **90** for 90%.
- W** Specifies the *high\_water\_mark\_log*. The high water mark stops collection of data when the log-used size reaches the specified percentage. Valid values are **1 - 100**.  
The default parameter is **90** for 90%.  
This option is in UPPERCASE.
- x** This option turns off `trendpm` capability.  
This option is in UPPERCASE.
- y** This option disables the calculation of `total_count` for `bcp_gateway` metrics.
- Y** Specifies which clock to use to calculate Delta Time.  
Valid values are:
  - **1**: Directs the procedure to use System Uptime to calculate Delta Time.
  - **0**: Directs the procedure to use the `received_ts` column for the calculation.
 The default is **0**.  
This option is in UPPERCASE.  
This option is equivalent to the `@bDelta_time` parameter in `trendpm`.

- z** Sets the debug level for child collector processes.  
Values of **0**, **1**, **2**, or **3** are valid. The higher the number, the more detailed the information.  
The default is **0**, which means no debug output.  
Debug output is written to standard out. You should only use this option for testing in coordination with HP Technical Support due to the additional overhead it places on `pa_collect`.
- Z** This option specifies whether to turn on debugging or logging or both for the `bcp_gateway` process. When the logging option is turned on, the information is written to the log file. The name of the log file varies depending on which environment variable is set.  
If the `$TREND_LOG` environment variable is set, then the log is written in the directory specified by the `$TREND_LOG` environment to a log file named `bcp_gateway_dbg.log`; the path and file name is `$TREND_LOG/bcp_gateway_dbg.log`.  
If the `$TREND_LOG` environment variable is not set, then the log is written to the same log file name in the `$DPIPE_HOME/log` directory; the path and file name is `$DPIPE_HOME/log/bcp_gateway_dbg.log`.  
Valid values for this option are:
- **1**: Turn on logging.
  - **2**: Turn on debugging.
  - **3**: Turn on both logging and debugging.
- This option is in UPPERCASE.

## Usage Notes

- There is only one child collector, which is `dpipe_pa`, for `pa_collect`. This command works in conjunction with another utility, which is `pa_discovery`; see [pa\\_discovery](#) on page 351 for more information.

To collect data with this command, the collector module for the data table must be **SR**.

- `pa_collect` uses **InitialHistoryPoll** as the interval to collect data from PA agents. The default value for **InitialHistoryPoll** is 120 minutes. When `pa_collect` is run for the first time, it calculates the **start\_time** and **end\_time** for collection based on **InitialHistoryPoll**. **start\_time** is **current\_time-60** and **end\_time** is current time. The **start\_time** and **end\_time** are relative to the system on which `pa_collect` is run.

There may be cases where `pa_collect` does not collect any data from the agents with the **start\_time** and **end\_time**. This situation can arise if the agents have a different time or time zone and the **start\_time** and **end\_time** on local system may be future data for the system on which agents are running.

Since all subsequent `pa_collect` collection are based on initial collection , no data is collected in such scenarios.

**InitialHistoryPoll** must, therefore be modified according to the time zones in which the agents, from which `pa_collect` is going to collect data, reside.

- When you collect data from performance agents by using the `pa_collect` command with **-E** option set to 0, the `dpipe_pa` child collector fails to collect the boundary data. You can add a new parameter called **TimeLag** to the `pa_rpt.cnfg` file to make sure that `dpipe_pa` collects the boundary data.

To do so, follow these steps:

- a Locate the `pa_rpt.cnfg` file, which is stored in `$DPIPE_HOME/data`.  
`$DPIPE_HOME` is the directory into which you installed PI.
- b Open the file using a text editor and add the following parameter at the end of the file.

**TimeLag=<time\_lag\_in\_seconds>**

For example, TimeLag=60

- c Save the file.

With these changes, `pa_collect` collects boundary data from performance agents when `-E` option is set to 0.

## 24 pa\_discovery

The `pa_discovery` command is a utility that enables you to discover HP Performance Agent (PA) or HP Operations agent on a PI system. It does the following:

- Retrieves a list of managed nodes via the PI `node_manager`.
- Determines if the agent exists.
- Gathers information from the existing agents.
- Creates a data file that `ee_collect` uses to load the data into PI.

### Requirements or Restrictions

- Requires access to the PI `node_manager`.
- An error occurs if the **-file** and the **-node** options appear on the command line at the same time.

# Syntax

The `pa_discovery` command uses the following syntax:

```
pa_discovery    [ -d debug_level ]  
  
                  [ { -file file_name  
                    { -node node_name } } ]  
  
                  [ -help ]  
  
                  [ -p ]  
  
                  [ -t trace_level ]  
  
                  [ { -v  
                    { -version } } ]
```

# Options

The `pa_discovery` command has the following options:

- d**            Use this option to set the debug output level. Values of **0**, **1**, **2**, or **3** are valid. The higher the number, the more detailed the information.
- Debug output is written to standard out. You should only use this option for testing in coordination with HP Technical Support due to the additional overhead it places on `pa_discovery`.
- The default is **0**, which means no debug output.
- file**        This option identifies the name of an existing file that contains a list of nodes to discover. You must specify the fully qualified path to the file. See [node\\_manager](#) on page 307 for a description of the format.
- This option may not appear on the command line with the **-node** option.



<b>-help</b>	This option is the help option, which displays the command-line options for the <code>pa_discovery</code> command.
<b>-p</b>	Supresses ping to the host.
<b>-node</b>	This option identifies the name of a single node to discover. The value may be the IP address or the hostname. This option may not appear on the command line with the <b>-file</b> option.
<b>-t</b>	Use this option to set the trace level, which is the level of detail to put in the status files generated at run time. Valid values are as follows: <ul style="list-style-type: none"> <li>• <b>0</b>: Tracing level is off and no status files are created.</li> <li>• <b>1</b>: Error and Warning messages appear in the status file.</li> <li>• <b>2</b>: Error, Warning, and Info messages appear in the status file.</li> <li>• <b>3</b>: Error, Warning, Info, and Trace level 1 messages appear in the status file.</li> <li>• <b>4</b>: Error, Warning, Info, and Trace level 1 and 2 messages appear in the status file.</li> </ul> The default is <b>0</b> .
<b>-v</b>	Use this option to display the current version of <code>pa_discovery</code> . This option is in UPPERCASE.
<b>-version</b>	Use this option to display the current version of <code>pa_discovery</code> .

# Usage Notes

This utility is similar to any discovery utility that locates nodes in the system; however, it is different because it tests each node to determine if the node has a specific type of data to collect. It discovers nodes or agents that have historical data such as HP Performance Agent.

After it discovers that a node or agent has historical data, it adds the host name with the relevant information to a data file. This file has the name `pa_discovery.data` and is in the `DPIPE_HOME/data` directory. The report pack uses this file to populate the database with `ee_collect`.

There is only one child collector, which is `dpipe_pa`, for `pa_discovery`. This utility works in conjunction with another command, which is `pa_collect`; see [pa\\_collect](#) on page 337 for more information.

## Processing Considerations

Typically, you can run the `pa_discovery` utility without parameters, which means that it checks every node. If you want to check specific nodes, you can use an option.

- If you want to check a single node, use the **-node** option.
- If you want to check a specific list of nodes, use the **-file** option. When you use this option, you can omit nodes that would not contain the desired data, for example, routers and hubs.

## Configuration File Settings

The `pa_discovery` program uses a configuration file to set various parameters. Some of these parameters override the defaults set by the `PI pa_discovery` and `dpipe_pa` programs. The configuration file has the name `pa_rpt.cnfg` and is in the `$DPIPE_HOME/data` directory.

You may need to fine-tune the system, depending on the installation environment and number of systems that you are discovering. You may have to modify the `ThreadCount` and `ThreadTimeout` parameters in the configuration file.

## Syntax

The following rules apply to the `pa_rpt.cfg` configuration file.

- The parameters are case sensitive.
- The values may be case sensitive.
- The parameters can appear in any order.
- The format is `parameter=value`.

## Parameters

This list of parameters in alphabetical order that provides the descriptions.

**Table 19 Configuration File Parameters**

Parameter	Description
<b>CODArequestTimeout</b>	Use this parameter to set the number of minutes that must pass before the CODA agent data request connection times out. The default value is 60.
<b>doping</b>	Use this parameter to indicate if pingging a node is enabled before you collect data. Valid values are as follows: <ul style="list-style-type: none"><li>• yes: To enable ping requests which are performed prior to collection of data from agents. The ping requests are meant to identify the node up or down status.</li><li>• no: To disable ping requests which are performed prior to collection of data from agents. If the collection is happening from agents located across a firewall that does not allow ping requests, you should set the doping flag to "no". Otherwise status of those nodes might wrongly be identified as down.</li></ul>
<b>https</b>	Use this parameter to enable or disable HTTPS communication. The valid values are yes and no. The default value is no.
<b>InitialHistoryPoll</b>	Use this parameter to set the number of minutes of historical data you want to poll, when you run the data collection for the first time. The Default value is 120.

**Table 19 Configuration File Parameters**

Parameter	Description
<b>inputDateFormat</b>	<p>Use this parameter to set the format in which you want to type the date.</p> <p>Valid values are <i>mm/dd/yy</i>, <i>dd/mm/yy</i>, and <i>yy/mm/dd</i>.</p> <p>The default date format is <i>mm/dd/yy</i>.</p>
<b>MaxHistoryPoll</b>	<p>This parameter is not applicable for initial data collection.</p> <p>When data is not collected for a long time and you do not want to collect all the data from the last poll information in the database, use this parameter to configure the minutes of historical data you want to poll.</p> <p>Default value is -1, which indicates that the parameter is not used.</p>
<b>outputTimeFormat</b>	<p>Use this parameter to set the date format in the standard output or files.</p> <p>The valid values are UTC and LTZ, where UTC is universal time stamp and LTZ is local time zone.</p> <p>The default value is LTZ.</p>
<b>Proxy</b>	<p>Use this parameter to specify whether a proxy server is used to connect to the CODA agent.</p> <p>Valid values are yes and no.</p> <p>The default value is no.</p>
<b>proxyhost</b>	<p>Use this parameter to set the name of the proxy server that the CODA agent uses for communication.</p> <p>The default value is null.</p>

**Table 19 Configuration File Parameters**

Parameter	Description
<b>proxyport</b>	Use this parameter to set the port number of the proxy server that the CODA agent uses for communication. The default value is 0.
<b>ThreadCount</b>	Use this parameter to set the number of threads that the pa_discovery process uses while running. The default value is 35.
<b>ThreadTimeout</b>	Use this parameter to set the number of minutes the pa_discovery process waits for a thread to complete. The default value is 1.
<b>traceLevel</b>	Use this parameter to set the trace level, which is the level of detail to put in the status files generated at run time. Valid values are as follows: <ul style="list-style-type: none"><li>• 0: Tracing level is off and no status files are created.</li><li>• 1: Error and Warning messages appear in the status file.</li><li>• 2: Error, Warning, and Info messages appear in the status file.</li><li>• 3: Error, Warning, Info, and Trace level 1 messages appear in the status file.</li><li>• 4: Error, Warning, Info, and Trace level 1 and 2 messages appear in the status file.</li><li>• 5: Error, Warning, Info, and Trace level 1, 2, and 3 messages appear in the status file.</li></ul> Valid values are between 0 and 5. The default value is 0.

## 25 piadmin

You can use the `piadmin` command to open the PI Management Console. The Management Console lets you manage most features of your PI system.

You can start the management console from the Start menu or the command line. See the *HP Performance Insight Administration Guide* for more information about using the Management Console.

This chapter describes the `piadmin` command, which is the command that you enter to start the Management Console from the command line.

### Syntax

The `piadmin` command uses the following syntax:

```
piadmin  [ -laf]  
          [ -h]  
          [ -help]  
          [ -u]  
          [ -p password]  
          [ -server Application server host]  
          [ -port port ]  
          [ -file reportname]  
          [ -debug ]  
          [ -log logfile name]  
          [ -mode (local | remote)]
```

**[-directory** *directory path*]

**[-params]**

**[-protocol** ( *http* | *https* )]

## Options

The piadmin command has the following options:

- |                |  |
|----------------|--|
| <b>-laf</b>    | This option specifies the look and feel for the application's user interface.<br>The following options are available: Metal (Java), Motif (UNIX), and Windows.   |
| <b>-h</b>      | Displays the syntax for the command.   |
| <b>-help</b>   | You can use the abbreviated version, <b>-h</b> , or the spelled out version, <b>-help</b> .<br>These option are available only on UNIX systems.  |
| <b>-u</b>      | This option specifies the username for the login process. This username must have administrative privileges.<br>If you do not use this option with the <b>-p</b> option, the system prompts for the username and password. |
| <b>-p</b>      | This option specifies the password for the login process.<br>If you do not use this option with the <b>-u</b> option, the system prompts for the username and password.  |
| <b>-server</b> | This option specifies the Web Access Server host name.<br>The default for this option is the server host name supplied during the PI installation.   |
| <b>-port</b>   | This option specifies the Web Access Server port.<br>The default for this option is the port number supplied during the PI installation, which is port number 80.  |



- file** Use this option to specify the name of the report you want to open automatically when you run `piadmin`. You can use the absolute or relative path with the name of the report file. For a remote file, you need to give the remote location in reference to its deployed location.
- debug** This option enables diagnostic messages, which are messages with an extra level of detail that appear in the log file. The **-log** option specifies the name of the log file.
- log** This option enables logging and specifies which log file to open. You can use this option to change the log file.  
If the file exists, the system appends the new messages to the end; otherwise, the system creates a new file with the specified name. If you include a directory with the name, it must exist in the `DPIPE_HOME` directory. If you want the file to be in the `log` directory, you must include `/log/` with the name; otherwise, the file is saved in the `DPIPE_HOME` directory. Remember to include the preceding slash (or backslash) with every file name.  
The default log file is `console.log`, which is in the `DPIPE_HOME/log` directory.
- mode** Use this option to specify the location for the file you want to access. There are two values; they are:
- **local**: when the file is on the local system.
  - **remote**: when the file is on the Web Access Server.
- The default is **local**.  
Use the **-file** option to specify the name and location of the file. You must use the **-file** option with this option to open a remote file automatically when you run `piadmin`.
- directory** This option specifies the directory that contains the reports to open.

**-params**

Use this option to specify the report parameters to change report defaults at run time. A parameter has the following format: *parameter=value*. This is a global option; it applies to every report you open that has the parameter specified.

When you specify more than one parameter, separate the parameters with a comma (,). When a parameter value contains a space, enclose all the parameters in one set of quotation marks. The following example shows multiple parameters with one parameter that has a space in the value.

```
-params "INTERFACE=92,CUSTOMER=All Telco"
```

When a parameter value contains a character that is special to the command interface (shell) such as a comma, precede the character with a backward slash (\), for example:

```
-params "INTERFACE=92,CUSTOMER=Telco\,North"
```

See the *HP Performance Insight Guide to Building and Viewing Reports* for details about how to create and view the parameters associated with a report using Report Builder, and for details about how to view and modify the parameters associated with a report using the Web Access Server.

**-protocol**

Use this option to specify the communication protocol. Valid values are http or https. The default value is **http**.

# Usage Notes

If you enter the `piadmin` command without any options, the system starts the application and display the Login dialog box.

If you want to start the Management Console automatically so that it does not display the Login dialog box, enter the command with the username and password options.

To enter the command, do the following steps for the appropriate operating system.

## UNIX

From a shell window, enter the following command with the desired options, defined in [Options](#) on page 360.

```
piadmin
```

## Windows

From a Command Prompt window, enter the following command with the desired options, defined in [Options](#) on page 360.

```
piadmin
```

# Examples

The following examples illustrate some uses of the **piadmin** tool.

## Example 1

This example starts the application without displaying the Login dialog box from the user account, **user1**, which has the password **test1**. Note that user account must have administrative privileges.

```
piadmin -u user1 -p test1
```

## Example 2

This example starts the application on a different server, **ferrari1**, even though the default server host name is **powder2**.

```
piadmin -server ferrari1
```

## 26 schedule

The `schedule` command is a tool that enables you to configure schedules on a PI system. It enables you to add, remove, and list entries in the schedule.

You can also add, modify, and remove schedule entries from the Web Access Server. The Web Access Server is the graphical user interface (GUI) for PI. See the *HP Performance Insight Guide to Building and Viewing Reports* for more information.

### Requirements and Restrictions

- An error occurs if more than one of the following options appears on the command line at the same time: **-list**, **-remove**, **-rn**.
- All modes require the following options with their corresponding values: **-host**, **-port**, **-pass**, **-user**, and **-schedule**.
- In add mode, the **-rn** option requires the following options with their corresponding values: **-format**, **-trigger**, and **-title**.
- In remove mode, the **-remove** option requires the **-uid** option to specify the report entry to remove from the schedule.
- If the command string is too long, you can use a continuation character to complete the entry.
  - For UNIX, use the back slash (\) as the last character on the line.
  - For Windows, use the character above the number 6 (^).

# Syntax

A parameter less `schedule` command displays the command syntax.

```
schedule    [ -behalf username ]  
             [ -desc "rpt_desc" ]  
             [ -drilldepth level ]  
             [ -format type ]  
             -host hostname  
             {  
               -list  
               -remove  
               -rn rpt_name  
             }  
             [ -p parameter=value [:parameter=value] ]  
             -pass password  
             -port port_num  
             [ -query hh:mm ]  
             [ -retain num_days ]  
             -schedule sched_name ]  
             [ -title "event_id" ]  
             [ -trigger keyword=value [:keyword=value] ... ]  
             [ -uid uid_num ]  
             -user rpt_username
```

# Options

The `schedule` tool has the following options:

- behalf** This option enables an administrator to schedule a report on behalf of the named user. This means that the administrator runs a report for specific web site login user and then puts the report results into a folder with the user's name and a sub-folder with the schedule file name.  
The user must have an existing schedule so that the administrator can add the report to it.  
If you are using the **-behalf** option and this is a new schedule name, precede this name with the username specified in the **-behalf** option and the appropriate slash for the operating system.  
Use this option only with the **-rn** option.
- desc** Describes the report. You must enclose the description in quotation marks.  
This appears in the Description field on the Schedule Listing page.  
Use this option only with the **-rn** option.
- drilldepth** Specifies the number of queries to the database, which is the resultant number of rows to show, when you link elements in a report. For more information about drill-down depth, see the *HP Performance Insight Guide to Building and Viewing Reports*.  
A value of **all** or **-1** generates all possibilities, which may result in a very large file set. For this reason, you may want to avoid using this value.  
The default value is **5**.  
Use this option only with the **-rn** option.

**-format** Specifies the format type you want for the generated report. Valid values are:

- **csv**
- **html**
- **pdf**
- **srep**

You may specify only one format type for a report.

This is a required option with the **-rn** option.

**-host** Specifies the host name of the Web Access Server where the schedule resides.

This is a required option.

**-list** Returns a list of report entries in a specified schedule. Each report entry includes its unique identification number (UID), the title of the report, and the description of the report.

This is the same list that appears on the Schedule Listing page.

The output from this option shows the following:

```
uid_num | event_id | rpt_desc
```

where, *uid\_num* is the UID for the report.

*event\_id* is the event identification from the **-title** option.

*rpt\_desc* is the description of the report from the **-desc** option. This field shows `None` if you did not include the **-desc** option in the schedule definition for the report.

This option should not be on the command line with either the **-remove** or **-rn** option; if it is, the system shows the list.



- p** Specifies the report parameters to change report defaults at run time. Uses the format *parameter=value*.
- When you specify more than one parameter, separate the parameters with a colon (:). When a parameter value contains a space, enclose all the parameters in one set of quotation marks. The following example shows multiple parameters with one parameter that has a space in the value.
- p "DEVICETYPE=WAN:REGION=NORTHEAST USA"**
- When a parameter value contains a character that is special to the command interface (shell) such as a comma, precede the character with a backward slash (\), for example:
- p DEVICETYPE=WAN:REGION=NORTH\, EAST**
- If the parameter has a colon (:) in its value, you need to precede the colon with a backslash (\).
- Use this option only with the **-rn** option.
- pass** Specifies the corresponding password to the PI username required to access the Web Access Server and modify the schedule.
- This is a required option.
- port** Specifies the Web Access Server port where the schedule resides.
- The default value is **80**.
- This is a required option.

- query** Specifies the time for the report generator to examine the schedule so that it can identify which reports to run. Use the 24-hour format for the time. For example, if you want the generator to examine the schedule at 3:34 p.m. every day, use the following option on the command string.
- query 15:34**
- If you want the generator to examine the schedule on an hourly basis, use the value, **-1**, as follows.
- query -1**
- This option is required if you want the reports to run automatically, according to the triggers you define. If you do not specify a query time, you must use the `generate` command to run the reports. See [generate](#) on page 177 for more information.
- This appears in the Examination Time field on the Schedule Listing page.
- Use this option only with the **-rn** option.
- remove** Removes a report entry from a schedule. Specify the entry using the **-uid** option, followed by the report's UID. For example, to remove the entry from the schedule with the UID equal to `trendadm-934786`, use the following option on the command string.
- remove -uid trendadm-934786**
- Do not use this option when the **-list** or **-rn** option is on the command line.
- retain** Specifies the number of days to retain the generated report results in the Results folder. For example, if you want to delete the report results after 5 days, use the following option on the command string.
- retain 5**
- The default is to retain the results forever.
- Note that you should enter a value for this option since the default consumes a large amount of disk space per report request.
- Use this option only with the **-rn** option.

- rn** Specifies the path and name of the report to generate.  
Do not use this option when the **-list** or **-remove** option is on the command line.
- schedule** Specifies the name of the schedule for which you are adding an entry, removing an entry, or listing all existing entries. If the schedule's name is more than one word, enclose it in quotation marks. For example, if you want to create a schedule with the name, Executive Summary, use the following option on the command string.
- schedule "Executive Summary"**
- If you are using the **-behalf** option and this is a new schedule name, precede this name with the username specified in the **-behalf** option and the appropriate slash for the operating system.
- This name appears in the Schedule File field on the Schedule Listing page.
- This is a required option.
- title** Specifies the title identifier for the event you are scheduling. You must enclose the title in quotation marks. This is the name of the entry that appears in the specified schedule.
- This name appears in the Title field on the Schedule *schedname* for user *username* page, and it is the name that appears in the results folder for each format type.
- This is a required option with the **-rn** option.

**-trigger** Specifies the time to generate the report. You can choose to run the report on a specific day of every week, day of every month, date, or quarter. The keywords for this option are the following:

Keyword	Valid Values
<b>date=</b> <i>yyyy/mm/dd</i>	<i>yyyy</i> 4-digit year <i>mm</i> 2-digit month, values are <b>01-12</b> <i>dd</i> 2-digit day, values <b>01-31</b> depending on the month
<b>day=</b> <i>day</i>	Specify the day of every week to generate the report. Valid values are: <ul style="list-style-type: none"><li>• <b>mon</b></li><li>• <b>tues</b></li><li>• <b>wed</b></li><li>• <b>thurs</b></li><li>• <b>fri</b></li><li>• <b>sat</b></li><li>• <b>sun</b></li></ul>
<b>month=</b> <i>num_day</i>	The value ranges from <b>1 - 31</b> to specify which day of every month to generate the report.

You can use these keywords in combination by placing a colon (:) between them. For example, to run the report every Monday and on the 25th of every month, use the following option in the command string.

**-trigger day=mon:month=25**

This information appears on the Triggers - *path\_reportname.rep* page and you can modify it by clicking the **Edit Event** icon.

This is a required option with the **-rn** option.

**-uid** Used to specify an entry to be removed from the schedule (with the **-remove** option). When you add a new entry to the schedule, the system assigns and displays its UID. The format is *user\_name-nnn* where: *user\_name* is the name of the user that views the report, which is the user named in the **-behalf** option when it is used. *nnn* is a unique identifier. Use this option only with the **-remove** option.

**-user** Specifies the PI username required to access the Web Access Server and modify the schedule. This user must be an administrator user when the **-behalf** option is on the command line. This is a required option.

# Usage Notes

After you create a report, you can set up a time to generate the report on a regular basis. To do this, you add an event to the schedule that specifies when to run the report, which report template to use, what the format is for the report output, the title for the event, and when the system should check the schedule to generate the report.

You can use the `schedule` command to configure schedules when you want to automate the process or create a script.

As an administrator, you can create schedules that generate reports for other users to view.

## Modes of Operation

The `schedule` command has three modes of operation: `add`, `list`, and `remove`.

### Add

The *add* mode provides the ability to add an entry to the schedule. If the specified schedule does not exist, the Schedule tool creates it. When you add an entry to the schedule, the system returns a UID for that entry.

### List

The *list* mode provides the ability to view the current entries in the schedule.

### Remove

The *remove* mode provides the ability to remove an entry from the schedule.

## Using the `schedule` Command

This section shows some formats of the command for the various modes. There is a minimum of five required options for the `schedule` command. Each mode shows the command with the required options along with the other options for

the particular task; however, only the definitions for the new options appear for each subsequent command. The definitions for the required options appear below.

- All schedule commands must have all the following options for each task:

```
schedule -user rpt_username -pass password -host host_name  
-port port_num -schedule sched_name
```

Where, *rpt\_username* is the user name for the report.

*password* is the password for the corresponding user name.

*host\_name* is the name of the host for the Web Access Server.

*port\_num* is the port number for the Web Access Server.

*sched\_name* is the name of the schedule that contains the added entry.

- If you enter the `schedule` command without any options, the system displays the help information. Use the following format.

```
schedule
```

## List

To view the list of current entries in the schedule, enter the following command:

```
schedule -user rpt_username -pass password -host host_name  
-port port_num -schedule sched_name -list
```

## Remove

To remove an entry from the schedule, enter the following command:

```
schedule -user rpt_username -pass password -host host_name  
-port port_num -schedule sched_name -remove -uid uid_num
```

where: *uid\_num* is the event identification number for the report.

## Add

The following formats show various options for adding an event to a schedule. Note that you can combine the additional options in any manner that meets your requirements.

- To add an entry to the schedule using only the required options, enter the following command:

```
schedule -user rpt_username -pass password -host host_name  
-port port_num -schedule sched_name -rn rpt_name -title  
"event_id" -format type -trigger key_value_set
```

where: *event\_id* is the title for the event in the schedule.

*rpt\_name* is the path and name of the report that the system generates.

*type* is format of the output.

*key\_value\_set* is the set of keywords and values that specify the time to generate the report.

- To add an entry to the schedule for another user to view, enter the following command:

```
schedule -user rpt_username -pass password -host host_name  
-port port_num -schedule sched_name -rn path_reportname -title  
"event_id" -format type -trigger key_value_set -behalf user_name
```

where: *user\_name* is the name of the user to access the report.

- To add an entry to the schedule that specifies the number of days to keep the generated report, enter the following command:

```
schedule -user rpt_username -pass password -host host_name  
-port port_num -schedule sched_name -rn path_reportname -title  
"event_id" -format type -trigger key_value_set -retain num_days
```

where: *num\_days* is the number of days to keep the generated report.

- To add an entry to the schedule that specifies the time the generator checks the schedule to generate the reports, enter the following command:

```
schedule -user rpt_username -pass password -host host_name  
-port port_num -schedule sched_name -rn path_reportname -title  
"event_id" -format type -trigger key_value_set -query hh:mm
```

where: *hh:mm* is the time the generator examines the schedule specified in a 24-hour time.



- To add an entry to the schedule that specifies the number of queries to the database for the linked elements in the report, enter the following command:

```
schedule -user rpt_username -pass password -host host_name  
-port port_num -schedule sched_name -rn path_reportname -title  
"event_id" -format type -trigger key_value_set -drilldepth level
```

where: *level* is the number of queries to the database for the linked elements in the report.

- To add an entry to the schedule that adds a description for the event, enter the following command:

```
schedule -user rpt_username -pass password -host host_name  
-port port_num -schedule sched_name -rn path_reportname -title  
"event_id" -format type -trigger key_value_set -desc desc
```

where: *desc* is the description for the report.

- To add an entry to the schedule that changes the report defaults, enter the following command:

```
schedule -user rpt_username -pass password -host host_name  
-port port_num -schedule sched_name -rn path_reportname -title  
"event_id" -format type -trigger key_value_set -p param_val_set
```

where: *param\_val\_set* is the set of parameters and values that change the report defaults.

# Examples

The following examples illustrate some uses of the **schedule** command.

## Example 1

To add an entry with the title **rpt1** to the schedule called **test1a** that runs every Thursday, use the following command.

```
schedule -user user1 -pass test1 -host powder2 -port 80  
-rn /system/SystemResource/quickview.rep -trigger day=thu  
-format csv -title "rpt1" -schedule test1a
```

This entry in schedule test1a appears in the **user1** directory on the **powder2** host. It has the **quickview** report from the **SystemResource** directory with a title of **rpt1**. The output is a file in **csv** format, which means that it has comma-separated values. When you press **Enter** after typing this command, the system returns the UID number for this entry, which is **ID:user1-1064459800046** in this case.

Note that you can create schedules only for the deployed reports.

For example,

- if **quickview.rep** is deployed under **/users/trendadm**, then with **-rn** option, you must mention the path as **/users/trendadm/quickview.rep**.
- if **quickview.rep** is deployed under **/system/test\_folder**, then with **-rn** option, you must mention the path as **/system/test\_folder/quickview.rep**.

## Example 2

To show all the entries in the schedule called test1a for user/password pair of **user1/test1** from host **powder2** and port **80**, use the **-list** option as in the following command.

```
schedule -user user1 -pass test1 -host powder2 -port 80  
-schedule test1a -list
```

The output shows the list of entries in the specified schedule. Each entry has three parts with each part separated by a bar (|). Each entry shows the UID, which is the user name followed by a dash and a unique number, the title of the report, which is from the **-title** option, and the description of the report from the **-desc** option.

The output for this example follows.

```
user1-1064459800046 | rpt1 |  
user1-1064529136962 | rpt2 |  
user1-1064526276978 | rpt3 |  
user1-1064526237368 | rpt4 |  
user1-1064526622837 | null | None
```

Note that when the **-title** option is missing from the command line during the add function, the title of the report is null.

### Example 3

To delete a report called rpt1 from the schedule called test1a for user/password pair of **user1/test1** from host powder2 and port 80, use the **-remove** option with the **-uid** option as in the following command.

```
schedule -user user1 -pass test1 -host powder2 -port 80  
-schedule test1a -remove -uid user1-1064459800046
```

### Example 4

To add an entry called CP to the schedule named john that runs on the 15th of every month, use the following command.

```
schedule -user jmk -pass jmk -host rover -port 80  
-schedule john  
-rn \system\Interface_Reporting\Interface\  
Capacity_Planning.rep -trigger month=15 -format srep  
-retain 3 -title "CP" -p INTERFACE=92:CUSTOMER=Telco
```

The format of the report is srep, and the system keeps the report results for 3 days. When CP runs, the system sets the INTERFACE and CUSTOMER parameters to 92 and Telco, respectively.

When you press **Enter** after typing this command, the system displays the entry's UID. If you ever want to delete this entry from the schedule, you need to use this UID.

# Error Messages

This section describes some of the messages that can occur from the `schedule` command. Each message has the following format:

- A brief description about why the message appears. Each new message description starts with a check box.
- The actual message that appears with parameters for any information that may be missing.
- A suggestion about the action to do so that the message does not appear again.

- ❑ If the following error message appears, the value for the **-rn** option is missing.

```
A value is required for argument rn : Report file name to generate.
```

Verify that the **-rn** option has the name of the report to generate with a fully qualified path.

- ❑ If the following error message appears, the **-port** option is missing.

```
Option "-port" must be specified.
```

Verify that the command has the **-port** option with the number of the port.

- ❑ If the following error message appears, the schedule name specified with the **-schedule** *sched\_name* option with the **-user** *user\_name* option on the command line does not exist.

```
Schedule user_name \ sched_name not found.
```

Verify the spelling of the schedule name or that the schedule exists in the specified location.

- ❑ If the following error messages appear, the **-user** *user\_name* or the **-pass** *password* option is incorrect.

```
You are not authorized to view this website.  
Non-successful HTTP command received: 401
```

Verify the following items.

- The spelling of the *user\_name* or *password* is correct.
- The combination of the user name/password pair is correct.

- ❑ If the following error messages appear, the system cannot find the specified UID number for the specified schedule.

```
User user_name is removing from schedule owned by user_name.  
Looking for schedule sched_name.  
Can't find schedule.  
Non-successful HTTP command received: 206
```

Verify the entry for **-uid uid\_num** option is on the command line with the **-remove** option and is correct.

- ❑ If the following error messages appear, the **-remove** option is not on the command line.

```
Report not found.  
Non-successful HTTP command received: 206
```

Verify the **-remove** option is on the command line with the **-uid** option.

- ❑ If the following error message appears, there is an invalid option on the command line.

```
-option is not valid for this program.
```

Verify the spelling of the *option* on the command line.

- ❑ If the following error message appears, the value for the **-host** option is missing.

```
A value is required for argument host : Trend Application  
Server host.
```

Verify that the **-host** option has the name of the host that has the Web Access Server where the schedule resides.

- ❑ If the following error message appears, the **-host** option is missing.

```
Option "-host" must be specified.
```

Verify that the command has the **-host** option with the name of the host that has the Web Access Server where the schedule resides.

- If the following error message appears, the specified option is missing.

Option "*-option*" must be specified.

Verify that the command has the specified *option* with its corresponding value.

- If the following error message appears, the value for the specified option is missing.

A value is required for argument *option* : *value\_description*.

Verify that the specified option has the corresponding value with it. The *value\_description* describes the corresponding value.

## 27 snmpv2dis

The SNMP V2 Discovery Utility (`snmpv2dis`) identifies devices that support the SNMP V2 protocol on a PI system.

### Syntax

The SNMP V2 Discovery Utility (`snmpv2dis`) has the following syntax.

```
snmpv2dis  [ -c category_name ]  
            [ -d debug_level ]  
            [ -E ]  
            [ -g group_name ]  
            [ -p num_nodes ]  
            [ -s server_name ]  
            [ -u ]  
            [ -v ]
```

# Options

The SNMP V2 Discovery Utility (`snmpv2dis`) has the following options:

- c** Specifies the category name.  
The default value is **ksi\_managed\_node**; however, if you specify a managed-object group name using the **-g** option, the default value is **view**.
- d** Specifies a debug output level. Values of **1**, **2**, or **3** are valid. The higher the number, the more detailed the information.  
The default is no debug output.  
Debug output writes to standard output.
- E** Echoes the progress of `snmpv2dis` to standard output, which is typically the screen.  
This option is in UPPERCASE.
- g** Specifies the name of the managed-object group to discover. This is equivalent to the view name defined using Polling Policy Manager.
- p** Specifies the number of nodes to test concurrently, which is the number of active sessions.  
The default is **10**.
- S** Specifies the name of the server that contains the list of SNMP devices to check. The default uses the server identified in the default database entry in the `systems.xml` file.  
This option is in UPPERCASE.
- u** Displays the list of command line options (help).
- v** Displays the version number of the product.  
This option is in UPPERCASE.



