



# Using HP Data Protector Debug Log Collector Tool

“Best Practices”

## Table of contents

Executive summary.....	2
Debugging basics.....	3
Debugging using the Data Protector GUI .....	3
Debugging using the trace configuration file .....	3
Debugging using the scheduler.....	3
Changing the location of debug files.....	4
Debug syntax .....	4
Names and locations of debug files .....	5
Using Debug Log Collector (omnidlc).....	5
omnidlc Command Syntax .....	6
Options .....	7
Examples of omnidlc “Debug Log Collector”.....	9
Default collection .....	9
Verbose Collection.....	10
Collection from specific clients to specific directory .....	10
Cleanup hosts.....	10
Show needed disk space .....	10
Unpack dlc.pck.....	10
Use Case.....	11
Enable Debugging.....	11
Run Debug Session.....	12
Collect Debugs.....	12
Send package to support .....	13
Fix problem .....	13
Cleanup Systems .....	13
Using the dlc.pck file .....	14
Conclusions .....	15
Limitations.....	15
Appendix .....	16
HP Data Protector 6.0 differences .....	16
For more information.....	16
Visit.....	16
HP Data Protector guides .....	16

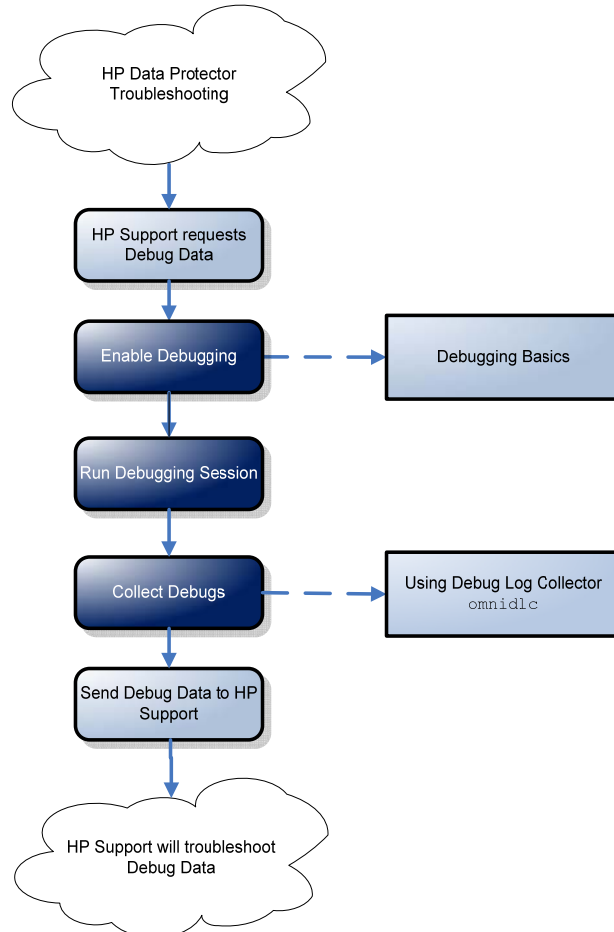
## Executive summary

The HP Customer Support Service might ask you to gather and send those data they need to resolve a technical issue. Since Data Protector operates in large network environments, the data might sometimes be difficult to gather. The Data Protector *omnidlc* command is a tool for collecting and packing log, debug, config and getinfo files.

This white paper provides information on how to use the HP Data Protector feature “Debug Log Collector” (*omnidlc*) and will give you the best practices for this feature. It will also cover some debugging basics, but the whole range of debugging features, troubleshooting hints and guidelines is documented in the “HP Data Protector troubleshooting guide” which can be found on the HP Data Protector DVD, in the /docs directory on your Cell Manager or can be downloaded from the official HP Data Protector webpage @ [www.hp.com/go/dataprotector](http://www.hp.com/go/dataprotector).

This white paper is based on the HP Data Protector 6.1 release and its *omnidlc* features. Some options are not available in earlier releases (see Appendix).

**Figure 1:** Flowchart Debugging



The main flow of collecting and sending debugs is shown in the above chart and will be described in detail in this document.

