

HCP tools コマンドリファレンス

目次

| | | |
|--------|----------------------|----|
| 1 | コマンドリファレンス | 15 |
| 1.1 | hcpd | 15 |
| 1.1.1 | f, foreground | 15 |
| 1.1.2 | V, version | 16 |
| 1.1.3 | h, help | 16 |
| 1.1.4 | t, configtest | 16 |
| 1.1.5 | q, quit | 19 |
| 1.1.6 | c, configfile | 19 |
| 1.1.7 | l, logfile | 19 |
| 1.1.8 | L, statlogfile | 20 |
| 1.1.9 | p, pidfile | 21 |
| 1.1.10 | k, license | 21 |
| 1.1.11 | investigation | 22 |
| 1.2 | hcpd_winserv | 22 |
| 1.2.1 | バージョン | 22 |
| 1.2.2 | 設定ファイル | 22 |
| 1.2.3 | ログファイル | 23 |
| 1.2.4 | 統計ログ | 23 |
| 1.2.5 | ライセンス | 23 |
| 1.3 | hcp | 23 |
| 1.3.1 | p, permission | 25 |
| 1.3.2 | R, recursive | 25 |
| 1.3.3 | Y, anydirs | 25 |
| 1.3.4 | g, regex | 26 |

| | | |
|--------|------------------------|----|
| 1.3.5 | y, verify | 26 |
| 1.3.6 | z, compress | 27 |
| 1.3.7 | s, copylinkfile | 27 |
| 1.3.8 | S, followlinkdir | 27 |
| 1.3.9 | e, noemptyfile | 28 |
| 1.3.10 | E, noemptydir | 28 |
| 1.3.11 | d, nocopydotfile | 28 |
| 1.3.12 | D, nocopydotdir | 28 |
| 1.3.13 | l, copyhidden | 29 |
| 1.3.14 | A, archivecheck | 29 |
| 1.3.15 | r, resume | 29 |
| 1.3.16 | N, nosend | 30 |
| 1.3.17 | n, nodiskio | 30 |
| 1.3.18 | U, udp | 31 |
| 1.3.19 | v, verbose | 32 |
| 1.3.20 | V, version | 34 |
| 1.3.21 | t, configtest | 35 |
| 1.3.22 | h, help | 37 |
| 1.3.23 | m, copymode | 37 |
| 1.3.24 | o, overwrite | 38 |
| 1.3.25 | a, failaction | 39 |
| 1.3.26 | u, user | 39 |
| 1.3.27 | w, password | 40 |
| 1.3.28 | H, remotehost | 40 |
| 1.3.29 | P, remoteport | 40 |

| | | |
|--------|---------------------|----|
| 1.3.30 | f, sourcefile | 41 |
| 1.3.31 | c, configfile..... | 41 |
| 1.3.32 | l, logfile..... | 42 |
| 1.3.33 | L, statlogfile..... | 42 |
| 1.3.34 | investigation..... | 43 |
| 1.3.35 | multi-run..... | 43 |
| 1.4 | hrm..... | 44 |
| 1.4.1 | f, force..... | 45 |
| 1.4.2 | R, recursive..... | 46 |
| 1.4.3 | d, dir | 46 |
| 1.4.4 | i | 46 |
| 1.4.5 | I | 47 |
| 1.4.6 | v, verbose | 47 |
| 1.4.7 | n, nodiskio..... | 47 |
| 1.4.8 | U, udp..... | 48 |
| 1.4.9 | V, version..... | 48 |
| 1.4.10 | t, configtest..... | 48 |
| 1.4.11 | h, help | 50 |
| 1.4.12 | u, user | 50 |
| 1.4.13 | w, password | 50 |
| 1.4.14 | H, remotehost..... | 50 |
| 1.4.15 | P, remoteport | 51 |
| 1.4.16 | c, configfile..... | 51 |
| 1.4.17 | l, logfile..... | 51 |
| 1.4.18 | L, statlogfile..... | 51 |

| | | |
|--------|-----------------------|----|
| 1.4.19 | investigation..... | 51 |
| 1.4.20 | multi-run..... | 51 |
| 1.5 | hcp-ls..... | 52 |
| 1.5.1 | q, querycmdname | 52 |
| 1.5.2 | o, cmdoptions | 53 |
| 1.5.3 | v, verbose | 53 |
| 1.5.4 | U, udp..... | 53 |
| 1.5.5 | V, version..... | 54 |
| 1.5.6 | t, configtest | 54 |
| 1.5.7 | h, help | 56 |
| 1.5.8 | u, user | 56 |
| 1.5.9 | w, password | 56 |
| 1.5.10 | H, remotehost..... | 56 |
| 1.5.11 | P, remoteport | 57 |
| 1.5.12 | c, configfile..... | 57 |
| 1.5.13 | l, logfile..... | 57 |
| 1.5.14 | L, statlogfile..... | 57 |
| 1.5.15 | investigation..... | 57 |
| 1.5.16 | multi-run..... | 57 |
| 1.6 | hmkdir..... | 57 |
| 1.6.1 | p, parents..... | 58 |
| 1.6.2 | v, verbose | 58 |
| 1.6.3 | U, udp..... | 59 |
| 1.6.4 | V, version..... | 59 |
| 1.6.5 | t, configtest | 59 |

| | | |
|--------|----------------------|----|
| 1.6.6 | h, help | 61 |
| 1.6.7 | u, user | 61 |
| 1.6.8 | w, password | 61 |
| 1.6.9 | H, remotehost | 62 |
| 1.6.10 | P, remoteport | 62 |
| 1.6.11 | c, configfile | 62 |
| 1.6.12 | l, logfile | 62 |
| 1.6.13 | L, statlogfile | 63 |
| 1.6.14 | investigation | 63 |
| 1.6.15 | multi-run | 63 |
| 1.7 | hpwd | 63 |
| 1.7.1 | I, logical | 63 |
| 1.7.2 | P, physical | 64 |
| 1.7.3 | U, udp | 64 |
| 1.7.4 | V, version | 64 |
| 1.7.5 | t, configtest | 64 |
| 1.7.6 | h, help | 66 |
| 1.7.7 | u, user | 67 |
| 1.7.8 | w, password | 67 |
| 1.7.9 | c, configfile | 67 |
| 1.7.10 | l, logfile | 67 |
| 1.7.11 | L, statlogfile | 67 |
| 1.7.12 | investigation | 67 |
| 1.7.13 | multi-run | 67 |
| 1.8 | hmv | 67 |

| | | |
|--------|-------------------------|----|
| 1.8.1 | f, force..... | 68 |
| 1.8.2 | i, interactive | 68 |
| 1.8.3 | N, no-overwrite..... | 69 |
| 1.8.4 | v, verbose | 69 |
| 1.8.5 | U, udp..... | 69 |
| 1.8.6 | V, version..... | 70 |
| 1.8.7 | t, configtest..... | 70 |
| 1.8.8 | h, help | 72 |
| 1.8.9 | u, user | 72 |
| 1.8.10 | w, password | 72 |
| 1.8.11 | H, remotehost..... | 72 |
| 1.8.12 | P, remoteport | 73 |
| 1.8.13 | c, configfile..... | 73 |
| 1.8.14 | l, logfile..... | 73 |
| 1.8.15 | L, statlogfile..... | 73 |
| 1.8.16 | investigation..... | 73 |
| 1.8.17 | multi-run..... | 73 |
| 1.9 | hln..... | 73 |
| 1.9.1 | f, force..... | 75 |
| 1.9.2 | i, interactive | 75 |
| 1.9.3 | N, no-dereference | 75 |
| 1.9.4 | s, symbolic | 75 |
| 1.9.5 | I, logical..... | 76 |
| 1.9.6 | B, physical..... | 76 |
| 1.9.7 | v, verbose | 76 |

| | | |
|--------|---------------------------------------|----|
| 1.9.8 | U, udp..... | 77 |
| 1.9.9 | V, version..... | 77 |
| 1.9.10 | t, configtest..... | 77 |
| 1.9.11 | h, help | 79 |
| 1.9.12 | u, user | 79 |
| 1.9.13 | w, password | 79 |
| 1.9.14 | H, remotehost..... | 80 |
| 1.9.15 | P, remoteport | 80 |
| 1.9.16 | c, configfile..... | 80 |
| 1.9.17 | l, logfile..... | 80 |
| 1.9.18 | L, statlogfile..... | 81 |
| 1.9.19 | investigation..... | 81 |
| 1.9.20 | multi-run..... | 81 |
| 2 | 設定リファレンス | 81 |
| 2.1 | hcpd.conf | 81 |
| 2.1.1 | TCPListenAddress | 84 |
| 2.1.2 | UDPListenAddress..... | 85 |
| 2.1.3 | UDPServiceExtensionBufferSize | 86 |
| 2.1.4 | UseServerCertificateSecurity | 87 |
| 2.1.5 | RequireServerCertificateSecurity..... | 87 |
| 2.1.6 | ServerKeyFile | 87 |
| 2.1.7 | ServerCertificateFile | 88 |
| 2.1.8 | ServerCertificateChainFile | 89 |
| 2.1.9 | LocalPasswordAuthentication..... | 89 |
| 2.1.10 | PAMAuthentication..... | 90 |

| | | |
|--------|--|-----|
| 2.1.11 | PubkeyAuthentication | 90 |
| 2.1.12 | WinLogonUserAuthentication | 90 |
| 2.1.13 | PerformSystemAuthenticationRegardlessUsers | 91 |
| 2.1.14 | UserDirectoryFallbackAvailable | 91 |
| 2.1.15 | RejectOnUserHomeDirectoryNotFound | 92 |
| 2.1.16 | UsePrivilegeSeparation | 92 |
| 2.1.17 | PrivilegeSeparationMinimumUID | 93 |
| 2.1.18 | PrivilegeSeparationMinimumGID | 93 |
| 2.1.19 | PrivilegeSeparationUser | 94 |
| 2.1.20 | ApplyUserPermission | 94 |
| 2.1.21 | AuthorizedKeySearchDir | 95 |
| 2.1.22 | AuthorizedKeyFile | 95 |
| 2.1.23 | CACertificateFile | 96 |
| 2.1.24 | CACertificatePath | 96 |
| 2.1.25 | CARevocationFile | 97 |
| 2.1.26 | CARevocationPath | 97 |
| 2.1.27 | OCSPRevocationEnabled | 97 |
| 2.1.28 | LocalUserFile | 98 |
| 2.1.29 | LocalPasswordFile | 98 |
| 2.1.30 | AcceptableCryptMethod | 99 |
| 2.1.31 | AcceptableDigestMethod | 99 |
| 2.1.32 | RequireDataIntegrityChecking | 100 |
| 2.1.33 | TransportCharEncoding | 101 |
| 2.1.34 | HostEncoding | 101 |
| 2.1.35 | HeaderCompress (予約) | 102 |