## Usage Notes

The SNMP V2 Discovery Utility (`snmpv2dis`) identifies devices that support the SNMP V2 protocol. After this utility determines that a device supports the SNMP V2 protocol, it does not check that device again.

If you enter the `snmpv2dis` command without any options, it checks the SNMP devices on the database server identified in the default database entry in the `systems.xml` file. This means that the `snmpv2dis` utility checks all the nodes identified as SNMP devices in the `ksi_managed_node` property table that do not have SNMP V2 identification. In this case, there was no category name specified with the `-c` option and no managed-object group name specified with the `-g` option.

If you specify a managed-object group name with the `-g` option and do not specify the category name with the `-c` option, then the category name is **view**. You can find managed-object group names from Polling Policy Manager. See the *HP Performance Insight Administration Guide*.

If there are any errors, they appear in `trend.log`.



---

## 28 tpmaint

The Time Periods Maintenance Utility, `tpmaint`, is a PI utility that populates *time-period* tables. Each of these tables contains a list of time periods for a specified number of days, which is the retention value for the table. There are time-period tables for various time categories, such as day, hour, 5-minute, and 30-second. These tables enable optimized searches of the database.

### Requirements or Restrictions

If you change the retention period for any data table to a retention period value longer than the default value for the corresponding time-period table by 1.5 or more times, it negatively impacts performance.

# Syntax

The `tpmaint` command uses the following syntax:

```
tpmaint      [ -d debug_level ]  
              [ -e "end_date" ]  
              [ -F ]  
              [ -h ]  
              [ -s "start_date" ]  
              [ -S server_name ]  
              [ -t time_period_table_name ]  
              [ -u ]  
              [ -v ]
```

# Options

The `tpmaint` command has the following options:

- d** Specifies the type of debug output. The values are:
  - **0**: Provides no debug output. This is the default value.
  - **1**: Provides general debug output.Debug output is in standard output. You should only use this option for testing in coordination with HP Technical Support.
- e** This option specifies the end date. The format of the *end\_date* is *mm-dd-yyyy*.  
Where, *mm* is a 2-digit month, *dd* is a 2-digit day, and *yyyy* is a 4-digit year.  
Enclose the date in quotation marks, since the format includes dashes.
- F** This option forces `tpmaint` to continue processing when there are more rows to process from the table than two times the default retention period for that table.  
This option is in UPPERCASE.
- h** Displays the syntax for this utility.
- s** This option specifies the start date. The format of the *start\_date* is *mm-dd-yyyy*.  
Where, *mm* is a 2-digit month, *dd* is a 2-digit day, and *yyyy* is a 4-digit year  
Enclose the date in quotation marks, since the format includes dashes.  
The default for this option sets the start date to today.
- S** This option specifies the database *server\_name*. It overrides the value of the default database entry in the `systems.xml` file for the current process.  
This option is in UPPERCASE.

- t      Used to specify the *time\_period\_table\_name* to be processed.  
The default for this option causes all time period tables to be processed sequentially.
- u      Displays the syntax for this utility.
- v      Displays the version of this utility.  
This option is in UPPERCASE.

## Usage Notes

The `trendtimer` program should invoke the `tpmaint` utility on a daily basis. The utility verifies that each time-period table contains data for the specified number of retention days for the time category. For example, if the time-period table for the hour time category has a retention of 400 days, `tpmaint` verifies that there is data for at least 400 days in that table.

To improve performance, PI applications verify that the appropriate entries exist in the corresponding time-period table for a data table. They do this by comparing the begin and end dates of the source table to the begin and end dates of the corresponding time-period table. If the begin and end dates of the source table are within the range of the begin and end dates for the corresponding time-period table, the program continues processing; otherwise, it invokes `tpmaint` to update the corresponding time-period table. Note that if there are gaps in the time-period table for the range of dates from the source table, the program invokes `tpmaint` to cover that time frame.



You can change the retention period for any table in any time category; however, setting the retention period value longer than the default value of the time category (1.5 or more times) negatively impacts performance.

There are predefined limits for the number of rows to keep in time-period tables. Exceeding those numbers results in performance problems. When `tpmaint` populates a time-period table and it tries to increase the number of rows to more than twice the recommended amount, it generates the following messages in `trend.log`.

```
Adding the requested date range to the total of numberdays to
table table_name will exceed the total default of retention
days by more than 2 times.
```

Reduce the date range specified or use **-F** to override.

If these error messages appear in `trend.log` while running `tpmaint`, you need to rerun **tpmaint** with the **-F** option to override the default and continue processing. Be aware that this causes performance penalties.

Note that if you terminate `tpmaint` with **Ctrl+c**, you need to terminate any associated stored procedures that are currently running on the server separately.

## Examples

### Example 1

The following command causes the `tpmaint` utility to populate the time-period tables with the date starting on September 14, 2003.

```
tpmaint -s "09-14-2003"
```



Note that a check is performed before adding a row to prevent inserting duplicate records.

### Example 2

The following command causes the table named `dsi_local_time_period_hour` to be processed.

```
tpmaint -t dsi_local_time_period_hour
```





## 29 transform\_maint

You can use the `transform_maint` command to perform maintenance tasks related to transformations on a PI system. A *transformation* is a set of parameters with a corresponding procedure that changes the data from one form to another. In the case of summary transformations, a transformation is the set of parameters that `trend_sum` uses to generate a procedure that changes the delta data in one table to the hourly data in another table.

### Requirements and Restrictions

- When you use the **-delete** option, you must include the **-name** option on the command line.
- One of the following options must appear on the command line: **-delete**, **-list**, **-refresh**, or **-remove**.

# Syntax

A parameter less `transform_maint` command displays the following syntax:

```
transform_maint  [ -all ]  
                  [ -database server_name ]  
                  [ -d debug_level ]  
                  [ -debug debug_level ]  
                  [ -delete ]  
                  [ -h ]  
                  [ -help ]  
                  [ -l ]  
                  [ -list ]  
                  [ -name name ]  
                  [ -refresh ]  
                  [ -remove ]  
                  [ -v ]  
                  [ -version ]
```

# Options

The `transform_maint` command has the following options:

- all** Use this option to list the information for a transformation definition from the database. This information includes the transformation and procedure names, the source and destination table names, the first day of week and baseline days specifications, and the hysteresis and lag time settings. The format of the output has a tag enclosed in angle brackets followed by the value for the tag.  
This option must appear with the **-list** option on the command line.
- database** Use this option to specify the remote database name.  
The default for this option is the default database specified in the `systems.xml` file.
- d** Use this option to set the debug output level. The higher the number, the more detailed the information. Debug output writes to standard out. Use this option only for testing in coordination with Technical Support due to the additional overhead it places on `transform_maint`. The valid values are **0**, **1**, **2**, or **3**. The default is no debug output, which is **0**.
- debug**
- delete** Use this option to remove a transformation definition. If the corresponding procedure is in use, the `transform_maint` utility does not delete the transformation definition.  
You must use the **-name** option with this option.
- h** This option is the help option, which displays the command-line syntax for the `transform_maint` command.
- help** This option overrides all other options on the command line.
- l** Use this option to list the names of the transformation definitions in the database. The output has the following format: `<name> 'transformation_definition_name'`.
- list**
- name** Use this option to specify the name of the transformation definition to remove or generate. You can use the **-list** option to get the name. This is a required option when the **-delete** option appears on the command line.

<b>-refresh</b>	Use this option to generate new procedures for the corresponding transformation definitions.
<b>-remove</b>	Use this option to remove the procedures that are no longer associated to any transformation definition. When you use this option, the <code>transform_maint</code> utility does not remove any procedures that are running.
<b>-v</b> <b>-version</b>	Use this option to display the current version of the <code>transform_maint</code> utility. This option overrides all other options on the command line, except the <b>-help</b> option. The <b>-v</b> option is in UPPERCASE.

## Usage Notes

The commands that create transformations use a set of specified parameters to generate a corresponding procedure that changes the data from one form to another. For example, `trend_sum` creates summary transformations that change delta data in one table to hourly data in another table. It does this by using the settings from the `.sum` file and generating a procedure to aggregate the data. It also stores this information as a transformation in the database. When there are changes to the settings in the `.sum` file for a particular summary transformation, the system generates a new procedure for that transformation.

Currently, the `transform_maint` utility performs the following tasks related to summary transformation definitions.

- Force `trend_sum` to create new summary procedures.
- Remove all the summary procedures that are no longer associated to any summary transformation definition.
- List the names of the summary transformation definitions with the record information from the database, which includes the following elements.
  - transformation and corresponding procedure name
  - source and destination table names
  - first day of week and baseline days designations
  - hysteresis and lag time settings

- Remove a specific summary definition and its associated procedure.

## Using the `transform_maint` Command

This section shows some formats of the command for the various tasks.

- If you enter the `transform_maint` command without any options, the system displays the help information.
- To display the syntax and options for this command, enter one of the following commands.

**`transform_maint -h` or `transform_maint -help`**

- To display the version information for this command, enter one of the following commands.

**`transform_maint -V` or `transform_maint -version`**

- To generate new procedures for all transformation definitions available on the default database, enter the following command.

**`transform_maint -refresh`**

- To generate new procedures for all transformation definitions available on a specific database, enter the following command.

**`transform_maint -refresh -database` *server\_name***

Where, *server\_name* is the name of the database that has the transformation definitions.

- To generate a new procedure for a specific transformation definition on a specific database, enter the following command.

**`transform_maint -refresh -name` *tname* **`-database`** *server\_name***

Where, *tname* is the name of the specific transformation definition.

*server\_name* is the name of the database that has the transformation definitions.

- To display only the names of all transformation definitions on a specific database, enter the following command.

**`transform_maint -list -database` *server\_name***

Where, *server\_name* is the name of the database that has the transformation definitions.

- To display the record information for a specific transformation definition on a specific database, enter the following command.

```
transform_maint -list -all -name tname -database server_name
```

Where, *tname* is the name of the specific transformation definition.

*server\_name* is the name of the database that has the transformation definitions.

- To display the record information for all transformation definitions on a specific database, enter the following command.

```
transform_maint -list -all -database server_name
```

Where, *server\_name* is the name of the database that has the transformation definitions.

- To remove a specific transformation definition on a specific database, enter the following command.

```
transform_maint -delete -name tname -database server_name
```

Where, *tname* is the name of the specific transformation definition.

*server\_name* is the name of the database that has the transformation definitions.

- To remove all the procedures that are no longer associated to any transformation definition on a specific database, enter the following command.

```
transform_maint -remove -database server_name
```

Where, *server\_name* is the name of the database that has the transformation definitions.

# Examples

The following examples illustrate some uses of the `transform_maint` tool.

## Example 1

To list the name for each transformation definition on the `powder2` database, enter the following command.

```
transform_maint -list -database powder2
```

## Example 2

To list the record information for the transformation definition with the name `SDIRCust_All_SDIRDevPorts_SDIRCust` on the `powder2` database, enter the following command.

```
transform_maint -list -name  
SDIRCust_All_SDIRDevPorts_SDIRCust -database powder2 -all
```

The output has the following tags with their corresponding values:

<name>, <source>, <target>, <first day of week>, <baseline days>, <hysteresis units>, <lagtime units>, and <procedure name>. The output follows:

```
<name> 'SDIRCust_All_SDIRDevPorts_SDIRCust' <source>  
'SDIRDevPorts' <target> 'SDIRCust' <first day of week> 'Mon'  
<baseline days> '0' <hysteresis units> '-1' <lagtime units>  
'-1' <procedure name> 'null'
```

# Error Messages

This section describes some of the messages that can occur from `transform_maint`. Each message has the following format:

- A brief description about why the message appears. Each new message description starts with a check box.
  - The actual message that appears with parameters for any information that may be missing.
  - A suggestion about the action to do so that the message does not appear again.
- If the following error message appears, the specified command requires the specified option.

Missing required option *option* for command *command\_name*.

Verify that one of the following conditions occurs:

- At least one of the following options is on the command line: **-delete**, **-list**, **-refresh**, or **-remove**.
- The **-delete** and **-name** options are on the same command line.

- If the following error message appears, the specified *server\_name* may be missing in the `systems.xml` file.

The connection to the database *server\_name* could not be established : Connection URL not found.

Verify the spelling of the *server\_name* specified in the **-database** option on the command line. Check the System Manager on the Web Access Server to determine if the specified *server\_name* is in the `systems.xml` file. See the *HP Performance Insight Administration Guide* for more information about System Manager.

- If the following error message appears, the default database server name may be missing in the `systems.xml` file.

The connection to the default database could not be established.

Check the System Manager on the Web Access Server to determine if the default server name is in the `systems.xml` file. See the *HP Performance Insight Administration Guide* for more information about System Manager.



- ❑ If the following error message appears, the syntax for at least one of the options on the command line is incorrect.

The syntax of the command is incorrect.

Verify the format for each option on the command line. See [Syntax](#) on page 394.

- ❑ If the following error message appears, a name enclosed in angle brackets (< >) appears on the command line.

The system cannot find the file specified.

Verify that each option on the command line has the correct format and that a value for an option does not have angle brackets (< >).

- ❑ If the following error message appears, the name for the transformation definition is not in the database.

Transformation definition *tname* not found in the PI system.

Verify the spelling of the name for the transformation definition. You can use the **-list** option to locate the name.



# 30 trendcopy

The `trendcopy` command enables the user to copy data from one PI database to another.

## Requirements or Restrictions

- This version of `trendcopy` copies data to or from databases in the following manner:

From	To
PI, Sybase	PI, Sybase or Oracle
PI, Oracle	PI, Oracle

It does not copy data from databases with PI on Oracle to any PI database on Sybase.

- `trendcopy` only copies *new* data, which is data that is later than the latest `ta_period` value in the destination table for a particular object (keyid).
- `trendcopy` can copy tables that have by-variables different from `target_name` and `table_key`. These tables can also have foreign keys.
- Data transported by `trendcopy` goes from data table to data table. Data copied in this way is accessible only by PI.
- `trendcopy` copies the property and data tables to another database only when the property table of the destination table has an identity column.
- `trendcopy` only copies the definitions of property and data tables to another database when the property table of the destination table is missing an identity column (such as a Report Package demo). In this case, use either the BCP or BCPfile utility to complete the transfer.

- `trendcopy` copies a source data table constructed as a view to an existing destination data table constructed as a view; otherwise, it creates the destination data table as a table.
- `trendcopy` copies a data table with a corresponding property table that is a view only when the view is a recognized view. The following table lists the recognized property table views.
- `trendcopy` is not configured to automatically copy data from satellite servers to the central server or vice-versa. To allow data copy, you must create and enable copy policy for the table. For more information about copy policies, see the *HP Performance Insight Administration Guide*.

**Table 20 Property Table Views Recognized by `trendcopy`**

Kmon_ethstatsdata_	Kmon2_alhostdata_	Kmone_ethstatsdata_
Kmon_histdata_	Kmon2_almatrixdata_	Kmone_histdata_
Kmon_histtrmdata_	Kmon2_nlhostdata_	Kmone_histtrmdata_
Kmon_histtrpdata_	Kmon2_nlmatrixdata_	Kmone_histtrpdata_
Kmon_hostdata_	Kmon2_statsdata_	Kmone_hostdata_
Kmon_matrixdata_		Kmone_matrixdata_
Kmon_trmstatsdata_		Kmone_trmstatsdata_
Kmon_trpstatsdata_		Kmone_trpstatsdata_
Kmon_trroutestatsdata_		Kmone_trroutestatsdata_
Kmon_trstationdata_		Kmone_trstationdata_

# Syntax

The format for `trendcopy` follows.

```
trendcopy  [ -c data_column_name:operator:value ]  
             [ -C [table_name.]property_column_name:operator:value ]  
             [ -d debug_level ]  
             [ -D ]  
             [ -e yyyymmdd[:yyyymmdd] ]  
             [ -E seconds ]  
             [ -f day_of_week[:day_of_week] ]  
             [ -g hour[:hour] ]  
             [ -h ]  
             [ -I mode_value ]  
             [ -K retries ]  
             [ -M ]  
             [ -N ]  
             [ -P ]  
             [ -q ]  
             [ -Q ]  
             [ -s source_server ]  
             -S destination_server  
             [ -t source_table [:destination_table] ]  
             [ -u ]  
             [ -v ]  
             [ -w high_water_mark ]
```

[ **-x** *level* ]  
[ **-Y** ]  
[ **-Z** *max\_rows* ]

## Option Categories

The following table lists the options in categories that you might use.

**Table 21 trendcopy Option Categories**

Category	Options
Typical:	-t, -s, -S
Specific:	-D, -P, -M, -N, -Y
Row Filter:	-e, -f, -g
Limits:	-Z, -w
Debug:	-d, -q, -x
Copy Policy on Central Server:	-t, -s
Copy Policy on Satellite Server:	-t
Miscellaneous:	-h, -Q, -u, -V

# Options

You can run the `trendcopy` command-line interface with the following options:

**-c** Provides filtering according to the criteria specified for the data table column.

The format for the entry is *data\_column\_name:operator:value* where:

*data\_column\_name* is the name of data table column for the filter.

*operator* is the boolean operator to apply.

Valid operators are:

- = equal
- < less than
- <= less than or equal
- > greater than
- >= greater than or equal
- !=not equal

*value* is the value of the filter for the specified column. If the SQL type of the column is CHAR or VARCHAR, enclose the value in single quotation marks.

- C** Provides filtering according to the criteria specified for the property table column.  
The format for the entry is  
*[table\_name.]property\_column\_name:operator:value*  
where:  
*table\_name* is the name of the property table.  
*property\_column\_name* is the name of property table column for the filter.  
*operator* is the boolean operator to apply.  
Valid operators are:
- = equal
  - < less than
  - <= less than or equal
  - > greater than
  - >= greater than or equal
  - != not equal
- value* is the value of the filter for the specified column. If the SQL type of the column is CHAR or VARCHAR, enclose the value in single quotation marks.
- d** Sets the debug output level.  
Valid values for *debug\_level* are 1, 2, and 3.  
The higher the number, the more detailed the information. Debug output is written to the standard output destination. Use this option only for testing in coordination with HP Technical Support.
- D** Copies the new rows into the data table for each specified data table only when there are entries in the corresponding keymap. It does not check, copy, or update the property table or the keymap. See [Using the Specific and Debug Options](#) on page 418.  
This option is in UPPERCASE.



- e** Provides filtering by date, which copies the data table rows that match the specified date. The format for the date is *yyyymmdd*, Where, *yyyy* is the 4-digit year, *mm* is the 2-digit month, and *dd* is the 2-digit day.
- The `trendcopy` command accepts multiple instances of this option. See [Using the Row Filter Options To Copy Data by Date](#) on page 417.
- E** Specifies the number of seconds the system waits before it tries to obtain a database lock on a table.
- f** Provides filtering by day of week, which copies the data table rows that match the specified day of the week.
- Valid *day\_of\_week* values may be upper or lower case and are the following:
- **su**: Sunday
  - **mo**: Monday
  - **tu**: Tuesday
  - **we**: Wednesday
  - **th**: Thursday
  - **fr**: Friday
  - **sa**: Saturday
- The *day\_of\_week* range is Sunday through Saturday, and does not wrap around to the following week. To specify a *day\_of\_week* range that wraps around to the following week, use two command-line options.
- `trendcopy` accepts multiple instances of this option. See [Using the Row Filter Options To Copy Data by Date](#) on page 417.

- g** Provides filtering by hour of the day, which copies the data table rows that match the specified hour.
- Valid *hour* values are 0-23, where 13 means 1 p.m. and 0 (zero) means 12 midnight. This value does not wrap around to the following day. To specify an hour range that wraps around to the following day, use two command-line options.
- The value 9:14 selects the data table rows where *ta\_period* matches 9 a.m. to 2 p.m.
- trendcopy* accepts multiple instances of this option.
- See [Using the Row Filter Options To Copy Data by Date](#) on page 417.
- h** Displays a listing of all *trendcopy* options. This option is the same as the **-u** option.
- I** Use this option to specify the mode to run *indexmaint*, which maintains the indexes on the tables. Valid values are:
- **0**: Does not run *indexmaint*.
  - **1**: To run *indexmaint* in maintenance mode.
  - **2**: To run *indexmaint* in force mode.
- The default is **0**.
- This option is in UPPERCASE.
- K** Specifies the number of retries the system attempts to obtain a database lock on a table.
- M** Synchronizes the corresponding keymap for each data table specified. It updates the property table keymap by mapping only the keyids that are on both the source and destination servers. That is, the keymap contains the intersection of keyids on the servers. The updated keymap is on both servers. This option does not invoke the copy of the property table or the data table. See [Using the Specific and Debug Options](#) on page 418.
- This option is in UPPERCASE.

- N**      Transcribes the keymap, for the corresponding property table for each data table specified, from the destination server (**-S**) to the source server (**-s**) after processing all other tasks in the command. Note that this is an add-on option, which means that *trendcopy* processes this option after it processes all other options on the command line.

This option is in UPPERCASE.
- P**      Copies the new rows into the corresponding property table for each data table specified. It also updates the corresponding keymap. It does not copy the data table. See [Using the Specific and Debug Options](#) on page 418.

This option is in UPPERCASE.
- q**      Checks and displays the list of tasks that *trendcopy* generates. Note that if the specified table does not exist on the destination server (**-S**), *trendcopy* creates the table and the corresponding property table without copying the data and then display the list of tasks. Use this option only for testing in coordination with HP Technical Support.
- Q**      Launches the *datapipe\_manger* to create tables that that do not exist on the destination server.

This option is in UPPERCASE.
- s**      Identifies the source server, which is the server (or database) that contains the data tables to copy. This name must be an entry in the *systems.xml* file with the tag **<Name>**.

The default value for the *source\_server* is default database entry in the *systems.xml* file.

**trendcopy** uses the last instance of this option that appears in the command line.

This option is not required, but when it is used, it readily identifies the location for the data.

See [Using Various Options to Copy Database Tables](#) on page 416.

- S** Identifies the destination (or target) server, which is the server (or database) to receive the copied data tables. There is no default for *destination\_server*. This name must be an entry in the *systems.xml* file with the tag **<Name>**.  
trendcopy uses the last instance of this option that appears in the command line.  
This is a required option.  
This option is in UPPERCASE.  
See [Using Various Options to Copy Database Tables](#) on page 416.
- t** Identifies the data or property table to copy. If you specify a property table, you must also use the **-P** option. This is mandatory; if not, an error is generated. The parameters are the following:
- *source\_table* contains the rows to copy
  - *destination\_table* is the data table to receive the copied rows
- If the destination table name is missing, trendcopy uses the source table name (*source\_table\_name* => *destination\_table\_name*).  
Enter the name of the table as shown in the SQL Name column of the **Database Table Management** display from the management console. Note that, if the *destination\_table* name is from the Alias Name column, trendcopy processes the command as if the table name is an SQL name.  
Using this option, you can run multiple trendcopy instances simultaneously, improving throughput and reducing run times.  
If this option is missing, trendcopy copies all data tables with the corresponding property tables serially, which may result in excessive run times.  
If the *destination\_table* does not exist on the destination server, trendcopy creates the table with the same property table assignment as the source data table. If the destination data table has a different property table than the source data table, then you need to create the destination tables with the appropriate property table assignment before running trendcopy.  
See [Using Various Options to Copy Database Tables](#) on page 416.
- u** Displays a listing of all trendcopy options. This option is the same as the **-h** option.

- v** Displays the `trendcopy` version.  
This option is in UPPERCASE.
- w** `trendcopy` determines the current size of the database to be copied and does not run if the database-used size exceeds the percentage specified in this parameter. The default is 90 for 90%.
- x** Requests task timings to appear in the log. Enter any integer for the *level* value. The output is in the `trendcopy_dbg.log` file. Use this option only for testing in coordination with Technical Support.
- Y** Requests that `trendcopy` not verify the keymap table.  
This option is in UPPERCASE.
- z** Specifies the number of maximum rows allowed for a BCP batch file; enter an integer for the *max\_rows* parameter. See [Using the Specific and Debug Options](#) on page 418.  
The default is **1000** rows.  
This option is in UPPERCASE.

# Usage Notes

The basic function of `trendcopy` is to copy data tables from one PI database to another. For each table it copies, it copies the data that has a later timestamp than the latest timestamp in the corresponding destination table for each managed object. If there is a gap in the timestamps for the data, `trendcopy` does not copy the data to fill in the gap.

When `trendcopy` copies a data table, it verifies the keymap first and then copies the corresponding property table before copying the new data. If a data table does not exist, `trendcopy` creates the missing table. Some of the options allow you to modify the process. See [Options](#) on page 407 for the descriptions of the options. See [Keymap Tables](#) on page 415 for more information about keymaps.

Some applications for `trendcopy` follow.

- In a PI system that uses satellite servers to process data, use `trendcopy` to copy the processed data from a PI satellite server to the PI central server.
- In a PI system that includes an archive server, use `trendcopy` to copy data from the central server to the archive server.

## Performance Notes

- If property tables are static, this means that there are no new objects added to the property table, use the `-D` option.
- If you are satisfied with the integrity of the keymaps and you do not change them manually, use the `-Y` option.
- Before you copy a large data table, verify that the indexes of the table are in order by running `indexmaint` in force mode. See [indexmaint](#) on page 261.
- By employing the `-t` processing option, you can run multiple `trendcopy` instances simultaneously, thus improving throughput and reducing run times. If you do not specify a specific table (with the `-t` option), `trendcopy` copies all the data tables with their corresponding property tables, that satisfy all the other processing options on the command line, serially. This may result in excessive run times.

- Consider the limits of your system when you run concurrent processes. Take into account which processes are already running or scheduled when you plan to run or schedule multiple `trendcopy` commands at the same time.

## Capabilities

`trendcopy` can:

- Copy all the rows in a table to a table in another PI database.
- Create a new data table in a database using information copied from a table in another database.
- Filter the data it copies by date or time of day.

## Keymap Tables

`trendcopy` copies tables using a keyid mapping scheme, which synchronizes just the new keyids to provide fast execution. In summary, this scheme maintains a keymap on the destination server, which maps all the keyids in a property table between source and destination servers.

For instance, suppose a keyid for a `Kib_ii_ifentry` table on server A has a value of 123 while that same object on destination server B has a keyid value of 234. Then, there would be a row in the server B keymap table for `Kib_ii_ifentry` having the values of server A, 123, and 234.

`trendcopy` insures the integrity of a keymap table. If the integrity check fails for a server, `trendcopy` assumes that all rows in the keymap table for that server are invalid and regenerates them.

# Examples

The following examples show various methods for copying PI database tables from one server to another.

## Using Various Options to Copy Database Tables

### Example 1

This example shows the command to specify multiple tables on a single command line. If, for example, you want to copy three tables named `mib-II_ifEntry`, `mib-II_system`, and `rmon_history` from the `PRIMARY_DB` server to the `BACKUP_DB` server, use the following command.

```
trendcopy -t mib-II_ifEntry -t mib-II_system  
-t rmon_history -s PRIMARY_DB -S BACKUP_DB
```

### Example 2

This example shows the command to copy the data table `SDIRCustDevice` from the server `xyz_in` to a table named `test_tbl_b` on the server `XYZ_out`. It copies 100 rows at a time in a batch file.

```
trendcopy -t SDIRCustDevice:test_tbl_b -s xyz_in -S  
XYZ_out -Z 100
```

### Example 3

This example shows the command to copy the data table `SDIRConfig` from the server `xyz_in` to an existing table named `test_tbl_2` on the server `XYZ_out`.

```
trendcopy -s xyz_in -S XYZ_out -t SDIRConfig:test_tbl_2
```

### Example 4

Note that this example produces an error because the table named `test_tbl_2x` does not exist on the server `XYZ_out`. The `-Q` option prevents `trendcopy` from creating the missing table on the destination server.

```
trendcopy -s xyz_in -S XYZ_out -t SDIRConfig:test_tbl_2x  
-Q
```



## Using the Row Filter Options To Copy Data by Date

You can use `trendcopy` to copy data filtered by date. The following examples explain the use of the `-e`, `-f`, and `-g` options for filtering by date, day of week, and hour of day. You can specify more than one filter; each filter can be a single value or a range of values.

All of the following examples copy the data from the source server named `xyz_in` to the destination server named `XYZ_out`. They use all the data tables to locate the data since the `-t` option is missing.

### Example 1

The following command copies any data for which the `ta_period` is 1:00 p.m. (13) on either Monday (`mo`) or Friday (`fr`):

```
trendcopy -f mo -f fr -g 13 -s xyz_in -S XYZ_out
```

### Example 2

The following command copies data with `ta_period` greater than or equal to Monday (`mo`) and less than or equal to Wednesday (`we`); and `ta_period` greater than or equal to October 1, 2002 (20021001), and less than or equal to November 1, 2002 (20021101):

```
trendcopy -f mo:we -e 20021001:20021101 -s xyz_in -S XYZ_out
```

### Example 3

The following command copies data from a Wednesday (`we`) through the following Monday (`mo`). It specifies a day range that wraps around to the following week and uses two command-line options:

```
trendcopy -f we:sa -f su:mo -s xyz_in -S XYZ_out
```

### Example 4

The following command copies data with `ta_period` greater than or equal to 1 a.m. (1) and less than or equal to 3 a.m. (3) and with a `ta_period` greater than or equal to October 1, 2002 (20021001), and less than or equal to November 1, 2002 (20021101):

```
trendcopy -g 1:3 -e 20021001:20021101 -s xyz_in -S XYZ_out
```

#### Example 4

The following command copies data from 8:00 p.m. (20) to 7:00 a.m. (7); it specifies an hour range that wraps around to the following day; that is, through midnight (0), with two command-line options.

```
trendcopy -g 20:23 -g 0:7 -s xyz_in -S XYZ_out
```

## Using the Specific and Debug Options

#### Example 1

This example shows the command to copy only the new rows in the property table for the rate data table `Rasic_interface_info_` from a satellite server to the central server. It copies 10,000 rows at a time in a batch file.

```
trendcopy -t Rasic_interface_info_ -Z 10000 -P  
-s satellite_DB -S central_DB
```

#### Example 2

This example shows the command to copy only the new rows in the rate data table `Rasic_interface_info_` from a central server to the satellite server. It copies all the data, 10,000 rows at a time.

```
trendcopy -t Rasic_interface_info_ -Z 10000 -D  
-s central_DB -S satellite_DB
```

#### Example 3

This example shows the command to synchronize the keymaps on the servers `xyz_in` and `XYZ_out` for `test_tbl_2`. It updates the keymaps to contain only the keys that are common on both servers.

```
trendcopy -t test_tbl_2 -s xyz_in -S XYZ_out -M
```

#### Example 4

This example shows the command to copy only the new rows in the data table `test_tbl_2` from the server `xyz_in` to the server `XYZ_out`. It uses the existing keymap on the destination server `XYZ_out`. It uses the keymap created in if that was the last `trendcopy` command for this table and these servers.

```
trendcopy -t test_tbl_2 -s xyz_in -S XYZ_out -D
```

### Example 5

This example shows the command to copy only the new rows in the rate data table `Rasic_interface_info_` with the keymap from a central server, `xyz_in`, to the satellite server, `XYZ_out`. It copies only the keys that are common on both servers. Note that it is more efficient to use the `-M` option first and then the `-D` option as in the previous examples.

```
trendcopy -t Rasic_interface_info_ -D -M -s xyz_in -S  
XYZ_out
```

### Example 6

This example shows the command to skip the validation of the keymap and copy only the new rows in the data table `test_tbl_7` from the server `xyz_in` to the server `XYZ_out`. In this example, `trendcopy` also copies the corresponding property table.

```
trendcopy -t test_tbl_7 -s xyz_in -S XYZ_out -Y
```

# Error Messages

This section describes some of the messages that can occur from `trendcopy`. Each message has the following format:

- A brief description about why the message appears. Each new message description starts with a check box.
- The actual message that appears with parameters for any information that may be missing.
- A suggestion about the action to do so that the message does not appear again.

## General

- ❑ The following warning message appears when the same command is already running.

`Trendcopy for table table_name in db server_name is already running.`

If you want to run this command again, you must re-enter it when it is not already running. When this message appears, the system does not process or schedule the command.

- ❑ The following error message appears when the `trendcopy` command originates from a PI system that is not using the 4.6 version of it.

`The TREND version on server_name is unknown.`

Verify that the database you originate the `trendcopy` command from is using PI version 4.6.

## Server Name Error Messages

- ❑ The following error message appears when the source and destination server names are the same and the `-M` option appears on the command line. The keymap table is not copied.

`Copy of keymap is not allowed when source db = destination db`

Verify the entries and spelling of the source (**-s**) and destination (**-S**) servers.

- ❑ The following error message appears when the *host\_name* does not exist in the `DPIPE_HOME/data/systems.xml` file.

Error: failed to get ip information for host *host\_name*.

Verify the *host\_name* entry is in the `DPIPE_HOME/data/systems.xml` file.

- ❑ The following error message appears when the *src\_server\_name* and the *dest\_server\_name* entries in the `DPIPE_HOME/data/systems.xml` file have the same address.

Error: *src\_server\_name* and *dest\_server\_name* have the same network address of *address*.

Verify the entries and spelling of the source (**-s**) and destination (**-S**) server names. Verify the server entries in the `DPIPE_HOME/data/systems.xml` file.

- ❑ The following error message appears when the format of the name of the server on the command line does not match the format of the name in the `DPIPE_HOME/data/systems.xml` file.

Database '*db\_name*' was not found in `systems.xml` file.

Verify that the format and spelling of the source (**-s**) and destination (**-S**) servers is the same as the format of the server name entries in the `DPIPE_HOME/data/systems.xml` file. For example, if the *server\_name* for the **-s** or **-S** option is in lowercase, replace the *server\_name* in UPPERCASE. When this message appears, `trendcopy` continues processing the command.

## Source Server

- ❑ The following error message appears when the source server is not available.

Unable to connect to source db (*server\_name*).

Verify the following:

- The source server specified in the **-s** option exists and has an entry in the `DPIPE_HOME/data/systems.xml` file.
- The spelling of the server name is correct.