## Debugging basics

When HP Support is asking you to send debug data, there are several methods to generate this data with Data Protector. This chapter will give you just an overview of the different ways of running Data Protector in debug mode, further details can be found in the Data Protector troubleshooting guide (chapter 11 – Before calling support). Details how to generate Inet debugs and CRS debugs are documented in the HP Data Protector troubleshooting guide and not part of this document.

**Note:** Collect debugs only when the support organization requires them to resolve a technical issue. When Data Protector runs in the debug mode, it creates debug information that consumes a **large amount of disk space**.

Consult the support organization about the required detail level and environmental conditions for debugging.

When debug files are not created like expected, the *debug.log* file on the problem host needs to be checked and you need to consult your support representative.

Debugging can be enabled in the listed different ways:

- Debugging using the Data Protector GUI
- Debugging using the trace configuration file
- Debugging using the scheduler
- Debugging using command line option *-debug*

### Debugging using the Data Protector GUI

In the **File** menu, click **Preferences**, and then click the **Debug** tab. Specify the debug options and restart the GUI. The GUI will restart in the debug mode.

When using the JAVA GUI a GUI restart is not required, but you need to make sure to disable debugging afterwards.

### Debugging using the trace configuration file

Edit the trace configuration file, located in:

Windows Server 2008: *Data\_Protector\_program\_data\Config\server\Options\trace*  
Other Windows systems: *Data\_Protector\_home\Config\server\Options\trace*  
UNIX systems: */etc/opt/omni/server/options/trace*

### Debugging using the scheduler

To debug scheduled sessions, edit the schedule file, located in:

Windows Server 2008:

*Data\_Protector\_program\_data\Config\server\Schedules* or  
*Data\_Protector\_program\_data\Config\server\Barschedules*

Other Windows systems:

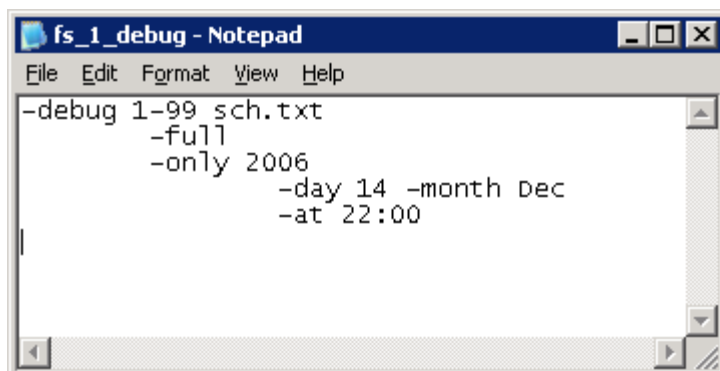
*Data\_Protector\_home\Config\server\Schedules* or  
*Data\_Protector\_home\Config\server\Barschedules*

UNIX systems: */etc/opt/omni/server/schedules* or  
*/etc/opt/omni/server/barschedules*

Add debugging parameters in the first line of the file.

**Note:** Before you edit the file, make a copy of it, as the changes have to be reverted when debugging is no longer desired.

**Figure 2:** Example – Schedule with debug enabled



## Changing the location of debug files

To change the location of debug files on a per system basis the omnirc variable OB2DBGDIR=<pathname> can be used. You can find the use of this variable in the omnirc.tmpl file.

```
...
#     OB2DBGDIR=<pathname>
#     Default: none
#     This variable is used to change the location of debug files on a per
#     system basis. You have to specify a fully qualified path of an existing
#     directory. This variable has precedence over the paths specified by the
#     postfix parameter.
#     By default, this variable is not set. If this variable is not set, the
#     pathname is set as /tmp (UNIX) or <Data_Protector_home>\tmp (Windows).
...
```

The specified directory needs to exist and it needs to have the necessary permissions.

## Debug syntax

Almost all Data Protector commands can be started with an additional `-debug` parameter that has the following syntax:

```
-debug 1-999[,C:n][,T:s][,U] XYZ [host]
```

where:

- 1-999 is the debug range. Specify the range 1-200 unless instructed otherwise by HP Data Protector support. (Default in GUI is 1-99).  
Specify optional parameters as a part of the range parameter, separated by commas:
  - C:n limits the size of debug files to n kilobytes. The minimum value is 4 (4kB) and the default value is 1024 (1 MB).  
For more information, see “Limiting the maximum size of debugs” chapter in the Data Protector troubleshooting guide.
  - T:s is the timestamp resolution, where the default value is 1, 1000 means the resolution is one millisecond and 0 means timestamps are turned off. On some platforms (Novell NetWare, MPE/iX), millisecond resolution is not available.