| | | |
|--------|-----------------------------------|-----|
| 2.1.36 | ContentCompress (予約) | 102 |
| 2.1.37 | MaxConcurrentThread | 102 |
| 2.1.38 | MaxTotalConnection | 102 |
| 2.1.39 | MaxTcpConnection | 102 |
| 2.1.40 | MaxUdpConnection..... | 103 |
| 2.1.41 | MaxConnectionPerUser..... | 103 |
| 2.1.42 | MaxConnectionPerSec | 103 |
| 2.1.43 | MaxReceiveFileSize..... | 104 |
| 2.1.44 | MaxSendFileSize | 104 |
| 2.1.45 | MaxRequestFileEntryAtOnce | 105 |
| 2.1.46 | MaxTotalBufferSize..... | 105 |
| 2.1.47 | MaxBufferSizePerConnection | 105 |
| 2.1.48 | MaxTotalReceiveRate..... | 106 |
| 2.1.49 | MaxTotalSendRate | 106 |
| 2.1.50 | MaxReceiveRatePerConnection | 107 |
| 2.1.51 | MaxSendRatePerConnection..... | 107 |
| 2.1.52 | InitHeaderBlockSize | 107 |
| 2.1.53 | InitContentBlockSize | 108 |
| 2.1.54 | MaxHeaderBlockSize..... | 108 |
| 2.1.55 | MaxContentBlockSize | 109 |
| 2.1.56 | FileLock | 109 |
| 2.1.57 | FileLockRetries | 110 |
| 2.1.58 | FileLockRetryInterval | 110 |
| 2.1.59 | TransportTimeout..... | 111 |
| 2.1.60 | IdleTimeout..... | 111 |

| | | |
|--------|---|-----|
| 2.1.61 | DocPoint..... | 111 |
| 2.1.62 | DocPath..... | 112 |
| 2.1.63 | PermitAccessRead..... | 112 |
| 2.1.64 | PermitAccessWrite | 113 |
| 2.1.65 | PermitAccessOverwrite | 113 |
| 2.1.66 | PermitAccessDelete | 114 |
| 2.1.67 | PermitAccessRandomRead (予約) | 114 |
| 2.1.68 | PermitAccessRandomWrite (予約) | 115 |
| 2.1.69 | AccessList | 115 |
| 2.1.70 | Allow..... | 115 |
| 2.1.71 | Deny..... | 116 |
| 2.1.72 | SyslogOption..... | 117 |
| 2.1.73 | SyslogFacility..... | 117 |
| 2.1.74 | SystemLog | 118 |
| 2.1.75 | SystemLogLevel..... | 120 |
| 2.1.76 | ApplicationStatLog..... | 121 |
| 2.1.77 | TransportStatLog..... | 122 |
| 2.1.78 | SystemStatLog..... | 124 |
| 2.1.79 | FileOperationLog..... | 125 |
| 2.1.80 | CallbackScript | 129 |
| 2.1.81 | EnsureDestinationInFileTransfer | 133 |
| 2.1.82 | StatLogPerUserInPrivilegeSeparation | 133 |
| 2.2 | クライアントコマンド共通設定項目 | 134 |
| 2.2.1 | RequireServerCertificateSecurity..... | 135 |
| 2.2.2 | RejectFallbackServerCertificateSecurity | 136 |

| | | |
|--------|--|-----|
| 2.2.3 | IgnoreCertificateCNInvalid | 136 |
| 2.2.4 | IgnoreCertificateDateInvalid..... | 137 |
| 2.2.5 | IgnoreUnknownCA..... | 137 |
| 2.2.6 | IgnoreRevocation..... | 137 |
| 2.2.7 | CACertificateFile..... | 138 |
| 2.2.8 | CACertificatePath (予約) | 138 |
| 2.2.9 | CARevocationFile..... | 139 |
| 2.2.10 | CARevocationPath (予約) | 139 |
| 2.2.11 | OCSPRevocationEnabled | 140 |
| 2.2.12 | StrictHostKeyChecking..... | 140 |
| 2.2.13 | LocalPasswordAuthentication..... | 141 |
| 2.2.14 | PAMAuthentication..... | 141 |
| 2.2.15 | PubkeyAuthentication | 141 |
| 2.2.16 | WinLogonUserAuthentication..... | 142 |
| 2.2.17 | PrivateKeySearchDir | 142 |
| 2.2.18 | PrivateKeyFile..... | 143 |
| 2.2.19 | AcceptableCryptMethod | 144 |
| 2.2.20 | AcceptableDigestMethod..... | 144 |
| 2.2.21 | DisableDataIntegrityChecking..... | 145 |
| 2.2.22 | AcceptDataIntegrityCheckingOnRejection | 145 |
| 2.2.23 | TransportCharEncoding | 146 |
| 2.2.24 | HostEncoding..... | 147 |
| 2.2.25 | CompressLevel | 147 |
| 2.2.26 | HeaderCompress..... | 147 |
| 2.2.27 | ContentCompress | 148 |

| | | |
|--------|---------------------------------------|-----|
| 2.2.28 | MaxConcurrentThread | 148 |
| 2.2.29 | MaxReceiveFileSize..... | 148 |
| 2.2.30 | MaxSendFileSize | 149 |
| 2.2.31 | MaxRequestFileEntryAtOnce | 149 |
| 2.2.32 | MaxBufferSize | 150 |
| 2.2.33 | MaxReceiveRate | 150 |
| 2.2.34 | MaxSendRate..... | 150 |
| 2.2.35 | InitHeaderBlockSize | 151 |
| 2.2.36 | InitContentBlockSize..... | 151 |
| 2.2.37 | MaxHeaderBlockSize..... | 152 |
| 2.2.38 | MaxContentBlockSize | 152 |
| 2.2.39 | FileLock | 152 |
| 2.2.40 | FileLockRetries..... | 153 |
| 2.2.41 | FileLockRetryInterval | 153 |
| 2.2.42 | TransportTimeout..... | 153 |
| 2.2.43 | ApplicationLog..... | 153 |
| 2.2.44 | ApplicationLogLevel..... | 154 |
| 2.2.45 | ApplicationStatLog..... | 154 |
| 2.2.46 | TransportStatLog..... | 155 |
| 2.2.47 | UDPTransportExtensionBufferSize | 156 |
| 2.3 | hcp.conf..... | 157 |
| 2.3.1 | Include..... | 157 |
| 2.4 | hrm.conf..... | 158 |
| 2.5 | hcp-ls.conf..... | 158 |
| 2.6 | hmkdir.conf | 158 |

| | | |
|-------|-----------------------|-----|
| 2.7 | hpwd.conf..... | 158 |
| 2.8 | hmv.conf | 158 |
| 2.9 | hln.conf..... | 158 |
| 2.10 | users..... | 158 |
| 2.11 | passwd..... | 159 |
| 3 | 統計リファレンス | 160 |
| 3.1 | アプリケーション統計 | 160 |
| 3.2 | トランスポート統計 | 163 |
| 3.2.1 | TCP 統計..... | 163 |
| 3.2.2 | HpFP 統計..... | 165 |
| 3.3 | システム統計..... | 167 |
| 4 | エラーコードリファレンス | 169 |
| 4.1 | コマンド終了ステータス | 169 |
| 4.2 | 設定ファイルエラーコード | 170 |
| 4.3 | ファイル処理理由コード..... | 171 |
| 5 | コマンド実行モード | 174 |
| 5.1 | 単一起動モード..... | 174 |
| 5.2 | 多重起動モード..... | 175 |
| 6 | HCP tools 制約事項一覧..... | 177 |
| 7 | 改訂履歴 | 178 |

1 コマンドリファレンス

1.1 hcpd

Usage: hcpd [OPTION]...
or : hcpd -q

hcpd デーモンは、リモート（クライアント）からの hcp（ファイル転送）コマンドや hrm（ファイル削除）コマンドの指示を受け付け、その処理を実行するためにサーバ側で待機するソフトウェアです。Linux 版で提供されます。

| 短縮名 | オプション名 | 概略 |
|-----|---------------|----------------------------|
| f | foreground | フォアグラウンド実行 |
| V | version | バージョンを表示 |
| h | help | ヘルプを表示 |
| t | configtest | 入力パラメータ及び設定ファイルのロード結果を表示する |
| q | quit | 停止シグナルを送信する |
| c | configfile | 設定ファイル |
| l | logfile | ログファイル |
| L | statlogfile | 統計ログファイル |
| p | pidfile | PID ファイル |
| k | license | ライセンスキーファイル |
| | investigation | 調査モード |

1.1.1 f, foreground

=====

対応 OS : Linux

書式 : -f | --foreground

=====

hcpd デーモンを、フォアグラウンドモードで起動します。

--

例 :

[root@localhost ~]# hcpd -f ...

--

1.1.2 V, version

```
=====
対応 OS : Linux
書式 : -V | --version
=====
```

hcpd デーモンのバージョンを表示します。

```
--
例 :
[root@localhost ~]# hcpd -V
hcp server (hcpd) 1.2.4b_30 / Linux (HpFP2 2.0.0.85)
--
```

1.1.3 h, help

```
=====
対応 OS : Linux
書式 : -h | --help
=====
```

hcpd デーモンのヘルプを表示します。

```
--
例 :
[root@localhost ~]# hcpd -h
--
```

1.1.4 t, configtest

```
=====
対応 OS : Linux
書式 : -t | --configtest
=====
```

hcpd デーモンの入力パラメータ及び設定情報を出力します。

```
--
例 :
[root@localhost ~]# hcpd -t
...

Number of Logical Processors : 4
Number of Physical Processors : 4

Command parameters
foreground: disable
version   : disable
```



```
verbose    : disable
help       : disable
configtest: enable
quit       : disable
configfile: disable [/etc/hcp/hcpd.conf] default
logfile    : disable [/var/log/hcpd.log] default
statlogfile : disable [/var/tmp/.hcp.statistics] default
pidfile    : disable [/var/run/hcpd.pid] default
license    : disable [/etc/hcp/license.key] default
```