- The server is running.

## Destination Server

- ❑ The following error message appears when the destination server name (**-s**) is missing.

A destination server needs to be specified.

Verify that the **-s** option is on the command line with a server name. Note that the **-s** option is in UPPERCASE.

- ❑ The following error message appears when the destination server is not available.

Unable to connect to destination db (*server\_name*).

Verify the following:

- The destination server specified in the **-s** option exists and has an entry in the `DPIPE_HOME/data/systems.xml` file. Note that the **-s** option is in UPPERCASE.
- The spelling of the server name is correct.
- The server is running.

## Table Name Error Messages

- ❑ The following error message appears when the source data table name does not exist in the PI dictionary and there is no property table for the data table.

No entry in *dictionary\_table* for *table\_name* on *server\_name*.

Verify that the source table specified in the **-t** option exists in the PI database on the specified server and the spelling of the table name is correct.

- ❑ The following error message appears when the property table name for the source data table does not exist.

No property table name for *table\_name* on *server\_name*.

When this message appears, trendcopy does not copy the data table. Call Technical Support.

- ❑ The following error message appears when the source property table is missing from the database dictionary table (sysobjects) and the data table exists.

Source property table *table\_name* does not exist on *server\_name*.

Verify that the property table for the specified data table in the **-t** option exists in the database. Determine if the name of the source property table changed.

- ❑ The following error message appears when the source data table is missing from the database dictionary table (sysobjects). In this case, there is a property table for the specified data table.

Source data table *table\_name* does not exist on *server\_name*.

Verify that the source data table specified in the **-t** option exists in the database and the spelling of the table name is correct. Determine if the name of the source data table changed.

- ❑ The following error message appears when the source and destination table names are the same and the source and destination server names are the same. The data table is not copied.

Copy of *source\_table* to *destination\_table* is not allowed when source db = destination db.

Verify any of the following:

- The spelling of the source and destination table names specified in the **-t** option are correct.
- The names of the source and destination servers are correct.
- If the **-s** option for the source server is missing, check to see if the destination server is the same server identified in the default database entry in the *systems.xml* file.

- ❑ The following error message appears when the system cannot find any tables to copy.

No tables found in *dictionary\_table* on *server\_name*.

Verify the tables exist in the database on the specified server.

- ❑ The following error message appears when trendcopy has to create a table but the **-Q** option is on the command line, which prevents trendcopy from creating the table by executing *datapipe\_manager*.

No table build by DPM for: *datapipe\_manager\_command*.

Verify that the table specified in the *datapipe\_manager\_command* exists in the database and the spelling of the table name is correct, or remove the **-Q** option from the command line so that trendcopy can create the missing table.

- ❑ The following error message appears when trendcopy copies many large property tables to an Oracle database system.

Reached maximum number of open cursors.

Run trendcopy again, and continue to run trendcopy until it copies all the tables and the message no longer appears.

## View Error Messages

- ❑ The following error message appears when the destination table is a view instead of a table or the destination property table is not a recognized view. The data table is not copied.

*destination\_table* is a view; trendcopy does not copy into views.

See [Table 20](#) on page 404 for a list of recognized views.

Verify the table name specified in the **-t** option is data table name and not a view name. Note that if the box in the Is View column on the Database Table Management display has a check mark (4), then that table name is a view.

## By-variable Error Messages

- ❑ The following error message appears when the data type of the by-variable is not character or numeric.

Unacceptable data type of *data\_type* in BuildPropCols for *column\_name* in prop table *property\_table\_name* on *server\_name*.

Verify the spelling of the specified data table name, and that the corresponding property table has by-variables with character or numeric data types. If the by-variables have any other data types, trendcopy does not copy it or the data table.



- ❑ The following error message appears when the by-variables for a property table are missing.

Error: failed to find any by-variables in *property\_table\_name* on *server\_name*.

Verify the spelling of the specified data table name, and that the corresponding property table has at least one by-variable. If the property table does not have any by-variables, *trendcopy* does not copy it or the data table.

- ❑ The following error message appears when the number of by-variables for the source property table is different than the number of by-variables for the destination property table.

Error: *source\_property\_table* on *source\_server* has *num1* by-variables while *destination\_property\_table* on *destination\_server* has *num2*.

Verify the source and destination table names are correct, and that the corresponding property tables have the same number of by-variables; otherwise, *trendcopy* does not copy the property or data table.

- ❑ The following error message appears when the by-variables for the source property table are different from the by-variables for the destination property table.

Error: *source\_property\_table* on *source\_server* has *source\_byvar* as by-var *num* while *destination\_property\_table* on *destination\_server* has *destination\_byvar*.

Verify the source and destination table names are correct, and that the corresponding property tables have the same by-variables; otherwise, *trendcopy* does not copy the property or data table.

## Foreign Key Error Messages

- ❑ The following error message appears when the foreign key reference column is missing from the foreign key table.

foreign key reference column *fkey\_col\_name* for column *column\_name* in property table *property\_table\_name\_1* is not in foreign property table *fkey\_table\_name\_2*.

When this message appears, *trendcopy* does not copy the data table. Call Technical Support.

- ❑ The following error message appears when the foreign key reference column does not contain `dsi_key_id`.

Error: foreign-key reference column *fkey\_name* is not *dsi\_key\_id* for column *col\_name* in prop table *table\_name* on *server\_name*.

When this message appears, trendcopy does not copy the data table. Call Technical Support.

## Option Error Messages

- ❑ The following error message appears when the calendar date option (**-e**) contains a non-numeric value.

Invalid value for the -e option. The date must be in `yyyymmdd` format.

Verify the value specified for the **-e** option has a valid numeric value in the format `yyyymmdd`.

- ❑ The following error message appears when the day-of-week option (**-f**) contains an invalid value.

Invalid value for the -f option. The day of the week value must be one of the following: `MO`, `TU`, `WE`, `TH`, `FR`, `SA`, `SU`.

Verify the value specified for the **-f** option has one of the following values:

<b>MO</b>	for	Monday
<b>TU</b>	for	Tuesday
<b>WE</b>	for	Wednesday
<b>TH</b>	for	Thursday
<b>FR</b>	for	Friday
<b>SA</b>	for	Saturday
<b>SU</b>	for	Sunday

- ❑ The following error message appears when the hour-of-day option (**-g**) contains any value less than 0 and greater than 23.

Invalid value for the -g option. The time value must be between 0 and 23.

Verify the value specified for the **-g** option has a number greater than or equal to 0 (zero) and less than 24.

- ❑ The following error message appears when the **-P** option is on the same command line with either the **-D** or **-M** option.

Selection of both -P and *option\_letter* is unacceptable.

If you want to copy the property table only, then use the **-P** option without the **-D** or **-M** options. If you want to copy a data table with its corresponding property table, do not use the **-P**, **-D**, or **-M** options since that is the default behavior.



## 31 trend\_discover

`trend_discover` enables you to conduct a search that can:

- Find the nodes on your system.
- Ascertain whether each node is SNMP manageable.
- Determine the node type; for example, router, hub, switch.
- Automatically update the tables or views that control data collection.

As a result, `trend_discover` gives you the option of using an automated process to define your network.



Note that the use of `trend_discover` is optional. You can also populate and update your various node tables manually, if you choose.

## Requirements or Restrictions

- `trend_discover` must run from a Performance Insight (PI) server.
- `trend_discover` defaults to the `trendadm` user to run if the `USER` environment variable is not set.
- Type Discover is set to run automatically at least once a day from the PI scheduler.
- Type Discover requires `.dis` files to run. These files must exist in a specified directory or the `DPIPE_HOME/scripts` directory.
- The Management Console provides GUI access to both types of Discover. Use the SNMP Discovery option from the Tools menu for IP Discover. Use the Type Discovery option from the Tools menu for Type Discover. See *HP Performance Insight Administration Guide* for more information about the GUI tools.

## IP Discover Syntax

IP Discover locates the nodes on your system and determines whether they are SNMP manageable.

```
trend_discover    [ -a delete_age ]  
                   [ -c community_string ]  
                   [ -C ping_packets ]  
                   [ -d debug_level ]  
                   [ -D ]  
                   [ -E ]  
                   [ -f community_names_file ]  
                   -h end_IP_range  
                   [ -H ]  
                   [ -k ]  
                   -l start_IP_range  
                   [ -m ]  
                   [ -o SNMP_timeout ]  
                   [ -O ping_timeout ]  
                   [ -p max_entries ]  
                   [ -P SNMP_port_number ]  
                   [ -q database_server ]  
                   [ -r SNMP_retries ]  
                   -s network_subnet  
                   [ -S ping_packet_size ]  
                   [ -u username ]  
                   [ -V ]  
                   [ -z ]
```

The minimum syntax required for the IP Discover command is shown below:

```
trend_discover -l start_IP_range -h end_IP_range -s network_subnet
```

## IP Discover Options

IP Discover has the following options:

- a** Specifies the *delete\_age*, which is the number of consecutive attempts to ping a node that must fail before automatically deleting the node from the *dsi\_nodes* and *dl\_type* views.  
The default is **10**.
- c** Specifies the single community string to use for all SNMP GETs.
- C** Specifies the *ping\_packets*, which is the number of packets that can be in a discover ping.  
The default is **1**.  
This option is in UPPERCASE.
- d** Specifies a debug output level. Values of **1**, **2**, or **3** are valid. The higher the number, the more detailed the information.  
The default is no debug output.  
Debug output writes to standard output.
- D** Suppresses domain names. This option truncates the translated host names starting from the first period in the host name.  
This option is in UPPERCASE.
- E** Echoes the progress of Discover to standard output, which is typically the screen.  
This option is in UPPERCASE.
- f** Specifies the *community\_names\_file*, which is a file containing community strings that Discover uses for SNMP GETs. Discover looks for the specified file in the `$DPIPE_HOME/scripts` directory unless you specify a path for the file.

- h** Defines the *end\_IP\_range*, which is the maximum value of the discover range. If the value is incomplete, Discover appends zeros to the end of that value. For example, if you input the value **134.70**, Discover assumes that input to be **134.70.0.0**.
- H** This is the help option, which displays a list of available Discover command line options to the screen.  
This option is in UPPERCASE.
- k** Disables the execution of the `snmpv2dis` utility.
- l** Defines the *start\_IP\_range*, which is the minimum value of the discover range. If the value is incomplete, Discover appends zeros to the end of that value. For example, if you input the value **134.70**, Discover assumes that input to be **134.70.0.0**.
- m** Disables the collection of a node's mac address (physical address) when Discover runs.  
  
For example, if a system has two IP addresses, Discover uses the mac address and treat this system as one node instead of two. If you specify the **-m** option, Discover treats this system as two nodes with two different IP addresses.
- o** Sets the *SNMP\_timeout*, which determines how long Discover waits before timing out an SNMP GET request.  
The default is **1** second.
- O** Sets the *ping\_timeout*, which specifies the time in milliseconds that must pass before Discover times out a ping when Discover is running from a Windows platform.  
The default is **750** milliseconds.  
This option is in UPPERCASE.
- p** Resets the maximum number of entries that can be in a Protocol Data Unit (pdu).  
The default is **20**.
- P** Defines the *SNMP\_port\_number*, which is the port number to use.  
The default is **161**.  
This option is in UPPERCASE.
- q** Specifies the name of the database in which the discovery information is stored. If the **-q** option is not used, the default database is used. Use the database name not its hostname.



- r** Defines the number of times Discover retries its SNMP GET if there is no response.
- s** Defines the *network\_subnet*, which is the Subnet Mask used in the Discover process. If the value is incomplete, Discover appends zeros to the end of that value. For example, if you input the value **134.70**, Discover assumes that input to be **134.70.0.0**.
- S** Specifies the *ping\_packet\_size*, which is the packet size of a discover ping.  
The default is **1** byte.  
This option is in UPPERCASE.
- u** Defines the PI username of the user conducting the discover. If you do not use this option, Discover attempts to get the *username* from the `USER` environment variable. If Discover cannot locate a `USER` environment variable, it uses `trendadm` for the *username*.
- v** Displays the version number of the product.  
This option is in UPPERCASE.
- z** Specifies that Discover attempts to translate a discovered IP address to a host name.



Note that when the node translation (**-z**) option is on and a ping is successful, Discover performs a host name lookup. A previously discovered translated name can only be replaced by another translated name, not an IP address. In other words if the host name lookup fails to return translated name, Discover does not replace the current name with an IP address.

# IP Discover Usage Notes

IP Discover is a manually initiated process; you can initiate it either from the user interface or from the command line. The purpose of IP Discover is to find the devices on a network, and determine whether they are SNMP manageable.



Note that any PI user can initiate Discover, but only from a PI server. On PI clients, Discover is disabled.

## How IP Discover Works

You can initiate an IP Discover by defining a range of IP addresses, a Subnet mask, and one or more read community strings that the system searches. Discover then pings each IP address (except beginning subnet addresses [0] and broadcast addresses [255]) in that IP address range and one of the following occurs:

- If there is no device at the IP address, there is no response.
- If there is a device at that IP address that recognizes the ping, it responds to Discover. From this response, Discover knows that there is a device at that IP address. Discover then:
  - Sends an SNMP GET message to the responding device.
  - Attempts to discover the host name of the device (if you have specified host name translation).

If the device recognizes and responds to the GET, Discover identifies the device as SNMP manageable.

If your system has a protocol to translate IP addresses to host names, and it finds a host name, Discover associates it with the discovered device. If it does not find a host name, Discover lists the node by its IP address.

## IP Discover View Population

When Discover gets a response from an SNMP capable device, it inserts the host name (or IP address) of the device into the `dsi_nodes` view. After the host name exists in `dsi_nodes`, PI can access the discovered device.



Note that `trend_discover` only places primary IP addresses, or host names if using translation, into `dsi_nodes`.

## Specifying Community Strings for an SNMP GET Request

When Discover performs an SNMP GET request, the community strings it uses can be defined in the following ways:

- If you do not specify a community string, Discover uses the default, which is public.
- If you want to specify a single community string for all the SNMP GETs in a discover, you can use the `-c community_string` option.
- Whatever character string you input with the `-c` option is used as the identifying community string for the GET requests.
- If you want to specify multiple community strings for the SNMP GETs, you can use the `-f` option.

This option can be used in two ways:

- If you use the syntax `-f file_name`, where *file\_name* is the name of a file, but does not include the file path, Discover looks for that file only in the `$DPIPE_HOME/scripts` directory.
- If you use the syntax `-f file_path`, where *file\_path* is the name and path of a file (including either `/` or `\`), Discover looks for that file only in the specified path.

This file is a regular ASCII text file, with one read community string per line. Discover tries the community strings in this file until the first successful SNMP request is performed.

## Community Strings Files

Discover uses the entries in the Community Strings file in the order in which they appear. The sequence is as follows:

- 1 Discover polls the first node, using the strings identified in the Community Strings file.

Discover polls using the first string. If there is no response, Discover tries the second string, then the third, and so on.

When the node responds to a poll, Discover classifies that device as responding to that community string, and goes on to the next node.

- 2 Discover does the same for each node.

It is up to you to determine in which order the community strings appear in the Community Strings file.

► Note that in some cases, it is possible for devices on a network to respond to multiple community strings.

### Example

Some RMON vendors allow the use of public as the read community string for MIB-II tables, but require different community strings for RMON tables. This community string can also be used to GET MIB-II tables. In this case, the community strings file should list the RMON community string above the public community string.

► If a node is discovered with a community string other than the one it had when it was originally discovered, its community string is updated only if the original community string was public.

When you create a Community Strings file, it is recommended that you list the more common community strings first; this can expedite the discover process. However, you should consider the possibility that your system may include devices that respond to multiple community strings, as described above.

# SNMP Type Discover Syntax

The only valid command line arguments for SNMP Type Discover are:

```
trend_discover    [ -d debug_level ]  
                   [ -E ]  
                   [ -F type_definition_file ]  
                   [ -H ]  
                   [ -k ]  
                   [ -o SNMP_timeout ]  
                   [ -p max_entries_pdu ]  
                   [ -P SNMP_port_number ]  
                   [ -r SNMP_retries ]  
                   [ -t ]  
                   [ -u username ]  
                   [ -v view_name ]  
                   [ -V ]
```

The minimum syntax required for the SNMP Type Discover command is shown below:

```
trend_discover -t
```

# SNMP Type Discover Options

SNMP Type Discover has the following options:

- d** Specifies a debug output level. Values of 1, 2, or 3 are valid. The higher the number, the more detailed the information. The default is no debug output. Debug output writes to standard output.
- E** Echoes the progress of Discover to standard output, which is typically the screen.  
This option is in UPPERCASE.
- F** Names the type definition file. Discover looks for the specified file in the `$DPIPE_HOME/scripts` directory unless you specify a path for the file.  
If this option is missing, Discover looks for `.dis` files in the directory, `DPIPE_HOME/scripts`. If the `.dis` files are missing, an error message appears.  
This option is in UPPERCASE.
- H** Displays the list of available Discover command line options (help).  
This option is in UPPERCASE.
- k** Disables the execution of the `snmpv2dis` utility.
- o** Sets the *SNMP\_timeout*, which determines how long Discover waits before timing out an SNMP GET request.
- p** Resets the maximum number of entries that can be in a Protocol Data Unit (pdu).  
The default is **20**.
- P** Defines the *SNMP\_port\_number*, which is the port number to use.  
The default is **161**.  
This option is in UPPERCASE.
- r** Defines the number of times Discover retries its SNMP GET if there is no response.
- t** Causes Discover to run an SNMP Type discover.

- u** Defines the PI user name of the user conducting the discover. If you do not use this option, Discover attempts to get the *username* from the `USER` environment variable. If Discover cannot locate a `USER` environment variable, it uses `trendadm` for the *username*.
- v** Specifies the name of the view to discover.
- v** Displays the version number of the product.  
This option is in UPPERCASE.

## Naming Convention

Type discover files must have the `.dis` extension and reside in the `DPIPE_HOME\scripts` directory.

## Type Discover Usage Notes

SNMP Type Discover typically runs automatically, once a day from the PI scheduler. You can initiate it in the following ways:

- Package Manager, when you use it to install a report pack
- From the command line
- From the Management Console, Tools menu

The purpose of SNMP Type Discover is to identify the nature of the devices in the `dsi_nodes` view.

In particular, SNMP Discover determines the Type of a discovered device. SNMP Type Discover interrogates nodes to determine what kind of device they are, and records the information in the PI database. This identification enables PI to target a specific type of device for data collection.

In SNMP Type Discover, the Discover function reads the `dsi_nodes` and `dl_type` views to identify the nodes that require type validation.

Discover uses special files, identified by the extension `.dis`, to determine whether a device is of a certain type. Each `.dis` file contains the necessary protocols to test a node for a single, specific type. See [Type Definition Files](#) on page 440 for more information about `.dis` files.

## SNMP Type Discover View Population

There are two database views that are affected by SNMP Type Discover: `dsi_nodes` and `dl_type`.

`dsi_nodes`: The `dsi_nodes` view contains a row for each recognized IP node on your system. In addition, one column of `dsi_nodes` identifies whether a node is SNMP or RMON (default is SNMP). When SNMP Type Discover identifies an RMON-manageable device, it updates this column of `dsi_nodes`.

`dl_type`: For each node on your network, the `dl_type` view identifies the type of device it is. SNMP Type Discover populates this view by interrogating each new node in `dsi_nodes`, and adding an entry to `dl_type`.

## Type Definition Files

When you run an SNMP Type Discover, the Discover function tries to assign different types to the nodes specified in `dsi_nodes`. After a node is determined to be of a particular type, it is inserted into `dl_type`.

SNMP Type Discover is driven by information defined in Type Definition files. The path of a Type Definition file can be specified via the `-F` option. By default, `trend_discover` searches for `*.dis` files in the `$DPIPE_HOME/scripts` directory. If `.dis` files are missing, an error message appears. Either install a report pack or create a custom report with a corresponding `.dis` file before running Type Discover again.



Note that after a node is determined to be of particular type, `trend_discover` does not try to rediscover that node for the same type again.

## Syntax

The syntax rules for a Type Definition file follow.

- Every line in a Type Definition file must end with a semi-colon (`;`), except for comments.



- Comments begin with a slash and an asterisk (/\*) and end with an asterisk and a slash (\* /), as shown in the following example:  

```
/* this is a comment */
```
- Blanks are ignored in the input file, unless they are inside a string (between single quotation marks), as shown in the following example:  

```
` this is a string `
```
- Each action begins with a keyword. The valid keywords for Type Definition files are `typename`, `collection`, and `nodetype`. Use of the `collection` and `nodetype` keywords is optional. The syntax for each keyword follows:

Syntax	Description
<b>typename:</b> <i>name_of_type</i> ;	Assigns a name to a type. Example: <b>typename:</b> <b>cisco_routers</b> ;
<b>collection:</b> <b>no</b> ; or <b>collection:</b> <b>yes</b> ; The default is <b>no</b> .	Specifies whether the polling policy controls Type Discover. If the value is <b>yes</b> , then <code>trend_discover</code> performs type discover only if there is a polling policy defined for this type (Group Name).
<b>nodetype:</b> <i>collect_type</i> ; where, <i>collect_type</i> is <b>rmon</b> or <b>snmp</b> . The default is <b>snmp</b> .	Specifies whether this is an RMON or regular SNMP table.

## SNMP Tests

The actual definition of the type defines one or more SNMP Tests that a device must pass to be of the given type. If multiple SNMP Tests are defined for a given type, all of them must pass; for example, AND function is used. The following tests are available:

## Collection Table

This test checks that the device supports the Collection Table specified, to belong to the type.

Syntax:           *collectiontable\_name*;  
                  where *collectiontable\_name* is the alias name defined in  
                  *dsi\_tab\_alias*.

Example:        **mib-II\_ifentry**;

## Simple Object Identifier (Oid)

This test checks that the device supports the OID. To pass this test, the device must support the OID.

Any value returned by the device for that OID indicates that the device supports the OID.

Syntax:           *oid*;

Example:        **1.3.6.1.2.1.1.1**;

## Simple NOT Object Identifier (Oid)

This test checks that the device does not support the OID. To pass this test, the device must NOT support the OID.

Syntax:           ~*oid*;

Example:        ~**1.3.6.1.2.1.1.1**;

## Value

To pass this test, the device must support the given OID and the return value must satisfy the defined expression. This test has 4 expressions:

- This first expression checks that the OID is equal to the specified string.
- This second expression checks that the OID is not equal to the specified string.
- This third expression checks that the OID is equal to the specified number.

- This fourth expression checks that the OID is not equal to the specified number.

If the *string\_value* contains wildcard characters, then the compare is not case-sensitive; otherwise, the compare is case-sensitive.

Syntax:        *oid* = '*string\_value*';  
                  or  
                  *oid* != '*string\_value*';  
                  or  
                  *oid* = *numerical\_value*;  
                  or  
                  *oid* != *numerical\_value*;

Example1:    1.3.6.1.2.1.1.1 = 'cisco router';  
                  /\* this is regular string compare, case  
                  sensitive \*/

Example2:    1.3.6.1.2.1.1.1 = '\*cisco\*';  
                  /\* this is wildcard character compare, when  
                  wildcard characters are used trend\_discover will  
                  perform case insensitive compare \*/

Example3:    1.3.6.1.2.1.1.1 != 'bay router';  
                  /\* this is a not equal test \*/

Example4:    1.3.6.1.2.1.1.1=453;

## Type Definition File Examples

### Example 1

```
/* this file defines 3com_device type */
/* the test is done on sysDescr oid */
typename:3com_devices;
collection:no;
1.3.6.1.2.1.1.1='*3com*';
```

## Example 2

```
/* this file defines 3com_routers */
/* the test is done on sysDescr oid and ipForwarding oid */
typename:3com_router;
collection:no;
1.3.6.1.2.1.1.1='*3com*';
1.3.6.1.2.1.4.1=1;
```

## Example 3

```
/* this file will define types that support mib-II_ifEntry */
/* collectiontable_name and have ipForwarding turned on */
typename:mibII;
collection:no;
mib-II_ifEntry;
1.3.6.1.2.1.4.1=1;
```

## Example 4

```
/* this file will define rmon1 ether stats type */
/* this type is limited to rmon1 devices that do not support
*/
/* rmon2 extensions */
typename:rmon1ethstats;
collection:no;
nodetype:rmon;
~1.3.6.1.2.1.16.1.4.1.1;
/* does not support rmon2 extension defined for rmon1 */
1.3.6.1.2.1.16.1.1.1.1; /* must support etherStatsIndex oid
*/
```

## 32 trendexec

The `trendexec` program uses the `trend_sum` program to run the `trend_sum` procedures listed in the database; it performs these functions when the time interval specified on the command line matches the time interval in the database. `trendtimer` contains entries for the `trendexec` program to run on an intermittent basis. You can launch the `trendexec` program to run at a different time from the command line.

### Requirements or Restrictions

A PI database server must be available.

### Syntax

The `trendexec` command uses the following syntax:

```
trendexec    [ -b database_server ]  
              [ -d debug_level ]  
              [ -h ]  
              -i interval  
              [ -v ]
```

# Options

The `trendexec` command has the following options:

- b**      Used to specify an alternate *database\_server*.
- d**      Sets the debug output level in the `trend_sum` program.  
Valid values for *debug\_level* are: **0, 1, 2, 3, 4**, and **5**. The default is **0**.  
The higher the number, the more detailed the information. Debug output is written to the standard output destination. Use this option only for testing in coordination with HP Technical Support.
- h**      Displays the command line options.
- i**      Used to specify the time interval in minutes. This option is required.
- v**      Displays version information. This option is in UPPERCASE.

## Example

When executed, the command below locates and executes procedures to be run every 5 minutes.

```
trendexec -i 5
```

## 33 trend\_export

This chapter describes the `trend_export` utility. `trend_export` enables you to run a PI report against your database and to store or export the output in ASCII text format.

This chapter has four parts:

- [Overall Process and Functionality](#) on page 448
- [Controls and Switches](#) on page 449
- [Input Parameter File](#) on page 452
- [Output Format](#) on page 460

### Syntax

The `trend_export` command has the following syntax:

```
trend_export      [-t database table name]  
                  [-r Report name]  
                  [-o Output File]  
                  [-s Database server]  
                  [-f delimiter [tab|comma]]  
                  [-a include column headers]  
                  [-m include recent data only]  
                  [-i batch export]
```

**[-p** *Max rows to export*]

**[-d** *debug mode*]

**[-quotes** *put double quotes around string data*]

## Overall Process and Functionality

`trend_export` generates ASCII files of data extracted from the PI database using the SQL query defined in any PI report. The ASCII file generated can be saved in any location.

The utility uses the source SQL in the file - including the constraints - as the basis for a database query, and stores the rows returned as an ASCII file. The query can be limited to a single device/instance pair or run against a set of pairs. The output can be saved in a separate file for each pair or all the data can be combined into a single output file. `trend_export` can be invoked on a single PI report or for a set of reports by using an input parameter file.

The ASCII files generated can be in either a tab separated format or a comma separated values (CSV) format. The files can contain or omit column headers. A facility for generating data for just the most recent time period or between to specific dates is also available.



The number of lines to be processed can be limited, which prevents the generation of overly large output files.

Lastly, each report to be run, either individually from the command line or through a parameter file, can query any database server system accessible from the user's machine.



# Controls and Switches

There are multiple parameters to control the execution of the program. The command line options are:

Flag	Description	Argument	
<b>-a</b>	Include column headers. Headers include both a line for the column names and a line of hyphens (-) underlining each column name.	y	The ASCII file includes column headers (default).
		n	The ASCII file does not include column headers.
<b>-d</b>	Show debug output. If this option is included, the entries made in the log file are also displayed on your screen. This option should only be used for testing in coordination with HP Technical Support.	0	No debug output.
		1, 2, or 3	The higher the number, the more detailed the debug output.
<b>-f</b>	Specifies the format of the ASCII output file.	tab	Generate a tab separated ASCII output file (default).
		comma	Generate a comma separated ASCII output file. Character data is enclosed in quotation marks.
<b>-h</b>	Help. Display command line options.	N/A	
<b>-i</b>	The full path and file name of the input parameter file.	The name and directory of any valid parameter file.	

Flag	Description	Argument	
<b>-m</b>	Only include data with the most recent timestamp; that is, the maximum <code>ta_period</code> value in the source data table.	y	Only include data with the most recent timestamp.
		n	Include all data regardless of timestamp (default).
<b>-o</b>	The name of the ASCII output file. To ensure that existing files are not overwritten, the actual output file name has a timestamp appended to it in the format <code>YYYYMMDDhhmm</code> . If <b>-o myfile.txt</b> is specified and <code>trend_export</code> is run on April 10, 2001 at 3:30pm, the output file name is <code>myfile.txt.20010410153</code> .	The complete path of the ASCII output file to be generated.	
<b>-p</b>	The maximum number of lines to be generated in the output file.	Any positive integer (default = 5000).	
<b>-r</b>	The name of the PI report ( <code>.rep</code> ) to be used.	The complete path of any PI report.	
<b>-quotes</b>	Specifies whether to put double quotation marks around string data	y	<i>Put double quotation marks around string data</i>
		n	<i>Do not put double quotation marks around string data</i>
<b>-s</b>	The name of the database server from which the data is extracted.	The name of the database server. (default = Current value of the user's <code>DSQUERY</code> environment variable.)	

Note that whenever the **-i** option is used, all other command line options are ignored except the **-q** option. In this case, the named input parameter file controls ASCII file generation exclusively.

The utility requires one environment variable set: `DPIPE_HOME`. This variable contains the path of the directory where all system components reside. The utility writes all of its messages to the `trend.log` file and looks for the export control parameter files in the `DPIPE_HOME/lib` directory.

Here are a few examples of launching `trend_export` from the command line.

#### Example 1

```
trend_export -r /OVPI/reports/Daily_Lan_Inventory.rep  
-o /daily_lan_report.txt -f comma
```

This would use the PI report `Daily_Lan_Interface_Inventory.rep` as input and produce an output ASCII file with the name `daily_lan_interface_inventory.txt`. Notice that the fully qualified path names were specified for both the report and the ASCII output file. In addition, note the use of the **-f** option specifying that the ASCII file should be comma delimited.

#### Example 2

```
trend_export -i /OVPI/reports/ascii_parm.dat
```

This invocation would use the input parameter file `ascii_parm.dat` found in the directory named `$DPIPE_HOME/lib` to drive the ASCII file creation process. See [Input Parameter File](#) on page 452 for the parameter file format.

#### Example 3

```
trend_export -i /OVPI/reports/ascii_parm.dat -q hidekel
```

This invocation would work like the previous one but would use the database server `hidekel` as the source of the data generated during processing.

# Input Parameter File

When using the **-i** option to point the process to an input parameter file all other command line options are ignored except for **-q** (DSQUERY). This is the preferred method for using the utility. Only by using the input parameter file option can multiple reports and multiple instances of those reports be generated. Here are the guidelines for the parameter file:

The rules for formatting a parameter file are:

- One entry per line.
- All entries of keywords with arguments use the format:  
*keyword=argument*.
- Each report to be run must be blocked off with a start and end statement.
- An asterisk (\*) in the first column indicates a comment.
- Any keyword not listed here is ignored.

Valid keywords for the parameter file are:

Keyword	Associated Command Line Option	Required	Valid Values	Default
<b>start</b>		Yes	N/A	N/A
<b>end</b>		Yes	N/A	N/A
<b>report</b>	<b>-r</b>	Yes	Any qss file path	None
<b>output</b>	<b>-o</b>	Yes	ASCII file path	None
<b>dsquery</b>	<b>-q</b>	Yes	valid database server name	DSQUERY environment variable value

<b>Keyword</b>	<b>Associated Command Line Option</b>	<b>Required</b>	<b>Valid Values</b>	<b>Default</b>
<b>headers</b>	<b>-a</b>	No	y n	<b>y</b> (yes)
<b>format</b>	<b>-f</b>	No	comma tab	tab
<b>max</b>	<b>-m</b>	No	y n	<b>n</b> (no)
<b>exception</b>		No	y n	<b>n</b> (no)
<b>instances</b>		No	valid target name/ table key pairs, or file containing target name/table key pairs. See below for format.	None
<b>maxlines</b>	<b>-p</b>	No	any positive integer	5000
<b>condense</b>	<b>-c</b>	No	y n	<b>n</b> (no)
<b>style</b>		No	individual combine	individual
<b>begindate</b>		No	date in US format	None
<b>begintime</b>		No	timestamp hh:mm	00:00
<b>finishdate</b>		No	date in US format	None

Keyword	Associated Command Line Option	Required	Valid Values	Default
<b>finishtime</b>		No	timestamp hh:mm	23:59
<b>frequency</b>		No	daily weekly monthly	None
<b>*</b>		No	Any text. This is a comment line.	

The format for an instances line is:

```
instances = (target_name, table_key);(target_name,table_key)...
```

Multiple instances lines can be added for any report (see examples).

```
instances = /tmp/exception_list
```

This usage of the keyword points to a file containing a list of target name/table key pairs in the form:

```
target-name1, table-key.1
target-name1, table-key.2
target-name2, table-key.1
```

Note that the instances file can be generated by specifying a report as an exception report. When a report specification block contains the **exception=y** keyword, the ASCII output file generated is simply a list, in the above described format, which is the matching target name/table key pairs from the exception criteria. This allows for the results from an exception report to be used to drive drilldown reports. See [Example 4](#) on page 458.

The **style** keyword determines whether the rows returned for multiple instances are combined into a single output file written to separate files, one for each target name/table key. If no instances are listed the **style** keyword is

ignored. If **style = individual** is specified, each output file name has *target\_name.table\_key* appended to its name in addition to the timestamp described above.

The **condense** keyword causes *trend\_export* to remove trailing blanks from each column included in the SQL select statement.

If the **max** keyword is used, then the **begindate**, **finishdate**, **begintime**, and **finishtime** constraint keywords are ignored. Also, any embedded **start** or **end** date constraints in the original *.rep* file are ignored.

If the **frequency** keyword is used (**daily**, **weekly**, or **monthly**) then any other date constraint is ignored. That is, if in the same report specification block the **max** keyword or the **begindate** or the **finishdate** keywords are used, then they are bypassed.