- `U` is the Unicode flag. If it is specified, the debug files on Windows are written in the Unicode format.
- `XYZ` is the debug postfix, for example `DBG_01.txt`.
- `host` is a list of clients where debugging is turned on. Use this option to run the debugging only on the clients specified. Delimit multiple clients by spaces. Enclose the list in quotes, for example: `"computer1.company.com computer2.company.com"`.

## Names and locations of debug files

The debug postfix option is used for creating debug files in the following directory:

Windows Vista, Windows Server 2008: `Data_Protector_program_data\tmp`

Other Windows systems: `Data_Protector_home\tmp`

UNIX systems: `/tmp`

Novell NetWare: `SYS:\USR\OMNI\TMP`

The files are named

`OB2DBG_did__Program_Host_pid_XYZ`

where:

- `did` (debugging ID) is the process ID of the first process that accepts the debugging parameters. This is the ID of the debugging session and is used by all further processes.
- `Program` is the code name of the Data Protector program writing the debug file.
- `Host` is the client where the debug file is created.
- `pid` is the process ID.
- `XYZ` is the postfix as specified in the `-debug` parameter.

Once the backup or restore session ID `sid` is determined, it is added to the file name:

`OB2DBG_did_sid_Program_Host_pid_XYZ`

Processes that add the `sid` are BMA/RMA, xBDA/xRDA, and other processes started by the session, but not by the BSM/RSM itself.

## Using Debug Log Collector (`omnidlc`)

After Data Protector debug data has been generated, the `omnidlc` command can be used to collect Data Protector debug, log, and `getinfo` files from the Data Protector cell (by default, from every client). The command transfers the data from selected clients to the Cell Manager where it is then packed.

The command can also selectively collect the data, for example, only log files from a certain client, or only debug files that were created during a particular Data Protector session.

A Data Protector debug session will create debug files in the Data Protector `tmp` directory on every client which has been participating in the Data Protector session in debug mode.

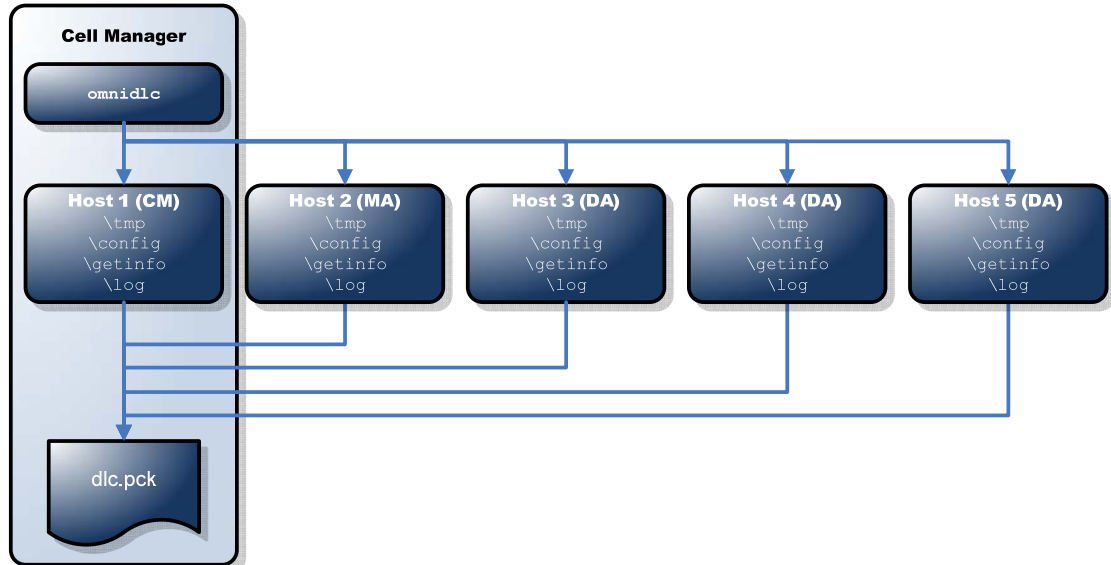
This can be a bunch of data and when you run several sessions hard to identify which are the needed files. All the collection can be automatically done by the `omnidlc` command.