Configuration parameters

```
WinLogonUserAuthentication      : yes
PubkeyAuthentication           : no
PAMAuthentication              : yes
LocalPasswordAuthentication    : no
PerformSystemAuthenticationRegardlessUsers : no
UserDirectoryFallbackAvailable : no
RejectOnUserHomeDirectoryNotFound : no
UsePrivilegeSeparation         : yes
PrivilegeSeparationMinimumUID  : 1
PrivilegeSeparationMinimumGID  : 1
PrivilegeSeparationUser        : nobody
ApplyUserPermission            : yes
UseServerCertificateSecurity    : yes
RequireServerCertificateSecurity : yes
HeaderCompress                 : yes
ContentCompress                : yes
OCSPRevocationEnabled          : yes
AuthorizedKeySearchDir         : - [/etc/hcp/authkeys]
AuthorizedKeyFile               : - [~/.hcp/authorized_keys]
LocalUserFile                  : - [/etc/hcp/users]
LocalPasswordFile              : - [/etc/hcp/passwd]
ServerKeyFile                  : - [/etc/hcp/key/server.key]
ServerCertificateFile           : - [/etc/hcp/cert/server.crt]
ServerCertificateChainFile      : - [/etc/hcp/cert/chain.crt]
CACertificateFile               : - [/etc/hcp/cacert.pem]
CACertificatePath               : - [/etc/ssl]
CARevocationFile               : - [/etc/hcp/crl.pem]
CARevocationPath               : - [/etc/ssl]
ProtocolVersion                 : 2
MaxConcurrentThread             : 0
MaxTotalConnection             : 10
MaxTcpConnection               : 5
MaxUdpConnection               : 5
MaxConnectionPerUser           : 1
MaxConnectionPerSec             : 1
MaxRequestFileEntryAtOnce      : 50
```

HCP tools コマンドリファレンス

```

MaxReceiveFileSize           : 107374182400
MaxSendFileSize              : 107374182400
MaxTotalBufferSize          : 2147483648
MaxBufferSizePerConnection  : 104857600
MaxTotalReceiveRate          : 100000000000
MaxTotalSendRate             : 100000000000
MaxReceiveRatePerConnection : 100000000000
MaxSendRatePerConnection    : 100000000000
InitHeaderBlockSize         : 51200
InitContentBlockSize        : 51200
MaxHeaderBlockSize          : 51200
MaxContentBlockSize         : 51200
TransportTimeout             : 180
IdleTimeout                  : 600
FileLock                     : yes
FileLockRetries              : 0
FileLockRetryInterval        : 3
TCPListenAddress             : 0.0.0.0:874
UDPListenAddress             : 0.0.0.0:884[HpFP port=65520, sndbuf=1
04857600, rcvbuf=209715200, mss=-1]
UDPServiceExtensionBufferSize : 2147483648
AcceptableCryptMethod        : AES256/CBC AES128/CBC
AcceptableDigestMethod       : SHA256 SHA160
RequireDataIntegrityChecking : yes
TransportCharEncoding         : UTF8
DocPoint                     : /home
  DocPath = /home
  PermitAccessRead = yes
  PermitAccessWrite = yes
  PermitAccessOverwrite = yes
  PermitAccessDelete = yes
  PermitAccessRandomRead = no
  PermitAccessRandomWrite = no
DocPointEnd
HostEncoding                  : UTF8
SyslogOption                  : LOG_CONS | LOG_PID
SyslogFacility                : LOG_DAEMON
SystemLogLevel                : INFO Rotation[size=no:0:0, pattern=n
o:]
ApplicationStatLog            : no Rotation[size=no:0:0, pattern=no:]
TransportStatLog              : no Rotation[size=no:0:0, pattern=no:]
SystemStatLog                 : yes Rotation[size=no:0:0, pattern=n
o:]
FileOperationLog              : no -[/var/log/hcpd.file.operation.lo
g] Rotation[size=no:0:0, pattern=no:]
CallbackScript                 : no Script[~/hcp/callback.sh, data=~/
hcp/callback]

```

```
EnsureDestinationInFileTransfer : yes
StatLogPerUserInPrivilegeSeparation : no
```

--

1.1.5 q, quit

=====
対応 OS : Linux

書式 : -q | --quit
=====

hcpd デーモンを停止するシグナルを送信します。

pidfile オプションで指定されたファイルもしくは既定のファイルからプロセス ID を取得し、このプロセス ID を指定してシグナルを送信します。

--

例 :

```
[root@localhost ~]# hcpd -q ...
```

--

1.1.6 c, configfile

=====
対応 OS : Linux

書式 : -c <config-file-path> | --configfile=<config-file-path>

config-file-path

既定値 : なし

値の範囲 : ファイルシステムのパス文字列
=====

hcpd デーモンが使用する設定ファイルのパスを指定します。

--

例 :

```
[root@localhost ~]# hcpd -c /etc/hcp/hcpd2.conf ...
```

--

このオプションが指定されなかった場合は、次のファイルから設定を読み込みます。

/etc/hcp/hcpd.conf

1.1.7 l, logfile

=====
対応 OS : Linux

書式 : -l <log-file-path> | --logfile=<log-file-path>

log-file-path

既定値 : なし

値の範囲 : ファイルシステムのパス文字列

=====

hcpd デーモンのログを指定したパスのファイルに出力します。

--

例 :

```
[root@localhost ~]# hcpd -l /var/log/hcpd.log ...
```

--

このオプションが指定されなかった場合は、標準出力にログを出力します。

1.1.8 L, statlogfile

=====

対応 OS : Linux

書式 : -L <log-file-path> | --statlogfile=<log-file-path>

log-file-path

既定値 : なし

値の範囲 : ファイルシステムのパス文字列

=====

hcpd デーモンが出力する統計ログの基準パスを指定します。

--

例 :

```
[root@localhost ~]# hcpd -L /var/tmp/.hcp.statistics2 ...
```

--

指定されたパスに接尾辞を付加して各統計ログが出力されます。

<指定されたパス>.system (システム統計)

<指定されたパス>.application (アプリケーション統計)

<指定されたパス>.transport.tcp.service_<サービス番号>.<サービスポート番号>.thead_<スレッド番号> (TCP トランスポート統計)

<指定されたパス>.transport.hpfp.service_<サービス番号>.<サービスポート番号>.thead_<スレッド番号> (HpFP トランスポート統計)

非特権ユーザで後述の StatLogPerUserInPrivilegeSeparation が有効(yes)の場合は、次の様なパスに記録されます。

<指定されたパス>.application.<UID>_<GID>

<指定されたパス>.application.<Username> (Windows)

指定しない場合は、次のパスに出力します。

`/var/tmp/.hcp.statistics`

1.1.9 p, pidfile

=====
対応 OS : Linux

書式 : `-p <pid-file-path> | --pidfile=<pid-file-path>`

pid-file-path

既定値 : `/var/run/hcpd.pid`

値の範囲 : ファイルシステムのパス文字列
=====

hcpd デーモンプロセスのプロセス ID を出力するファイルパスを指定します。

--

例 :

`[root@localhost ~]# hcpd -p /var/run/hcpd2.pid ...`

--

複数のデーモンプロセスを起動する場合もしくは既に他のデーモンプロセスが起動している場合は、同じファイルパスを使用しないようにご注意ください。

1.1.10 k, license

=====
対応 OS : Linux

書式 : `-k <license-key-path> | --license=<license-key-path>`

license-key-path

既定値 : `/etc/hcp/license.key`

値の範囲 : ファイルシステムのパス文字列
=====

ライセンスキーのパスを指定します。

--

例 :

`[root@localhost ~]# hcpd -k /etc/hcp/license2.key ...`

--

指定しない場合は、次のパスからライセンスキーを読み込みます。

`/etc/hcp/license.key`

ライセンスキーが読み込めない場合は、トライアルライセンスが次の要件で適用されます。

合計スループット制限 1Gbps

スループット制限 1Gbps

TCP 最大接続数 3

HpFP 最大接続数 3

1.1.11 investigation

=====
対応 OS : Linux

書式 : --investigation
=====

hcpd デーモンを調査モードで起動します。期待しない動作をする場合などに、調査のために詳細なログを出力します。多量のログが出力されるため、長時間の使用もしくは性能が必要とされるケースでは使用しません。また、取得したログはユーザが可読な内容ではないため、本製品提供元などへ調査目的等のため送付してください。

1.2 hcpd_winserv

hcpd_winserv は、リモート（クライアント）からの hcp（ファイル転送）コマンドや hrm（ファイル削除）コマンドの指示を受け付け、その処理を実行するためにサーバ側で待機するソフトウェアです。Windows 版でサービスプログラムとして提供されます。また、本プログラムはコマンドプロンプトで実行する機能は提供しません。

1.2.1 バージョン

hcpd_winserv のバージョンは、後述のログファイルにサービス起動時に記録されます。

--

例 :

hcp server (hcpd) 1.2.4b_30 / Windows (HpFP2 2.0.0.85)

--

1.2.2 設定ファイル

hcpd_winserv は、設定を次のファイルから読み込みます。

C:/ProgramData/Clealink/HCP Tools/hcpd.conf

1.2.3 ログファイル

アプリケーションのログは次のファイルに書き込まれます。

C:/ProgramData/Clealink/HCP Tools/hcpd.log

1.2.4 統計ログ

Windows 版では、統計ログは次のパスに出力されます。追加される接尾辞については、前述の Linux 版と同じ文字列が使用されます。

C:/ProgramData/Clealink/HCP Tools/Temp/_hcp.statistics

1.2.5 ライセンス

次のパスからライセンスキーを読み込みます。

C:/ProgramData/Clealink/HCP Tools/license.key

ライセンスキーが読み込めない場合は、Linux 版と同じトライアルライセンスが適用されます。

1.3 hcp

Local : hcp [OPTION]... <SRC1>[<SRC2> ...] <DEST>
or : hcp [OPTION]... -f <SRC_LIST_PATH> <DEST>

Remote :
Push : hcp [OPTION]... <SRC1>[<SRC2> ...] <HOST>:<PORT>:<DEST>
Pull : hcp [OPTION]... <HOST>:<PORT>:<SRC1>[<HOST>:<PORT>:<SRC2> ...]
<DEST>