If you plan on running the same report on a daily, weekly, or monthly basis, then use the **frequency** keyword for the required time period. That way, no modification to the control parameter file is needed. By specifying the daily, weekly, or monthly **frequency** value, the following actions are done:

- for **daily** => the previous day's data is queried for. For example, if the current date is 6/1/2001, then the report would query for data from 5/31/01 00:00 to 5/31/01 23:59. No matter when the report is run, the previous day's data is queried for.
- for **weekly** => the previous week's data is queried for. For example, if today was Wednesday May 23, 2001, then the report would query for data from Monday May 14, 2001 00:00 to Sunday May 20, 2001 23:59. Again, no matter when the report is actually run, the previous week's data is queried.
- for **monthly** => the previous month's data is queried for. For example, if today was May 01, 2001, then the report would query for data from April 1, 2001 00:00 to April 30, 2001 23:59.

# Examples

## Example 1

```
* this is a comment line
*
start
report = /apps/OVPI/reports/Mib-II/
Daily_LAN_Interface_Inventory.rep
output = /tmp/daily_lan_interface_inventory
headers = n
format = comma
dsquery = HARDWARE_SYBASE
condense = n
maxlines = 2000
style = individual
end
```

**Example 1 would run the PIREport Daily\_LAN\_Interface\_Inventory.rep and place the output in /tmp/daily\_lan\_interface\_inventory.timestamp. The ASCII file generated would have no column headers and would be in a condensed, comma delimited format. Lastly, the query would be run against the database server HARDWARE\_SYBASE.**

## Example 2

```
* this is a comment line*
start
report = /apps/OVPI/reports/Mib-II/
Daily_LAN_Interface_Inventory.rep
output = /tmp/daily_lan_interface_inventory
headers = n
format = comma
dsquery = HARDWARE_SYBASE
end

start
report = /apps/OVPI/reports/Mib-II/Serial_Interface_Usage.rep
output = /tmp/serial_interface_usageendstart
report = /myreports/lan_interface_load.rep
output = /myreports/lan_interface_load
```



```

dsquery = HARDWARE_SYBASE
max = y
instances = (mynode1,1); (mynode1,2); (mynode2,1); (mynode2,2)
instances = (mynode3,3); (mynode4,1)
end

start
report = /myreports/lan_interface_load.rep
output = /myreports/lan_interface_load
max = y
instances = (mynode1,1); (mynode1,2); (mynode2,1); (mynode2,2)
instances = (mynode3,3); (mynode4,1)
style = combine
format = comma
dsquery = HARDWARE_SYBASE
condense = y
end

```

Example 2 causes trend\_export to run a set of reports.

- 1 First it would run the same report as in Example 1.
- 2 The second report run would be Serial\_interface\_Usage.rep. Its output would be a tab separated, uncondensed data saved in a file called /tmp/serial\_interface\_usage.timestamp. Notice that all the default values would be used for this second report.
- 3 The third report to be run, lan\_interface\_load.rep would produce six tab separated, uncondensed ASCII files because the keyword **style = individual** is used. The .rep report source file is used as a template to run the query for each of the listed instances. Note that the instances are on multiple lines. They could have been all listed on the same line. Also note that the **max** keyword was added to indicate that the ASCII files only contain data for the most recent period available for each instance. The output files are named:

```

lan_interface_load.mynode1.1.timestamp
lan_interface_load.mynode1.2.timestamp
lan_interface_load.mynode2.1.timestamp
lan_interface_load.mynode2.2.timestamp
lan_interface_load.mynode3.3.timestamp
lan_interface_load.mynode4.1.timestamp

```

- 4 The fourth report would produce the same data as the third report, but it would only produce a single output file called **lan\_interface\_load.timestamp**. In addition, the output file would be comma delimited and condensed.

### Example 3

```
start
report = /myreports/lan_interface_load.rep
output = /myreports/lan_interface_load
max = n
begindate = 03/01/01
finishdate = 03/31/01
instances = (mynode1,1); (mynode1,2); (mynode2,1); (mynode2,2)
instances = (mynode3,3); (mynode4,1)
style = combine
format = comma
dsquery = HARDWARE_SYBASE
condense = y
end
```

Example 3 causes `trend_export` to run a set of reports, in a combined format; that is, all the data from the listed instances is combined into one output file, for only data whose `ta_period` lies between March 1st and March 31st.

### Example 4

```
start
report = /myreports/critical_interfaces.rep
output = /tmp/critical_interfaces_list
frequency = daily
exception = y
max = n
begindate = 03/01/01
finishdate = 03/31/01
format = comma
dsquery = HARDWARE_SYBASE
end
start
report = /myreports/lan_interface_performance_report.rep
output = /myreports/heavy_hitters
frequency = daily
max = n
```

```
instances = /tmp/critical_interfaces_list
format = comma
dsquery = HARDWARE_SYBASE
style = individual
end
```

Example 4 demonstrates the use of the **exception** keyword. In the first report the **exception** keyword has been turned on. This causes the system to generate a target name/table key listing in the file /tmp/critical\_interfaces\_list. The generation of this list is based upon whatever criteria was specified in the exception report. The output from this exception report is then used as an instances input list for the next report. Note that the second report is set to **style = individual**, which means that for each target name/table key pair a separate output file is generated. Also note that the **frequency = daily** keyword was used. This means that both the exception report and the drilldown report only queries for the previous day's data.

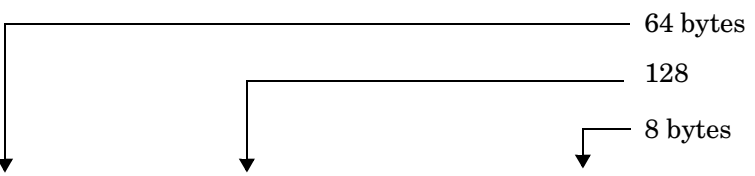
# Output Format

As stated earlier, the ASCII files can be generated in two formats, tab or comma separated and both formats can be generated either in a condensed or uncondensed mode. Here are examples of how the output would look in the various possible combinations of these methods. They are based on a PI report which utilized the following columns:

Column Name	Size in Database
dsi_target_name	64 bytes
dsi_table_key	128 bytes
ifinoctets018	8 bytes

## Tab Separated Format: Not Condensed

The tab format consists of columns of data aligned with the next in a series of standard, equally spaced tab stops. That is, the actual amount of separation between columns is the number of spaces from the end of the column to the next tab stop. Because this example is not condensed, the amount of space occupied by each column is based on the field size defined in the database. Below is an example of a tab separated ASCII output file with the headers option set to **yes**:



dsi_target_name	dsi_table_key	ifinoctets018
your-router	1	123456
your-router	2	123456
your-router2	1	1234567

The output generated if the **headers = n** keyword was specified would be the same data in the same format but without the column headings:

your-router	1	123456
your-router	2	123456
your-router2	1	1234567

## Comma Format: Not Condensed

The comma format consists of columns of data separated by commas. Each column occupies the number of bytes specified for it in the database. Note that whenever a comma delimited file is requested, all character-based data is enclosed in quotation marks. With **headers = y** set, the output would look like this:

dsi_target_name	, dsi_table_key	, ifinoctets018
'your-router'	, '1'	, 123456
'your-router'	, '2'	, 123456
'your-router2'	, '1'	, 1234567

Without headers, the same data would be generated, but without the column headings:

'your-router'	, '1'	, 123456
'your-router'	, '2'	, 123456
'your-router2'	, '1'	, 1234567

## Tab Format: Condensed

The condensed mode of the tab separated format generates a tab delimited file, but the columns are only as wide as the actual data rather than being padded to the defined field width. The columns are aligned with the next tab stop. With **headers = y** the output would be:

```
dsi_target_name dsi_table_key ifinoctets018
your-router 1 123456
your-router 2 123456
your-router2 1 1234567
```

This same output with **headers=n** would be:

```
your-router 1 123456
your-router 2 123456
your-router2 1 1234567
```

## Comma Format: Condensed

The condensed comma format is a series of columns that are only as wide as the actual data they contain, separated by commas. Note that whenever a comma delimited file is requested, all character-based data is enclosed in quotation marks.

With **headers**, the ASCII file generated would be:

```
dsi_target_name,dsi_table_key,ifinoctets018
'your-router','1',123456
'your-router','2',123456
'your-router2','1',1234567
```

If the **headers=no** option was used, then this would appear as:

```
'your-router','1',123456
'your-router','2',123456
'your-router2','1',1234567
```

## 34 trend\_label

You can use the `trend_label` command to populate one or more columns in a property table with data from its counterpart data table.



Note that you cannot use `trend_label` with summarized data tables. That is, columns from tables generated by `trend_sum` does not populate columns in a property table.

Each data table in the PI database is associated with a property (key) table. By default, most of the columns in a data table are not present in its associated property table. However, you may find it useful to include some of these data columns in the property table. `trend_label` enables you to do this.

You can run `trend_label` from the command line, and schedule it to run regularly in `trendtimer.sched`.

# Syntax

The `trend_label` command uses the following syntax:

```
trend_label [ -c [ alias= ] column ]  
              [ -d debug_level ]  
              [ -e column ]  
              [ -h ]  
              -k key_table  
              [ -n ]  
              [ -o hour ]  
              [ -r [ alias= ] column ]  
              -t source_data_table  
              [ -v ]
```



# Options

The `trend_label` command has the following options:

- c**      Use this option to identify the column in the source data table that you want to use to populate the column in the property table. The *alias* parameter is the name assigned to the column in the property table and the *column* parameter is the name of the column in the source data table. When you use this option, `trend_label` searches the property table for the column named by *alias*. If the column named *alias* exists, and if its value is null or blanks, `trend_label` populates the column with data from the column named *column* in the source data table. If the column exists and its value is non-null or non-blank, the existing value in the target property table row is not changed.

If the column named *alias* does not exist in the property table, `trend_label` creates it and then populates it in the manner described above.

You can repeat this option multiple times in the `trend_label` statement to identify different property and data table columns or to concatenate column substrings or columns in the source data table to populate the property table column.

If you omit *alias=* from the command, the property table column is assigned the same name as the column in the source data table.

See [Usage Notes](#) on page 467 and [Examples](#) on page 470 for a complete description of this option with examples.
- d**      Sets the debug output level. Valid values are 1, 2, and 3. The higher the number, the more detailed the information. Debug output is written to the standard output destination. This option is for development purposes.
- e**      Identifies the column in the source data table that provides the data to populate the `dsi_descr` column in the property table.
- h**      Displays the syntax of the `trend_label` command.
- k**      Identifies the destination property table. Specify the SQL name of the table.

- n** Use this option to specify that the default target row in the source data table is the one where the maximum `ta_period` that is less than or equal to the current time (the minutes and seconds portions of the `ta_period` value are set to zeroes) minus 1 hour on today's date.

If you omit both this option and the **-o** option, the default is the latest `ta_period`.

This option cannot be on the same command line as the **-o** option.

See [Locating the Target Row in the Source Data Table](#) on page 467 for a detailed explanation of how the target row is located.
- o** Use this option to specify the hour of the previous day. Valid values are **0** (midnight) through **23**. The system uses this value to locate the row in the source data table associated with the `dsi_key_id` that it uses to populate the property table columns.

If you omit both this option and the **-n** option, the default is the latest `ta_period`.

This option cannot be on the same command line as the **-n** option.

See [Locating the Target Row in the Source Data Table](#) on page 467 for a detailed explanation of how the target row is located.
- r** Uses the same syntax and application as the **-c** option with the following exception: The target value from the source data table populates the associated column in the property table even if that column already exists and has a non-null or non-blank value.
- t** Identifies the SQL name of the source data table. The table is typically a rate table (but can be a raw table). Summary tables cannot be used as a source.
- v** Displays the current `trend_label` command version. For example, a return value of 4.0 means version 4.0.

# Usage Notes

This section provides information on various aspects of the `trend_label` command usage.

Note that omitting `alias=` from the command may produce an ambiguous error message such as "prog-name date/time - Ambiguous column name *XXXX*", which indicates a situation where the property table and data table both have a column called *XXXX*. The query uses both property and data tables in the `from` clause and uses only *XXXX* in the `select` clause; it does not specify the owner table name.

To resolve this situation, do the following:

- 1 Start a SQL session with user **dsi\_dpipe**.
- 2 Perform **sp\_help** on the property table to get the column-name listing.
- 3 Drop the improper column from the property table with the following syntax:  

```
alter table tablename drop column_name
```
- 4 Exit the SQL session.
- 5 Correct the `trend_label` syntax to use `alias=column`.

## Populating the dsi\_descr Column

When you generate a report, the report title appears above the graph and the legend appears below the graph. If the value of the `dsi_descr` column in the property table that the report uses is null or blank, the legend simply displays the name of the node on which the report is based. However, if the `dsi_descr` column is populated, the legend displays the description that appears in the column. This can make graphical reports easier to identify and read.

## Locating the Target Row in the Source Data Table

The source data table contains one row for each unique `dsi_key_id` for each collection period. Thus, if collections occur at 15 minute intervals, there are 96 rows for each unique `dsi_key_id`, which is four rows per hour \* 24 hours per day equals 96 rows.

If the **-o** option is used, the system compares the hour (0-23) specified in the option to the `ta_period` timestamp in the rows of the source data table as follows:

- 1 Get the timestamp for the current time minus 24 hours. Thus, if the current time is 11:22:35 on June 12, 2001, the resulting timestamp is 11:22:35 on June 11.
- 2 Make the hour portion of the timestamp equal to the value (0-23) specified in the **-o** option. Thus, if the `trend_label` command line has the option **-o 4** on it, the timestamp is 04:22:35 on June 11.
- 3 Set the minutes and seconds portions of the timestamp to zeroes. In this example, 04:22:35 on June 11 becomes 04:00:00 on June 11. Thus, the target record in the source data table in this example is the one for the `dsi_key_id` where `ta_period` is Jun 11 2001 04:00:00.
- 4 If there is no row with the target time period, `trend_label` decrements the hour by 1 until it finds a match. For example, if there is no row for the `dsi_key_id` where the `ta_period` time is 04:00:00, `trend_label` looks for a time of 03:00:00, decrementing the hour by 1 in this manner until it finds a match.

If the **-n** option is used, the target row in the source table is the one for `dsi_key_id` where the maximum `ta_period` timestamp is less than or equal to the current time (with hours and minutes zeroed) minus 1 hour. For example, if the current time is 10:30 AM on June 12, 2001:

- 1 Get the `dsi_key_id` record in the source data table where the `ta_period` timestamp is Jun 12 2001 09:00:00 AM.
- 2 If no record exists for 09:00:00, `trend_label` decrements the hour by 1 until it finds a match. For example, if there is no row for the `dsi_key_id` where the `ta_period` time is 09:00:00, `trend_label` looks for a time of 08:00:00, decrementing the hour by 1 in this manner until it finds a match.

If neither the **-o** option nor the **-n** option is used, the default is the last `ta_period` timestamp in the rows of the source data table.

## Ensuring Property Table/Data Table Compatibility

You must ensure that the specified data table (**-t** option) uses the specified property table (**-k** option) in the PI database. The `trend_label` command does not check for this.

## Update Restrictions

You cannot update the values in the property table columns `dsi_key_id`, `dsi_target_name`, or `dsi_table_key`.

## Extracting Substrings from Column Values

When you use the **-e**, **-c**, and **-r** options, you can specify a substring that `trend_label` extracts from the source column and use to populate the destination property table column with the following syntax:

*column:offset,length*

Where, *column* is the column name in the source data table.

*offset* is the starting character position of the substring in the column.

*length* is the number of characters to include in the substring beginning with the offset column position.

For example, if you want to extract a substring that is 8 characters long and begins in position 13 of the `ta_period` value, use the following entry.

`ta_period:13,8`

This entry replaces the *column* parameter in the **-e**, **-c**, or **-r** option.

## Concatenating Column Values

You can populate a property table column with a value that is concatenated from the values of multiple columns or column substrings in the source data table by repeating the **-e**, **-c**, and **-r** options using the same *alias=* and different source column names or substrings. For example, if you want to create, if necessary, and populate a property table column named

dsi\_eurodate with the concatenated substrings from positions 5-7, 1-4, and 8-11 of the ta\_period column in the source data table, you can use the following trend\_label command:

```
trend_label -k Ksi_dbstats_ -t Rsi_dbstats_  
-c dsi_eurodate=ta_period:5,3 -c dsi_eurodate=ta_period:1,4  
-c dsi_eurodate=ta_period:8,4
```

In this case, if the value of ta\_period in the source data table is **Jul 29 2001 04:00:00:000AM**, then the value used to populate dsi\_eurodate in the property table is **29 Jul 2001**.

## Examples

### Example 1

If you want to populate the dsi\_descr column in the Kib\_ii\_ifentry\_ property table with the latest data contained in the ifdescr010 column from the mib\_ii\_ifentry\_ source data table, use the following command.

```
trend_label -k Kib_ii_ifentry_ -t mib_ii_ifentry_  
-e ifdescr010
```

### Example 2

If you want to populate three columns in the Kib\_ii\_ifentry\_ property table with the latest data contained from the mib\_ii\_ifentry\_ source data table, use the following command.

```
trend_label -k Kib_ii_ifentry_ -t mib_ii_ifentry_  
-e ifdescr010 -c speed=ifspeed013 -c type=iftype011
```

In this example, trend\_label populates the dsi\_descr column in the Kib\_ii\_ifentry\_ property table with the latest data contained in the ifdescr010 column from the mib\_ii\_ifentry\_ source data table. In addition, it creates the column (if it does not already exist) and populates the property table column named speed with the value for ifspeed013 in the target source data table row and the property table column named type with the value for iftype011 in the target source data table row if the property table column values are null or blank. If the value in either column is non-null or non-blank, the current value remains the same.

### Example 3

If you want to populate the `dsi_descr` column in the `Ksi_dbstats` property table with a concatenation of the value for the `user_name` column and the first three characters of the `ta_period` value in the target source table row, you can use the following command.

```
trend_label -k Ksi_dbstats_ -t Rsi_dbstats_ -e user_name  
-e ta_period:1,3
```

In this example, if the `user_name` value is **Jones** and the first three characters of the `ta_period` value are **Nov**, then the value **JonesNov** is the value in the `dsi_descr` column in `Ksi_dbstats`, only if the existing value is null or blanks. If the existing `dsi_descr` value is non-null or non-blank, the value is not changed.

### Example 4

If you want to populate the `dsi_descr` column in the `Ksi_dbstats` property table in the same way as described for with the exception that the existing value of `dsi_descr` is replaced by the concatenation even if that value is non-null or non-blank, you can use the following command.

```
trend_label -k Ksi_dbstats_ -t Rsi_dbstats_ -o 4  
-r dsi_descr=user_name -r dsi_descr=ta_period:1,3
```

In this example, the `-r` option causes unconditional replacement of the value; whereas, the `-e` option replaces the value only if the existing value in the target property table record is null or blanks. Furthermore, the target record in the source data table is the `dsi_key_id` record where the `ta_period` value is 04:00:00AM on the previous day.

### Example 5

If you want to do the following:

- use the data from the `Rsi_dbstats_ data` table where `ta_period` for the `dsi_key_id` equals 04:00:00 AM on the previous day
- concatenate the `user_name` and the first three characters of the `ta_period` value in `Rsi_dbstats_ data` table and place the result in the `dsi_descr` column in the `Ksi_dbstats_ property` table

- create a column named `dsi_eurodate` in the `Ksi_dbstats_` property table, if the column does not already exist, and unconditionally update that column with the result of concatenating positions 5-7, 1-4, and 8-11 from the `ta_period` value

You can use the following command:

```
trend_label -k Ksi_dbstats_ -t Rsi_dbstats_ -o 4
-e user_name -e ta_period:1,3
-r dsi_eurodate=ta_period:5,3
-r dsi_eurodate=ta_period:1,4
-r dsi_eurodate=ta_period:8,4
```

### Example 6

If you want to populate the `dsi_descr` column in the `proptbl1` property table with the data contained in the `ifdescr010` column from the `datatbl2` source data table that has an earlier or equal timestamp to the current time, use the following command.

```
trend_label -k proptbl1 -t datatbl2 -e ifdescr010 -n
```

If the time when the `trend_label` command executes is 11:45 AM, the `dsi_key_id` record with a `ta_period` value of today's date at 10:00:00 AM is used. If there is no record for 10:00:00 AM, `trend_label` searches for a record for 09:00:00 AM and so on backward in 1-hour increments until a match is found, which is the maximum `ta_period` value that is equal to or less than the current hour minus 1 hour.



## 35 trendpm

The `trendpm` command manages stored procedures used by PI.

### Syntax

The `trendpm` command has the following syntax:

```
trendpm      [ -c code_gen_file ]  
              [ -cs "command_string" ]  
              [ -d ]  
              [ -db trace_level ]  
              [ -e ]  
              [ -f file_name ]  
              [ -g ]  
              [ -h ]  
              [ -ot target_table_name ]  
              [ -pe "parameter1=value [, parameter2=value] ..." ]  
              [ -pg "parameter1=value [, parameter2=value] ..." ]  
              [ -r ]  
              [ -s db_server_name ]  
              [ -t table_name, [ proc_app_type ], [ proc_class ], [ proc_type ] ]  
              [ -v ]
```

# Options

The `trendpm` command has the following options:

- c** Specifies the name of the code generated file.
- cs** This option contains command string options from another program such as `trend_sum`.
- d** Delete procedure.
- db** Set trace parameters. Valid values for the trace level are **1 - 5** where **1** is the lowest verbosity and **5** is the highest verbosity.
- e** Executes procedure synchronously.
- f** Specifies the name for the legacy file associated with the procedure to be registered.  
The *file\_name* contains the full path with the file name.
- g** Generates SQL procedure code.
- h** List help options.
- ot** Specifies target table name.

**-pe** Specifies the execution parameters for raw-to-delta procedures. The format for the valid parameters are:

**@bArchive**=*turn\_off\_archiving*  
**@bCheck\_index**=*check\_index*  
**@bDelta\_time**=*delta\_time*  
**@bSuppress\_spike**=*suppress\_spikes*  
**@bVerbose**=*verbose\_value*  
**@debug\_level**=*debug\_level\_pm*  
**@hwm\_data**=*high\_water\_mark*  
**@hwm\_log**=*high\_water\_mark\_log*  
**@line\_suppress\_value**=*min\_filter\_value*  
**@max\_tran\_row\_cnt**=*num\_rows*  
**@retry\_count**=*retry\_count*  
**@retry\_interval**=*retry\_interval*  
**@user\_proc\_name**=*'proc\_name'*  
**@zerror**=*clock\_error\_value*

Do not use any spaces in the parameter list; however, there is a space after the **pe** option. See [pe Option Parameters](#) on page 476 for the descriptions of the parameters.

**-pg** Optional parameter values to replace default values in the generated output SQL procedure. The format for the valid parameters are:

- **@column\_suppress**=*'column\_name'*
- **@column\_suppress\_value**=*suppress\_value*

The *column\_name* is the name of the column containing counter data to suppress. The *suppress\_value* is the value in the column to suppress.

Do not use any spaces in the parameter list; however, there is a space after the **pg** option. This option only applies for raw-to-delta procedures.

**-r** Generates and registers procedure.

- s** Specifies database server name.
- t** Specifies table name, procedure application type, procedure class, and procedure type. Note that the procedure application type, procedure class, and procedure type values are optional and positional. When you use this option, you must enter at least the *table\_name* and the commas. See [Procedure Application Type Values](#) on page 479 for the valid values.
- v** Displays version stamp for `trendpm`.



Note that the **-pe** and **-pg** options require double quotation marks (") to enclose the parameters. Use single quotation marks (') to enter any character data from the parameters.

## pe Option Parameters

The descriptions for the **pe** option parameters follow.

- |                      |  |
|----------------------|--|
| <b>@bArchive</b>     | Enables archiving of raw data. The archive function stores the collected data in a raw data table. A value of <b>1</b> archives the raw data. A value of <b>0</b> does not archive the raw data.<br>The default is <b>1</b> .  |
| <b>@bCheck_index</b> | Specifies whether to use existing indices on the upload table or to drop existing indices and then recreate them. The value <b>1</b> means that the existing indices on the upload table are used. The value <b>0</b> means that the existing indices are dropped and then recreated. If the value is <b>1</b> and the proper indices are missing then the <code>trendpm</code> invocation fails.<br>The default is <b>0</b> . |

<b>@bDelta_time</b>	<p>Specifies which clock to use to calculate Delta Time. The value of <b>1</b> directs the procedure to use System Uptime to calculate Delta Time. The value of <b>0</b> directs the procedure to use the Agent Clock Column, which is the received_ts column, for the calculation.</p> <p>The default is <b>0</b>.</p>
<b>@bSuppress_spike</b>	<p>Specifies whether to reject samples if there are spikes. A spike occurs when the value of any counter suddenly goes too high, which is when the difference of two consecutive samples from a counter exceeds the spike threshold. The value of the spike threshold is <math>2^{31}</math> for 32-bit counters or <math>2^{51}</math> for 64-bit counters. Remember if the difference of the values is negative, account for the rollover of the counter by adding <math>2^{32}</math> for 32-bit counters or <math>2^{52}</math> for 64-bit counters.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> <li>• <b>1</b>: Rejects samples if a spike occurs.</li> <li>• <b>0</b>: Does not reject samples.</li> </ul> <p>The default is <b>1</b>.</p>
<b>@bVerbose</b>	<p>Sets the debug output level for the raw-to-delta procedures.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> <li>• <b>0</b>: Suppress all messages</li> <li>• <b>1</b>: Show row suppression messages</li> </ul> <p>The default is <b>1</b>.</p>
<b>@debug_level</b>	<p>Sets the debug output level for the trendpm process of mw_collect. Values of <b>0</b>, <b>1</b>, <b>2</b>, or <b>3</b> are valid. The higher the number, the more detailed the information.</p> <p>The default is <b>0</b>, which means no debug output.</p> <p>Debug output is written to standard output. You should only use this option for testing in coordination with HP Technical Support due to the additional overhead it places on mw_collect.</p>

<b>@hwm_data</b>	<p>Specifies the <i>high_water_mark</i>. The high water mark denotes the level at which the database-used size reaches the specified percentage and data collection stops. Valid values are <b>1–100</b>.</p> <p>The default value is <b>90</b> for 90%, which means data aggregation stops when the database is 90% full.</p>
<b>@hwm_log</b>	<p>Specifies the <i>high_water_mark_log</i>, which is the high water mark for the log. Valid values are <b>1–100</b>.</p> <p>The default value is <b>90</b> for 90%, which means data aggregation stops when the log is 90% full.</p>
<b>@line_suppress_value</b>	<p>Sets the minimum filter value. The procedure rejects the sample if the delta value of a counter falls below this value.</p> <p>The default value is <b>-1</b>, which means to accept the entire sample.</p>
<b>@max_tran_row_cnt</b>	<p>Sets the size of the batch to commit, which is the number of rows to commit per transaction.</p> <p>The default value is <b>0</b>, which means the system calculates this value dynamically per table.</p> <p>HP recommends that you use the default value. If this value is too low, it may affect performance; if this value is too high, it may affect concurrency.</p>
<b>@retry_count</b>	<p>Sets the number of times the procedure tries to acquire a lock on the specified object.</p> <p>The default value is <b>60</b>, which is 60 retries.</p>
<b>@retry_interval</b>	<p>Sets the number of seconds the procedure must wait to acquire a lock on the specified object.</p> <p>The default value is <b>60</b>, which is 60 seconds.</p>

<b>@user_proc_name</b>	<p>Specifies the name of the custom procedure to run at the end. You can register it as a dependent procedure with the following command:</p> <pre><b>trendpm -r -t input_table,C,D,R</b> <b>-ot output_table -c SQL_file.</b></pre> <p>The <i>SQL_file</i> includes the custom procedure.</p> <p>The default value is <b>NULL</b>, which means that there is no custom procedure.</p>
<b>@zerror</b>	<p>Sets the percentage level for valid data. The value is the percentage of difference between the delta values of two Received Timestamps and two System Uptimes. These statistics come from two consecutive raw data samples.</p> <p>For example, if r1 and s1 are the Received Timestamp and System Uptime for the first sample, and r2 and s2 are the Received Timestamp and System Uptime for the second sample, then the calculation for the value is <math>((r2 - r1) - (s2 - s1)) * 100 / (r2 - r1)</math>. During processing, if the calculated value for the samples exceeds the value set by this option then the samples are rejected.</p> <p>This value is an integer that is less than 255.</p> <p>The default value is <b>10</b>.</p>

## Procedure Application Type Values

The following table provides a list of valid procedure application types and descriptions for the corresponding procedure types.

**Table 22 Valid Values for Procedure Application Type with Procedure Type**

<b>Procedure Type</b>	<b>Valid Procedure Application Type</b>	<b>Procedure Application Type Description</b>
C	C	Custom application type: Custom procedure of type, <b>custom</b> .
R	C	Custom application type: Normalization procedure
R	S	Asset management procedure
R	R	Raw-to-delta procedure
R	Z	Copy stored procedure
U	C	Custom application type: Summary procedure
U	N	Ranking procedure
U	T	Legacy trendit replacement procedure
U	G	General summary procedure
U	F	Forecast summary procedure
U	B	Baseline summary procedure



## 36 trend\_proc

The `trend_proc` command is a utility that enables you to group together multiple interrelated commands on a PI system.

### Requirements and Restrictions

- Limit nesting levels to avoid straining the resources.
- Avoid accessing the same database table at the same time.

### Syntax

The `trend_proc` command uses the following syntax.

```
trend_proc  [ -d debug_level ]  
             -f file_name  
             [ -h ]  
             [ -n ]  
             [ -v ]
```

# Options

The descriptions for the command line options of the `trend_proc` command follows:

- d** Use this option to set the debug output level for `trend_proc`. Values of **0**, **1**, **2**, **3**, or **4** are valid. The higher the number, the more detailed the information.  
The default is **0**, which means no debug output. Debug output is written to standard out. You should only use this option for testing in coordination with HP Technical Support due to the additional overhead it places on `trend_proc`.
- f** Use this option to specify the name of the text file that contains the instructions. If the file is not in the current working directory, you must specify the fully qualified path to the file.  
This is a required option.  
See [The `trend\_proc` Input File](#) on page 483 for a description of the contents for the file.
- h** Use this option to display the command-line syntax for the `trend_proc` command.
- n** Use this option to skip the `trend_lock` locking check. When you use this option, the system can run multiple instances of the same `trend_proc` command at the same time.  
This option applies only to the command where it appears; it does not apply to the commands in the `trend_proc` input file.
- v** Use this option to display the current version of `trend_proc`.  
This option is in UPPERCASE.

# Usage Notes

The concept behind `trend_proc` is relatively simple:

- 1 Take a sequence of processes and string them together in such a way that you can run them with a single command.  
In `trend_proc` terminology, this series of sequential commands makes up a block, which is the equivalent of a macro.
- 2 Next, make it possible to string blocks together like individual commands.
- 3 When one block finishes its assigned tasks, `trend_proc` automatically launches the next block and continues this process until it launches all of the blocks.
- 4 Then, to make the process more powerful, make it possible to launch multiple blocks concurrently, so that they run in parallel.
- 5 Finally, make it possible to launch the entire process with a single entry in `trendtimer.sched`.

For example, users whose PI networks include satellite servers must export data to their central database every day. Typically, the export process occurs separately from the data roll-up process. If you want to export the data as soon as the system aggregates it, you can put both the roll-up and export processes in a single `trend_proc` file. This way, as soon as the system aggregates the data, `trend_proc` automatically exports the data to the proper database.

## The `trend_proc` Input File

This section defines the elements of the input file and describes the characteristics of it. This section also describes how the system processes the input files.

### Input File Definitions

A `trend_proc` file contains one or more blocks of commands. A block is a sequential list of commands to run. For example, if there are six commands in a block, `trend_proc` runs the first command, and wait for that command to be

completed. As soon as the first command is complete, `trend_proc` initiates the second command. When the second command finishes, `trend_proc` starts the third, and so on until it completes all of the commands in the block.

Most `trend_proc` files contain multiple blocks, arranged in a linear progression. It is necessary for the blocks to be in the order of desired processing, because `trend_proc` reads the blocks sequentially.

However, `trend_proc` is able to run commands from different blocks concurrently. This is so because `trend_proc` recognizes two different kinds of blocks in a `trend_proc` file: `wait` and `nowait`.

- A `wait` block is so called because it causes `trend_proc` to wait until the block has completed its processing in full before moving on to another block.
- A `nowait` block permits `trend_proc` to go to the next block in the sequence, and begin executing the commands in that block. The processing in a `nowait` block does not finish before `trend_proc` starts processing the next block in the file.

The order of the blocks determines how `trend_proc` runs the commands from different blocks concurrently. For example, if the input file has three blocks with the first block as a `wait` block and the other blocks are `nowait` blocks, then the last two blocks runs concurrently. On the other hand, if the input file has three blocks with the first two blocks as `nowait` blocks and the third block is a `wait` block, then all three blocks runs concurrently. Note that the commands within the `wait` and `nowait` blocks run sequentially.

## Input File Characteristics

The `trend_proc` command requires an input file with the following characteristics.

- It is a text file with the extension `.pro`.
- It must contain at least one block, and the blocks must adhere to a particular format, as follows.
  - A block begins with the **begin** statement; it has the following syntax.

```
begin: blockname [ { -wait  
                   { -nowait } ]
```

Where, *blockname* is a unique name for the block.

- If you do not specify `wait` or `nowait`, the default for the block is `wait`.
- A block can contain zero or more statements that `trend_proc` runs sequentially. These statements can be any executable commands, which may or may not be PI commands. You must provide the fully qualified path for the command.
- If a block does not contain a command, it is an empty block. This does not affect the performance of the `trend_proc`, so long as the populated blocks have the correct format.
- A block ends with the **end** statement; it has the following format.  
**end:** [ *blockname* ]  
 Putting the block name in the **end** statement is optional. It can, however, be useful for identification purposes, especially if you nest one `trend_proc` inside of another. (Note that the brackets are not part of the syntax.)
- The words **begin** and **end** must be in lower case.
- There should not be a space between the statement, **begin** or **end**, and the colon.
- Comments within a block begin with the number character (#).

## Processing the Input File

This section describes how `trend_proc` processes an input file. For example, a `trend_proc` file contains five blocks of commands, two of which are `wait` blocks. Each block has a set of commands and appears in the following order in the input file.

Block 1: six commands (`wait`)

Block 2: seven commands (`nowait`)

Block 3: four commands (`nowait`)

Block 4: eleven commands (`wait`)

Block 5: eight commands (`nowait`)

Note that each block must have a unique name. The names used above are representative of a naming pattern. If you intend to create your own `trend_proc` files, it is a good idea to establish a naming convention for your `trend_proc` blocks.

The processing for this `trend_proc` file requires four steps, as follows.