Let's have an example. When debugging a Data Protector backup session running on three hosts where Disk Agent is used, one additional host used as Media Agent, you have in total 5 systems involved in this example backup session.

- 1 x Cell Manager (CM)
- 1 x Media Agent (MA)
- 3 x Disk Agent (DA)

In this scenario you will get debug files on every of these 5 systems. With the help of the `omnidlc` command you are able to collect all debug files and additional needed information with the execution of only one command. No need to travel around with ftp.

**Figure 3:** Scenario – `omnidlc` collects debug data from 5 systems (incl. CM)



## omnidlc Command Syntax

The Debug Log Collector tool is a command line tool. The command `omnidlc` has several options which are described in detail in this chapter.

```
C:\>omnidlc -help
Usage: omnidlc -version | -help
Usage: omnidlc {-session sessionID | -did debugID
  | -postfix string | -no_filter}
  [-hosts list]
  [-pack filename | -depot directory | -space | -delete_dbg]
  [-no_logs] [-no_getinfo] [-no_compress] [-no_config]
  [-no_debugs | -debug_loc Dir1 [Dir2 ...]] [-verbose]
  [-add_info [-any | host] path]
Usage: omnidlc -localpack [filename]
Usage: omnidlc -unpack [filename]
Usage: omnidlc -uncompress filename
Usage: omnidlc [-hosts list] -del_ctracelog
```

## Options

`-version`

Displays the version of the `omnidlc` command.

`-help`

Displays the usage synopsis of the `omnidlc` command.

`-session sessionID`

Limits the collected debug files to those that were produced during the Data Protector session identified by the *sessionID*. Note that on OpenVMS, the `omnidlc` command run with the `-session` parameter does not collect the debug files produced during specified session, because session names are not part of the OpenVMS debug filename. Instead, all available logs are collected.

`-did debugID`

Limits the collected debug files to those identified by the *debugID*.

Example:

```
OB2DBG_3212_2008-03-04-1_BMA_test.testdomain.test_3760-536_Debug.txt
```

In this example the *debugID* is 3212.

`-postfix string`

Limits the collected debug files to the specified debug postfix.

`-no_filter`

Does not limit (select) the collected debug files.

`-hosts list`

Limits the files to be collected to the clients specified in the *list*. The hostnames must be separated by spaces. The debug files collected are still subject to `-session`, `-did` or `-postfix` options.

`-pack filename`

All collected files are, by default (if this option is not specified), packed and saved in the current directory as the `dlc.pck` file. If this option is specified, the collected files are packed and saved in the specified file in the current directory on the Cell Manager. If the full path name is specified, the files are packed and saved in the specified file in the specified directory.

To add files other than the collected files to the package, copy the files to one of the following directories before running the command:

```
dlc/client/getinfo, dlc/client/log, or dlc/client/tmp (on UNIX),  
or .\dlc\client\getinfo, .\dlc\client\log, or .\dlc\client\tmp
```

(on Windows). You cannot add directories, but only files. If the files are not copied to one of the specified directories, the package cannot be unpacked during the unpack phase.

`-depot [Directory]`

If the *Directory* is specified, the collected files are not packed and are saved to the `dlc` directory of the specified directory. If the *Directory* is not specified, the files are saved on the Cell Manager in the directory

```
Data_Protector_program_data\tmp\dlc (Windows Server 2008),  
Data_Protector_home\tmp\dlc (other Windows systems), or /tmp/dlc  
(UNIX systems).
```

`-space`  
 Displays the disk space required on the Cell Manager for the collected files.

`-delete_dbg`  
 Deletes the by filter (`-session`, `-did`, ...) selected files on clients. If the option `-no_filter` is selected all debug files will be deleted. On OpenVMS, if run together with the `-session` parameter, the command does not delete any debugs from the debug files directory.