Remote using source file list :
Push : hcp [OPTION]... -f <SRC_LIST_PATH> <HOST>:<PORT>:<DEST>
Pull : hcp [OPTION]... -H <HOST> -P <PORT> -f <SRC_LIST_PATH> <DEST>

hcp コマンドは、リモート（サーバ、hcpd デーモン）との間でファイル転送を行うコマンドです。ローカルのファイルコピー機能もサポートします。

| 短縮名 | オプション名 | 概略 |
|-----|------------|-------------------------------------|
| p | permission | 転送元のパーミッションを保持する |
| R | recursive | 再帰的にコピーする |
| Y | anydirs | ディレクトリはすべて探索する（ワイルドカード、正規表現の適用から外す） |

| | | |
|---|---------------|---|
| g | regex | ファイル名の指定に正規表現を使用する（ワイルドカードを使用しない） |
| y | verify | データをダイジェストにより検証する |
| z | compress | 圧縮する |
| s | copylinkfile | シンボリックリンクを解決(Dereference)してファイルをコピーする |
| S | followlinkdir | シンボリックリンクのディレクトリを解決してファイルをコピーする |
| e | noemptyfile | 空のファイルをコピーしない |
| E | noemptydir | 空のディレクトリをコピーしない |
| d | nocopydotfile | ドットで始まるファイル名のファイルをコピーしない |
| D | nocopydotdir | ドットで始まるディレクトリ名のディレクトリをコピーしない |
| I | copyhidden | 隠し属性ファイルをコピーする(Windows) |
| A | archivecheck | アーカイブ属性が設定されているファイルをコピーする(Windows) |
| r | resume | 再開処理を行う（前回の実行記録が必要） |
| N | nosend | トランスポート上で送受信はせずディスク I/O 性能を計測する |
| n | nodiskio | ディスク I/O 操作を行わず通信プロトコルのネットワーク I/O 性能を計測する |
| U | udp | UDP (HpFP)プロトコルを使用する |
| v | verbose | 実行記録を標準出力に出力する |
| V | version | バージョンを表示する |
| t | configtest | 入力パラメータ及び設定ファイルのロード結果を表示する |
| h | help | コマンドラインヘルプを表示する |
| m | copymode | コピーモード |
| o | overwrite | 上書きモード |
| a | failaction | 処理エラー時動作指定 |
| u | user | ユーザ名 |
| w | password | パスワード |

| | | |
|---|---------------|-----------------------|
| H | remotehost | リモートホスト指定（ソースファイル使用時） |
| P | remoteport | リモートポート指定（ソースファイル使用時） |
| f | sourcefile | ソースファイルリスト |
| c | configfile | 設定ファイル |
| l | logfile | ログファイル |
| L | statlogfile | 統計ログファイル |
| | investigation | 調査モード |
| | multi-run | 多重起動モード |

1.3.1 p, permission

=====
 対応 OS : Linux / Windows

書式 : -p | --permission
 =====

転送元のファイルパーミッション（UID/GID, 実行権フラグなど）やファイル属性（更新日時など）を転送先で保持します。アクセス権などの理由で転送先でパーミッションやファイル属性を更新できない場合は、ファイル書込みを行った際に OS により適用された属性などがそのまま保持されます。

--

例 :

```
[user@localhost ~]$ hcp -p ...
```

--

1.3.2 R, recursive

=====
 対応 OS : Linux / Windows

書式 : -R | --recursive
 =====

転送元のディレクトリを再帰的に探索してファイル転送を行います。

--

例 :

```
[user@localhost ~]$ hcp -R ...
```

--

1.3.3 Y, anydirs

=====
 対応 OS : Linux / Windows

書式 : -Y | --anydirs

=====

regex オプションやワイルドカードなどによるパターンマッチングが指定された際に、ディレクトリ探索でこのパターンマッチングを無視して探索を行います。

--

例 :

```
[user@localhost ~]$ hcp -Y ...
```

--

1.3.4 g, regex

=====

対応 OS : Linux / Windows

書式 : -g | --regex

=====

転送元のパスを正規表現として解釈します。このオプションを使用しない場合は、転送元のパスはワイルドカード(*)を含むパターン文字列表現として解釈します。

正規表現（もしくはワイルドカードを含むパターン文字列）による評価は、指定した転送元のパスがディレクトリ及びファイルとしても存在しない場合に行われます。この評価は、ファイル探索時のファイル名及びディレクトリ名に対して適用されます。パターンにマッチしなかったファイルやディレクトリはスキップされます。パターン文字列は、最後のパス区切り文字（/など）以降の文字列から抽出されます。

--

例 :

```
[user@localhost ~]$ hcp -g ...
```

--

1.3.5 y, verify

=====

対応 OS : Linux / Windows

書式 : -y | --verify

=====

転送するデータブロック及びファイルに対してメッセージダイジェストの検査を行います。

--

例 :

```
[user@localhost ~]$ hcp -y ...
```

--

1.3.6 z, compress

=====
対応 OS : Linux / Windows

書式 : -z | --compress
=====

転送するデータブロックを圧縮します。

--

例 :

[user@localhost ~]\$ hcp -z ...

--

1.3.7 s, copylinkfile

=====
対応 OS : Linux / Windows

書式 : -s | --copylinkfile
=====

シンボリックリンクが参照するファイルをコピーします。このオプションを使用しない場合は、転送先でシンボリックリンクの作成を行います。

--

例 :

[user@localhost ~]\$ hcp -s ...

--

1.3.8 S, followlinkdir

=====
対応 OS : Linux / Windows

書式 : -S | --followlinkdir
=====

シンボリックリンクが参照するディレクトリを探索します。

本オプションを使用する場合は、サイクリックなディレクトリ探索やリンクを解決しない場合に確認される転送元パス配下の情報（ファイル数や容量など）からは予想しにくい多量のデータ処理などが生じ得る点に注意が必要です。

--

例 :

[user@localhost ~]\$ hcp -S ...

--

1.3.9 e, noemptyfile

```
=====
対応 OS : Linux / Windows
書式 : -e | --noemptyfile
=====
```

空ファイル（サイズが 0 のファイル）のコピーを抑制します。

```
--
例 :
[user@localhost ~]$ hcp -e ...
--
```

1.3.10 E, noemptydir

```
=====
対応 OS : Linux / Windows
書式 : -E | --noemptydir
=====
```

空ディレクトリ（配下にファイルもしくはディレクトリを全く含まないディレクトリ）のコピーを抑制します。

```
--
例 :
[user@localhost ~]$ hcp -E ...
--
```

1.3.11 d, nocopydotfile

```
=====
対応 OS : Linux / Windows
書式 : -D | --nocopydotfile
=====
```

ドット(.)から始まるファイルのコピーを抑制します。

```
--
例 :
[user@localhost ~]$ hcp -d ...
--
```

1.3.12 D, nocopydotdir

```
=====
対応 OS : Linux / Windows
書式 : -D | --nocopydotdir
=====
```

ドット(.)から始まるディレクトリのコピーを抑制します。

--

例 :

```
[user@localhost ~]$ hcp -D ...
```

--

1.3.13 I, copyhidden

=====
対応 OS : Windows

書式 : -I | --copyhidden
=====

隠し属性が設定されたファイルをコピーするか指定します。

--

例 :

```
[user@localhost ~]$ hcp -I ...
```

--

1.3.14 A, archivecheck

=====
対応 OS : Windows

書式 : -A | --archivecheck
=====

ファイルのアーカイブ属性が設定されているファイルをコピーします。また、転送先でアーカイブ属性を維持します。

--

例 :

```
[user@localhost ~]$ hcp -A ...
```

--

1.3.15 r, resume

=====
対応 OS : Linux / Windows

書式 : -r <run-record-path> | --resume=<run-record-path>

run-record-path

既定値 : なし

値の範囲 : ファイルシステムのパス文字列
=====

hcp コマンドの実行記録 (.hcp.out に出力されるファイル転送の実行記録) を使用して、コマンドの再開 (レジューム) を行います。

```
--
例 :
[user@localhost ~]$ hcp ...
[user@localhost ~]$ mv .hcp.out .hcp.in
[user@localhost ~]$ hcp -r .hcp.in ...
--
```

1.3.16 N, nosend

```
=====
対応 OS : Linux / Windows
書式 : -N | --nosend
=====
```

ネットワークへのデータ送信を抑制します。

このオプションは、転送元でのディスク I/O やデータ処理の性能を確認する目的で使用します。

```
--
例 :
[user@localhost ~]$ hcp -N ...
--
```

1.3.17 n, nodiskio

```
=====
対応 OS : Linux / Windows
書式 : -n <bench_spec> | --nodiskio=<bench_spec>
      bench_spec := <number_of_files>:<file_size>
-----
```

number_of_files

既定値 : なし

値の範囲 : 符号なし整数

file_size

既定値 : なし

値の範囲 : 符号なし倍長整数

```
=====
```

転送元及び転送先でのローカル I/O を抑制します。

このオプションは、ネットワーク通信の性能を確認する目的で使用します。オプションの各パラメータの意味は次の通りです。

number_of_files := 送信するファイル数

file_size := ファイル毎に送信するサイズ (単位 バイト)

オプションの値を“0:0”で指定した場合は、転送元のパスを通常の動作で探索してファイル転送を行います（`permission` オプションのためのファイル属性情報の参照やファイルからのデータの読み込みなどの処理は行われません）。

```
--
例：
[user@localhost ~]$ hcp -n 1000:1048576 ...
--
```

1.3.18 U, udp

```
=====
対応 OS : Linux / Windows
書式 : -U <hpfp_options> | --udp=<hpfp_options>
      hpfp_options := <hpfp_udp_port>:<hpfp_cong_mode>:<hpfp_sndbuf>:<hpfp_rcvbuf>:<hpfp_mss>
-----

hpfp_udp_port
既定値 : 65520
値の範囲 : 1 - 65535
-----

hpfp_cong_mode
既定値 : FAIR
値の範囲 : DEFAULT(D), FAIR(F), MODEST(M), FAIR_FAST_START(S), AGGRESSIVE(A)
※括弧内は省略表記
-----

hpfp_sndbuf
書式 : ( DEFAULT | D | <decimal_number>[[(T|G|M|K)]B] )
既定値 : 100MB
値の範囲 : バイト換算で符号なし倍長整数の範囲。D は DEFAULT の略記。
-----

hpfp_rcvbuf
書式 : ( DEFAULT | D | <decimal_number>[[(T|G|M|K)]B] )
既定値 : 200MB
値の範囲 : バイト換算で符号なし倍長整数の範囲。D は DEFAULT の略記。
-----

hpfp_mss
書式 : ( DEFAULT | D | NONE | N | <decimal_number>[[(T|G|M|K)]B] )
既定値 : NONE
値の範囲 : バイト換算で符号なし整数の範囲。D は DEFAULT の略記。N は NONE の略記。
=====
```

トランスポートに HpFP プロトコルを使用するように指示します。

`hpfp_udp_port` は、HpFP プロトコルで使用する UDP ポート番号を指定します。

`hpfp_cong_mode` は、輻輳制御モードを指定します。

`hpfp_sndbuf` は、HpFP プロトコル送信バッファサイズを指定します（バイト単位）。

`hpfp_rcvbuf` は、HpFP プロトコル受信バッファサイズを指定します（バイト単位）。

`hpfp_mss` は、HpFP プロトコルの MSS(Maximum Segment Size)を指定します。MTU サイズからプロトコルヘッダ 44 バイトを引いたサイズが最適値です。D を指定すると既定値を使用します。N を指定した場合、HpFP プロトコルによる MTU 探索を行い値を決定します。

輻輳制御モードについては、指定したモードをサーバが許可しない場合は、FAIR が適用されます。この動作により異なるアルゴリズムが適用される場合は、次の様なログにより通知されます。

```
2018/07/05 13:20:44 00007f9a2dfa0700:INFO :Your requested congestion mode
of HpFP will not be used (requested=AGGRESSIVE, actual=FAIR).
```

--

例 :

```
[user@localhost ~]$ hcp -U D:S:4MB:8MB:8956B ...
```

--

1.3.19 v, verbose

```
=====
対応 OS : Linux / Windows
書式 : -v | --verbose
=====
```