### Step 1

When the system launches `trend_proc` from the appropriate `trend_proc` entry in `trendtimer.sched`, for example, it executes the first command listed in Block 1. When the first command completes its processing, `trend_proc` executes the second command in Block 1. Since this block is a wait block, `trend_proc` must complete the entire sequence of six commands in Block 1 before beginning the sequence of commands contained in Block 2.

### Step 2

After `trend_proc` executes the final command in Block 1, it begins immediately to run the commands in Block 2. Again, `trend_proc` runs the commands in Block 2 in their proper sequence. However, since Block 2 is a `nowait` block, `trend_proc` also begins to run the sequence of commands in Block 3. Since Block 3 is also a `nowait` block, `trend_proc` also begins to run the commands in Block 4. At this point, `trend_proc` is executing commands from three different blocks at the same time.

### Step 3

Block 4 is a wait block; so once again, `trend_proc` must complete the commands in Block 4 before beginning the processes in Block 5. However, it is possible that `trend_proc` runs all of the commands in Block 4 before blocks 2 and 3 are completed. In this circumstance, `trend_proc` does not have to wait for blocks 2 and 3, after Block 4 has completed its processing, `trend_proc` starts with Block 5.

### Step 4

When `trend_proc` executes the commands in Block 5, the sequence is finished. The processing for the `trend_proc` file is complete after all blocks finish processing.

Depending on how you format a `trend_proc` file, you can run commands from several blocks at once. In the example above, we see `trend_proc` run commands from three different blocks that could be running at the same time. This enables a large part of the individual `trend_proc` file to be running in parallel.

## Processing with Multiple `trend_proc` Files

A `trend_proc` file can have one or more additional `trend_proc` commands in it. In fact, multiple instances of the `trend_proc` command can run concurrently, each running parallel processes.

This process of creating a `trend_proc` file that contains one or more `trend_proc` commands can be especially useful. Since it is possible for `trend_proc` commands to conflict with each other, it is sometimes necessary to ensure that one `trend_proc` command completes its processing before the system launches another. If, however, you create a `trend_proc` file whose blocks are `trend_proc` commands, you can launch an entire sequence of PI procedures with one entry in the `trendtimer.sched` file.



Note that while it is often a good idea to have a master `trend_proc` controlling your `trend_proc` files, you should limit this nesting to two levels. Otherwise, you increase the chance of creating a conflict between your various processes.

Note that `trend_proc` has an automatic `trend_lock` locking check that prevents the system from processing more than one instance of the same command string at the same time. This means that if you have a `trend_proc` file and the initial `trend_proc` command runs automatically and periodically from `trendtimer`, the system does not start the same command at the next time period if the previous command did not finish processing first.

You can override the `trend_lock` locking check with the `-n` option. If you use the `-n` option on a `trend_proc` command that has an input file with multiple `trend_proc` commands in it, the option only applies to the `trend_proc` command where it appears. The effect of the option does not propagate down to any nested `trend_proc` or executable commands.

### Processing Example with Multiple `trend_proc` Files

You have a situation where you are obtaining data from various datapipes with `ee_collect` on an hourly basis. The data is coming from multiple locations and you want to process that data on an hourly basis.

You can create different `trend_proc` files to perform a particular function. For example, you can create a `trend_proc` file that collects the data from the various locations. This file could have `ee_collect` or `trendcopy` commands for each location. The name of the file could be `CollectData`. The next `trend_proc` file would process the collected data; so, this file would have `trend_sum` commands and any other commands necessary to process the data.

The name of this file could be `ProcessData`. The third `trend_proc` file would create the reports from the processed data; so, this file would have all the necessary commands to generate the reports. The name of this file could be `ReportData`. The last `trend_proc` file would distribute the reports; so, this file would have all the necessary commands to distribute the reports to the various locations. The name of this file could be `DistributeData`.

You could create a separate entry in `trendtimer.sched` for each `trend_proc` file, provided you spaced them out over time to ensure that each `trend_proc` finished before the system launched the next one.

Rather than this, you create a fifth `trend_proc` file, such as `ProjectA`, to contain the other four. Its blocks would be as follows:

Block A: `CollectData` (wait)

Block B: `ProcessData` (wait)

Block C: `ReportData` (wait)

Block D: `DistributeData` (wait)

You then put an entry into the `trendtimer.sched` file to run `ProjectA` for each hour of the day. You can add the `-n` option to the entry in the `trendtimer.sched` file so that the system can begin processing the same command at the next hour since the first internal `trend_proc` command should be finished. Note that the `trend_proc` commands inside `ProjectA` do not use the `-n` option so that the system does not start processing the command if the previous one has not finished; in this case, the system skips the command if the previous one is still processing.

## Creating a `trend_proc` File

Creating your own `trend_proc` files is a four-step process. You must:

- 1 Define the commands you want to run.
- 2 Determine the order to run the commands.
- 3 Create the `trend_proc` file.

You need to make sure you define your blocks in the correct order using the `wait` and `nowait` options; otherwise, the processing may be different than you expect.

- 4 Schedule `trend_proc` to launch.



## Scheduling a trend\_proc File

After you have created your `trend_proc` file, you can run it from the command line or place an entry in the `trendtimer.sched` file to run it at the appropriate time. The syntax for this entry follows:

```
trend_proc -f filename
```

Where, *filename* is the name of your `trend_proc` file.

If you do not use a fully qualified file name, the system looks for the file in the current working directory.

If you have a large or complicated `trend_proc` file that `trendtimer` runs on a regular basis, you may want to use the `-n` option to prevent the `trend_lock` locking check.

## Examples

The following examples show some possible applications of `trend_proc`.

### Example 1

PI uses the `trend_sum` application to maximize the efficiency of data aggregation. If you want to copy that data to the central server from a satellite server, you can use the `trendcopy` command to copy data after the `trend_sum` command finishes its processing. Such a `trend_proc` block might look like the one below:

```
begin: block1 wait
trend_sum -e SDifentry -f trendsum.sum
trendcopy -s src_db -S dest_db -t SDifentry
end: block1
```

In this example, the `trend_sum` command accesses the table, `SDifentry`. It uses the instructions in the input file `trendsum.sum`, which is in the same directory where this `trend_proc` file is running. After the `trend_sum` processing is complete, the system invokes the `trendcopy` command to copy the same table from the satellite server, which is `src_db`, to the central server, which is `dest_db`.

## Example 2

In this example, the first block, block0, is a wait block; so the system completes processing the `trend_proc` command in this block before it starts processing the `trend_sum` command in the next block, block1. Since block1 is a `nowait` block, the system also starts processing the `trend_sum` command in the next block, block2.

```
begin: block0 wait
{DPIPE_HOME}/bin/trend_proc -f {DPIPE_HOME}/scripts/
abc_hr.pro
end: block0
begin: block1 nowait
{DPIPE_HOME}/bin/trend_sum -f {DPIPE_HOME}/scripts/
day_abc_loc.sum
{DPIPE_HOME}/bin/trend_sum -f {DPIPE_HOME}/scripts/
month_abc_loc.sum
{DPIPE_HOME}/bin/trend_sum -f {DPIPE_HOME}/scripts/
abc_loc_fore.sum
{DPIPE_HOME}/bin/trend_sum -f {DPIPE_HOME}/scripts/
abc_loc_DOW.sum
end: block1
begin: block2 nowait
{DPIPE_HOME}/bin/trend_sum -f {DPIPE_HOME}/scripts/
day_a2z_loc.sum
{DPIPE_HOME}/bin/trend_sum -f {DPIPE_HOME}/scripts/
month_a2z_loc.sum
end: block2
```

## Example 3

In this example, the first block, block1, is a wait block; so the system completes processing the `trend_sum` commands in this block before it starts processing the `trendcopy` commands in the next block, block2. Note that the input file for the `trend_sum` command is in a different directory than the command; both the command and the input file have fully qualified path names since they may not be in the current working directory. After the `trend_sum` processing is complete, the system invokes the `trendcopy` command to copy the same tables from the satellite server, which is `SAT_DB`, to the central server, which is `CENTRAL_DB`.

```
begin: block1 wait
{DPIPE_HOME}/bin/trend_sum -f {DPIPE_HOME}/scripts/
Device1.sum
```

```

{DPIPE_HOME}/bin/trend_sum -f {DPIPE_HOME}/scripts/
Device2.sum
{DPIPE_HOME}/bin/trend_sum -f {DPIPE_HOME}/scripts/
Device3.sum
end:block1

begin: block2 wait
{DPIPE_HOME}/bin/trendcopy -t Device1 -s SAT_DB -S CENTRAL_DB
{DPIPE_HOME}/bin/trendcopy -t Device2 -s SAT_DB -S CENTRAL_DB
{DPIPE_HOME}/bin/trendcopy -t Device3 -s SAT_DB -S CENTRAL_DB
end: block2

```

#### Example 4

In this example, the block is a wait block. This block has a `trend_proc` command in it that must complete its processing before the system continues processing this block with the Perl program that follows. After the Perl program finishes, the system processes the `trend_sum` commands that follow in the order that they appear in the block. When the processing for the last `trend_sum` command is complete, the system processes the Perl program again. When the Perl program finishes, the processing for this block is complete.

```

begin: test1 wait

{DPIPE_HOME}/bin/trend_proc -f {DPIPE_HOME}/scripts/procl.pro

{DPIPE_HOME}/bin/perl {DPIPE_HOME}/scripts/Perfl.pl
-p "Location_Summary" -t day -v start

{DPIPE_HOME}/bin/trend_sum -t SH_Loc -e SD_Loc
-f {DPIPE_HOME}/scripts/SD_Loc.sum
{DPIPE_HOME}/bin/trend_sum -t SD_Loc -e fore -y 42
-f {DPIPE_HOME}/scripts/SD_Loc_forecast.sum
{DPIPE_HOME}/bin/trend_sum -t SDCustLoc -e dow -y 42
-f {DPIPE_HOME}/scripts/SD_Loc_forecast_dow.sum
{DPIPE_HOME}/bin/trend_sum -t SD_Loc -e SM_Loc
-f {DPIPE_HOME}/scripts/SM_Loc.sum

{DPIPE_HOME}/bin/perl {DPIPE_HOME}/scripts/Perfl.pl
-p "Location_Summary" -t day -v stop

end: test1

```



## 37 trend\_sum

You can use `trend_sum` to conduct a statistical analysis of source data on a PI Performance Insight (PI) system. It calculates various statistics such as averages, percentiles, and forecast values. It then uses these calculations to populate summary tables to monitor the behavior of the data for a particular window of time or to project the future behavior of that data.

### Requirements or Restrictions

- The `trend_sum` command requires at least one option.
- The source table must be a rate table or a summary table.
- The column specification must be an *alias* if the column expression contains more than one source table column name. See the **column:** keyword on [page 498](#) for usage of the *alias* parameter.
- The maximum number of nested functions in an expression is ten.
- The required input file must contain at least one by-variable and column specification; see [Input Files](#) on page 508.
- The command-line options may appear only once on the command line unless otherwise specified.

# Syntax

PI provides two methods for supplying information to `trend_sum`. One method is by using the command-line options. The other method is by using an input file, which is the only method for entering by-variables and column specifications. See [Input File Keywords](#) on page 495 for the syntax. Note that some options are only available by using the keywords in the input file while some other options are only available from the command line. See [Command-Line Options](#) on page 494. For more information about processing considerations when using both the command-line options and the input file together, see [Input Files](#) on page 508.

## Command-Line Options

`trend_sum` uses the following command-line syntax:

```
trend_sum    [ -C num_rows ]  
              [ -d debug_level ]  
              [ -e dest_name ]  
              [ -f input_file ]  
              [ -F ]  
              [ -h ]  
              [ -H hysteresis_units ]  
              [ -L log_hwm_number ]  
              [ -P lag_time_minutes ]  
              [ -R ]  
              [ -S server_name ]  
              [ -t src_table_name ]  
              [ -U ]  
              [ -V ]  
              [ -W db_hwm_number ]
```

```
[ -x stored_proc_name]
[ -y number_of_days ]
[ -z ]
```

## Input File Keywords

The following table shows the syntax of the keywords for the `trend_sum` input file with their equivalent command line options. The descriptions for the keywords are the same as for the corresponding command line options. See [Options](#) on page 503 for the descriptions and [Input Files](#) on page 508 for more information about input files.

**Table 23** `trend_sum` Input File Keywords

Keyword	Command Line Option
<b>baseline_days:</b> <i>number_of_days</i>	<b>-y</b>
<b>by_variable:</b> <i>by_var_expr</i>	
<b>column:</b> <i>column_spec</i>	
<b>destination_table:</b> <i>dest_name</i>	<b>-e</b>
<b>first_day_of_week:</b> <i>day_of_week</i>	
<b>name:</b> <i>sum_def_name</i>	
<b>source_table:</b> <i>src_table_name</i>	<b>-t</b>
<b>syntax_version:</b> <i>version_num</i>	
<b>time_hysteresis:</b> <i>hysteresis_units</i>	<b>-H</b>
<b>time_lag:</b> <i>lag_time_minutes</i>	<b>-P</b>

## Option Categories

The following table lists the options in categories that you might use.

**Table 24 trend\_sum Option Categories**

Category	Options
Data	-e, -f, -H, -P, -S, -t, -y
Switches	-F, -R, -U
Process	-C, -x, -z
Maintenance	-L, -W
Miscellaneous	-d, -h, -v

## Keywords

The `trend_sum` input file uses the following keywords:

**baseline\_days:** Specifies the number of days in the baseline period. See [Rolling Baseline Table](#) on page 511 for more information about this keyword. The default is set during the procedure building.



**by\_variable:**

The specification for the grouping by-variable which has the format:

[ *alias* = ] *column* or *bukeyword*

where: *column* is the name of a column in the source table.

*bukeyword* has one of the following values:

- **keyid** Group by the column named **dsi\_key\_id\_**
- **ta\_period** Group by *ta\_period*
- **hour** Group by hour
- **day** Group by day
- **week** Group by week
- **month** Group by month
- **quarter** Group by quarter
- **year** Group by year
- **day\_of\_week** Group by day of week for baseline table
- **day\_of\_week\_by\_hour** Group by day of week, by hour for baseline table
- **hour\_of\_day** Group by hour of day for baseline table

The *trend\_sum* summary process requires the *by\_variable* keyword in the input file.

The input file may contain multiple by-variables. Each by-variable that appears in the input file must be unique. The time by-variable (if used) must appear last in the list.

For more information, see [Input Files](#) on page 508 and [By-Variables](#) on page 509.

**column:**

Use this keyword to generate a column in the destination table for each statistic that appears in the *column\_spec*.

The name of each column in the destination table has the following form: *prefix* + *alias\_name* or *prefix* + *column\_name*.

The *prefix* has three uppercase characters and appears as a column in [Table 26](#) on page 515 with the corresponding statistic. See [Example 1](#) on page 518 for an example of derived column names. See [Input Files](#) on page 508 for more information about this keyword.

The *column\_spec* has one of the formats shown below.

### Single Statistic Format

[*alias* =] *column\_expr*:*statistic* ]

or

[!][*alias* =] *column\_expr*:*statistic* ]

where: *alias* is another name for the column.

*column\_expr* is either a column name or an arithmetic expression. The arithmetic expression requires the *alias* parameter to provide a name for this column. See [Divide-By-Zero Error](#) on page 514 to avoid a divide-by-zero error in an expression.

where, *statistic* is at least one of the statistics that appear in [Table 26](#) on page 515.

! (exclamation point) suppresses the *prefix*, derived from the statistics in [Table 26](#) on page 515, from the derived column name.

### Multiple Statistics Format

```
[alias =]column_expr:statistic1[ ,statistic2  
,statistic3, ... ,statisticN ]
```

or

```
[!][alias =] column_expr:statistic1  
[!][alias =] column_expr:statistic2  
[!][alias =] column_expr:statistic3  
:  
[!][alias =] column_expr:statisticN
```

When you use the exclamation point (!) to suppress the addition of the statistic prefix to the column name, you can only specify one statistic for that **column** argument. If you want to generate additional statistics from the same input when you suppress the prefix, you must specify the same expression in multiple **column** lines, as shown above, with an alias.

The `trend_sum` summary process requires the **column** keyword in the input file.

The input file may contain multiple **column** specifications. Each **column** specification that appears in the input file must be unique.

**destination\_table:** This keyword has multiple formats; the format for a standard table is different than the format for a rolling baseline table.

#### Standard Table

The *dest\_name* is the SQL name of the destination table for the `trend_sum` output. This keyword is required for a non-baseline table.

#### Rolling Baseline Table

The *dest\_name* is an optional suffix for the destination table with the `trend_sum` baseline output.

For a baseline table, the full name of the destination table contains the following parts in the specified order:

**SD<sub>nn</sub>** where *nn* is the number of days used for the value of the **-y** option or the **baseline\_days:** input file parameter.

*source\_table* name from the **-t** option or the **source\_table:** input file parameter.

*dest\_name* which is this suffix.

See [Rolling Baseline Table](#) on page 511 for more information about this keyword.

**first\_day\_of\_week:** Use this keyword to specify the first day of the week. Valid values are: **Mon, Tue, Wed, Thu, Fri, Sat,** or **Sun**. Note that these values are not case-sensitive; however, this keyword is in lowercase and is case-sensitive.

The default is **Mon**.

**name:** Specifies the unique name of the summary definition. The length of this name can be up to 128 characters.

**source\_table:** Use this keyword to specify the name of the source table that is providing the input to `trend_sum`. The `trend_sum` summary process requires either the **-t** option on the command line or the **source\_table:** keyword in the input file.

**syntax\_version:** Use this keyword to specify the version of this input file.

**time\_hysteresis:**

Specifies the amount of time to go back to reprocess data to insure that all samples were included. The value is the number of *hysteresis\_units* to reprocess. The destination table determines the *hysteresis\_units* to apply such as hours for an hourly table or days for a daily table.

For example, if the destination table is a daily table, this value is the number of days, and PI recalculates the data for the specified number of days.

If the destination table is a table with a long duration such as a year, the value should be 0; otherwise, PI reprocesses the data for the specified duration, which is the number of years. If the value is 0 (zero), PI does not reprocess any data; it continues processing from the last processed sample. For more information, see [Determine the Last Processed Sample for Hysteresis](#) on page 509.

This keyword identifies the beginning of the time range to process data from the source table. PI subtracts the number of corresponding *hysteresis\_units* from the last processed sample in source table and selects the samples that occur starting from that time. See the **-P** option for identifying the end of the time range.

The default is **-1**, which uses the default values for each type of table in the database.

You cannot use the **time\_hysteresis:** keyword with the **-U** option.

**time\_lag:**

Specifies the minimum lag time, which is the number of minutes to delay processing to insure most of the data has arrived before processing. The value is the minimum number of minutes earlier than the value of the calculated timestamp that a sample must be time stamped to be processed.

For example, if the value is 20 minutes and the calculated timestamp for the last sample was 10 minutes ago, PI does not process the data.

However, if the calculated timestamp for the last sample was 30 minutes ago, PI processes the last sample.

This keyword identifies the ending of the time range to process data from the source table. PI subtracts the specified number of minutes from the calculated timestamp and selects the samples from the source table that occur before that time and that correspond to the destination table boundary. See the **time\_hysteresis:** keyword for identifying the beginning of the time range. For more information about lag time, see [Lag Time](#) on page 510.

The default is **-1**, which uses the default values for each type of table in the database.

You cannot use the **time\_lag:** keyword with the **-U** option.

# Options

The `trend_sum` command has the following options:

- c**       Sets the size of the batch to commit, which is the number of rows to commit per transaction.  
The default value is **0**, which means the system calculates this value dynamically per table.  
HP strongly recommends that you use the default value. If this value is too low, it may affect performance; if this value is too high, it may affect concurrency.  
This option is in UPPERCASE.
- d**       Specifies the type of debug output. The values are:
- **0**: Provides no debug output. This is the default value.
  - **1**: Provides general debug output.
  - **2**: Provides debug output for each record. This value only applies to the stored procedures.
  - **3**: Provides debug output
- Debug output is written to standard output.
- e**       See the keyword, **destination\_table**: on [page 499](#), for the description of this option.
- f**       Name of an input file containing the aggregation statements.
- F**       If `trend_sum` has to run `tpmaint`, `trend_sum` passes the **-F** option to `tpmaint` so that it continues processing when there are more rows to process from the source table than two times the default retention period for that table. See [tpmaint](#) on [page 387](#) for more information.  
This option is in UPPERCASE.
- h**       Displays the syntax for this utility.
- H**       See the keyword, **time\_hysteresis**: on [page 501](#), for the description of this option.  
You cannot use the **-H** option with the **-U** option.  
This option is in UPPERCASE.

- L** Specifies the high water mark for the log. The high water mark denotes the level for when the database-used log size reaches the specified percentage and data processing stops. Valid values are **1 - 100**.

The default value is **90** for 90%, which means the data processing stops when the log is 90% full.

This option is in UPPERCASE.
- P** See the keyword, **time\_lag:** on [page 502](#), for the description of this option.

You cannot use the **-P** option with the **-U** option.

This option is in UPPERCASE.
- R** Use this option to specify that `trend_sum` replace the summary definition in the dictionary if the new definition is different than the old one.

This option is in UPPERCASE.
- S** This option specifies the database server name. It overrides the value in the `system.xml` file for the current process.

This option is in UPPERCASE.
- t** See the keyword, **source\_table:** on [page 500](#), for the description of this option.
- U** Specifies that `trend_sum` ignore the time-period range restrictions and process all the data in the source table.

You cannot use the **-U** option with the **-H** or **-P** option.

This option is in UPPERCASE.
- V** Displays the version of this utility.

This option is in UPPERCASE.
- W** Specifies the *high\_water\_mark* for the segment where the table resides. The high water mark denotes the level for when the segment-used size of the database reaches the specified percentage and data processing stops. Valid values for the *high\_water\_mark* are **1 - 100**.

The default value is **90** for 90%, which means the data processing stops when the database is 90% full.

This option is in UPPERCASE.



- x**      Use this option to remove the named stored procedure.
- y**      See the keyword, **baseline\_days**: on [page 496](#), for the description of this option.
- z**      Use this option to remove and then regenerate the related stored procedure.

## Naming Conventions

- A `trend_sum` input file generally has an extension of `.sum`, such as `ifexceptions.sum`; note that this is not a requirement.
- The source table name for `trend_sum` input should start with an UPPERCASE **R** or **S**.
- The destination table SQL name for a `trend_sum` baseline or forecast output table should have the following characteristics:
  - a Start with the UPPERCASE letter **S**.
  - b The second UPPERCASE letter designates the roll-up interval such as **H** for hour, **D** for day, **W** for week, **M** for month, **Q** for quarter, or **Y** for year.
  - c The number of baseline days follows the rollup interval.
  - d An alphanumeric name follows the baseline days. This name is the *src\_table\_name* from the **-t** option followed by the *dest\_name* from the **-e** option.
- The destination table Alias name for a `trend_sum` baseline or forecast output table should have the following characteristics:
  - a Start with **hourofday\_**, **dayofweek\_**, or **dayofweekbyhour\_** for a baseline table or **sum\_** for a forecast table as a prefix.
  - b The table category from the input table follows the prefix.
  - c The SQL name of the output table is the last part.
- The destination table SQL name for a `trend_sum` standard grouping output table should have the following characteristics:
  - a Start with the UPPERCASE letter **S**.

- b The second UPPERCASE letter designates the rollup interval such as **H** for hour, **D** for day, **W** for week, **M** for month, **Q** for quarter, or **Y** for year.
  - c An alphanumeric name follows the rollup interval. This name is the *dest\_name* from the **-e** option.
- The destination table Alias name for a `trend_sum` standard grouping output table should have the following characteristics:
  - a Start with **sum\_** as a prefix.
  - b The table category from the input table follows the prefix.
  - c The SQL name of the output table is the last part.
- The name of the summary definition that `trend_sum` generates varies depending on the parameters, as follows.

- If the **name** keyword is not in the input file and there are no options on the command line, `trend_sum` generates the following name:

*FileName\_srcF\_destF*

Where, *FileName* is the name of the input file.

*srcF* is the *src\_table\_name* from the **source\_table** keyword in the corresponding input file.

*destF* is the *dest\_name* from the **destination\_table** keyword in the corresponding input file.

- If the **name** keyword is not in the input file and there are options on the command line, `trend_sum` generates the following name:

*FileName\_srcC\_destC*

where: *FileName* is the name of the input file.

*srcC* is the *src\_table\_name* from the **-t** option on the command line.

*destC* is the *dest\_name* from the **-e** option on the command line.

- If the **name** keyword is in the input file and there are no options on the command line, `trend_sum` generates the following name:

*NameF*

where: *NameF* is the *src\_def\_name* from the **name** keyword in the input file.

- If the **name** keyword is in the input file and there are options on the command line, `trend_sum` generates the following name:

*NameF\_srcC\_destC*

Where, *NameF* is the *src\_def\_name* from the **name** keyword in the input file.

*srcC* is the *src\_table\_name* from the **-t** option on the command line.

*destC* is the *dest\_name* from the **-e** option on the command line.

## Usage Notes

You must use an input file to provide most parameters to `trend_sum`. The combination of all the parameters from the input file, command-line options, and versions of the associated programs that `trend_sum` invokes make up a *summary definition* that `trend_sum` stores in the database with a unique name. Another name for the summary definition is *transformation*.

You can provide the name with the **name** keyword in the input file; otherwise, `trend_sum` generates a name depending on the parameters provided. See [Naming Conventions](#) on page 505 for more information.

The associated programs that `trend_sum` invokes are `datapipe_manager` and `tpmaint`. See [datapipe\\_manager](#) on page 85 and [tpmaint](#) on page 387 for more information.

After `trend_sum` registers the summary definition in the database, it generates a stored procedure, associates the procedure with the definition, and then executes the procedure.



Note that if you terminate `trend_sum` with **Ctrl + c**, you need to terminate any associated stored procedures that are currently running on the server separately.

There are two modes that `trend_sum` uses to process its input files. If the **checkmode** value is **false**, which is the default, `trend_sum` determines the name of the summary definition and then check the database for it. If `trend_sum` finds the definition in the database, it runs the associated procedure without any sum file verification; otherwise, `trend_sum` creates the procedure, store it, and run it. If the **checkmode** value is **true**, `trend_sum` determines the name of the summary definition and then check the database

for it. If `trend_sum` finds the definition in the database, it validates the existing definition by comparing it to the new definition. If the definitions are the same, `trend_sum` runs the procedure associated with the existing definition. If the definitions are not the same, and the `-R` option is on the command line, then `trend_sum` replaces the existing definition with the new definition, regenerate the procedure, store it, and then run it; otherwise, if the `-R` option is not on the command line, `trend_sum` displays an error message and exit. The **`trendsum_checkmode`** entry is in the `be_defaults.prp` file in the `DPIPE_HOME/data` directory.

To improve performance, `trend_sum` uses time-period tables. It verifies that the appropriate entries exist in the corresponding time-period table. It does this by comparing the begin and end dates of the source table to the begin and end dates for the corresponding time-period table. If the begin and end dates of the source table are within the range of the begin and end dates for the corresponding time-period table, then `trend_sum` continues processing. Otherwise, `trend_sum` invokes `tpmaint` to update the corresponding time-period table. Note that if there are gaps in the time-period table for the range of dates from the source table, then `trend_sum` invokes `tpmaint`.



Note that the `trend_sum` and `tpmaint` programs require that the system have the correct date and time for the system date.

## Input Files

An input file includes statements with keywords that correspond to specific options on the command line. [Table 23](#) on page 495 lists the keywords and their equivalent command-line options.

The file name for the input file must contain the full path in the specification of the `-f` option unless the file is present in the current directory or the `DPIPE_HOME/scripts` directory.

If you use a keyword in an input file and the corresponding command line option on the command line, `trend_sum` executes the command line option, and ignores those keywords in the input file.

## By-Variables

A by-variable indicates a time period or a table element such as `dsi_key_id`. At least one by-variable is required to specify the aggregation level. If both a table element and a time by-variable are given, the time by-variable must be the last one specified. If a time by-variable is not specified, `trend_sum` defaults to the `ta_period` time by-variable.

When the source and destination data tables share a property table, `keyid` (or its equivalent) must be present as the only non-time by-variable. Similarly, when the data tables have separate property tables, `trend_sum` does not allow `keyid` (or its equivalent) as the by-variable.

Each non-time by-variable must exist as a column in the source data or property table. Each non-time by-variable must also exist as a column in the destination property table. If the column name differs between the source and destination tables, use the **alias** feature. For example: **by\_variable: h\_key=frcircuitnumber**. In this example, the name of the column in the source table is `frcircuitnumber` and the name of the column in the destination property table is `h-key`.

If the only by-variable specified is a time by-variable then the value for the first non-time by-variable, which is `dsi_target_name`, defaults to the string “**All Devices**”; the second non-time by-variable, which is `table_key`, defaults to the string “**0**”.

All *bukeyword* values except **keyid** for the **by\_variable:** input file keyword are time by-variables. See [page 497](#) for the list of *bukeyword* values.



Note that there are only three by-variables that you can use to populate a baseline table. They are `day_of_week`, `day_of_week_by_hour`, and `hour_of_day`. See [Rolling Baseline Table](#) on page 511 for more information about baseline tables.

## Determine the Last Processed Sample for Hysteresis

To identify the beginning of the time period range for hysteresis, `trend_sum` determines what the last processed sample timestamp is and then subtracts the hysteresis value from that timestamp.

To determine the last processed sample timestamp, `trend_sum` uses the minimum timestamp from the following parameters:

- The last processed sample timestamp for the destination table.
- The last processed sample timestamp for a particular managed object for the destination table.
- The earliest not processed sample timestamp for a particular managed object that does not appear in the destination table.
- The earliest sample timestamp in the source table if the destination table is empty.
- The earliest sample timestamp in the source table if there are samples for more days in the source table than the number of retention days for the source table.

In the case of the last two parameters, `trend_sum` does not need to subtract the hysteresis value since it processes the entire source table.

## Lag Time

PI uses a minimum lag time to establish the end point for processing data from the source table. The lag time is the amount of time to hold off performing the summarizing process until there is little chance that missing data arrives. For example, if there are only 3 of the 4 samples for an hour in the source table and the interface is no longer available, the remaining sample never shows up. After the lag time hour has ended, PI summarizes the data that is in the table but did not occur during the lag time interval.

During the summarization process, `trend_sum` only includes data for rollup periods. For example, `trend_sum` only processes the data for the last hour available for an hourly table. Similarly, it only processes the data for the last day available for a daily table.

To establish the end point for processing the data, `trend_sum` subtracts the lag time value from the current timestamp on the server. It then uses the calculated timestamp to process the data in the source table with timestamps that are earlier than the calculated timestamp.

By default, the minimum lag time value varies depending on the time type of table. The `-P` option (UPPERCASE) is available to provide a different minimum lag time value.

To calculate the end point for processing the data, do the following steps:

- 1 Use the current timestamp and subtract the minimum lag time value.
- 2 Round the difference in step 1 to the rollup interval such as hourly or daily.
- 3 If the value in step 2 is equal to the element's last `ta_period`, nothing happens.
- 4 If the value in step 2 is greater than the element's last `ta_period`, the processing includes the row.

For example, if `trend_sum` is processing a daily summary from a rate table with a 60-minute lag time, the following results can occur. If `trend_sum` runs at 07/03/01 00:30:00, the difference for step 1 is 07/02/01 23:30:00. The rollup interval is a day, so `trend_sum` rounds the value in step 1 to 07/01/01 00:00:00, which is the last complete day. Since the value for step 2 is equal to the last `ta_period`, which is 07/01/01 00:00:00, nothing happens.

However, if `trend_sum` runs at 07/03/01 01:30:00 instead, the difference for step 1 is 07/03/01 00:30:00. The system rounds the value in step 1 to 07/02/01 00:00:00, which is the most recent day. Since the value for step 2 is greater than the last `ta_period`, which is 07/01/01 00:00:00, then `trend_sum` includes the row in the summarization process.

## Reprocessing Data

After `trend_sum` determines the time range and identifies the samples from the source table, it then determines if it really needs to reprocess those samples. It only reprocesses those samples when there are more cumulative samples in the source table for a particular grouping than the number of cumulative samples recorded in the corresponding destination row for the same grouping.

## Rolling Baseline Table

Use the command line option **-y** or the input file keyword **baseline\_days:** to specify the number of days in the rolling baseline period. (When you specify the **-y** option, the **-e** option is not required. Likewise, if you use the input file parameter **baseline\_days:**, the input file parameter **destination\_table:** is not required.)

After `trend_sum` establishes a rolling baseline for a particular destination table, you cannot change the name of the source table used for the rolling baseline or the number of days in the baseline period.

Use of the **-y** option or the input file parameter **baseline\_days:** places restrictions on the destination table name. The destination table name always begins with SD42 (assuming a 42-day baseline). Normally, the remainder of the name is the source table name. For example, when the source table name is SDifexceptions the destination table name becomes SD42SDifexceptions.

You may use the **-e** option or the input file parameter **destination\_table:** to append the specified value to the name. For instance, the following option string yields SD42SDifexceptionsfore as the destination table name:

```
-y42 -t SDifexceptions -e fore
```

An input file that performs the same function would be as that shown below:

```
source_table: SDifexceptions
destination_table: fore
column: volume=(ifinoctets+ifoutoctets):tot
baseline_days: 42
by_variable: keyid
by_variable: hour_of_day
```

A rolling baseline is inherently a daily summary. Therefore, no time by-variable greater than **day** is accepted. However, you may specify a lower-level summarization level such as **hour\_of\_day** to get 24 summary rows of data per day. When summarizing into a baseline table, if you omit the time by-variable, **day** is the default.



When using the optional day-of-week or day-of-week-by-hour summarization level, it is strongly recommended that the number of days for the baseline period be evenly divisible by seven; otherwise, some summary rows summarize a different number of days than others. For example, one summary row might be for seven Mondays while another row might be for six Tuesdays.



## NULL Handling

The database rules for NULL handling apply to `trend_sum`.

In the event that NULL values exist in a column of data, `trend_sum` calculates most functions by ignoring the NULL values. For most functions, if a NULL value does appear as a result, it means that all the values in the column were NULL. For example, to calculate the average of two values, 1 and NULL, the result is 1 not 0.5.

There is one function that does not ignore NULL values in the column; it is `last`. This function may produce a NULL value as a result depending on the location of the NULL value. For example, if there are 20 values and the last value is NULL, then the result of the function `last` is NULL. Resultant NULL values may indicate that the data is incorrect.