`-no_getinfo`  
 Excludes the `getinfo` file from the selection. For OpenVMS, this parameter is not applicable as OpenVMS systems do not have the `get_info` utility.

`-no_config`  
 Excludes the configuration information from the selection.

`-no_logs`  
 Excludes the log files from the selection.

`-no_debugs`  
 Excludes the debug files from the selection.

`-no_compress`  
 Disables the compression of the collected files on clients. By default, the compression is enabled.

`-debug_loc dir1 [dir2]...`  
 Includes debugs not only from the default debug files directory but also from other directories, `dir1`, `dir2`, ... Note that the subdirectories are excluded from the search. If a specified directory does not exist on a particular client, the directory is ignored.  
 This option is valid only if the `-no_debugs` option is not specified.

`-verbose`  
 Enables verbose output. By default, verbose output is disabled.

`-add_info path`  
 Includes the additional information (for example, screenshots, pictures and the like) from a directory on client identified by `path`.  
 The `-any` option is used when the directory path is the same for all clients. It is important to make sure the path is not host-specific before using this option.

`-localpack [filename]`  
 Packs the directory structure from the current directory (must be the directory containing the `dlc` directory generated by the `-depot` option) to the `filename`.  
 If the `filename` is not specified, the `dlc.pck` file is created in the current directory.  
 This option is equivalent to the `-pack` option, but is to be used only if the data is collected using the `-depot` option.  
 To add files other than the collected files to the package, copy the files to one of the following directories before running the command:  
`dlc/client/getinfo`, `dlc/client/log`, or `dlc/client/tmp` (on UNIX),  
 or `.\dlc\client\getinfo`, `.\dlc\client\log`, or `.\dlc\client\tmp`  
 (on Windows). You cannot add directories, but only files. If the files are not copied to one of the specified directories, the package cannot be unpacked during the unpack phase.



`-unpack [filename]`

Creates the `dlc` directory in the current directory, and unpacks the contents of the *filename* to the `dlc` directory. If the *filename* is not specified, the `dlc.pck` file in the current directory is unpacked.

Use this option when the collected (compressed or uncompressed) data was packed on the Cell Manager either using the `-pack` option or the `-localpack` option.

`-uncompress filename`

Uncompresses the unpacked compressed single file in the current directory.

Use this option after the packed data is unpacked using the `-unpack` option.

`[-hosts list] -del_ctracelog`

Deletes `ctrace.log` files containing the information where (on which clients) debug logs are generated and which debug prefixes are used. If the `-hosts list` option is specified, the command deletes `ctrace.log` files on specified clients only. Otherwise, `ctrace.log` files on all clients in a cell are deleted.

## Examples of `omnidlc` “Debug Log Collector”

### Default collection

To collect and compress all debug, log, and getinfo files from the cell and pack them in the `dlc.pck` file in the current directory on the Cell Manager run:

```
omnidlc -no_filter
```

```
C:\>omnidlc -no_filter
Collection started: pack.
Collection finished. Pack file dlc.pck
```

`dlc.pck` is the default package file which will be put in the current directory, any other output file and location can be chosen. This is an advantage if you have several debug sessions then you may use the `-pack` option of the `omnidlc` command and pack them to the file `C:\debug\hostA.pck` on the Cell Manager. To do so, run:

```
omnidlc -pack C:\debug\hostA.pck
```

```
C:\>omnidlc -no_filter -pack C:\debug\hostA.pck
Collection started: pack.
Collection finished. Pack file C:\debug\hostA.pck
```

This will put the packed file `hostA.pck` in the directory `C:\debug`, this directory needs to exist, if not the `omnidlc` command will send an error.

## Verbose Collection

To collect and compress all debug, log, and getinfo files from the cell and pack them in the `dlc.pck` file in the current directory on the Cell Manager, using verbose output, run:

```
omnidlc -no_filter -verbose
```

```
C:\>omnidlc -no_filter -verbose
Collection started: pack.
Collection from test.testdomain.test started.
Packing file: cons_ftsstats.log
Packing file: ctrace.log
. . .
Packing file: classspec
Packing file: userlist
Packing file: webaccess
Packing file: xcopy.cfg
Packing file: omni_format
Packing file: omni_info
Files from test.TESTDOMAIN.TEST packed.
Collection finished. Pack file dlc.pck
```

In the command window you will see which file is packed into the package.