ファイル転送の実行記録を標準出力に出力します。本オプションで出力される情報は、`resume` オプションで必要となります。

--

例 :

```
[user@localhost ~]$ hcp -v ...
SRC /home/user/Desktop/hcp_src5
DST 127.0.0.1:11112:/home/user/Desktop/hcp_dst5
OK 0000 FS 80000001 /home/user/Desktop/hcp_src5/file1.txt
OK 0000 FT 00000001 /home/user/Desktop/hcp_src5/file1.txt
EXIT 0 REASON 0000
```

--

本オプションを指定しない場合は、実行記録は実行ディレクトリの次のファイルなどに記録されます。

.hcp.out
_hcp.out (Windows)

FT (File Transfer)は、ファイル転送処理を表します。書式は次の通りです。

<result> <reason> FT <sequence> [<src_label>]<path>

result は、処理結果(OK もしくは NG)が記録されます。

reason は、その処理結果の理由として後述のファイル処理理由コードが記録されます。

sequence は、この処理に割り当てられた番号（シーケンス番号）が記録されます。

src_label は、この処理で扱われるファイルに対応するソースのラベルが次の書式で記録されます。

SRC<src_index>

src_index は、コマンドで指定したソースの添え字が記録されます。

path は、この処理で扱われるファイルパスが記録されます。コマンドを実行しているホストのパスが記録されます。

複数のソースが指定された場合は、次の様な内容で進捗情報が記録されます。

—

例：

```
SRC0 /home/user/Desktop/hcp_src5
SRC1 /home/user/Desktop/hcp_src6
DST 127.0.0.1:11112:/home/user/Desktop/hcp_dst5
OK 0000 FS 80000001 SRC0 /home/user/Desktop/hcp_src5/file1.txt
OK 0000 FS 80000002 SRC0 /home/user/Desktop/hcp_src5/file2.txt
OK 0000 FS 80000003 SRC1 /home/user/Desktop/hcp_src6/file3.txt
OK 0000 FS 80000004 SRC1 /home/user/Desktop/hcp_src6/file4.txt
OK 0000 FT 00000001 SRC0 /home/user/Desktop/hcp_src5/file1.txt
OK 0000 FT 00000002 SRC0 /home/user/Desktop/hcp_src5/file2.txt
OK 0000 FT 00000003 SRC1 /home/user/Desktop/hcp_src6/file3.txt
OK 0000 FT 00000004 SRC1 /home/user/Desktop/hcp_src6/file4.txt
EXIT 0 REASON 0000
```

—

FS (File Sync)は、ファイル同期処理を表します。転送先のファイルが転送元に存在せず本同期処理によって削除される場合は、次の様に記録されます。

—
例：

```
SRC /home/user/Desktop/hcp_src5
DST 127.0.0.1:11112:/home/user/Desktop/hcp_dst5
OK 0000 FS 80000001 /home/user/Desktop/hcp_src5/file1.txt
OK 0000 FS 80000002 /home/user/Desktop/hcp_src5/file2.txt
OK A001 FS 80000003 /home/user/Desktop/hcp_src5/file3.txt
OK 0000 FT 00000001 /home/user/Desktop/hcp_src5/file1.txt
OK 0000 FT 00000002 /home/user/Desktop/hcp_src5/file2.txt
EXIT 0 REASON 0000
```

—

ソースが複数ある場合は、ソースラベルの添え字が“?”で表記されます。

—
例：

```
SRC0 /home/user/Desktop/hcp_src5
SRC1 /home/user/Desktop/hcp_src6
DST 127.0.0.1:11112:/home/user/Desktop/hcp_dst5
OK 0000 FS 80000001 SRC0 /home/user/Desktop/hcp_src5/file1.txt
OK 0000 FS 80000002 SRC0 /home/user/Desktop/hcp_src5/file2.txt
OK 0000 FS 80000003 SRC1 /home/user/Desktop/hcp_src6/file3.txt
OK 0000 FS 80000004 SRC1 /home/user/Desktop/hcp_src6/file4.txt
OK A001 FS 80000005 SRC? /home/user/Desktop/hcp_src6/file5.txt
OK 0000 FT 00000001 SRC0 /home/user/Desktop/hcp_src5/file1.txt
OK 0000 FT 00000002 SRC0 /home/user/Desktop/hcp_src5/file2.txt
OK 0000 FT 00000003 SRC1 /home/user/Desktop/hcp_src6/file3.txt
OK 0000 FT 00000004 SRC1 /home/user/Desktop/hcp_src6/file4.txt
EXIT 0 REASON 0000
```

—

最後の行にこの実行の終了ステータスと理由コードが次の書式で記録されます。

EXIT <exit_status> REASON <reason_code>

1.3.20 V, version

```
=====
対応 OS : Linux / Windows
書式 : -V | --version
=====
```

hcp コマンドのバージョンを表示します。

```
--
例 :
[user@localhost ~]$ hcp -V
hcp client (hcp) 1.2.4b_30 / Linux (HpFP2 2.0.0.85)
--
```

1.3.21 t, configtest

```
=====
対応 OS : Linux / Windows
書式 : -t | --configtest
=====
```

hcp コマンドの入力パラメータ及び設定情報を出力します。

```
--
例 :
[user@localhost ~]$ hcp -t
...
Number of Logical Processors : 4
Number of Physical Processors : 4

Command parameters
permission      : disable
recursive       : disable
anydirs         : disable
regex           : disable
verify          : disable
compress        : disable
copylinkfile    : disable(don't copy symlink file)
followlinkdir   : disable(don't follow symlink directory)
noemptyfile     : disable(copy)
noemptydir      : disable(copy)
nodotfile       : disable(copy)
nodotdir        : disable(copy)
hiddenfile      : disable(don't copy)
archivecheck    : disable(don't check archive property)
resume          : disable
nosend          : disable
nodiskio        : disable (0, 0)
copymode        : disable [ALLCOPY]
overwrite       : disable [FORCE]
failaction      : disable [HALT]
remotehost      : disable
remoteport      : disable
```

HCP tools コマンドリファレンス

```
sourcefile      : disable
verbose         : disable
udp             : disable (use TCP)
version         : disable
configtest      : enable
help            : disable
user            : disable
password        : disable
configfile      : disable
logfile         : disable
statlogfile     : disable
multi-run       : disable
```

Target source:

Target destination:

Configuration parameters

```
PubkeyAuthentication      : yes
WinLogonUserAuthentication : yes
PAMAuthentication        : yes
LocalPasswordAuthentication : yes
RequireServerCertificateSecurity : yes
RejectFallbackServerCertificateSecurity : yes
IgnoreCertificateCNInvalid : yes
IgnoreCertificateDateInvalid : no
IgnoreUnknownCA           : no
IgnoreRevocation           : no
CompressLevel              : -1
HeaderCompress             : yes
ContentCompress            : yes
OCSPRevocationEnabled      : yes
StrictHostKeyChecking      : ask
ApplicationStatLog         : no Rotation[size=no:0:0, pattern=no:]
TransportStatLog           : no Rotation[size=no:0:0, pattern=no:]
PrivateKeySearchDir        : - [/etc/hcp/keys]
PrivateKeyFile              : - [~/.hcp/id_rsa]
CACertificateFile          : - [/etc/hcp/cacert.pem]
CACertificatePath          : -
CARevocationFile           : - [/etc/hcp/crl.pem]
CARevocationPath           : -
ProtocolVersion            : 2
MaxConcurrentThread        : 0
UDPTransportExtensionBufferSize : 2147483648
MaxRequestFileEntryAtOnce  : 50
MaxReceiveFileSize         : 107374182400
```

```

MaxSendFileSize           : 107374182400
MaxBufferSize             : 1073741824
MaxReceiveRate            : 10000000000
MaxSendRate               : 10000000000
InitHeaderBlockSize       : 51200
InitContentBlockSize      : 51200
MaxHeaderBlockSize        : 51200
MaxContentBlockSize       : 51200
TransportTimeout          : 180
FileLock                  : yes
FileLockRetries           : 0
FileLockRetryInterval     : 3
AcceptableCryptMethod     : AES256/CBC AES128/CBC
AcceptableDigestMethod    : SHA256 SHA160
DisableDataIntegrityChecking : no
AcceptDataIntegrityCheckingOnRejection : yes
TransportCharEncoding     : UTF8
HostEncoding              : UTF8
ApplicationLog             : INFO Rotation[size=no:0:0, pattern=no:]
UseProperCopyAndSync      : yes
--

```

1.3.22 h, help

```

=====
対応 OS : Linux / Windows
書式 : -h | --help
=====

```

hcp コマンドのヘルプを表示します。

```

--
例 :
[user@localhost ~]$ hcp -h
--

```

1.3.23 m, copymode

```

=====
対応 OS : Linux / Windows
書式 : -m <mode_name> | --copymode=<mode_name>
-----
mode_name
既定値 : ALLCOPY
値の範囲 : ALLCOPY, UPDATE, DIFF, DIFF_STRICT, SYNC
=====

```

ファイルのコピー動作モードを指定します。

ALLCOPY は、全てのファイルをコピーします。

UPDATE は、転送先のファイルの更新日時を比較して転送元のファイルの更新日時の方が新しいファイルをコピーします。

DIFF は、UPDATE に加えてファイルのサイズ比較も行ってサイズに変化がある場合もコピーを行います。

DIFF_STRICT は、DIFF に加えてファイル毎にハッシュの比較を行い、異なるファイルをコピーします。

SYNC は、転送元に存在しない転送先のファイルを削除します（同期）。

```
--  
例 :  
[user@localhost ~]$ hcp -m UPDATE ...  
--
```

1.3.24 o, overwrite

```
=====
対応 OS : Linux / Windows
書式 : -o <overwrite_name> | --overwrite=<overwrite_name>
-----
overwrite_name
既定値 : FORCE
値の範囲 : CONFIRM, FORCE, RENAME, BACKUP
=====
```

ファイルの上書き動作を指定します。

CONFIRM は、上書きの確認を行います。

FORCE は、確認せずに上書きを行います。

RENAME は、上書きを行う前に古いファイルを次の様な接尾辞を付加してリネームします。

.YYMMDD_HHMMSS.NNN (NNN は連番)

BACKUP は、上書きする前に古いファイルを次の様なバックアップディレクトリへ移動します。

./YYMMDD_HHMMSS/

このディレクトリ名は、トランザクション形成時にプロセスユニークに生成されます。

```
--
例 :
[user@localhost ~]$ hcp -o CONFIRM ...
--
```