## Troubleshooting

If `trend_sum` fails, always check the `trend.log` file for errors relating to `trend_sum`, `tpmaint`, or `datapipe_manager`. If `tpmaint` or `datapipe_manager` fails then `trend_sum` fails.

### Date Range Error

There are predefined limits for the number of rows to keep in time-period tables. Exceeding those numbers results in performance problems. When `tpmaint` populates a time-period table and it tries to increase the number of rows to more than twice the recommended amount, it generates the following messages in `trend.log`.

```
Adding the requested date range to the total of numberdays to
table table_name will exceed the total default of retention
days by more than 2 times.
```

```
Reduce the date range specified or use -F to override.
```

If these error messages appear in `trend.log` while running `trend_sum` or `tpmaint`, you need to rerun `trend_sum` with the `-F` option to override the default and continue processing. Be aware that this causes performance penalties.

## Divide-By-Zero Error

It is possible to create a column definition that has an expression with a zero in the denominator. This causes the database engine to generate a runtime error. You can avoid this error by using one of the following expressions:

- **NULLIF(*D*, 0)** where *D* is the denominator of an expression.

This function produces a NULL value for the expression when the denominator is zero. The denominator, *D*, can be another expression. Remember that the maximum number of nested functions is ten.

- ***D* + *V*** where *D* is the denominator of an expression and *V* is a very small value.

The value, *V*, can be any small value with respect to the data so that the denominator is not zero.

## Statistical Formulas and Variables

Table 25 defines the formula variables. Table 26 on page 515 shows the formulas for computing statistics. It also shows the corresponding prefix for the formula.



Note that in all formulas where it appears, *val* is the resolved value of the expression to which the function is applied. The *val* is considered to be *counter* data if more than one column appears in the expression or if a single-column expression is of counter data type. The *val* is considered to be *gauge* data only if the single-column expression is of gauge data type.

the following table lists the definitions for the variables that appear in Table 26 on page 515.

**Table 25 trend\_sum Formula Variables**

Variable	Formula
<b>a</b>	<b>ybar - b * xbar</b>
<b>b</b>	<b>(cnt * <math>\Sigma</math> (val * <math>\Delta t</math>) - tot * <math>\Sigma \Delta t</math>) / d</b>
<b>cnt</b>	Number of samples being summarized.
<b>d</b>	<b>cnt * <math>\Sigma</math> (<math>\Delta t</math> * <math>\Delta t</math>) - <math>\Sigma \Delta t</math> * <math>\Sigma \Delta t</math></b>

**Table 25 trend\_sum Formula Variables (cont'd)**

Variable	Formula
$\Delta t$	ta_period column value of this row minus the ta_period column value of the first row in the group.
$\Delta t_g$	ta_period column value of the last row in the group minus the ta_period column value of the first row in the group.
$\bar{x}$	$\Sigma \Delta t / \text{cnt}$
$\bar{y}$	$\text{tot} / \text{cnt}$

The following table shows the formulas for computing statistics in trend\_sum. Note that a column statement may contain multiple statistics, and each statistic may appear only once on a column statement. If a column statement has a statistic repeated, then an error occurs.

**Table 26 trend\_sum Statistic Formulas and Prefixes**

Statistic	Prefix	Description and Formula
<b>avg</b>	AVG	Average $\text{tot} / \text{cnt}$
<b>cnt</b>	CNT	Count of samples in the group
<b>dt</b> [value]	DTT	Number of days until <i>value</i> is reached where <i>value</i> is the threshold_value the user specifies. $\text{ceil}(((\text{threshold\_value} - a) / b - \Delta t_g) / 86400)$ trend_sum limits dt to $\pm 1000$ .
<b>f30</b>	F30	Projected value in 30 days $a + b * (\text{Dt} + 86400 * 30)$
<b>f60</b>	F60	Projected value in 60 days $a + b * (\Delta t_g + 86400 * 60)$
<b>f90</b>	F90	Projected value in 90 days $a + b * (\Delta t_g + 86400 * 90)$
<b>lst</b>	LST	Last value in the group

**Table 26 trend\_sum Statistic Formulas and Prefixes**

Statistic	Prefix	Description and Formula
<b>mad</b>	MAD	Delta time when the maximum value occurred
<b>mat</b>	MAT	Time when the maximum value occurred in the source table
<b>max</b>	MAX	Maximum value
<b>med</b>	MED	Median $vr[\text{floor}(.50 * (\text{cnt}))]$ where <b>vr</b> is an array of <b>vals</b> sorted into ascending order. The definition of this statistic has changed. It no longer subtracts <b>1</b> from <b>cnt</b> . The formula was <b>vr[floor(.50 * (cnt-1))]</b> .
<b>mid</b>	MID	Delta time when the minimum value occurred
<b>min</b>	MIN	Minimum value
<b>mit</b>	MIT	Time when the minimum value occurred
<b>per90</b>	P90	90th percentile $vr[\text{floor}(.90 * (\text{cnt}))]$ where <b>vr</b> is an array of <b>vals</b> sorted into ascending order. The definition of this statistic has changed. It no longer subtracts <b>1</b> from <b>cnt</b> . The formula was <b>vr[floor(.90 * (cnt-1))]</b> .
<b>per95</b>	P95	95th percentile $vr[\text{floor}(.95 * (\text{cnt}))]$ where <b>vr</b> is an array of <b>vals</b> sorted into ascending order. The definition of this statistic has changed. It no longer subtracts <b>1</b> from <b>cnt</b> . The formula was <b>vr[floor(.95 * (cnt-1))]</b> .

**Table 26 trend\_sum Statistic Formulas and Prefixes**

Statistic	Prefix	Description and Formula
<b>per98</b>	P98	98th percentile $vr[\text{floor}(.98 * (\text{cnt}))]$ where <b>vr</b> is an array of <b>vals</b> sorted into ascending order. The definition of this statistic has changed. It no longer subtracts <b>1</b> from <b>cnt</b> . The formula was <b>vr[floor(.98 * (cnt-1))]</b> .
<b>rct</b> [ <i>min - max</i> ]	RCT	Count of samples greater than or equal to the <i>min</i> value and less than the <i>max</i> value; that is, count of samples $\geq \text{min}$ and $< \text{max}$ .
<b>std</b>	STD	Standard deviation (the square root of the variance) $\sqrt{((\text{cnt} * \sum (\text{val} * \text{val}) - \text{tot} * \text{tot}) / (\text{cnt} * (\text{cnt}-1)))}$
<b>tct</b> [ <i>value</i> ]	TCT	Count of samples greater than or equal to <i>value</i> ; that is, count of samples $\geq \text{value}$ .
<b>tot</b>	TOT	Total $\sum \text{val}$ where <b>val</b> is the resolved value of the expression to which the <b>tot</b> function is being applied.
<b>vct</b> [ <i>value</i> ]	VCT	Count of samples equaling <i>value</i>
<b>wav</b>	WAV	Weighted Average $\sum (\text{val} * \Delta t) / (\sum \Delta t)$

# Examples

## Example 1

Run the following command to invoke `trend_sum` to read data from a rate table named `Rninterface_base_` and create a rolling baseline destination table named `SD42Rninterface_base_if` based on a 42-day rolling baseline period:

```
$DPIPE_HOME/bin/trend_sum -f $DPIPE_HOME/scripts/  
if_baseline_hourofday_volume_interface.sum
```

The following file contains the column and by-variable specifications to be included in the `SD42Rninterface_base_if` destination table:

```
$DPIPE_HOME/scripts/if_baseline_hourofday_volume_  
interface.sum
```

The contents of this input file are:

```
# trend_sum procedure to update hour_of_day baseline table  
for  
# combined network volume  
source_table: Rninterface_base_  
destination_table: if  
column: thru_put=(ifin octets+ifout octets):per95,std,wav  
column: volume=(ifin octets+ifout octets):tot,avg  
baseline_days: 42  
by_variable: keyid  
by_variable: hour_of_day
```

The `SD42Rninterface_base_if` destination table contains 24 rows, one row for each hour of the day, for each `keyid` present in the source table for each baseline period of 42 days.

The destination table created by the above process contains the following data columns:

- `P95thru_put`
- `STDthru_put`
- `WAVthru_put`
- `TOTvolume`
- `AVGvolume`

## Example 2

Run the following command to invoke `trend_sum` to read a daily summary table named `SDifexceptions` and create a rolling baseline destination table named `SD42SDifexceptions` with a 42-day rolling baseline period:

```
$DPIPE_HOME/bin/trend_sum -f $DPIPE_HOME/scripts/  
if_forecast.sum
```

The input file `$DPIPE_HOME/scripts/if_forecast.sum` contains the column and by-variable specifications to be included in the `SD42SDifexceptions` destination table:

```
# trend_sum procedure to create days to threshold  
calculations  
# from the daily summary table.  
source_table: SDifexceptions  
column: utilcurrent=P95utilization:per95  
column: inutilcurrent=P95waninutil:per95  
column: oututilcurrent=P95wanoututil:per95  
column: errorcurrent=AVGerror_pct:avg  
column: discardcurrent=AVGdiscard_pct:avg  
column: utiltrend=P95utilization:dtc[40],f30,f60,f90  
column: wanutiltrend=P95wanutil:dtc[40],f30,f60,f90  
column: waninutiltrend=P95waninutil:dtc[40],f30,f60,f90  
column: wanoututiltrend=P95wanoututil:dtc[40],f30,f60,f90  
column: discardtrend=P95discard_pct:dtc[5],f30,f60,f90  
column: errorstrend=P95error_pct:dtc[10],f30,f60,f90  
baseline_days: 42  
by_variable: keyid
```

The `SD42SDifexceptions` destination table contains one row for each keyid present in the source table for each baseline period of 42 days.





## 38 trendtimer

The `trendtimer` program is the PI scheduler that invokes other PI processes at scheduled times. It runs as a service and you can change the scheduled processes by adding, modifying, or deleting entries in the `trendtimer.sched` file; see [The Schedule File](#) on page 523 for more information.

### Requirements and Restrictions

- The `trendtimer.sched` file has a limit of 100 command entries that `trendtimer` schedules.
- The interval and offset for each command in the `trendtimer.sched` file should be on a 5-minute boundary.
- Avoid using an interval of **24:00** without an offset for a command entry.
- The default location, and the only location on a Windows system, for the `trendtimer.sched` file is in the `$DPIPE_HOME/lib` directory.
- Note that you do not need to stop and restart the `ovpi_timer` UNIX process or the `OVPI Timer` Windows service for any changes to take effect.

## Syntax

The typical usage for `trendtimer` does not require using any options on the command line.

The `trendtimer` command uses the following syntax on a UNIX system:

```
trendtimer  [ -h ]  
              [ -s schedule_file ]  
              [ -v ]
```

## Options

The `trendtimer` command has the following options:

- h**      This option is available on UNIX systems only.  
          Use this option to display the help information for the `trendtimer` command. This option is not valid with any other options.
- s**      This option is available on UNIX systems only.  
          Use this option to specify the path of the schedule file.  
          The default is `$DPIPE_HOME/lib/trendtimer.sched`.
- v**      Use this option to display the version information for `trendtimer`.  
          This option is not valid with any other options.  
          This option is in UPPERCASE.

# Usage Notes

The `trendtimer` program invokes other PI applications that run at specific times or regular intervals. It runs on a 5-minute cycle. Every hour on the hour, and at 5-minute intervals in between, it examines the contents of its schedule file and determines which processes to run at the current time. It runs these processes one at a time, in the order in which they appear in the schedule file.

The contents of the default schedule file vary to enhance performance and manage resources. The default schedule file on a central server contains processes for various applications, such as the collection, log backup, maintenance, and package processes. The default schedule file on a remote poller contains only the collection and log backup processes.

## The Schedule File

When `trendtimer` runs, it determines what it is going to do by reading a schedule file. The name of the schedule file is `trendtimer.sched` and it is in the `$DPIPE_HOME/lib` directory. On a UNIX system, you can override this file with the `-s` command line option. There is no override option for this file on a Windows system.

## Schedule File Syntax

The schedule file has the following elements.

- It contains a maximum of 100 command entries in the following format:

*interval* - - *command\_line*

Where, *interval* is how often the command on this line are run.

*command\_line* is the entire command that `trendtimer` runs.

- The hyphens ( - ) in the syntax are placeholders for future arguments. They must be included.
- The time specification uses a 24-hour clock, and `trendtimer` runs the commands in the schedule file according to local time.
- Comment lines in the file begin with a #.

The *interval* has multiple formats. Typically, it specifies the hours or minutes when the command runs. However, it can contain an *offset* that specifies a time between hours or a specific time that the command runs. All intervals are counted in *interval* minutes or hours and minutes from unix 0 time, which is **midnight, Thursday, January 1, 1970**. The interval has the following formats.

- *MM* specifies the number of minutes the command starts during the day. For example, **20** means that the command starts every 20 minutes during the day.
- *HH:00* specifies the number of hours the command starts during the day. For example, **2:00** means that the command starts every 2 hours during the day. Do not use **24:00** since it produces unexpected results.
- *HH:00+MM* specifies the number of hours and minutes the command starts during the day. For example, **2:00+20** means that the command starts every 2 hours and 20 minutes during the day.
- **24:00+HH:00** specifies a specific hour of each day that the command runs. For example, **24:00+2:00** means that the command starts every day at 2:00 a.m.
- **24:00+HH:00+MM** specifies a specific hour and minutes of each day that the command runs. For example, **24:00+2:00+20** means that the command starts every day at 2:20 a.m.
- **wkday+HH:00** specifies a specific hour of the specified weekday that the command runs. Valid values for *wkday* are the first two letters of the day of the week in English, as follows: **SU** is Sunday, **MO** is Monday, **TU** is Tuesday, **WE** is Wednesday, **TH** is Thursday, **FR** is Friday, and **SA** is Saturday. For example, **MO+2:00** means that the command starts every Monday at 2:00 a.m. If you want to run a program at midnight for a specific day, note that midnight starts the day; so, if you want to run a program every Saturday night at midnight; that is, the midnight between Friday and Saturday, the entry is **SA+24:00**.
- **MONTHx+HH:00** specifies a specific day in the month and the hour of that day that the command runs. For example, **MONTH5+2:00** means that the command starts on the fifth day of every month at 2:00 a.m.

Intervals should divide evenly into 60 minutes to ensure that the process runs at a predictable time. For example, an interval of 5 minutes means the program invoked runs on the hour, and at 5, 10, 15, 20, and so on, minutes after every hour because these are 5-minute intervals since midnight, Thursday, January 1, 1970. A program to be run every 2 hours has an interval

of **2:00** and runs at midnight, 2 a.m., 4 a.m., and so on. You can specify interval values in minutes or in hours; for example, you can use **120** minutes or **2:00** hours since they have the same value. The weekday and month intervals can also have offsets in minutes in the interval.

Note that an offset of **0:00** is invalid; use **24:00** instead. To run exactly at midnight, use **24:00+24:00**, since **24:00+0:00** is invalid.

The *command\_line* is the entire command that trendtimer runs. This is the rest of the line in the file. It does not need to be inside quotation marks. Environment variables can be used in the command line by placing them inside the braces, { }, and omitting the leading dollar sign (\$) in the UNIX variable name or the leading and trailing percent sign (%) in the Windows variable name.

Note that trendtimer schedules only the first 100 command entries in the trendtimer.sched file.



Be careful that you do not overload your system. When you install a report pack, the installation process typically adds commands to the trendtimer.sched file. Note also that a single command can start multiple processes.

## Schedule File Example

A sample trendtimer.sched file is:

```
# trendtimer.sched
24:00+5:00 - - {DPIPE_HOME}/bin/trendexec -i 1440
MO+5:00 - - {DPIPE_HOME}/bin/trendexec -i 10080
MONTH2+5:00 - - {DPIPE_HOME}/bin/trendexec -i 44640

5 - - {DPIPE_HOME}/bin/mw_collect -i 5 -n -K 1
10 - - {DPIPE_HOME}/bin/mw_collect -i 10 -n -K 1
15 - - {DPIPE_HOME}/bin/mw_collect -i 15 -n -K 1
20 - - {DPIPE_HOME}/bin/mw_collect -i 20 -n -K 1
60 - - {DPIPE_HOME}/bin/mw_collect -i 60 -n -K 1

24:00+1:00 - - {DPIPE_HOME}/bin/tpmaint
24:00+1:00 - - {DPIPE_HOME}/bin/transform_maint -r
24:00+2:00 - - {DPIPE_HOME}/bin/trend_discover -t
24:00+24:00 - - {DPIPE_HOME}/bin/db_delete_data
# eof
```

In this example:

- The system starts `trendexec` every day at 5 a.m., every week on Monday morning at 5 a.m., and on the second day of every month at 5 a.m.
- The system starts `mw_collect` every 5, 10, 15, 20, and 60 minutes for data collection.
- The system starts `tpmaint` and `transform_maint` every day at 1:00 a.m.
- The system starts `trend_discover` every day at 2:00 a.m.
- The system starts `db_delete_data` every day exactly at midnight.

Note that `trendtimer` runs the commands in the schedule file according to local time.

## Starting trendtimer

You can run `trendtimer` at any time, but you should invoke it at system startup. The PI installation adds a command to the system startup file to run the `ovpi_timer` script with the **start** action; this command invokes `trendtimer`. The **su** command in the startup file quotes the entire `trendtimer` command. This ensures that the **su** command runs the entire `trendtimer` command line rather than interpreting the `trendtimer` options as options to **su**.

## UNIX Platforms

If you need to start `trendtimer` manually, use the method described below. This method uses scripts for each platform that allow you to supply the action to perform, which is **start** in this case, as follows:

- 1 Open a new shell on the system where the application is installed.
- 2 Make sure that you are root.
- 3 Run the following command:

— HP-UX:

```
/sbin/init.d/ovpi_timer start
```

— Solaris:

```
/etc/init.d/ovpi_timer start
```

## Windows Platform

If you need to start `trendtimer` manually, use the procedure listed below to start it.

- 1 From the Windows Desktop, access the Services window via the Control Panel. This procedure varies depending on the type of Windows environment.
- 2 Locate the PI Timer entry and select it.
- 3 Click the **Start** button.

The Status field for PI Timer shows `Started`.

## Stopping `trendtimer`

After you stop `trendtimer`, remember that when you want to restart `trendtimer`, you need to restart it manually or reboot your system to restart it automatically.

## UNIX Platforms

Use the following procedure to stop `trendtimer`.

- 1 Shut down the PI application.
- 2 Open a new shell on the system where the application is installed.
- 3 Make sure that you are **root**.
- 4 Stop the server by entering:

— HP-UX:

```
/sbin/init.d/ovpi_timer stop
```

— Solaris:

```
/etc/init.d/ovpi_timer stop
```

## Windows Platform

Use the procedure listed below to stop `trendtimer`.

- 1 From the Windows Desktop, access the Services window via the Control Panel. This procedure varies depending on the type of Windows environment.
- 2 Locate the PI Timer entry and select it.
- 3 Click the **Stop** button.

The Status field for PI Timer appears blank.

- 4 Close the Services and Control Panel windows.

If you want to completely stop all PI processes, do the following steps.

- 1 Exit from the PI application if it is running.
- 2 Select the Processes tab from Task Manager to verify that no processes are running.

## Example

To start `trendtimer` using the schedule file `/tmp/mytimer.sched` on a UNIX system, user `trendadm` executes the command:

```
trendtimer -s /tmp/mytimer.sched
```

For an example of a `trendtimer.sched` file, see the [Schedule File Example](#) on page 525.



## 39 userctl

You can use the `userctl` command to add, modify, or delete a single user account that accesses the Web Access Server on a PI system. You can use this command instead of the User Administration page on the Web Access Server; see the *Performance Insight Administration Guide* for more information about using the GUI tools. If you want to add, modify, or delete a large number of users at the same time, you can use the `userimport` command that requires a file in XML format; see [userimport](#) on page 543 for more information.

### Requirements and Restrictions

- If there are spaces in the value for any option, enclose the entire value in double quotation marks.
- If you specify a value for an option, it must be a minimum of one (1) character and a maximum of 255 characters, unless otherwise specified; in most cases, it may not contain any of the following characters: ‘, “, &, comma (,), or space.
- Each time you invoke this utility, you must enter the required options: **-host**, **-mode**, **-port**, **-pwd**, **-tuser**, and **-user**.

# Syntax

The `userctl` command uses the following syntax:

```
userctl      [ -database hostname ]  
              [ -debug ]  
              [ -dept name ]  
              [ -email email_address ]  
              [ -groups group1 [, group2, ..., groupN] ]  
              [ -help ]  
              -host hostname  
              [ -interactive ]  
              -mode action  
              [ -name firstname_lastname ]  
              -port number  
              [ -protocol protocol ]  
              -pwd adm_pwd  
              [ -role type ]  
              [ -tele phone_num ]  
              [ -tpwd password ]  
              -tuser username  
              -user adm_user  
              [ -verbose ]  
              [ -version ]  
              [ -view viewname ]
```

# Options

The `userctl` command has the following options:

- database**      Use this option to specify the database the user uses for reporting. The specified database must exist in the Web Access Server.  
The default is the reporting database defined in the `systems.xml` file, which is the default database specified on the Performance Insight Systems page in the Web Access Server.
- debug**          Use this option to run the command in debug mode. To use this option, set the value to **true**.  
By default, debug mode is not set.
- dept**            Use this option to specify the company or department name for the user. If the name contains spaces, enclose it in double quotation marks.
- email**           Use this option to specify the email address of the user.
- groups**          Use this option to specify the Web Access Server groups of which the user is a member. Separate each group with a comma (,); if there are any spaces (such as between groups after the comma), enclose the entire value in double quotation marks. The specified groups must exist.  
Note that when you modify a user account, you must include this option if you want to keep the user as a member of the specified groups.
- help**            Use this option to display the syntax for the `userctl` command.
- host**            Use this option to specify the Web Access Server hostname where the transaction occurs.  
This is a required option.
- interactive**    Use this option to display the login box. When you use this option, the system displays the login box if the entry for one or both of the **-user** or **-pwd** options is incorrect; otherwise, if you do not use this option, you get an error message instead.

<b>-mode</b>	<p>Use this option to specify the type of transaction to perform.</p> <p>Valid entries are:</p> <ul style="list-style-type: none"> <li>• <b>add</b>: Use this mode to add a new user.</li> <li>• <b>delete</b>: Use this mode to delete an existing user.</li> <li>• <b>modify</b>: Use this mode to modify an existing user by changing its properties.</li> </ul> <p>This is a required option.</p>
<b>-name</b>	<p>Use this option to specify the actual full name for the user. If you use spaces in the name, enclose the entire name in double quotation marks.</p>
<b>-port</b>	<p>Use this option to specify the Web Access Server port number where the transaction occurs.</p> <p>You must enter this option even though the default for this option is the port number supplied during the PI installation, which is port number <b>80</b>, in most cases.</p> <p>This is a required option.</p>
<b>-protocol</b>	<p>Use this option to specify the communication protocol.</p> <p>Valid values are <b>http</b> or <b>https</b>.</p>
<b>-pwd</b>	<p>Use this option to specify the corresponding password for the username that has authorization to make the specified changes.</p> <p>This is a required option.</p>
<b>-role</b>	<p>Use this option to specify the role of the user as an administrative user. To use this option, set the value to <b>admin</b>.</p> <p>The default value is <b>user</b>.</p>
<b>-tele</b>	<p>Use this option to specify the telephone number of the user.</p> <p>The format of the number is <i>aaa-ppp-nnnn</i>:</p> <p>Where, <i>aaa</i> is the area code.</p> <p><i>ppp</i> is the prefix.</p> <p><i>nnnn</i> is the remainder of the telephone number.</p>

<b>-tpwd</b>	Use this option to specify the corresponding password for the username to add, modify, or delete. This password parameter must be a minimum of four (4) characters and a maximum of ten (10) characters. This is a required option with the <b>-mode add</b> option.
<b>-tuser</b>	Use this option to specify the username to add, modify, or delete. This is a required option.
<b>-user</b>	Use this option to specify the username that has authorization to make the specified changes. This username must have administrative privileges. This is a required option.
<b>-verbose</b>	Use this option to turn on verbose messaging.
<b>-version</b>	Use this option to display the current version of <code>userctl</code> .
<b>-view</b>	Use this option to assign a single catalog view to the user. The specified view must exist.

## Usage Notes

The purpose of this command is to manage a single user account. All PI clients such as Report Builder, Report Viewer, Web Access Server, and the Management Console require users to log on.

By default, PI creates one user account when you install it. This is the `trendadm` account; ideally, you should create a user account for each person who uses the PI client applications.

## Modes of Operation

The `userctl` command has three modes of operation: add, modify, and delete.

## Add

The add mode provides the ability to add a user account to access the Web Access Server.

## Modify

The modify mode provides the ability to change the properties of an existing user account that is used for accessing the Web Access Server.

## Delete

The delete mode provides the ability to remove an existing user account that enables access to the Web Access Server.

## Using the userctl Command

This section shows some formats of the command for the various modes. There is a minimum of six required options for the `userctl` command. Each mode shows the command with the required options along with the other options for the particular task; however, only the definitions for the new options appear for each subsequent command. The definitions for the required options appear below.

- All `userctl` commands must have all the following options for each task:

```
userctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -tuser username -mode action
```

Where, *host\_name* is the name of the host for the Web Access Server.

*port\_num* is the port number for the Web Access Server.

*adm\_user* is the administrative user name that has authorization to make the specified changes.

*adm\_pwd* is the corresponding password for the administrative user that has authorization to make the specified changes.

*username* is the name for the user account to add, modify, or delete.

*action* is the type of action to perform, such as add, modify, or delete.

- If you enter the `userctl` command without any options, the system displays the help information. Use the following format.

```
userctl
```

- If you want to display the version for the `userctl` command, enter the following command.

```
userctl -version
```

- If you want the login box to pop up if either of the required options, `-user` or `-pwd`, is incorrect, enter the following command.

```
userctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -tuser username -mode action -interactive
```

## Add

The following formats show various options for adding a user account to access the Web Access Server. Note that you can combine the additional options in any manner that meets your needs.

- To add a new user account without any user information, enter the following command:

```
userctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -tuser username -mode add -tpwd password
```

Where, *password* is the password for the new user account.

- To add a new user account as an administrative user without any user information, enter the following command:

```
userctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -tuser username -mode add -tpwd password  
-role admin
```

Where, *password* is the password for the new user account.

- To add a new user account with user information, enter the following command:

```
userctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -tuser username -mode add -tpwd password  
-name "name_of_user" -tele phone_num -email email_address  
-dept "name"
```

Where, *password* is the password for the new user account.

*name\_of\_user* is the actual name of the user. Use quotation marks to enter the first and last names with a space.

*phone\_num* is the telephone number for the user.

*email\_address* is the email address for the user.

*name* is the company or department name for the user.

- To add a new user account without any user information and include it in two groups, enter the following command:

```
userctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -tuser username -mode add -tpwd password  
-groups group1,group2
```

Where, *password* is the password for the new user account.

*group1* is the name for the first group where the user is a member.

*group2* is the name for the second group where the user is a member.

- To add a new user account without any user information and specify another database for reporting, enter the following command:

```
userctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -tuser username -mode add -tpwd password  
-database db_name
```

Where, *password* is the password for the new user account.

*db\_name* is the name for the database that exists in the Web Access Server.

## Modify

- To add user information to an existing user account that is not a member of any group, enter the following command:

```
userctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -tuser username -mode modify  
-name "name_of_user" -tele phone_num -email email_address  
-dept "name"
```

Where, *name\_of\_user* is the actual name of the user. Use quotation marks to enter the first and last names with a space.

*phone\_num* is the telephone number for the user.



*email\_address* is the email address for the user.

*name* is the company or department name for the user.

- To change the group membership for the user to one group and assign a view to the user, enter the following command:

```
userctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -tuser username -mode modify -groups group1  
-view viewname
```

where: *password* is the password for the new user account.

*group1* is the name of the group where the user is a member.

*viewname* is the name of the view assigned to the user.

- To change the password for a user with group membership in two groups and add administrative privileges with a department name, enter the following command:

```
userctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -tuser username -mode modify -tpwd password  
-groups group1, group2 -dept "name" -role admin
```

Where, *password* is the password for the new user account.

*group1* is the name for the first group where the user is a member.

*group2* is the name for the second group where the user is a member.

*name* is the company or department name for the user.

## Delete

To remove a user account, enter the following command:

```
userctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -tuser username -mode delete
```

# Examples

The following examples illustrate some uses of the `userctl` command that an Administrator, such as the `trendadm` user, can enter.

## Add

### Example 1: Add a User Account without Additional Information

To add a user account with the name `user_1` and a password `pass_1` on the `powder2` host without additional information about the user, you can use the following command.

```
userctl -host powder2 -port 80 -user trendadm -pwd  
trendadm -tuser user_1 -mode add -tpwd pass_1
```

### Example 2: Add a User Account with Administrative Privileges

To add a user account with the name `user_2` and a password `pass_2` on the `powder2` host with administrative privileges, you can use the following command.

```
userctl -host powder2 -port 80 -user trendadm -pwd  
trendadm -tuser user_2 -mode add -tpwd pass_2 -role admin
```

### Example 3: Add a User Account with User Information

To add a user account with the name `user_3` and a password `pass_3` on the `powder2` host with information about the user such as the user's name, telephone number, email address, and department name, you can use the following command.

```
userctl -host powder2 -port 80 -user trendadm -pwd  
trendadm -tuser user_3 -mode add -tpwd pass_3 -name "User  
Three" -tele 310-555-1234 -email user3@xxx.com -dept  
Operations
```

### Example 4: Add a User Account as a Member of a Group

To add a user account with the name `user_4` and a password `pass_4` on the `powder2` host as a member of an existing group with the name `group1`, you can use the following command.

```
userctl -host powder2 -port 80 -user trendadm -pwd  
trendadm -tuser user_4 -mode add -tpwd pass_4 -groups  
group1
```

#### Example 5: Add a User Account as a Member of Multiple Groups

To add a user account with the name `user_5` and a password `pass_5` on the `powder2` host as a member of two existing groups, `group1` and `group2`, you can use the following command.

```
userctl -host powder2 -port 80 -user trendadm -pwd  
trendadm -tuser user_5 -mode add -tpwd pass_5 -groups  
group3,group4
```

#### Example 6: Add a User Account with a View

To add a user account with the name `user_6` and a password `pass_6` on the `powder2` host and associate it with an existing catalog view such as `view1`, you can use the following command.

```
userctl -host powder2 -port 80 -user trendadm -pwd  
trendadm -tuser user_6 -mode add -tpwd pass_6 -view  
view_1
```

#### Example 7: Add a User Account to Access Another Database

To add a user account with the name `test_user_1` and a password `test_pass_1` on the `powder2` host and specify another database for reporting such as `test_system`, you can use the following command.

```
userctl -host powder2 -port 80 -user trendadm  
-pwd trendadm -tuser test_user1 -mode add  
-tpwd test_pass1 -database test_system
```

Note that the database must exist in the Web Access Server.

## Modify

#### Example 1: Change the Password of a User Account with a Group

To change the password of a user account with the name `user_4` and a current password `pass_4` to a password `pass_four` on the `powder2` host and adding administrative privileges and a department name of `Quality` while keeping the member in a group with the name `group1`, you can use the following command.

```
userctl -host powder2 -port 80 -user trendadm -pwd  
trendadm -tuser user_4 -mode modify -tpwd pass_four  
-dept Quality -groups group1 -role admin
```

#### Example 2: Add a User Account with User Information

To modify an existing user account with the name user\_1 on the powder2 host that does not have any group memberships by adding the name of the user, you can use the following command.

```
userctl -host powder2 -port 80 -user trendadm -pwd  
trendadm -tuser user_1 -mode modify -name "User One"
```

#### Example 3: Add a User Account with User Information

To modify an existing user account with the name user\_5 on the powder2 host that has membership in two groups to one group, you can use the following command.

```
userctl -host powder2 -port 80 -user trendadm -pwd  
trendadm -tuser user_5 -mode modify -groups group5
```

## Delete

To delete a user account with the name test\_user\_1 on the powder2 host, you can use the following command.

```
userctl -host powder2 -port 80 -user trendadm -pwd  
trendadm -tuser test_user1 -mode delete
```

# Error Messages

This section describes some of the messages that can occur from `userctl`. Each message has the following format:

- A brief description about why the message appears. Each new message description starts with a check box.
  - The actual message that appears with parameters for any information that may be missing.
  - A suggestion about the action to do so that the message does not appear again.
- ❑ If the following message appears, there is an incorrect value for the **-mode** option on the command line.

Error - Invalid application mode

Verify that the **-mode** option has the value **add**, **modify**, or **delete** on the command line.

- ❑ If the following message appears, there is an incorrect option on the command line.

Invalid Argument Supplied

Verify that all of the options are correct on the command line. See [Syntax](#) on page 530 for the valid options.

- ❑ If the following message appears, the user is unauthorized to create, modify, or delete a user account.

Unauthorized.

Verify the following:

- The user (**-user**) and password (**-pwd**) values are correct.
- The specified user has administrator privileges.



---

# 40 userimport

You can use the `userimport` command to add, modify, or delete Web Access Server users on a PI system. This utility is an extension of the Web Access Server User Accounts feature under the Administration link on the Management Console.

## Requirements or Restrictions

- There are five required command-line options that you must enter each time you invoke the utility.
- Any file to be imported must be in the Extensible Markup Language (XML) interchange format specified in this chapter.
- A group must already exist if you want to assign a user to it.

# Syntax

The `userimport` command uses the following syntax:

```
userimport  -c communications_protocol
              -f user_XML_file_name
              -h application_server_name
              [ -help ]
              -p application_server_port_number
              -P administrator_password
              -U administrator_username
              [ -v ]
```

# Options

The `userimport` commands have the following options:

- |              |   |
|--------------|---|
| <b>-c</b>    | The communications protocol that you want to use (http or https). If this option is not specified, http is used by default. |
| <b>-f</b>    | Name of the text file containing Web Access Server user information.  |
| <b>-h</b>    | Name of the Web Access Server.  |
| <b>-help</b> | Display the command-line options for the <code>userimport</code> command.   |
| <b>-p</b>    | Port number of the Web Access Server.   |
| <b>-P</b>    | Password of the PI administrator.<br>This option is in UPPERCASE.   |
| <b>-U</b>    | Username of the PI administrator.<br>This option is in UPPERCASE.   |
| <b>-v</b>    | Display version information for the <code>userimport</code> command.  |



# Usage Notes

When you want to assign a user to a group, the group must already exist. This implies that you must use the `groupctl` or `groupimport` commands from the command line or the Add New Group feature from the Group Administration page on the Web Access Server to create the group before you use this command to add a user to that group.