## Collection from specific clients to specific directory

To collect from the clients "client1.company.com" and "client2.company.com" to the directory "c:\depot" on the Cell Manager, in this example without compressing and packing the files, run:

```
omnidlc -no_filter -hosts client1.company.com client2.company.com -depot
c:\depot -no_compress
```

## Cleanup hosts

To delete all debug log files for the session with the ID "2007/08/27-9", run:

```
omnidlc -session 2007/08/27-9 -delete_dbg
```

## Show needed disk space

To display disk space needed on the Cell Manager for the uncompressed debug files with the debugID "2351" from the client "client.company.com", run:

```
omnidlc -did 2351 -hosts client.company.com -space -no_getinfo -no_logs
-no_compress
```

## Unpack dlc.pck

To unpack the "dlc.pck" file to the "dlc" directory of the current directory, run:

```
omnidlc -unpack
```

To get more examples refer to the HP Data Protector troubleshooting guide.

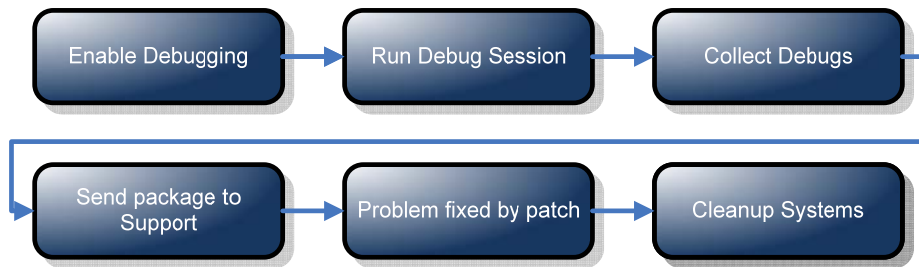
# Use Case

We will have an example which could fit into many support cases and can be used as a simple cookbook, which can be adapted with the above described options.

This use case will describe in detail the following scenario.

An error in Data Protector occurred and collecting of debugs is needed and requested by HP Support.

**Figure 4:** Use Case - Flow



## Enable Debugging

In this example we will enable the debugging using the Data Protector GUI

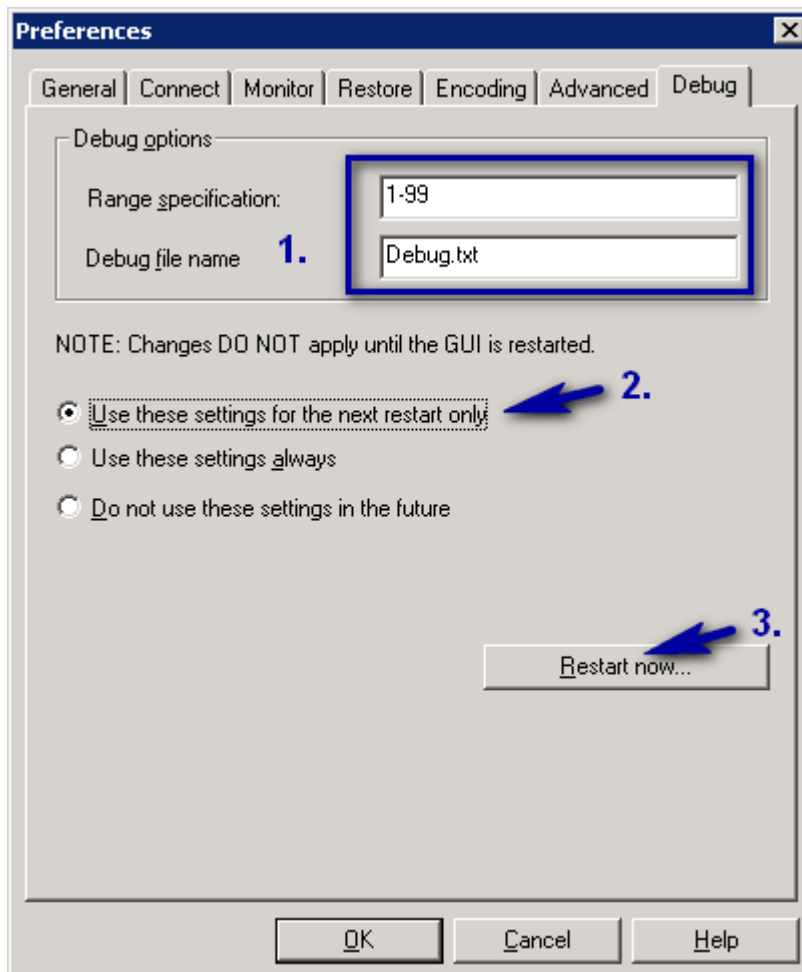
First open the Data Protector GUI and go to File → Preferences, ...