1.3.25 a, failaction

```
=====
対応 OS : Linux / Windows
書式 : -a <action_name> | --failaction=<action_name>
-----
```

action_name

既定値 : HALT

値の範囲 : HALT, SKIP

ファイルコピーでのエラー発生時の動作を指定します。

HALT は、ファイルコピーで継続不可能な事象が発生した場合にトランザクションを中断します。

SKIP は、ファイルのコピーでエラーが発生した場合に、同ファイルを飛ばして継続することが可能な場合は、処理を続けます。

```
--
例 :
[user@localhost ~]$ hcp -a SKIP ...
--
```

1.3.26 u, user

```
=====
対応 OS : Linux / Windows
書式 : -u <username> | --user=<username>
-----
```

username

既定値 : なし

値の範囲 : ユーザ名文字列

ユーザ認証で使用するユーザ名を指定します。

```
--
例 :
```

```
[user@localhost ~]$ hcp -u user ...  
--
```

1.3.27 w, password

=====
対応 OS : Linux / Windows

書式 : -w <password> | --password=<password>

password

既定値 : なし

値の範囲 : パスワード文字列
=====

ユーザ認証で使用する資格情報（パスワード、パスフレーズ）を指定します。

```
--  
例 :  
[user@localhost ~]$ hcp -w password ...  
--
```

1.3.28 H, remotehost

=====
対応 OS : Linux / Windows

書式 : -H <remote-host> | --remotehost=<remote-host>

remote-host

既定値 : なし

値の範囲 : IP アドレス もしくは ホスト名
=====

sourcefile オプションを指定する際に使用する接続先のリモートホストを指定します。

```
--  
例 :  
[user@localhost ~]$ hcp -H 192.168.100.100 ...  
--
```

1.3.29 P, remoteport

=====
対応 OS : Linux / Windows

書式 : -P <remote-port> | --remoteport=<remote-port>

remote-port

既定値 : なし

値の範囲 : ポート番号

=====

sourcefile オプションを指定する際に使用する接続先のリモートホストのサービスポート番号を指定します。

--

例 :

```
[user@localhost ~]$ hcp -P 1874 ...
```

--

1.3.30 f, sourcefile

=====

対応 OS : Linux / Windows

書式 : -f <source-path-list-file> | --sourcefile=<source-path-list-file>

source-path-list-file

既定値 : なし

値の範囲 : ファイルシステムのパス文字列

=====

転送元の一覧を記述したファイルを指定します。

例 1: ローカルが転送元

file1.txt

child_dir/file2.txt

child_dir2/

/home/user/file3.txt

例 2: リモートが転送元

192.168.100.100:874:file1.txt

192.168.100.100:874:child_dir/file2.txt

192.168.100.100:874:child_dir2/

192.168.100.100:874:/home/user/file3.txt

(IP アドレスは、-H remotehost 及び -P remoteport オプションでも指定可能)

--

例 :

```
[user@localhost ~]$ hcp -f source.list ...
```

--

1.3.31 c, configfile

=====

対応 OS : Linux / Windows

書式 : -c <config-file-path> | --configfile=<config-file-path>

config-file-path

既定値 : なし

値の範囲 : ファイルシステムのパス文字列

=====

hcp コマンドが使用する設定ファイルのパスを指定します。

hcp コマンドは、次の順に設定ファイルの読み込みを行います。

1. /etc/hcp/hcp.conf
2. <ユーザホームディレクトリ>/hcp/hcp.conf
3. <本オプションで指定されたパス>

ファイルが存在しない場合は、読み込みはスキップされます。いずれのファイルでも設定ファイルの読み込みが成功しなかった場合は、設定ファイルの読み込みエラーとなり動作は停止します。

--

例 :

```
[user@localhost ~]$ hcp -c hcp.conf ...
```

--

1.3.32 l, logfile

=====

対応 OS : Linux / Windows

書式 : -l <log-file-path> | --logfile=<log-file-path>

log-file-path

既定値 : なし

値の範囲 : ファイルシステムのパス文字列

=====

hcp コマンドのログを指定したパスのファイルに出力します。

--

例 :

```
[user@localhost ~]$ hcp -l hcp.log ...
```

--

1.3.33 L, statlogfile

=====

対応 OS : Linux / Windows

書式 : -L <log-file-path> | --statlogfile=<log-file-path>

log-file-path

既定値 : なし

値の範囲 : ファイルシステムのパス文字列

=====

hcp コマンドが出力する統計ログの基準パスを指定します。

指定されたパスに接尾辞を付加して各統計ログが出力されます。

<指定されたパス>.application (アプリケーション統計)

<指定されたパス>.transport.tcp (TCP トランスポート統計)

<指定されたパス>.transport.hpfp (HpFP トランスポート統計)

--

例 :

```
[user@localhost ~]$ hcp -L .hcp.statistics2 ...
```

--

1.3.34 investigation

hcpd コマンド参照

1.3.35 multi-run

=====

対応 OS : Linux / Windows

書式 : --multi-run=<record-path>[:<output-switch>]

record-path

既定値 : なし

値の範囲 : ファイルシステムのパス文字列

output-switch

書式 : ((A|T|R|L)[...] | THRU | FULL)

既定値 : THRU

=====

クライアントを多重起動モードで起動します。本オプションを指定した場合は、その指定に従い次の記録（ログ）を指定のディレクトリにユニークなファイル名で保存します。同一のディレクトリで複数のクライアントで作業する場合や、バックグラウンド実行を行う場合などに利用します。

- アプリケーション統計
- トランスポート統計
- 結果出力
- アプリケーションログ

record-path は、前述の記録（ログ）を保存するディレクトリを指定します。

output-switch は、前述の記録（ログ）の出力を制御する指示を指定します。

A はアプリケーション統計を表し、指定すると上記ディレクトリに次のファイル名でアプリケーション統計を記録します。

`hcp.mr.<YYYYMMDD_hhmmss_msec>.<pid>.statistics.application`

T はトランスポート統計を表し、指定すると上記ディレクトリに次のファイル名でトランスポート統計を記録します。

`hcp.mr.<YYYYMMDD_hhmmss_msec>.<pid>.statistics.application`

R は実行記録を表し、指定する上記ディレクトリに次のファイル名でコマンドの実行記録を記録します。

`hcp.mr.<YYYYMMDD_hhmmss_msec>.<pid>.out`

L はアプリケーションログを表し、指定すると上記ディレクトリに次のファイル名でログを記録します。

`hcp.mr.<YYYYMMDD_hhmmss_msec>.<pid>.log`

FULL は、“ATRL”を指定した場合と同じ動作をします。

THRU は、設定ファイルやコマンド引数の指定に従って記録の出力を行います。設定ファイルでアプリケーション統計（トランスポート統計）が有効に設定されている場合は、上記のファイル名にアプリケーション統計（トランスポート統計）を記録します。結果出力（アプリケーションログ）がファイル出力に設定されている場合（**-v** オプション未指定、**-l** オプション指定）は、上記のファイル名に実行記録（ログ）を記録します。

--

例：

`[user@localhost ~]$ hcp --multi-run=/var/tmp:ATR // ログは標準出力に出す。`
他は、`/var/tmp` 配下に出力

--

1.4 hrm

Usage: `hrm [OPTION]... <HOST>:[<PORT>:]<TARGET>`
or: `hrm -H <HOST> [-P <PORT>] [OPTION]... <TARGET>`

hrm コマンドは、リモート（サーバ、hcpd サーバ）上のファイルを削除するコマンドです。

| 短縮名 | オプション名 | 概略 |
|-----|---------------|---|
| f | force | 確認プロンプトを表示せずに削除 |
| R | recursive | 再帰的にファイルを削除 |
| d | dir | 空ディレクトリを削除 |
| i | | 削除毎に確認プロンプトを表示する |
| I | | 一度だけ確認プロンプトを表示する |
| v | verbose | 実行記録を標準出力に出力する |
| n | nodiskio | ディスク I/O 操作を行わず通信プロトコルのネットワーク I/O 性能を計測する |
| U | udp | UDP (HpFP) プロトコルを使用する |
| V | version | バージョンを表示する |
| t | configtest | 入力パラメータ及び設定ファイルのロード結果を表示する |
| h | help | ヘルプを表示する |
| u | user | ユーザ名 |
| w | password | パスワード |
| H | remotehost | リモートホスト指定 |
| P | remoteport | リモートポート指定 |
| c | configfile | 設定ファイル |
| l | logfile | ログファイル |
| L | statlogfile | 統計ログファイル |
| | investigation | 調査モード |
| | multi-run | 多重起動モード |

1.4.1 f, force

```
=====
対応 OS : Linux / Windows
書式 : -f | --force
=====
```

確認プロンプトを表示せずに削除を実行します。パーミッション等の理由で確認が必要な場合でも確認せずに削除を実行します。

--

例 :

```
[user@localhost ~]$ hrm -f ...
```

--

1.4.2 R, recursive

=====

対応 OS : Linux / Windows

書式 : -R | --recursive

=====

再帰的にファイルを削除します。

--

例 :

```
[user@localhost ~]$ hrm -R ...
```

--

1.4.3 d, dir

=====

対応 OS : Linux / Windows

書式 : -d | --dir

=====

空ディレクトリも削除します。

--

例 :

```
[user@localhost ~]$ hrm -d ...
```

--

1.4.4 i

=====

対応 OS : Linux / Windows

書式 : -i

=====

削除毎に確認プロンプトを表示します。

--

例 :

```
[user@localhost ~]$ hrm -i ...
```

--

1.4.5 I

```
=====
対応 OS : Linux / Windows
書式 : -I
=====
```

一度確認プロンプトを表示した後は、プロンプトを表示しない様にします。

```
--
例 :
[user@localhost ~]$ hrm -I ...
--
```

1.4.6 v, verbose

```
=====
対応 OS : Linux / Windows
書式 : -v | --verbose
=====
```

ファイル削除の実行記録を標準出力に出力します。

```
--
例 :
TARGET 127.0.0.1:11111:/home/user/Desktop/hcp_dst5
OK 0000 DE 00000001 ./
OK 0000 FR 00000002 ./file1.txt
OK 0000 DX 00000001 ./
EXIT 0 REASON 0000
--
```

FR (File Remove)は、ファイルの削除処理を表します。DE(Directory Enter)は、i オプションを指定したときに行われるディレクトリに対する削除を開始するかの確認処理を表します。DX(Directory eXit)は、i オプションに指定したときに行われるディレクトリに対する削除確認処理を表します。

```
--
例 :
[user@localhost ~]$ hrm -v ...
--
```

1.4.7 n, nodiskio

hcp コマンド参照

1.4.8 U, udp

hcp コマンド参照

1.4.9 V, version

```
=====
対応 OS : Linux / Windows
書式 : -V | --version
=====
```

hrm コマンドのバージョンを表示します。

```
--
例 :
[user@localhost ~]$ hrm -V
hrm client (hrm) 1.2.4b_30 / Linux (HpFP2 2.0.0.85)
--
```