This rest of this section describes the naming convention and file format for the text file that contains the XML tag sets when you use the `userimport` command.

## Naming Conventions

The `userimport` text file parameters must follow the naming conventions listed below:

- 1 The tag set parameter formats are as follows:
  - a The parameters may contain alphabetic characters (upper and lower case), numeric symbols, and special characters except those cited in Step 2.
  - b All parameters except *password* must be a minimum of one (1) character and a maximum of 255 characters.
  - c The *password* parameter must be a minimum of four (4) characters and a maximum of ten (10) characters.
  - d The *telephone* parameter can be in any format but cannot exceed 255 characters.
  - e The *e-mail* parameter can be in any format but cannot exceed 255 characters.
- 2 DO NOT use any of the following characters in the `userimport` text file:
  - a single quotation mark ( ' )
  - b double quotation mark ( " )
  - c ampersand ( & )
  - d comma ( , )
  - e space (blank)

## File Format

The `userimport` text file uses XML tag sets to define the data to be imported into the Web Access Server. The following figure shows all XML tag sets available for use in the `userimport` text files. [Table 27](#) on page 547 describes all XML tag sets available for use in the `userimport` text files.

**Figure 2** `userimport` Text File Format

```
<?xml version="1.0" encoding="UTF-8"?>
<Users>
  <User>
    <Action>action</Action>
    <ID>user_logon_name</ID>
    <Password>user_password</Password>
    <RealName>user_name</RealName>
    <Department>user_department</Department>
    <Phone>user_telephone</Phone>
    <Email>user_e-mail</Email>
    <Groups>
      <Group>user_group_name</Group>
    </Groups>
    <Role>admin</Role>
  </User>
</Users>
```



Note that all the XML tag sets include the angle brackets (< >) as part of the tag.

**Table 27 userimport File - XML Tag Definitions**

XML Tag Sets	Description
<?xml version="1.0" encoding="UTF-8"?>	Enables the parser to validate the XML format. It must appear as the first line in the file.
<Users></Users>	Opening tag for <Users></Users> tag set. Contains all <User></User> tag set information for all users defined in this file. Any number of users can be placed in this set using <User></User> tag sets inside the <Users></Users> tag set. This tag set is required.
<User></User>	Opening tag for the <User></User> tag set. Contains all tag sets; for example, <Action></Action>, <ID></ID>, and so on, that define a single user. This tag set is required.
<Action></Action>	Procedure tag set that defines the operation to perform. If a user exists and the <b>Add</b> action is specified, the request is ignored. Valid values are: <b>Add</b> , <b>Modify</b> , <b>Delete</b> . The default is <b>Add</b> .
<ID></ID>	Defines user's logon name. This tag set is required.
<Password></Password>	Defines user's password. This tag set is required only when the password is to be changed.
<RealName></RealName>	Defines actual name of the user.
<Department></Department>	Defines user's department or company name.
<Phone></Phone>	Defines user's telephone number.

**Table 27 userimport File - XML Tag Definitions**

XML Tag Sets	Description
<Email></Email>	Defines user's e-mail address.
<Groups></Groups>	<p>&lt;Groups&gt; is the opening tag for the &lt;Groups&gt;&lt;/Groups&gt; tag set.</p> <p>There can be multiple &lt;Group&gt;&lt;/Group&gt; tag sets inside the &lt;Groups&gt;&lt;/Groups&gt; tag set.</p> <p>All &lt;Group&gt;&lt;/Group&gt; tag sets are contained within the &lt;Groups&gt;&lt;/Groups&gt; tag set for the user being defined.</p> <p>This tag set is not required if the user does not belong to a group or groups.</p>
<Group></Group>	<p>&lt;Group&gt; is the opening tag for the &lt;Group&gt;&lt;/Group&gt; tag set. This tag set defines the group name that the user belongs to.</p> <p>There can be multiple &lt;Group&gt;&lt;/Group&gt; tag sets inside the &lt;Groups&gt;&lt;/Groups&gt; tag set.</p> <p>This tag set is not required if the user does not belong to a group or groups.</p>
<Role></Role>	<p>Defines the user as an administrator.</p> <p>Valid value = <b>admin</b></p> <p>This tag is required if the user requires administrative rights.</p>

## Example

The sample `userimport` file shown in [Figure 3](#) on page 550, when called from the command line by the command listed below, adds two users: The first user is added to the `NE_Reporting` group and has administrative rights to the Web Access Server. The second user is added to the Technical Services group.

```
userimport -h app_server -p app_server_port -U admin_username  
-P admin_password -f full_path_and_filename
```

**Figure 3 Sample userimport Text File**

```
<?xml version="1.0" encoding="UTF-8"?>
<Users>
  <User>
    <Action>Add</Action>
    <ID>lsmiley</ID>
    <Password>4822</Password>
    <RealName>Leonard Smiley</RealName>
    <Department>NE_Reporting</Department>
    <Phone>310-732-2144</Phone>
    <Email>lsmiley@wnmb.com</Email>
    <Groups>
      <Group>NE_Reporting</Group>
    </Groups>
    <Role>admin</Role>
  </User>
  <User>
    <Action>Add</Action>
    <ID>wbentley</ID>
    <Password>0494</Password>
    <RealName>William Bentley</RealName>
    <Department>Technical Services</Department>
    <Phone>310-732-9844</Phone>
    <Email>wbentley@wnmb.com</Email>
    <Groups>
      <Group>Technical Services</Group>
    </Groups>
  </User>
</Users>
```

# 41 vantage\_collect

You can use the `vantage_collect` command to collect SNMP data from two linked tables that have a control table and data table relationship, for example, on a PI system.

## Requirements and Restrictions

- In general, the collection by-variables (other than `target_name`) in a property table define the instances the poller polls for.
- The **MibIndexMap** statement in the TEEL definition file for the child table identifies the parent (source) table; this is the link between the two data tables for this command.
- To collect data with this command, the collector module for the data table must be **dsi\_hst**.

# Syntax

The `vantage_collect` command uses the following syntax:

```
vantage_collect [ -a ]  
                  [ -A ]  
                  [ -b ]  
                  [ -c max_processes ]  
                  [ -C wait_time ]  
                  [ -d debug_level ]  
                  [ -D thread_wait_time ]  
                  [ -e ]  
                  [ -E clock_error_value ]  
                  [ -F min_disk_pct ]  
                  [ -g ]  
                  [ -G debug_level_pm ]  
                  [ -H alternate_poller_name ]  
-i interval  
                  [ -I check_index ]  
                  [ -k ]  
                  [ -K suppress_spikes ]  
                  [ -L ]  
                  [ -M minimum_filter ]  
                  [ -n ]  
                  [ -N retry_interval ]  
                  [ -o timeout ]  
                  [ -p max_entries_per_pdu ]  
                  [ -P snmp_port ]
```



[ **-q** *log\_info\_level* ]  
[ **-Q** *name* ]  
[ **-r** *retries* ]  
[ **-R** *min\_rows* ]  
[ **-s** *round\_factor* ]  
[ **-S** *snmp\_version* ]  
[ **-u** ]  
[ **-v** ]  
[ **-w** *high\_water\_mark* ]  
[ **-W** *high\_water\_mark\_log* ]  
[ **-X** ]  
[ **-y** ]  
[ **-Y** *delta\_time* ]  
[ **-z** *child\_debug\_level* ]  
[ **-Z** *debug\_log\_bcpgateway* ]

# Options

The `vantage_collect` command has the following options:

- a** Directs the child collectors of `vantage_collect` to output the collected data in ASCII format. You may want to do this for debugging purposes.
- A** Enables archiving of raw data. The default is no archiving. When you use this option, it turns on archiving. The archive function stores the collected data in a raw data table. This option is in UPPERCASE. This option is equivalent to the **@bArchive=1** parameter in `trendpm`.
- b** Forces regeneration of definition files and worktable. The poller analyzes the structure of the table to be collected and creates definitions about how to load data into database for the process. This is a one-time process most of the time. However, when the poller realizes that the structures of the data table or related property tables have changed, it regenerates the `bcp_gateway` process definitions and worktable. The **-b** option forces the regeneration of these definition files, and cleans up any internal database objects used during data loading. See the discussion for the **-g** option lower in this table for more information. See the discussion for the **-k** option lower in this table for more information.
- c** Specifies the number of collection processes to run concurrently. When `vantage_collect` starts, it starts child processes that actually do the collections. The default is **5**. You can reduce SNMP collection cycles by increasing this number.
- C** Specifies the number of minutes that each child process can run. The system kills the child process if it runs longer than the specified number of minutes. The default is **30** minutes. This option is in UPPERCASE.

- d** Sets the debug output level for the parent instance of `vantage_collect`. Values of **0**, **1**, **2**, or **3** are valid. The higher the number, the more detailed the information.

The default is **0**, which means no debug output. Debug output is written to standard out. You should only use this option for testing in coordination with HP Technical Support due to the additional overhead it places on `vantage_collect`.

The **-z** and **-Z** options set the debug level for the child processes of `vantage_collect`. The **-d**, **-z**, and **-Z** options can be used together to control the debug level of parent and child processes independently.
- D** This option specifies how many seconds the parent thread should wait for signals from the collector and `bcp_gateway` threads.

It is normal to get `thread_wait` timeout messages in the log file when waiting for signals from the `bcp_gateway` or `trendpm` thread due to potential long running jobs when writing to the database. The program exits when a `thread_timeout` occurs, unless the `thread_timeout` occurred when all collector jobs are finished and only `bcp_gateway` or `trendpm` threads remained. In this case, the timeout only appears in the `trend.log` file.

The default is the same value as the **-C** timeout value, which is **1800** seconds. For example, if the **-C** value is **2** (2 minutes), then the **-D** value defaults to **120** (120 seconds).

This option is in UPPERCASE.
- e** Turns on `GETBULK` when using `SNMP-V2`.

When the poller uses `SNMP-V2`, you can opt to use the `GETBULK` request instead of `GETNEXT` when getting `SNMP` data.

- E      Sets the percentage level for valid data. The value is the percentage of difference between the delta values of two Received Timestamps and two System Uptimes. These statistics come from two consecutive raw data samples.  
          For example, if r1 and s1 are the Received Timestamp and System Uptime for the first sample, and r2 and s2 are the Received Timestamp and System Uptime for the second sample, then the calculation for the value is  $((r2 - r1) - (s2 - s1)) * 100 / (r2 - r1)$ . During processing, if the calculated value for the samples exceeds the value set by this option then the samples are rejected.  
          The default value is **10**.  
          This option is in UPPERCASE.  
          This option is equivalent to the **@zerror** parameter in `trendpm`.
- F      Specifies the minimum disk space (as percent of total disk space) that must exist on the disk where the `$COLLECT_HOME` directory resides. If less space exists on the directory, `vantage_collect` does not run.  
          The default is **5**, which represents 5 percent.  
          This option is in UPPERCASE.
- g      Use 3.5.x compatibility. That is, use the same method for inserting data into the table as version 3.5.x and earlier.  
          The **-g** option loads data into archive (raw) tables directly. The `bcp_gateway` definition file contains the setting for the **-g** option. This means that if you already invoked `vantage_collect` without the **-g** option and you need the **-g** option, then use the **-b** option to regenerate the definition file. Similarly, if you already invoked `vantage_collect` with the **-g** option and you do not need the **-g** option, then use the **-b** option to regenerate the definition file. Otherwise, `vantage_collect` uses the same setting for the **-g** option currently in effect when you originally generated the definition file.

- G** Sets the debug output level for the `trendpm` process of `vantage_collect`. Values of **0**, **1**, **2**, or **3** are valid. The higher the number, the more detailed the information.

The default is **0**, which means no debug output.

Debug output is written to standard out. You should only use this option for testing in coordination with HP Technical Support due to the additional overhead it places on `vantage_collect`.

This option is in UPPERCASE.

This option is equivalent to the `@debug_level` parameter in `trendpm`.
- H** Specifies an alternate poller name. When you run `vantage_collect` in distributed mode, with the `-n` option, the poller compares the local hostname to the Poll From field in the polling policy. When you use the `-H` option, `vantage_collect` compares the Poll From field in the polling policy to the alternate poller name.

This option is in UPPERCASE.

See [Distributed Polling](#) on page 301 for more information.
- i** Is the Collection ID. `vantage_collect` executes the entries in the polling policy that have this value in their Interval field. How frequently `vantage_collect` is actually run depends on the configuration of `trendtimer`, but the idea is to be consistent so that a collection request with a collection ID of **5** is run every 5 minutes. See [File Locks](#) on page 298 for additional information.

This option is required.
- I** Specifies whether to use existing indices on the upload table or to drop existing indices and then recreate them. The value **1** means that the existing indices on the upload table are used. The value **0** means that the existing indices are dropped and then recreated. If the value is **1** and the proper indices are missing then the raw-to-delta process fails.

The default is **0**.

This option is in UPPERCASE.

This option is equivalent to the `@bCheck_index` parameter in `trendpm`.

- k** Populates the property tables (but not the data tables) for the devices you are polling.

The `bcp_gateway` definition file contains the setting for the **-k** option. This means that if you already invoked `vantage_collect` without the **-k** option and you need the **-k** option, then use the **-b** option to regenerate the definition file. Similarly, if you already invoked `vantage_collect` with the **-k** option and you do not need the **-k** option, then use the **-b** option to regenerate the definition file. Otherwise, `vantage_collect` uses the same setting for the **-k** option currently in effect when you originally generated the definition file.
- K** Specifies whether to reject samples if there are spikes. A spike is defined when a counter is manually reset, the difference of two consecutive samples from a counter exceeds the spike threshold, and the second sample is less than the first sample. The value of the spike threshold is  $2^{31}$  for 32-bit counters or  $2^{51}$  for 64-bit counters. Remember if the difference of the samples is negative; account for the rollover of the counter by adding  $2^{32}$  for 32-bit counters or  $2^{52}$  for 64-bit counters.

Valid values for *suppress\_spikes* are:

  - 1** Rejects samples if a spike occurs.
  - 0** Does not reject samples. The default is **0**.

This option is in UPPERCASE.

This option is equivalent to the **@bSuppress\_spike** parameter in `trendpm`.
- L** Specifies that the collected data be stored locally instead of in the PI database. When you use this option, `vantage_collect` uses any previously saved collection definitions, and does not use the database.

This option is in UPPERCASE.
- M** Sets the *minimum\_filter* value. The procedure rejects the sample if the delta value of a counter falls below this value.

The default value is **-1**, which means to accept the entire sample.

This option is in UPPERCASE.

This option is equivalent to the **@line\_suppress\_value** parameter in `trendpm`.

- n** Enables distributed polling. If this option is used, `vantage_collect` executes the collection request only if the Poll From field in the polling policy record for this collection request matches the hostname of the machine on which `vantage_collect` is running. If you omit this option, `vantage_collect` executes all polling requests whose interval matches the value of the **-i** option, regardless of the hostname specified to do the polling in the polling instructions.  
  
You can set an alternate hostname to poll with the **-H** option. See [Distributed Polling](#) on page 301 for more information.
- N** Sets the *retry\_interval*, which is the number of seconds the procedure needs to wait to acquire a lock on an upload table.  
The default value is **10**, which is 10 seconds.  
This option is in UPPERCASE.  
This option is equivalent to the **@retry\_interval** parameter in `trendpm`.
- o** Number of seconds `vantage_collect` is to wait for a response after sending an SNMP request.  
The default is **1** second. (SNMP timeout.)
- p** The number of SNMP variables to include in the varbind list in the GETNEXT PDU (Protocol Data Unit) request. It is possible to generate a GETNEXT request that yields a response that is too long to transmit.  
The default value is **20**.
- P** This option enables the collection of SNMP data from the specified port rather than the default SNMP port of 161.  
This option is in UPPERCASE.

- q** Sets the log information level for the parent instance of `vantage_collect`.  
 Values of **0**, **1**, **2**, or **3** are valid. The higher the number, the more detailed the information.  
 The default is **0**, which means no log information for the parent instance of `vantage_collect`.  
 If the `$TREND_LOG` environment variable is set, then the log is written in the directory specified by the `$TREND_LOG` environment to a log file named `vantage_collect_dbg.log`; the path and file name is `$TREND_LOG/vantage_collect_dbg.log`.  
 If the `$TREND_LOG` environment variable is not set, then the log is written to the same log file name in the `$DPIPE_HOME/log` directory; the path and file name is `$DPIPE_HOME/log/vantage_collect_dbg.log`.  
 You can modify the log file name using the **-Q** option.
- Q** Specifies the *name* for the ROOT of the trace log file. You can use this option to change the name of the log file, as follows:  
`path/name_pid_ppid_polltime_dbg.log`  
 Where, *path* is `$TREND_LOG` if the environment variable is set; otherwise, it is `$DPIPE_HOME/log`.  
*name* is the name of the file that you supply.  
*pid* is the process id number for this process.  
*ppid* is the process id number of the parent process.  
*polltime* is the number of seconds since 01/01/1970 00:00.  
 When you use this option, set the **-q** option to a value greater than **0**.  
 This option is in UPPERCASE.
- r** Specifies the *retries*, which is the number of times `vantage_collect` resends an SNMP request before assuming the target node is not going to respond.  
 The default value is **5**. (SNMP retries)



- R**      Sets the minimum number of rows to collect before starting the loading of data into database. The number of rows is an approximation, since there is an assumption that each row is 500 bytes.  
             The default value is **1000** rows, which means that the loading of the file into the database starts when the size is 500,000 bytes (500 \* 1000).  
             This option is in UPPERCASE.
- s**      This option rounds off the collection time (ta\_period). If the vantage\_collect parent kicks off a collection at 3:07, and if you are using the default collection option of **300** seconds (5 minutes), the actual ta\_period value for the collection is recorded as 3:05.
- S**      Specifies the SNMP version. This option is in UPPERCASE.  
             Valid values are:
  - **1** for SNMP V1
  - **2** for SNMP V2C.
             This option overrides the values read from the database.
- u**      Displays the command line formats for vantage\_collect.
- V**      Displays the version stamp for vantage\_collect.  
             This option is in UPPERCASE.
- w**      Specifies the *high\_water\_mark*. The high water mark stops collection of data when the database-used size reaches the specified percentage.  
             Valid values are **1 - 100**.  
             The default parameter is **90** for 90%.
- W**      Specifies the *high\_water\_mark\_log*. The high water mark stops collection of data when the log-used size reaches the specified percentage. Valid values are **1 - 100**.  
             The default parameter is **90** for 90%.  
             This option is in UPPERCASE.
- x**      This option turns off trendpm capability.  
             This option is in UPPERCASE.
- y**      This option disables the calculation of total count for bcp\_gateway metrics.

- Y** Specifies which clock to use to calculate Delta Time.  
Valid values are:
- 1** Directs the procedure to use System Uptime to calculate Delta Time.
  - 0** Directs the procedure to use the `received_ts` column for the calculation.
- The default is **0**.  
This option is in UPPERCASE.  
This option is equivalent to the `@bDelta_time` parameter in `trendpm`.
- z** Sets the debug level for child collector processes.  
Values of **0**, **1**, **2**, or **3** are valid. The higher the number, the more detailed the information.  
The default is **0**, which means no debug output.  
Debug output is written to standard out. You should only use this option for testing in coordination with HP Technical Support due to the additional overhead it places on `vantage_collect`.
- Z** This option specifies whether to turn on debugging or logging or both for the `bcp_gateway` process. When the logging option is turned on, the information is written to the log file. The name of the log file varies depending on which environment variable is set.  
If the `$TREND_LOG` environment variable is set, then the log is written in the directory specified by the `$TREND_LOG` environment to a log file named `bcp_gateway_dbg.log`; the path and file name is `$TREND_LOG/bcp_gateway_dbg.log`.  
If the `$TREND_LOG` environment variable is not set, then the log is written to the same log file name in the `$DPIPE_HOME/log` directory; the path and file name is `$DPIPE_HOME/log/bcp_gateway_dbg.log`.  
Valid values for this option are:
- **1**: Turn on logging.
  - **2**: Turn on debugging.
  - **3**: Turn on both logging and debugging.
- This option is in UPPERCASE.

## Usage Notes

You can use the `vantage_collect` command to collect data from two data tables linked together with the `MibIndexMap` statement in the TEEL definition file for the child data table. Note that if there is no link to a second data table, `vantage_collect` collects data from the single data table. Sometimes, the reference to the second data table could be a parent table or a control table.

To collect data with this command, the setting for the `CollectorModule` statement in the TEEL definition file for the data table must be `dsi_hst`.

The `vantage_collect` command is a wrapper around the `mw_collect` command; this means that the `vantage_collect` command processes the options on its command line that are listed in this chapter first and then passes the results and any other `mw_collect` options that appear on its command line to `mw_collect` to complete the processing. See [mw\\_collect](#) on page 277 for more information.

Typically, the `vantage_collect` command generates two `mw_collect` commands to collect the data. It generates an `mw_collect` command for the parent or control data table to collect and process data for every device in the collection group; when it completes the processing of that command it generates the second `mw_collect` command to collect the data from the child data table. Note that if some devices in the collection group are very slow, you may want to create a separate collection group and polling policy for those devices, because the second `mw_collect` command does not start until the first one finishes its processing for each device in the group.

If the `MibIndex` statement in the TEEL definition file for the child data table has a time-related variable length, `vantage_collect` invokes a child process of `mw_collect` with the **method** parameter set to `MTD_HST`. Note that `mw_collect` runs the **method** parameter only if the `ByVarInfo` statement is set to **hst** in the TEEL definition file.

See the *Performance Insight TEEL Reference Guide* for more information about the TEEL statements.

## Example

In this example, the system locates the entries that have a collection ID of 15 for the hostname that matches the hostname where this command is running and then collect the data from the two tables identified in the policy. It rejects any sample that is a spike.

```
vantage_collect -n -i 15 -K 1
```

## 42 viewctl

You can use the `viewctl` command to add, delete, or modify a single catalog view on a PI system. A *catalog view* is a list of deployed reports in the system folder that a user can access. This view applies to any database that the user may access.

You can use this command instead of the Catalog Views page on the Web Access Server; see the *HP Performance Insight Administration Guide* for more information about using the GUI tools.

### Requirements and Restrictions

- You must be an administrative user to use this command.
- Each time you invoke this utility, you must enter the required options: **-host**, **-mode**, **-port**, **-pwd**, and **-user**, and **-view**.

# Syntax

The `viewctl` command uses the following syntax:

```
viewctl      [ -add rpt_name ]  
              [ -del rpt_name ]  
              [ -desc desc_text ]  
              [ -help ]  
              -host host_name  
              [ -interactive ]  
              -mode type  
              [ -name viewname ]  
              -port port_num  
              [ -protocol protocol ]  
              -pwd adm_pwd  
              -user adm_user  
              [ -verbose ]  
              [ -version ]  
              -view viewname
```

# Options

The `viewctl` command has the following options:

- |                     |  |
|---------------------|--|
| <b>-add</b>         | Use this option to specify the name of the report or folder to add to the view, relative to the system folder. You can add the reports or folders one at a time after you create the view.<br>Use this option with <b>modify</b> mode only.  |
| <b>-del</b>         | Use this option to specify the name of the report or folder to remove from the view, relative to the system folder.<br>Use this option with <b>modify</b> mode only.   |
| <b>-desc</b>        | Use this option to enter a description of the view. Enclose the text in double quotation marks if it contains any spaces.  |
| <b>-help</b>        | Use this option to display the syntax for the <code>viewctl</code> command.  |
| <b>-host</b>        | Use this option to specify the Web Access Server hostname where the transaction occurs.<br>This is a required option.  |
| <b>-interactive</b> | Use this option to display the login box. When you use this option, the system displays the login box if the entry for one or both of the <b>-user</b> or <b>-pwd</b> options is incorrect; otherwise, the system displays an error message.   |
| <b>-mode</b>        | This option specifies the type of transaction to perform that affects the entire view. For example, if you specify <b>delete</b> for this option, the system removes the view from the catalog.<br>Valid entries are: <ul style="list-style-type: none"><li>• <b>add</b>: Use this mode to add a new view to the catalog.</li><li>• <b>delete</b>: Use this mode to delete the view from the catalog.</li><li>• <b>modify</b>: Use this mode to modify an existing view by adding or deleting reports or folders.</li></ul> This is a required option. |

<b>-name</b>	<p>This option specifies the new name of the view. If the name contains spaces, enclose it in double quotation marks.</p> <p>Use this option with <b>modify</b> mode only.</p>
<b>-port</b>	<p>Use this option to specify the Web Access Server port number where the transaction occurs.</p> <p>You must enter this option even though the default for this option is the port number supplied during the PI installation, which is port number <b>80</b>, in most cases.</p> <p>This is a required option.</p>
<b>-protocol</b>	<p>Use this option to specify the communication protocol for the view.</p> <p>Valid values are <b>http</b> or <b>https</b>.</p>
<b>-pwd</b>	<p>Use this option to specify the corresponding password for the username that has authorization to make the specified changes.</p> <p>This is a required option.</p>
<b>-user</b>	<p>Use this option to specify the username that has authorization to make the specified changes. This username must have administrative privileges.</p> <p>This is a required option.</p>
<b>-verbose</b>	<p>Use this option to turn on verbose messaging.</p>
<b>-version</b>	<p>Use this option to display the current version of <code>viewctl</code>.</p>
<b>-view</b>	<p>Use this option to specify the name of the view to add, modify, or delete.</p> <p>This is a required option.</p>



# Usage Notes

The purpose of this command is to manage a single PI catalog view. The process is to create the view and then add reports or folders to it one at a time using the **-add** option.

To remove a report or folder from the view, use the **-del** option. The **-mode delete** option removes the entire view.

## Modes of Operation

The `viewctl` command has three modes of operation: add, modify, and delete.

### Add

The add mode enables you to add a view to the PI catalog on the Web Access Server.

### Modify

The modify mode enables you to change the contents of an existing PI catalog view on the Web Access Server.

### Delete

The delete mode enables you to remove an existing view from the PI catalog on the Web Access Server.

## Using the `viewctl` Command

This section shows some formats of the command for the various modes. The `viewctl` command requires the six options shown and defined in the first bullet below. The following discussions of the various modes (for example, **add**) show the command with these basic options, along with the options that are specific to each task.

- All `viewctl` commands must have all the following options for each task:

```
viewctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -view viewname -mode action
```

Where, *host\_name* is the name of the host for the Web Access Server.

*port\_num* is the port number for the Web Access Server.

*adm\_user* is the administrative user name that has authorization to make the specified changes.

*adm\_pwd* is the corresponding password for the administrative user that has authorization to make the specified changes.

*viewname* is the name for the PI catalog view to add, modify, or delete.

*action* is the type of action to perform, such as add, modify, or delete.

- If you enter the `viewctl` command without any options, the system displays the help information. Use the following format.

```
viewctl
```

- If you want to display the version for the `viewctl` command, enter the following command.

```
viewctl -version
```

- If you want the login box to pop up if either of the required options, `-user` or `-pwd`, is incorrect, enter the following command.

```
viewctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -view viewname -mode action -interactive
```

## Add

The following formats show various options for adding a view to the PI catalog on the Web Access Server. Note that you can combine the additional options in any manner that meets your requirements.

- To add a view to the PI catalog on the Web Access Server, enter the following command:

```
viewctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -view viewname -mode add
```

- To add a view to the PI catalog on the Web Access Server with a description, enter the following command:

```
viewctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -view viewname -mode add -desc "text"
```

Where, *text* is the description for the view.

## Modify

- To add a report or folder to an existing view in the PI catalog on the Web Access Server, enter the following command:

```
viewctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -view viewname -mode modify -add "rpt_name"
```

Where, *rpt\_name* is the name of the report or folder. Use quotation marks if there are any spaces in the name.

- To remove a report or folder from an existing view in the PI catalog on the Web Access Server, enter the following command:

```
viewctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -view viewname -mode modify -del "rpt_name"
```

Where, *rpt\_name* is the name of the report or folder. Use quotation marks if there are any spaces in the name.

## Delete

- To remove a view from the PI catalog on the Web Access Server, enter the following command:

```
viewctl -host host_name -port port_num -user adm_user  
-pwd adm_pwd -view viewname -mode delete
```

# Examples

The following examples illustrate some uses of the **viewctl** tool.

## Example 1: Add a View

To add a view with the name `view_1` to the PI catalog on the `powder2` host, you can use the following command.

```
viewctl -host powder2 -port 80 -user trendadm -pwd  
trendadm -view view_1 -mode add
```

## Example 2: Remove a Report from a View

To delete a report with the name `myrep.rep` from the view with the name `view_1` from the PI catalog on the `powder2` host, you can use the following command.

```
viewctl -host powder2 -port 80 -user trendadm -pwd  
trendadm -view view_1 -mode modify -del /system/dira/  
dirb/myrep.rep
```

## Example 3: Delete a View

To delete a view with the name `view_1` from the PI catalog on the `powder2` host, you can use the following command.

```
viewctl -host powder2 -port 80 -user trendadm -pwd  
trendadm -view view_1 -mode -mode delete
```

# Error Messages

This section describes some of the messages that can occur from `viewctl`. Each message has the following format:

- A brief description about why the message appears. Each new message description starts with a check box.
  - The actual message that appears with parameters for any information that may be missing.
  - A suggestion about the action to do so that the message does not appear again.
- If the following message appears, the user is unauthorized to create, modify, or delete a user account.

Unauthorized.

Verify the following:

- The user (**-user**) and password (**-pwd**) values are correct.
- The specified user has administrator privileges.

- If the following message appears, there is an incorrect option on the command line.

Invalid Argument Supplied

Verify that all of the options are correct on the command line. See [Syntax](#) on page 566 for the valid options.

- If the following message appears, there is an incorrect value for the **-mode** option on the command line.

Error - Invalid application mode

Verify that the **-mode** option has the value **add**, **modify**, or **delete** on the command line.



## 43 viewer

You can use the `viewer` command to start the Report Viewer client application from the command line on a PI system.

### Requirements and Restrictions

- When you connect to a different server with the **-server** option, use the Browse option to view the files on the specified server.
- When you use the **-mode remote** option, you must include the **-file** option on the command line at the same time.
- When you use the **-file** option without the **-mode** option on the command line, the system opens the specified file on the local system.
- When you change a parameter with the **-params** option, remember that it is a global option and it applies to every report you open that has the parameter specified.

# Syntax

The `viewer` command uses the following syntax:

```
viewer  [ -debug debug_value ]  
         [ -file path_reportname ]  
         [ -log logfile ]  
         [ -mode location ]  
         [ -p password ]  
         [ -params parameter1=value1 [, parameter2=value2, ... ] ]  
         [ -port number ]  
         [ -server servername ]  
         [ -u username ]
```



# Options

The `viewer` command has the following options:

- debug** Use this option to enable diagnostic messages, which are an extra level of detail included in the log file.  
Valid values are:
- **true** enables diagnostic messages
  - **false** does not enable diagnostic messages
- The default is **false**.
- file** Use this option to specify the name of the report you want to open automatically when you run `viewer`. You can use the absolute or relative path with the name of the report file. For a remote file, you need to give the remote location in reference to its deployed location.
- log** Use this option to specify the name of the log file to open. Include the path for the name. The name should have a slash as the first character in the name because the system adds the prefix `PI` to the specified name.  
The default log file is `viewer.log`.
- mode** Use this option to specify the location for the file you want to access. There are two values; they are:
- **local** when the file is on the local system.
  - **remote** when the file is on the Web Access Server.
- The default is **local**.  
Use the **-file** option to specify the name and location of the file. You must use the **-file** option with this option to open a remote file automatically when you run `viewer`.
- p** This option specifies the password for the login process.  
If you do not use this option with the **-u** option, the system prompts for the username and password.  
Use this option to specify the corresponding password for the username that has authorization to view the reports.

**-params** Use this option to specify the report parameters to change report defaults at run time. A parameter has the following format: *parameter=value*. This is a global option; it applies to every report you open that has the parameter specified.

When you specify more than one parameter, separate the parameters with a comma (,). When a parameter value contains a space, enclose all the parameters in one set of quotation marks. The following example shows multiple parameters with one parameter that has a space in the value.

**-params "INTERFACE=92,CUSTOMER=All Telco"**

When a parameter value contains a character that is special to the command interface (shell) such as a comma, precede the character with a backward slash (\), for example:

**-params "INTERFACE=92,CUSTOMER=Telco\,North"**

See the *HP Performance Insight Guide to Building and Viewing Reports* for details about how to view or modify the parameters associated with a report using Report Viewer, and for details about how to view and modify the parameters associated with a report using the Web Access Server.

**-port** Use this option to specify the port number of the Web Access Server that you want to access from the Report Viewer client application.

The default for this option is the port number supplied during the PI installation, which is port number **80**, in most cases.

**-server** Use this option to specify the host name of the Web Access Server that you want to access from the Report Viewer client application. If you want to access a system in a different domain, you need to specify the full domain name for the host name.

The default for this option is the server host name supplied during the PI installation.

**-u** This option specifies the username for the login process.

If you do not use this option with the **-p** option, the system prompts for the username and password.

## Usage Notes

You can use Report Viewer to open and view reports located on your system and on the Web Access Server, modify existing reports to change how the data is displayed, and save and print reports. See the *HP Performance Insight Guide to Building and Viewing Reports* for more information about using the Report Viewer client application.

If you want to browse the reports on a remote system when you run `viewer`, you can use the **-server** option alone. However, if you want to open a specific report file automatically when you run `viewer`, you can add the **-mode remote** option with the **-file** option.

## Examples

The following examples illustrate some uses of the **viewer** command.