**Figure 5:** Choose Preferences...



In the Preferences pop-up choose the 'Debug' tab and specify the, by HP Data Protector Support given, Debug options (1.), choose 'Use these settings for the next restart only' (2.) and push the 'Restart now...' button (3.).

Figure 6: Specify Debug Options – Windows GUI



### Run Debug Session

Choose your backup specification, which fails or gives the error and run this backup specification.

At the beginning of the session messages you will see the Session ID (from HP Data Protector 6.1 on).

```
[Normal] From: BSM@test.test.test "TEST" Time: 2/25/2009 11:32:51 AM
Backup session 2009/02/25-1 started.
...
```

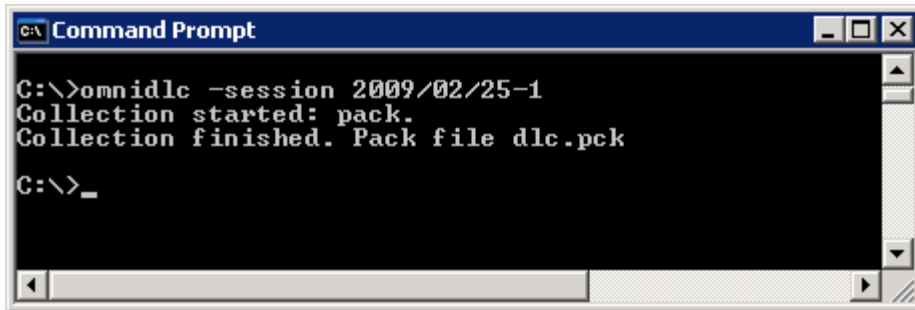
Remember the session ID to be able to collect only the debugs of the just run session.

### Collect Debugs

After the session is completed or failed, collect the debugs for this session with the `omnidlc` command. In this use case we use the option `-session`, which will collect only the debug information of the specified session. The `omnidlc` command needs to be executed on the Data Protector Cell Manager.

```
omnidlc -session 2009/02/25-1
```

Figure 7: omnidlc output



```
C:\>omnidlc -session 2009/02/25-1
Collection started: pack.
Collection finished. Pack file dlc.pck

C:\>_
```

### Send package to support

In the directory (here C:\) from where you executed the omnidlc you will find the dlc.pck file.

```
:>dir

...
04/28/2008 03:59 PM <DIR> dir_test1
02/25/2009 11:40 AM          5,281,291 dlc.pck
...
```

Send this file to the HP Data Protector Support Engineer who requested the debug information.

**Note:** It is important to transfer the .pck file in binary mode when using ftp, otherwise the file could get corrupted.

### Fix problem

Solve the problem with the proposed solution from HP Data Protector support (e.g. remove test binary, install SSP, etc.)

### Cleanup Systems

In order to delete the debug information from every client which was involved, you do not need to do this by your own, you can just use the omnidlc command with the -delete\_dbg option.

```
omnidlc -session 2009/02/25-1 -delete_dbg
```

```
C:\>omnidlc -session 2009/02/25-1 -delete_dbg
Deletion started.
Deletion finished.
```

## Using the dlc.pck file

In order to use the package file the support engineer will use the `-unpack` option of the `omnidlc` command.

The `omnidlc -unpack` creates the `dlc` directory in the current directory, and unpacks the contents of the `dlc.pck` to the `dlc` directory.