1.4.10 t, configtest

```
=====
対応 OS : Linux / Windows
書式 : -t | --configtest
=====
```

hrm コマンドの入力パラメータ及び設定情報を出力します。

```
--
例 :
[user@localhost ~]$ hrm -t
...
Number of Logical Processors : 4
Number of Physical Processors : 4

Command parameters
force      : disable
recursive  : disable
dir        : disable
prompt_every : disable
prompt_once : disable
nodiskio    : disable (0, 0)
verbose    : disable
remotehost  : disable
remoteport  : disable
udp         : disable (use TCP)
version     : disable
configtest  : enable
```


HCP tools コマンドリファレンス

```
help          : disable
user          : disable
password      : disable
configfile    : disable
logfile       : disable
statlogfile   : disable
multi-run     : disable
```

Target:

Configuration parameters

```
PubkeyAuthentication      : yes
WinLogonUserAuthentication : yes
PAMAuthentication        : yes
LocalPasswordAuthentication : yes
RequireServerCertificateSecurity : yes
RejectFallbackServerCertificateSecurity : yes
IgnoreCertificateCNInvalid : yes
IgnoreCertificateDateInvalid : no
IgnoreUnknownCA           : no
IgnoreRevocation           : no
CompressLevel              : -1
HeaderCompress             : yes
ContentCompress            : yes
OCSPRevocationEnabled      : yes
StrictHostKeyChecking      : ask
ApplicationStatLog         : no Rotation[size=no:0:0, pattern=no:]
TransportStatLog           : no Rotation[size=no:0:0, pattern=no:]
PrivateKeySearchDir        : - [/etc/hcp/keys]
PrivateKeyFile              : - [~/.hcp/id_rsa]
CACertificateFile          : - [/etc/hcp/cacert.pem]
CACertificatePath          : -
CARevocationFile           : - [/etc/hcp/crl.pem]
CARevocationPath           : -
ProtocolVersion            : 2
MaxConcurrentThread        : 0
UDPTransportExtensionBufferSize : 2147483648
MaxRequestFileEntryAtOnce  : 50
MaxReceiveFileSize         : 107374182400
MaxSendFileSize            : 107374182400
MaxBufferSize              : 1073741824
MaxReceiveRate             : 100000000000
MaxSendRate                : 100000000000
InitHeaderBlockSize        : 51200
InitContentBlockSize       : 51200
MaxHeaderBlockSize         : 51200
```

```
MaxContentSize          : 51200
TransportTimeout         : 180
FileLock                 : yes
FileLockRetries          : 0
FileLockRetryInterval    : 3
AcceptableCryptMethod    : AES256/CBC AES128/CBC
AcceptableDigestMethod   : SHA256 SHA160
DisableDataIntegrityChecking : no
AcceptDataIntegrityCheckingOnRejection : yes
TransportCharEncoding    : UTF8
HostEncoding             : UTF8
ApplicationLog            : INFO Rotation[size=no:0:0, pattern=no:]
--
```

1.4.11 h, help

```
=====
対応 OS : Linux / Windows
書式 : -h | --help
=====
```

hrm コマンドのヘルプを表示します。

```
--
例 :
[user@localhost ~]$ hrm -h
--
```

1.4.12 u, user

hcp コマンド参照

1.4.13 w, password

hcp コマンド参照

1.4.14 H, remotehost

```
=====
対応 OS : Linux / Windows
書式 : -H <remote-host> | --remotehost=<remote-host>
-----
remote-host
既定値 : なし
値の範囲 : IP アドレス もしくは ホスト名
=====
```

接続先のリモートホストを指定します。削除ターゲットのパスに適用されます。
このパラメータを指定した場合、このパスのホスト名を省略することができます。

```
--
例 :
[user@localhost ~]$ hrm -H 192.168.100.100 ...
--
```

1.4.15 P, remoteport

=====
対応 OS : Linux / Windows

書式 : -P <remote-port> | --remoteport=<remote-port>

remote-port

既定値 : なし

値の範囲 : ポート番号
=====

接続先のリモートホストのサービスポート番号を指定します。削除ターゲットのパスに適用されます。

```
--
例 :
[user@localhost ~]$ hrm -P 1874 ...
--
```

1.4.16 c, configfile

hcp コマンド参照

1.4.17 l, logfile

hcp コマンド参照

1.4.18 L, statlogfile

hcp コマンド参照

1.4.19 investigation

hcpd コマンド参照

1.4.20 multi-run

hcp コマンド参照

1.5 hcp-ls

Usage: hcp-ls [OPTION]... <HOST>[:<PORT>][:<FILE>]...
 or: hcp-ls -H <HOST> [-P <PORT>] [OPTION]... [<FILE>...]

hcp-ls コマンドは、リモート（サーバ、hcpd サーバ）上のファイル一覧を取得するコマンドです。

| 短縮名 | オプション名 | 概略 |
|-----|---------------|----------------------------|
| q | querycmdname | コマンド名問合せフラグ |
| o | cmdoptions | コマンドオプション |
| v | verbose | 実行記録を標準出力に出力する |
| U | udp | UDP (HpFP) プロトコルを使用する |
| V | version | バージョンを表示する |
| t | configtest | 入力パラメータ及び設定ファイルのロード結果を表示する |
| h | help | ヘルプを表示する |
| u | user | ユーザ名 |
| w | password | パスワード |
| H | remotehost | リモートホスト指定 |
| P | remoteport | リモートポート指定 |
| c | configfile | 設定ファイル |
| l | logfile | ログファイル |
| L | statlogfile | 統計ログファイル |
| | investigation | 調査モード |
| | multi-run | 多重起動モード |

1.5.1 q, querycmdname

=====
 対応 OS : Linux / Windows
 書式 : -q | --querycmdname
 =====

サーバ上で実行されるリストコマンドの名前を問合せます。ls もしくは dir(Windows)が表示されます。

```
--
例 :
[user@localhost ~]$ hcp-ls -q ...
--
```

1.5.2 o, cmdoptions

```
=====
対応 OS : Linux / Windows
書式 : -o <options> | --cmdoptions=<options>
-----
```

options
既定値 : なし
値の範囲 : **ls** もしくは **dir** コマンドの引数オプション

```
=====
```

コマンドのオプションを指定します。空白もしくはハイフン“-”を含む場合は、クオートします。

```
--
例 :
[user@localhost ~]$ hcp-ls -o "-al" ...
--
```

1.5.3 v, verbose

```
=====
対応 OS : Linux / Windows
書式 : -v | --verbose
=====
```

ファイルリスト取得の実行記録を標準出力へ出力します。リストの実行結果は常に標準出力へ出力されます。

```
--
例 :
FILE 127.0.0.1:11111:/home/user/Desktop/hcp_dst5
--
```

```
--
例 :
[user@localhost ~]$ hcp-ls -v ...
--
```

1.5.4 U, udp

hcp コマンド参照

1.5.5 V, version

=====
対応 OS : Linux / Windows

書式 : -V | --version
=====

hcp-ls コマンドのバージョンを表示します。

--

例 :

```
[user@localhost ~]$ hcp-ls -V
```

```
hcp-ls client (hcp-ls) 1.2.4b_30 / Linux (HpFP2 2.0.0.85)
```

--

1.5.6 t, configtest

=====
対応 OS : Linux / Windows

書式 : -t | --configtest
=====

hcp-ls コマンドの入力パラメータ及び設定情報を出力します。

--

例 :

```
[user@localhost ~]$ hcp-ls -t
```

...

```
Number of Logical Processors : 4
```

```
Number of Physical Processors : 4
```

Command parameters

```
querycmdname      : disable
```

```
cmdoptions        : -
```

```
nodiskio          : disable (0, 0)
```

```
remotehost        : disable
```

```
remoteport        : disable
```

```
udp               : disable (use TCP)
```

```
version           : disable
```

```
configtest        : enable
```

```
help              : disable
```

```
user              : disable
```

```
password          : disable
```

```
configfile        : disable
```

```
logfile           : disable
```

```
statlogfile        : disable
```

```
multi-run         : disable
```

```

Configuration parameters
PubkeyAuthentication          : yes
WinLogonUserAuthentication    : yes
PAMAuthentication            : yes
LocalPasswordAuthentication  : yes
RequireServerCertificateSecurity : yes
RejectFallbackServerCertificateSecurity : yes
IgnoreCertificateCNInvalid     : yes
IgnoreCertificateDateInvalid   : no
IgnoreUnknownCA               : no
IgnoreRevocation              : no
CompressLevel                 : -1
HeaderCompress                : yes
ContentCompress               : yes
OCSPRevocationEnabled         : yes
StrictHostKeyChecking         : ask
ApplicationStatLog             : no Rotation[size=no:0:0, pattern=no:]
TransportStatLog              : no Rotation[size=no:0:0, pattern=no:]
PrivateKeySearchDir           : - [/etc/hcp/keys]
PrivateKeyFile                 : - [~/.hcp/id_rsa]
CACertificateFile              : - [/etc/hcp/cacert.pem]
CACertificatePath              : -
CARevocationFile              : - [/etc/hcp/crl.pem]
CARevocationPath              : -
ProtocolVersion               : 2
MaxConcurrentThread           : 0
UDPTransportExtensionBufferSize : 2147483648
MaxRequestFileEntryAtOnce     : 50
MaxReceiveFileSize            : 107374182400
MaxSendFileSize               : 107374182400
MaxBufferSize                 : 1073741824
MaxReceiveRate                : 10000000000
MaxSendRate                   : 10000000000
InitHeaderBlockSize           : 51200
InitContentBlockSize          : 51200
MaxHeaderBlockSize            : 51200
MaxContentBlockSize           : 51200
TransportTimeout               : 180
FileLock                      : yes
FileLockRetries               : 0
FileLockRetryInterval         : 3
AcceptableCryptMethod          : AES256/CBC AES128/CBC
AcceptableDigestMethod         : SHA256 SHA160
DisableDataIntegrityChecking   : no
AcceptDataIntegrityCheckingOnRejection : yes
TransportCharEncoding          : UTF8

```

```
HostEncoding          : UTF8
ApplicationLog         : INFO Rotation[size=no:0:0, pattern=no:]
--
```

1.5.7 h, help

```
=====
対応 OS : Linux / Windows
書式 : -h | --help
=====
```

hcp-ls コマンドのヘルプを表示します。

```
--
例 :
[user@localhost ~]$ hcp-ls -h
--
```

1.5.8 u, user

hcp コマンド参照

1.5.9 w, password

hcp コマンド参照

1.5.10 H, remotehost

```
=====
対応 OS : Linux / Windows
書式 : -H <remote-host> | --remotehost=<remote-host>
-----
```

```
remote-host
既定値 : なし
値の範囲 : IP アドレス もしくは ホスト名
=====
```

接続先のリモートホストを指定します。リスト対象を指定するパスに適用されます。このパラメータを指定した場合、このパスのホスト名を省略することができます。

```
--
例 :
[user@localhost ~]$ hcp-ls -H 192.168.100.100 ...
--
```


1.5.11 P, remoteport

=====
対応 OS : Linux / Windows

書式 : -P <remote-port> | --remoteport=<remote-port>

remote-port

既定値 : なし

値の範囲 : ポート番号
=====

接続先のリモートホストのサービスポート番号を指定します。リスト対象を指定するパスに適用されます。

--

例 :

[user@localhost ~]\$ hcp-ls -P 1874 ...