### Example 1

You can use this example to log into the Report Viewer on a Windows operating system and display a report with the name, `Capacity_Planning.rep`, the `INTERFACE` parameter set to **92**, and the `CUSTOMER` parameter set to **Telco**.

```
viewer -file D:\software\trend\reports\deploy\system\  
Interface_Reporting\ Interface\Capacity_Planning.rep -u  
katter -p dusquesne -params INTERFACE=92,CUSTOMER=Telco
```

### Example 2

You can use this example to perform the same action as in the previous example, , except that the report is on the Web Access Server.

```
viewer -file /system/Interface_Reporting/Interface/  
Capacity_Planning.rep -u katter -p dusquesne -params  
INTERFACE=92,CUSTOMER=Telco -mode remote
```



# Index

## Numerics

- 32-bit counters, 285, 558
  - spike threshold, 477
- 3.5.x compatibility, 283, 342, 556
- 64-bit counters, 285, 558
  - spike threshold, 477

## Symbols

- \$DPIPE\_HOME
  - /lib directory, 156
  - /scripts, 431, 435, 438, 440
- .teel extension, 88

## A

- accounts
  - See* user accounts
- action
  - error messages, 47
  - syntax, 42
- action\_collect child collector, 45
- Action tag, 199, 547
- aging
  - criteria, 108
- Alias Name column
  - Table Manager, 107
- alias parameter, 90

- alter
  - datapipe, 96
  - parameter, datapipe\_manager, 91
- Analyze Table command, 268, 270
- angle brackets, XML tags, 198, 546
- archive server, 414
- archiving raw data, 280, 339, 476, 554
- ascii\_parm.dat, 451
- ASCII file
  - comma separated format
    - condensed, 462
    - uncondensed, 461
  - format for collection\_manager, 65
  - format for dip\_manager, 134
  - format for node\_manager, 312
  - formats, 448, 460
  - tab separated format
    - condensed, 462
    - uncondensed, 460
  - trend\_export, 447
- as-is mode, 95
- automatic scheduling, 44

## B

- backing up log files, 289
- back-up files *See* log\_backup
- backup option, 207, 222
  - example, 225
- baseline, 496

- bcg\_gateway
  - defined, 291
  - definition file, 283, 284, 342, 343, 556, 558
  - procedure, 292
  - turn on debugging/logging, 289, 348, 562
  - with local data storage, 300

- BCP batch file, 413

- bcg command, 325, 327, 329

- begin statement, 484

- bin directive, 234

- block
  - definition, 483
  - empty, 485
  - naming convention, 485
  - series of sequential commands, 483
  - types, 484
  - unique name, 485

- builder, 49
  - options, 51
  - syntax, 50

- by-variable, 403, 493
  - error messages, 424
  - grouping, 497
  - trend\_sum, 509

- ByVarInfo statement, 563

## C

- caching poller, 291

- calculate end point to process data, 511

- catalog view, 565

- category
  - group, 62, 64, 131, 132, 135, 213
    - defined, 215
  - type, 246

- central server, 414

- character data, entering, 476

- check syntax of statements in TEEL file, 92

- child collector
  - action\_collect, action, 45

- ChildGroups tag, 199

- child poller
  - defined, 290
  - responsibilities, 291

- child process, 42
  - bcg\_gateway, 291

- Child tag, 200

- collection
  - See also* data collection
  - from HP Performance Agent or HP Operations Agent, 337
  - policy, install with Package Manager, 245
  - processes, 150
    - concurrent, 280, 340, 554
  - request, 151, 156
  - retries, 43
  - routine, 152
  - stops, 478
  - store data locally, 285, 344, 558
  - time, rounding, 287, 346, 561
  - timeout, 43
  - without a database, 298

- collection\_manager
  - ASCII file with, 65
  - description, 57
  - error messages, 55, 74
  - examples, 71
  - modes of operation, 64
  - options, 59
  - requirements, 57
  - syntax, 58
  - using, 69

- collection\_name variable, 246

- collection\_table variable, 246

Collection ID, 151, 284, 304, 557  
    pa\_collect, 343  
Collector ID, 291  
CollectorModule statement, 563

commands  
    builder, 49  
    collection\_manager, 57  
    datapipe\_manager, 85  
    db\_delete\_data, 103  
    deploytool, 113  
    dip\_manager, 127  
    formdeploytool, 167  
    group\_manager, 205  
    groupctl, 183  
    groupimport, 195  
    grouping related commands, 481  
    indexmaint, 261  
    install.pkg, 231  
    log\_backup, 273  
    mw\_collect, 277  
    node\_manager, 307  
    ovpi\_bulk\_copy, 323  
    ovpi\_run\_sql, 331  
    pa\_collect, 337  
    pa\_discovery, 351  
    piadmin, 359  
    schedule, 365  
    snmpv2dis, 383  
    tpmaint, 387  
    transform\_maint, 393  
    trend\_discover, 429  
    trend\_label, 463  
    trend\_proc, 481  
    trend\_sum, 493  
    trendcopy, 403  
    trendexec, 445  
    trendpm, 473  
    trendtimer, 521  
    userctl, 529  
    userimport, 543  
    vantage\_collect, 551  
    viewctl, 565  
    viewer, 575

- comma separated file format
  - condensed, 462
  - uncondensed, 461
- comment character, install.pkg, 233
- community string
  - file, recommendation, 436
  - file containing, 431
  - multiple, 435
  - public, 436
  - RMON, 436
- compatibility mode, 95
- compatibility mode, setting, 88
- component type
  - list with file suffix, 252
  - PI SQL file, 254
  - SQL script, 252
  - TREND collector definition, 246
  - TREND database procedure, 235
  - TREND database view, 237
  - TRENDdiscover file, 252
  - TREND dll, 240
  - TREND documentation, 242
  - TREND documentation directory, 241
  - TREND form, 243
  - TREND form directory, 244
  - TREND library file, 245
  - TREND mibs, 245
  - TREND module, 235
  - TREND Other, 252
  - TRENDproc file, 252
  - TREND qss file, 252
  - TREND report link, 248
  - TREND software version, 253
  - TRENDsum file, 252
  - TREND temporary file, 255
  - TRENDtimer definition, 255
  - TREND uninstall run command, 256
  - TREND uninstall run sql script, 257
- constraint
  - keywords, 455

- Constraint tag, 202
- control parameter file, 455
- copy file to
  - \$DPIPE\_HOME/dll directory, 240
  - \$DPIPE\_HOME/docs directory, 242
  - \$DPIPE\_HOME/lib directory, 244
  - \$DPIPE\_HOME/mibs directory, 245
  - \$DPIPE\_HOME/scripts/db directory, 253
  - \$DPIPE\_HOME/scripts directory, 251
  - \$DPIPE\_HOME/tmp directory, 254
- counter
  - 32-bit, 285, 558
    - spike threshold, 477
  - 64-bit, 285, 558
    - spike threshold, 477
- counter data, suppressing, 475
- create
  - datapipe, 95
  - parameter, datapipe\_manager, 92
  - table, 412
- CSV file format, 448
- csv report format, specifying, 368, 378
- custom procedure, 479

## D

- database
  - delete data from, 108
  - dictionary table, 423
  - engine, 89
  - objects, 93
  - query, 448
  - size, 413
  - source, 93
  - statistics, 107
  - target, 93
  - type required by report pack, 237
- database\_procedure directive, 235
- database\_table directive, 236



- database\_view directive, 237
- database-full size, 152
- database server
  - data database, 291
  - topology database, 291
- Database Table Management
  - display, 412
- database-used size, 152
- data collection
  - See also* collection
  - granularity, 341
  - summarization level, 341
- data database, defined, 291
- datadb option, 207, 216
- datapipe
  - alter, 96
  - create, 95
- datapipe\_manager, 85
  - error messages, 97
  - example, 95
  - option categories, 87
  - options, 88
  - syntax, 86
- data processing stops, 504
- data storage, local, 298
- data stored longer than retention period, 108
- data table
  - associated with property table, 463
- Date Range Error, 513
- day range that wraps, 409, 417
- days data retained in table, 105
- db\_delete\_data, 103
  - error messages, 109
  - examples, 108
  - options, 105
  - syntax, 104
- db\_requirement directive, 237
- dbstats tables, 152
- debug output, 42
- debug output level, 150
  - group\_manager, 208
  - transform\_maint, 395
- default
  - aging value, 105
  - data in column, 90
  - public, 435
- default parameter, 90
- delay processing, 502
- delete
  - data from database, 108
  - parameter, datapipe\_manager, 92
- delta time, 517
  - calculating, 288, 347, 562
- Department tag, 547
- dependency directive, 238
  - legacy format, 239
  - version directive and, 258
- dependent procedure, 479
- Deployment Wizard, 113
- deploytool, 113
  - deploying reports, 121
  - error messages, 123
  - options, 114
  - syntax, 114
  - undeploying reports, 121
- derived groups
  - defined, 215
  - removing with force option, 209, 225
- desc parameter, 90
- destination server
  - specifying, 412
- determine size of database, 413

dictionary entries, 89

dip\_manager, 127

- ASCII file with, 134

- syntax rules, 135

- enumerated list groups, 132

- error messages, 142

- examples, 138

- modes of operation, 133

- options, 129

- requirements, 127

- syntax, 128

- using, 136

DIP groups

- categories, 131, 132

- enumerated list, 132

- exporting, 129, 133, 137, 138

- examples, 140

- group name, 131

- importing, 130, 133, 136

- example, 138

- managing, 127

- naming convention, 131

- removing, 130, 131, 134, 137

- examples, 141

- replacing contents of, 131, 133, 137

- example, 139

- rule-based

- converting to enumerated list, 132

- viewing contents of, 132

directed instance polling

- See* dip\_manager

- See* DIP groups

directives

- bin, 234

- database\_procedure, 235

- database\_table, 236

- database\_view, 237

- db\_requirement, 237

- dependency, 238

- dll, 240

- docdir, 241

- docs, 242

- form, 243

- formdir, 243

- install.pkg, 233

- lib, 244

- mibs, 245

- mw\_collection\_def, 246

- report\_dir, 247

- report\_pack, 248

- reportlink, 248

- reports, 249

- run\_command, 250

- run\_sql\_script, 251

- scripts, 251

- software\_version, 252

- sql, 253

- tmp, 254

- trend\_timer\_def, 255

- uninstall\_cmd, 256

- uninstall\_sql, 256

- version, 257

directory

- \$DPIPE\_HOME/dll, 240

- \$DPIPE\_HOME/docs, 241, 242

- \$DPIPE\_HOME/lib, 156, 245

- \$DPIPE\_HOME/mibs, 245

- \$DPIPE\_HOME/reports, 247, 249

- \$DPIPE\_HOME/scripts, 252, 257, 435, 440

- \$DPIPE\_HOME/tmp, 254

- create target for reports, 247, 249

- discover
  - community strings
    - file, 436
    - for an SNMP GET request, 435
  - HP Performance Agent or HP Operations Agent, 351
  - IP, 430
  - managed nodes, 351
  - SNMP Type, 437
    - naming convention, 439
  - type definition files, 440
- distributed polling, 151, 285, 301, 344, 559
- dl\_type table, 431, 439, 440
- dll directive, 240
- docdir directive, 241
- docs directive, 242
- double quotation marks, 476
- DPIPE\_HOME, 451
  - /dll directory, 240
  - /docs directory, 241, 242
  - /lib directory, 156, 245, 451
  - /mibs directory, 245
  - /reports, 247, 250
  - /scripts directory, 252, 257
  - /tmp directory, 254
- dpipe\_pa child collector, 344, 349, 354
- dsi\_descr column, populating with trend\_label, 467
- dsi\_nodes table, 431, 435, 439, 440
- duplicate records, preventing, 391

## E

- ee\_collect
  - command line, 153
  - error messages, 157
  - how to use, 153
  - pa\_discovery and, 354

- Email tag, 548
- empty block, 485
- end statement, 485
- enumerated list groups, 132
  - defined, 215
  - install.pkg, 246
- environment variable, 451
- error messages
  - action, 47
  - by-variable, 424
  - collection\_manager, 55, 74
  - datapipe\_manager, 97
  - db\_delete\_data, 109
  - deploytool, 123
  - dip\_manager, 142
  - ee\_collect, 157
  - formdeploytool, 174
  - group\_manager, 226
  - groupctl, 192
  - node\_manager, 320
  - ovpi\_bulk\_copy, 330
  - ovpi\_run\_sql, 336
  - schedule, 380
  - transform\_maint, 400
  - trendcopy, 420
  - userctl, 541
  - viewctl, 573
- execution mode, 91
- exporting
  - DIP groups, 129, 133, 137, 138
    - examples, 140
  - group definitions, 208, 216, 220, 221
    - examples, 224
  - nodes, 309, 312, 316
    - examples, 318
  - polling policies, 59, 71, 221, 222
    - examples, 73
  - polling policy definitions, 209
- Extensible Markup Language, 543

extension .teel, 88

## F

file

- .dis, 440
- copy to \$DPIPE\_HOME/bin directory, 234
- install.pkg, 231
- log backup, naming conventions, 274
- trendtimer.sched, 41, 44
- type definition, 440
  - naming, 438
- See also* ASCII file

file locks, 298

filter, 417

- by date, 409, 417
- by day of week, 409
- by hour of day, 410
- defining, 202
- minimum value, trendpm, 478

FilterConstraints tag, 201

force option, 209, 217

- example, 225

foreign\_key parameter, 90

foreign key, 403

- reference column, 425

format

- parameter file, 452

Form Deployment Wizard, 167

formdeploytool, 167

- error messages, 174
- examples, 173
- options, 169
- syntax, 168
- using, 171

formdir directive, 243

form directive, 243

forms

- deploying, 167, 171
- deployment examples, 173
- undeploying
  - multiple, 172
  - single, 172
- viewing, 171

frequency

- running ee\_collect, 151

frequency value, trend\_export, 455

frequency variable, 246

## G

generate

- how to use, 179
- options, 178
- syntax, 177

GETBULK, turning on, 282, 555

GET message, 434

GETNEXT, 282, 286, 345, 555, 559

## group

- See also* DIP groups
- backing up before removing, 207
- category, 62, 64, 135, 213
  - defined, 215
- defined, 214
- definition (XML) file, 218
- derived, 215
  - removing with force option, 209, 225
- enumerated list, 132, 215
  - install.pkg, 246
- exporting definitions, 208, 216, 220, 221
  - examples, 224
- importing definitions, 210, 215, 220
  - example, 223
- name, 548
  - conventions for, 62, 131, 213
  - specifying with snmpv2dis, 384, 385
  - XML tag, 199
- remove definitions, 211, 217, 222
  - examples, 225
- rule-based, 132, 215
- specifying owner, 212
- types, 215
- user-access, defined, 183
- Web Access Server, 195

## group\_manager

- description, 205
- error messages, 226
- examples, 223
- forcing removal of dependent groups, 209, 217
- how to use, 220
- modes of operation, 215
- options, 207
- requirements, 205
- syntax, 206

## groupctl, 183

- error messages, 192
- how to use, 188
- modes of operation, 187
- syntax, 184

## groupdb option, 210, 216

## groupimport, 195

- example, 203
- file format, 197
- naming conventions, 197
- options, 196
- syntax, 195

## grouping by-variable, 497

## Groups tag, 199, 548

## Group tag, 548

## H

## help, 42, 82

## help command

- trend\_sum, 503
- trendcopy, 412

## high water mark

- database-used size, 478
- log, trendpm, 478
- setting
  - mw\_collect, 288, 347
  - trend\_sum, 504
  - trendpm, 478
  - vantage\_collect, 561

## holding file, 292

## host name, translated from IP address, 433

## hour range that wraps, 410

## HP Operations agent, 351

## HP Operations agent, collecting data from, 337

## HP Performance Agent, 351, 354

HP Performance Agent, collecting data from, 337

html report format, specifying, 368

hysteresis

beginning of time period range, 509

units to reprocess, 501

## I

identity column, 403

ID tag, 547

ignore NULL values, 513

ignore time-period range restrictions, 504

importing

DIP groups, 130, 133, 136

example, 138

group definitions, 210, 215, 220

example, 223

nodes, 307, 310, 311, 315

examples, 317

polling policies, 60, 64, 70, 210, 215, 220

example, 71, 223

index

maintaining, 261

statistics, 107

indexmaint, 261

examples, 271

options, 262

Oracle, 268, 270

Sybase, 265, 270

syntax, 261

indices on upload table, 284, 343, 557

infile option, 210

input parameter file, 448, 451

install.pkg, 231

document conventions for directives, 234

format of directives, 233

processing of, 233

processing rules, 233

sample file layout, 259

syntax rules, 233

installation

collection policy, 245

command to run during, 250

SQL script to run during, 251

instance, 448

integrity checks, 415

interval

ee\_collect executes by, 148

executes, 151

field, 151

list of instructions, 153

polling entries, 156

schedule file definition, 523

interval polling, 301

IP

address, translating to host name, 433

range, defining, 432

isql command, 332, 334, 335

## K

key ID-based deletions, 106

keyid mapping scheme, 415

keymap, 411

integrity check, 415

keyword

condense, 455

frequency, 455

max, 455

style, 454

target name/table key pairs, 454

## L

- lag time, 510
- last processed sample, 509
- lib directive, 244
- list
  - trend\_sum options, 503
  - trendcopy options, 412
- local data storage, 298
- lock
  - trend\_proc automatic locking check, 487
- log, 413
  - entries, ee\_collect, 153
  - information level, pa\_collect, 345
- log\_backup, 273
  - options, 274
  - output file naming conventions, 274
  - syntax, 273
  - trendtimer.sched file, 275
- log files
  - backing up, 289
- log information level, 286, 560

## M

- macro, 483
- managed-object group name, 384, 385
- manual test, 44
- master trend\_proc, 487
- maximum
  - collection processes, 150
  - rows, 151, 413
- MaxRows statement, 151

- messages
  - action, 47
  - by-variable, 424
  - collection\_manager, 55, 74
  - datapipe\_manager, 97
  - db\_delete\_data, 109
  - deploytool, 123
  - dip\_manager, 142
  - ee\_collect, 157
  - formdeploytool, 174
  - GET, 434
  - group\_manager, 226
  - groupctl, 192
  - node\_manager, 320
  - ovpi\_bulk\_copy, 330
  - ovpi\_run\_sql, 336
  - schedule, 380
  - transform\_maint, 400
  - trendcopy, 420
  - userctl, 541
  - viewctl, 573

- MIB-II tables, 436

- MibIndexMap statement, 563

- mibs directive, 245

- MIB tables

- collecting values from a particular node, 302
  - polling application for, 290
  - specifying table to collect, 288, 346

- minimum filter value, 285, 344, 558

- minimum lag time, 502

- mode

- as-is, 95
  - compatibility, 88, 95
  - execution, 91
  - non-compatibility, 95

- mw\_collect, 277
  - configuration file, 292
  - parameters, 293
  - directory structure, 304
  - distributed polling, 301
  - examples, 305
  - file locks, 298
  - interval polling, 301
  - invoked by trendtimer, 301
  - multiple pollers, 302
  - polling instructions, 302
- mw\_collection\_def directive, 246

## N

- name
  - group, 548
  - sub-group, 200
- Name tag, 199
- nesting trend\_proc files, 487
- network subnet, 433
- node\_manager, 307
  - ASCII file with, 312
  - error messages, 320
  - examples, 317
  - modes of operation, 311
  - options, 308
  - requirements, 307
  - syntax, 308
  - using, 315

- nodes
  - ASCII file list, 312
  - deleting, 309, 311, 316
    - example, 319
  - discovering, 351
  - exporting, 309, 312, 316
    - examples, 318
  - importing, 307, 310, 311, 315
    - examples, 317
  - locating, 351
  - managing, 307
  - removing from type and view lists, 310, 311, 316
    - examples, 319
  - SNMP
    - profile assigned to, 314
    - support of protocol, 313, 314
  - type list, removing from, 310, 311, 316
    - examples, 319
  - view list, removing from, 310, 311, 316
    - examples, 319
- non-compatibility mode, 95
- nowait block, 484
- null
  - columns, 510
  - parameter, datapipe\_manager, 90
- NULL handling, 513
- number of characters in substring, 469

## O

- object by-variables, 135
- Object Manager
  - deploying forms to be viewed, 167
  - form deploy example, 173
- option
  - mutually exclusive, 107
- option categories, trendcopy, 406



## Oracle

- Analyze Table command, 268, 270
- copying data across databases, 403
- db\_requirement option, 238
- table indexes, 268

outfile option, 210

output file naming conventions, 274

override default, 513

override source statements, 152

ovpi\_bulk\_copy, 323

- error messages, 330
- examples, 329
- options, 325
- syntax, 324
- using, 327

ovpi\_run\_sql, 331

- error messages, 336
- example, 335
- options, 332
- syntax, 331
- using, 334

ovpi\_timer UNIX process, 521, 526

OVPI Timer Windows service, 521

## P

pa\_collect, 337

- dependence on trendtimer, 343
- options, 339
- syntax, 337

pa\_discovery, 351

- configuration file, 354
- ee\_collect and, 354
- options, 352
- syntax, 352

## Package Manager

- command to run when removing a report pack, 256
- insert entry in trendtimer.sched file, 255
- install.pkg file, 231
- processing rules for install.pkg directives, 233
- removes entry from trendtimer.sched file, 256
- running SNMP Type Discover, 439
- script to run when removing a report pack, 256

packets, ping, 431, 433

parameter

- e-mail, 545
- file
  - keyword, 452
  - valid keywords, 452
- max\_rows, 413
- password, 545
- telephone, 545

parameterless command

- ee\_collect, 148
- transform\_maint, 394

parent poller, 299

- defined, 290
- responsibilities, 291

parser to validate the XML format, 199, 547

password parameter, 545

Password tag, 547

pdf report format, specifying, 368

pdu, 432, 438

percentage level for valid data

- mw\_collect, 282
- trendpm, 479
- vantage\_collect, 556

performance penalties, 513

- performance problems
  - time-period tables, 390
- Phone tag, 547
- piadmin, 359
  - options, 360
  - syntax, 359
- PI database
  - dictionary entries, 89
- ping packets
  - number of, 431
  - size, 433
- ping timeout, 432
- PI SQL file, 254
- PI Timer Windows service, 527
- PI version required for report pack,
  - specifying, 252
- placeholders, required, 90
- poller
  - child, 290, 291
  - instance of, 291
  - parent, 290, 291, 299
- poller, alternate name, 283, 343, 557
- pollfrom computer, specifying, 61, 68, 211, 216
- polling
  - control table, 153
  - distributed, 151, 285, 301, 344, 559
  - dynamic list of nodes, 290
  - entering policy information, 153
  - instructions, 151
    - mw\_collect, 302
  - interval, 151, 301
  - node, 156
  - requests, 151
  - station for a polling policy, 61
  - with multiple pollers, 302
- polling interval, 43
- polling policies
  - backing up before removing, 207, 222
  - definition (XML) file, 219
  - exporting, 59, 71, 221, 222
    - examples, 73
  - exporting definitions, 209
  - importing, 60, 64, 70, 210, 215, 220
    - example, 71, 223
  - install with Package Manager, 245
  - managing, 57
    - with group\_manager, 205
  - modifying, 61, 64, 70
    - example, 72
  - removing, 61, 62, 65, 70, 74, 211, 217, 222
    - examples, 73, 225
    - specifying owner, 212
- polling policy
  - exporting definitions, 209
- Polling Policy Manager
  - importing nodes, 307
- populate property tables, 284, 343, 558
- port number, 43
- procedures
  - application type, specifying, 476
  - application types, described, 479
  - class, specifying, 476
  - custom, 479
  - delete option, 474
  - dependent, 479
  - execution parameters, 475
  - managing, 473
  - registering, 235, 475
  - type, specifying, 476
  - types, list, 479

- property table
  - associated with data table, 463
  - defined, 214
  - invalid name, 422
  - keyids mapped in, 415
  - populate, 284, 343, 558
  - populating with data from data table, 463
  - views, 404
- protocol data unit, 432, 438

## Q

- quotation marks, 476

## R

- r2d parameter usage, 95
- raw-to-delta processing, 95
- RealName tag, 547
- rebuild index, 264
- recognized view, 404
- regenerate definition files, 280, 339, 554
- regenerate stored procedure, 505
- registering
  - datapipe, 95
  - procedure, 235
  - table, 236
  - view, 237
- rejecting samples, 285, 558

- remove
  - DIP groups, 130, 131, 134, 137
    - examples, 141
  - group definitions, 211, 217, 222
    - examples, 225
  - nodes, 310, 311, 316
    - examples, 319
  - parameter, datapipe\_manager, 92
  - polling policies, 61, 62, 65, 70, 211, 217, 222
    - examples, 73, 74, 225
- rename index, 264
- replacing DIP groups, 131, 133, 137
  - example, 139
- report
  - query a database Server, 448
- report\_dir directive, 247
  - reports directive and, 250
- report\_pack directive, 248
  - dependency directive and, 239
  - version directive and, 258
- reportlink directive, 248
- report pack
  - directory structure, 232
  - installing, 231
  - primary identification for, 249
  - specify dependent report pack, 238
  - specify name of, 248
  - specify version of, 257
- Report Pack Selection Wizard, 238
- reports
  - create target directory for, 247, 249
  - deploying directories of, 121
  - specifying desired format, 368
- reports directive, 249
  - report\_dir and, 247
- retention period, 108
- retries, number of, 287, 346, 560

- retry interval, 286, 345, 559
- RMON community string, 436
- RMON-manageable device, 440
- Role tag, 548
- rolling baseline, 511
- round collection time, 43
- rule-based groups, 132
  - defined, 215
- run\_command directive, 236, 237, 250
- run\_sql\_script directive, 251

## S

- satellite server, 414
- schedule, 365
  - error messages, 380
  - file, trendtimer, 523
    - example, 525
  - how to use, 374
  - modes of operation, 374
  - options, 367
  - PI processes, 521
  - syntax, 366
- scheduling PI processes, 521
- scripts directive, 251
- server
  - archive, 414
  - central, 414
  - satellite, 414
  - source, 411
  - target, specifying, 412
- services
  - stopping, 527
- single quotation marks, 476
- size
  - database-full, 152
  - database-used, 152
  - size of batch to commit, 503
  - size parameter, 89
- SNMP
  - See also* SNMP V2
  - collection cycles, 280, 340, 554
  - GET, 433, 435, 438
  - GET message, 434
  - number of variables to include in
    - varbind list, 286, 345, 559
  - polling application, 290
  - port, specifying, 286, 345, 559
  - port number, defining, 432, 438
  - profile assigned to node, 314
  - protocol supported by node, 313, 314
  - Read Community string, 301
  - request retries, 287, 346, 433, 438, 560
  - tests, 441
  - timeout, 286, 345, 432, 438, 559
  - version, 279
  - version, specifying, 287, 346, 561
- SNMP V2
  - identifying devices that support protocol, 383
  - specifying version, 287, 346, 561
- snmpv2dis, 383
  - options, 384
  - specifying managed-object group name, 384, 385
  - syntax, 383
- software\_version directive, 252
- source
  - database, 93
  - data directory, 152
  - data file, single, 152
  - data table, 422
  - file disposition, 152
  - server, 411
  - table, 94
    - to copy, 412

- SourceDirectory, 152
- SourceDisposition, 152
- SourceFile, 152
- source statements, overriding, 152
- spike, 285, 558
  - threshold, 477
- SQL
  - generating procedure code, 474
  - Name column, 412
  - Table Manager, 106
- sql directive, 253
- sqlldr command, 326, 327, 329
- sqlname parameter, 89
- sqlplus command, 332, 334, 335
- SQL script
  - command that runs scripts on PI, 331
  - component type, 252
  - report pack removal, 256
  - run during installation, 251
- srep report format, specifying, 368, 379
- start\_date format, 389
- starting character position of substring, 469
- statement
  - begin, 484
  - end, 485
  - MaxRows, 151
  - SourceDirectory, 152
  - SourceDisposition, 152
  - SourceFile, 152
- statistics
  - update, 107, 108
- store data locally, 285, 344, 558
- subnet mask, 433
- substring, number of characters in, 469
- summarization level, 341

- summary definition, defined, 507
- suppressing counter data, 475
- Sybase
  - copying data across databases, 403
  - db\_requirement option, 238
  - table indexes, 265
  - update statistics command, 265, 270
- synchronous execution (trendpm), 474
- syntax
  - check TEEL file statements, 92
  - ee\_collect command, 148
- sysobjects, 423

## T

- ta\_period, 287, 346, 561
- table
  - element, 509
  - indexes
    - Oracle, 268
    - Sybase, 265
  - index maintenance, 261
  - registering, 236
  - retention period guideline, 390
  - source, 94, 412
  - target, 94
  - time-period, 387
  - upload, 476
- table\_category variable, 246
- table\_key, 403
- tab separated file format
  - condensed, 462
  - uncondensed, 460
- tag definitions, 199
- tag set parameter formats, 197

- target
  - database, 93
  - name, 454
  - row in table, locating, 467
  - server, specifying, 412
  - table, 94
  - table name, 474
- target\_name, 403
- task
  - list of, trendcopy, 411
  - timings, 413
- TEEL
  - file
    - name, 150
- TEEL file name
  - ee\_collect execution, 149
  - specifying, 88
- template for table creation, 95
- terminate associated stored procedures, 507
- test manually, 44
- ThreadCount parameter, 354
- ThreadTimeout parameter, 354
- time control options, 255
- timeout, ping, 432
- time period, 509
  - tables, populating, 387
- tmp directive, 254
- topology database, defined, 291
- tpmaint, 387
  - examples, 391
  - invoked by trendtimer, 390
  - options, 389
  - performance impact, 390
  - syntax, 388
  - trend.log message, 390
- trace level, setting, 353
- trace log file, 287, 346, 560
- trace parameters, setting, 474
- transcribe keymap for property table, 411
- transform\_maint, 393
  - error messages, 400
  - how to use, 397
  - syntax, 394
- transformation, defined, 507
- trend.log file
  - indexmaint, 264
  - tpmaint, 390
- trend\_discover, 429
  - syntax, 430
- trend\_export
  - Controls and Switches, 449
  - description, 447
  - examples, 451
  - functionality, 448
  - input parameter file, 452
  - parameters for controls and switches, 449
  - process, 448
- trend\_label, 463
  - options, 465
  - property table, 463
  - syntax, 464
- trend\_lock
  - locking check for trend\_proc, 487

- trend\_proc, 481
  - block, 483
    - definition, 483
    - format, 484
  - examples, 489
  - file
    - characteristics, 484
    - creating and applying, 488
    - description, 483
    - nesting limit, 487
    - processing, 485
    - scheduling, 489
    - syntax, 489
  - macro, 483
  - options, 482
  - processing multiple trend\_proc files, 487
  - scheduling, 489
  - syntax, 481
  - trend\_lock locking check, 487
  - trendtimer.sched file, 483
- trend\_sum, 493
  - destination table
    - optional suffix name, 499
  - input file description, 508
  - naming conventions, 505
  - period-end threshold, 510
  - requirements, 493, 507
  - trendexec and, 445
  - trendlabel and, 463
  - trendpm and, 474
- trend\_timer\_def directive, 255
- trendadm, 433, 439
- trendadm user account, 533
- TREND collector definition, 246
- trendcopy, 403
  - error messages, 420
  - options, 407
  - syntax, 405
  - verify keymap table, 413
- TREND database procedure, 235
- TREND database table, 236
- TREND database view, 237
- TRENDdiscover file, 252
- TREND dll, 240
- TREND documentation, 242
- TREND documentation directory, 241
- trendexec, 445
  - example, 446
  - options, 446
  - syntax, 445
- TREND form, 243
- TREND form directory, 244
- trendit
  - indexmaint and, 265, 268
- TREND library file, 245
- TREND mibs, 245
- TREND module, 235
- TREND Other, 252
- trendpm, 473
  - options, 474
  - procedure, 292
  - syntax, 473
- TRENDproc file, 252
- TREND qss file, 252
- TREND report link, 248
- TRENDsheet
  - trend\_export and, 447
- TREND software version, 253
- TRENDsum file, 252
- TREND temporary file, 255

- trendtimer, 41, 521
  - configuration, 151
  - mw\_collect and, 301
  - options, 522
  - sample command line, 528
  - schedule file, 156, 523
  - starting
    - at system startup, 526
  - stopping
    - on UNIX, 527
  - syntax, 522
  - tpmaint invocation, 390
  - trendexec and, 445
- trendtimer.sched
  - db\_delete\_data execution, 108
  - description, 523
  - indexmaint execution, 270
  - insert entry in file, 255
  - log\_backup execution, 275
  - log file backup, 289
  - override, UNIX only, 523
  - remove entry from, 256
  - trend\_label execution, 463
  - trend\_proc and, 483, 487, 489
- trendtimer.sched file, 44
- TRENDtimer definition, 255
- TREND uninstall run command, 256
- TREND uninstall run sql script, 257
- type
  - definition files, 440
    - naming, 438
  - parameter, 89
  - validation, 439
- type\_list\_name variable, 246

## U

- uninstall
  - command to run during, 256
  - SQL script to run during, 256
- uninstall\_cmd directive, 256
- uninstall\_sql directive, 256
- unique key, 297
- UNIX
  - stopping services, 527
- unix 0 time, 524
- update
  - keymap for property table, 410
  - statistics, 107, 108
  - time-period table, 508
- update statistics command, 265, 270
- upload table
  - use existing indices, 476
- user
  - defining as administrator, 548
  - group\_manager option, 212
- user-access group, defined, 183
- user accounts
  - maintaining, 529
  - trendadm, 533
- userctl, 529
  - error messages, 541
  - examples, 538
  - how to use, 534
  - modes of operation, 533
  - options, 531
  - syntax, 530
- userimport, 543
  - example, 549
  - naming conventions, 545
  - options, 544
  - syntax, 544
- users
  - add, modify, or delete, 543
- Users tag, 200, 547
- User tag, 201, 547



using input file and command line option,  
508

utility  
trend\_export, 447

## V

vantage\_collect, 551  
example, 564  
how to use, 563  
options, 554  
syntax, 552

verifyparms  
parameter, datapipe\_manager, 92

version  
minimum PI version required, 252  
report pack string, 257  
stamp, 43

version directive, 257  
report\_pack directive and, 249

view  
recognized, 404  
registering, 237

viewctl, 565  
add mode, 570  
delete mode, 571  
error messages, 573  
examples, 572  
how to use, 569  
modes of operation, 569  
modify mode, 571  
options, 567  
required options, 569  
syntax, 566

viewer, 575  
examples, 579  
options, 577  
syntax, 576

## W

wait block, 484

Web Access Server  
deploy file to, 242  
deploy forms to, 243  
deploy reports to multiple systems, 119  
form undeploy example, 173  
groups, 195  
maintaining user accounts, 529

## X

### XML

parser validates format, 199  
required for imported files, 195, 543  
tag definitions, 199  
tag sets  
groupimport, 197  
userimport, 546  
used in group definition files, 218  
used in polling policy definition files, 219

## Z

zero denominator, 514

# We appreciate your feedback!

If an email client is configured on this system, by default an email window opens when you click on the bookmark “Comments”.

In case you do not have the email client configured, copy the information below to a web mail client, and send this email to **docfeedback@hp.com**

**Product name:**

**Document title:**

**Version number:**

**Feedback:**