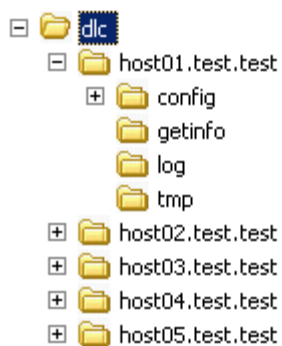
```
C:\>omnidlc -unpack
Unpacking dlc.pck
Processing: dlc/test.test.test/log/cons_ftsstats.log.gz
...
Processing: dlc/test.test.test/config/client/omni_info.gz
Processing: dlc/test.test.test/tmp/tpc077.smodomain.test_2009-02-25-
1_session_report.txt.gz
Unpack finished!
```

Inside the `dlc` directory exists one directory for each host for which information was collected.

The host directory contains the following directories:

- config
- getinfo
- log
- tmp

**Figure 8:** directory structure of unpacked `dlc.pck` file



You may use the `omnidlc` binary on any system to unpack the `dlc.pck` file.

## Conclusions

With the `omnidlc` tool you are able to collect and pack debug information and additional configuration data from several hosts with only one command.

You do not need to take care of the different installed operating systems on the different hosts in your Data Protector environment, different OS have different log directories.

Using the `omnidlc` command will give you the single point for collecting the logs, debugs and information of your whole environment.

The generated format can be used by every HP Data Protector support engineer, so there is no need to format or convert the files.

You do not need to grep the data by your own, this will save your time.

*Use one command – avoid problems – save time!*

## Limitations

- The `omnidlc` is a command line only tool. There is no GUI available to collect the debug files.
- `omnidlc` creates his own format, which cannot be unpacked by other packing utilities (WinZip, WinRAR, 7-Zip, etc.). The unpacking needs to be done by `omnidlc -unpack`.
- The `omnidlc` command cannot be used to collect the Data Protector installation execution traces. On how to create and collect these, see the HP Data Protector installation and licensing guide.
- The `omnidlc` command can only be run on Cell Managers.
- In a MoM environment, you can only collect data for each Data Protector cell separately by running the command from the respective Cell Manager.
- When a debug and logfile collector is used on HP OpenVMS, the following applies:
  - The OpenVMS ODS-2 disk structure file name can contain the maximum of 39 characters.
  - As OpenVMS systems do not have the `get_info` utility, the `get_info.out` file is blank and is not collected.
  - The `omnidlc` command run with the `-session` option does not collect the debug files produced during specified session, because session names are not part of the OpenVMS debug filename. Instead, all available logs are collected.
- The Data Protector GUI debug files for systems other than Cell Manager can only be gathered using the `-hosts` option.
- To collect debug files in a cluster, the command must be run using the `-hosts` option; the cluster nodes hostnames must be specified as the argument for the option. In a cluster, if the `-hosts` option is not specified, the data is collected from the active node.
- Absolute Path specified in postfix are not allowed.

# Appendix

## HP Data Protector 6.0 differences

As Data Protector is always improved some of the in this document described options of the `omnidlc` command are not available in HP Data Protector 6.0.

Unsupported options in Data Protector 6.0 are:

`-no_config`

Excludes the configuration information from the selection.

`-add_info path`

Includes the additional information (for example, screenshots, pictures and the like) from a directory on client identified by *path*.

The `-any` option is used when the directory path is the same for all clients. It is important to make sure the path is not host-specific before using this option.

`[-hosts list] -del_ctracelog`

Deletes `ctrace.log` files containing the information where (on which clients) debug logs are generated and which debug prefixes are used. If the `-hosts list` option is specified, the command deletes `ctrace.log` files on specified clients only. Otherwise, `ctrace.log` files on all clients in a cell are deleted.

## For more information

### Visit

Visit the following Data Protector online resources to get more information:

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

[www.hp.com/go/imhub/dataprotector](http://www.hp.com/go/imhub/dataprotector)

### HP Data Protector guides

To get more information on this topic consider the following Data Protector guides, available in the `/docs` directory on the HP Data Protector DVD, on your Installation Server or online.

- HP Data Protector A.06.10 Command line interface reference
- HP Data Protector A.06.10 Troubleshooting guide

## Technology for better business outcomes

© Copyright 2009 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. 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.

4AA2-4886ENA, May 2009