--

1.5.12 c, configfile

hcp コマンド参照

1.5.13 l, logfile

hcp コマンド参照

1.5.14 L, statlogfile

hcp コマンド参照

1.5.15 investigation

hcpd コマンド参照

1.5.16 multi-run

hcp コマンド参照

1.6 hmkdir

Usage: hmkdir [OPTION]... <HOST>[:<PORT>]:<DIRECTORY>...

or: hmkdir -H <HOST> [-P <PORT>] [OPTION]... <DIRECTORY>...

hmkdir コマンドは、リモート（サーバ、hcpd サーバ）上でディレクトリを作成するコマンドです。

| 短縮名 | オプション名 | 概略 |
|-----|---------------|----------------------------|
| p | parents | 中間ディレクトリ作成フラグ |
| v | verbose | 実行記録を標準出力に出力する |
| U | udp | UDP (HpFP)プロトコルを使用する |
| V | version | バージョンを表示する |
| t | configtest | 入力パラメータ及び設定ファイルのロード結果を表示する |
| h | help | ヘルプを表示する |
| u | user | ユーザ名 |
| w | password | パスワード |
| H | remotehost | リモートホスト指定 |
| P | remoteport | リモートポート指定 |
| c | configfile | 設定ファイル |
| l | logfile | ログファイル |
| L | statlogfile | 統計ログファイル |
| | investigation | 調査モード |
| | multi-run | 多重起動モード |

1.6.1 p, parents

指定されたパスの中間ディレクトリが存在しない場合に、それらのディレクトリも作成するように指示します。

```
--
例：
[user@localhost ~]$ hmkdir -p ...
--
```

1.6.2 v, verbose

```
=====
対応 OS : Linux / Windows
```

```
書式 : -v | --verbose
=====
```

ディレクトリ作成の実行記録を標準出力へ出力します。

--

例 :

```
[user@localhost ~]$ hmkdir -v ...
DIR0 127.0.0.1:1874:/home/user/Desktop/hcp_mkdir14
DIR1 127.0.0.1:1874:/home/user/Desktop/hcp_mkdir15
OK 0000 DC 00000001 /home/user/Desktop/hcp_mkdir14
OK 0000 DC 00000002 /home/user/Desktop/hcp_mkdir15
EXIT 0 REASON 0000
[user@localhost ~]$
```

--

--

例 :

```
[user@localhost ~]$ hmkdir -v ...
```

--

1.6.3 U, udp

hcp コマンド参照

1.6.4 V, version

=====
対応 OS : Linux / Windows

書式 : -V | --version
=====

hmkdir コマンドのバージョンを表示します。

--

例 :

```
[user@localhost ~]$ hmkdir -h
hmkdir client (hmkdir) 1.2.4b_30 / Linux (HpFP2 2.0.0.85)
```

--

1.6.5 t, configtest

=====
対応 OS : Linux / Windows

書式 : -t | --configtest
=====

hmkdir コマンドの入力パラメータ及び設定情報を出力します。

--

例 :

```
[user@localhost ~]$ hmkdir -t
...
```

HCP tools コマンドリファレンス

Number of Logical Processors : 4
Number of Physical Processors : 4

Command parameters

parents : disable
nodiskio : disable (0, 0)
verbose : disable
remotehost : disable
remoteport : disable
udp : disable (use TCP)
version : disable
configtest : enable
help : disable
user : disable
password : disable
configfile : disable
logfile : disable
statlogfile : disable
multi-run : disable

Configuration parameters

PubkeyAuthentication : yes
WinLogonUserAuthentication : yes
PAMAuthentication : yes
LocalPasswordAuthentication : yes
RequireServerCertificateSecurity : yes
RejectFallbackServerCertificateSecurity : yes
IgnoreCertificateCNInvalid : yes
IgnoreCertificateDateInvalid : no
IgnoreUnknownCA : no
IgnoreRevocation : no
CompressLevel : -1
HeaderCompress : yes
ContentCompress : yes
OCSPRevocationEnabled : yes
StrictHostKeyChecking : ask
ApplicationStatLog : no Rotation[size=no:0:0, pattern=no:]
TransportStatLog : no Rotation[size=no:0:0, pattern=no:]
PrivateKeySearchDir : - [/etc/hcp/keys]
PrivateKeyFile : - [~/.hcp/id_rsa]
CACertificateFile : - [/etc/hcp/cacert.pem]
CACertificatePath : -
CARevocationFile : - [/etc/hcp/crl.pem]
CARevocationPath : -
ProtocolVersion : 2

```
MaxConcurrentThread      : 0
UDPTransportExtensionBufferSize : 2147483648
MaxRequestFileEntryAtOnce : 50
MaxReceiveFileSize       : 107374182400
MaxSendFileSize          : 107374182400
MaxBufferSize            : 1073741824
MaxReceiveRate           : 10000000000
MaxSendRate              : 10000000000
InitHeaderBlockSize      : 51200
InitContentBlockSize     : 51200
MaxHeaderBlockSize       : 51200
MaxContentBlockSize      : 51200
TransportTimeout         : 180
FileLock                 : yes
FileLockRetries          : 0
FileLockRetryInterval    : 3
AcceptableCryptMethod    : AES256/CBC AES128/CBC
AcceptableDigestMethod   : SHA256 SHA160
DisableDataIntegrityChecking : no
AcceptDataIntegrityCheckingOnRejection : yes
TransportCharEncoding    : UTF8
HostEncoding             : UTF8
ApplicationLog            : INFO Rotation[size=no:0:0, pattern=no:]
--
```

1.6.6 h, help

```
=====
対応 OS : Linux / Windows
書式 : -h | --help
=====
```

hmkdir コマンドのヘルプを表示します。

```
--
例 :
[user@localhost ~]$ hmkdir -h
--
```

1.6.7 u, user

hcp コマンド参照

1.6.8 w, password

hcp コマンド参照

1.6.9 H, remotehost

=====
対応 OS : Linux / Windows

書式 : -H <remote-host> | --remotehost=<remote-host>

remote-host

既定値 : なし

値の範囲 : IP アドレス もしくは ホスト名
=====

接続先のリモートホストを指定します。ディレクトリを指定するパスに適用されます。このパラメータを指定した場合、このパス中のホスト名を省略することができます。

--

例 :

```
[user@localhost ~]$ hmkdir -H 192.168.100.100
```

--

1.6.10 P, remoteport

=====
対応 OS : Linux / Windows

書式 : -P <remote-port> | --remoteport=<remote-port>

remote-port

既定値 : なし

値の範囲 : ポート番号
=====

接続先のリモートホストのサービスポート番号を指定します。リディレクトリを指定するパスに適用されます。

--

例 :

```
[user@localhost ~]$ hmkdir -P 1874
```

--

1.6.11 c, configfile

hcp コマンド参照

1.6.12 l, logfile

hcp コマンド参照

1.6.13 L, statlogfile

hcp コマンド参照

1.6.14 investigation

hcpd コマンド参照

1.6.15 multi-run

hcp コマンド参照

1.7 hpwd

Usage: hpwd [OPTION]... <HOST>[:<PORT>]

hpwd コマンドは、リモート（サーバ、hcpd サーバ）上の作業ディレクトリを表示するコマンドです。

| 短縮名 | オプション名 | 概略 |
|-----|---------------|----------------------------|
| I | logical | 論理パス取得 |
| P | physical | 物理パス取得（シンボリックリンク解決） |
| U | udp | UDP (HpFP)プロトコルを使用する |
| V | version | バージョンを表示する |
| t | configtest | 入力パラメータ及び設定ファイルのロード結果を表示する |
| h | help | ヘルプを表示する |
| u | user | ユーザ名 |
| w | password | パスワード |
| c | configfile | 設定ファイル |
| l | logfile | ログファイル |
| L | statlogfile | 統計ログファイル |
| | investigation | 調査モード |
| | multi-run | 多重起動モード |

1.7.1 I, logical

=====

対応 OS : Linux / Windows

書式 : **-I | --logical**

=====

論理パスを返す様に指示します。**logical** オプション及び **physical** オプション共に指定されない場合は、本オプションが指定されたものとして動作します。

--

例 :

```
[user@localhost ~]$ hpwd -I ...
```

--

1.7.2 P, physical

=====

対応 OS : Linux / Windows

書式 : **-P | --physical**

=====

物理パスを返す様に指示します。中間パスに存在するシンボリックリンクは解決されます。

--

例 :

```
[user@localhost ~]$ hpwd -P ...
```

--

1.7.3 U, udp

hcp コマンド参照

1.7.4 V, version

=====

対応 OS : Linux / Windows

書式 : **-V | --version**

=====

hpwd コマンドのバージョンを表示します。

--

例 :

```
[user@localhost ~]$ hpwd -V
hpwd client (hpwd) 1.2.4b_30 / Linux (HpFP2 2.0.0.85)
```

--

1.7.5 t, configtest

=====

対応 OS : Linux / Windows

書式 : -t | --configtest

=====

hpwd コマンドの入力パラメータ及び設定情報を出力します。

--

例 :

```
[user@localhost ~]$ hpwd -t
```

...

Number of Logical Processors : 4

Number of Physical Processors : 4

Command parameters

```
logical      : disable
physical     : disable
nodiskio     : disable (0, 0)
udp          : disable (use TCP)
version      : disable
configtest   : enable
help         : disable
user         : disable
password     : disable
configfile   : disable
logfile      : disable
statlogfile  : disable
multi-run    : disable
```

Configuration parameters

```
PubkeyAuthentication      : yes
WinLogonUserAuthentication : yes
PAMAuthentication        : yes
LocalPasswordAuthentication : yes
RequireServerCertificateSecurity : yes
RejectFallbackServerCertificateSecurity : yes
IgnoreCertificateCNInvalid : yes
IgnoreCertificateDateInvalid : no
IgnoreUnknownCA           : no
IgnoreRevocation           : no
CompressLevel              : -1
HeaderCompress             : yes
ContentCompress            : yes
OCSPRevocationEnabled      : yes
StrictHostKeyChecking      : ask
ApplicationStatLog         : no Rotation[size=no:0:0, pattern=no:]
TransportStatLog           : no Rotation[size=no:0:0, pattern=no:]
PrivateKeySearchDir        : - [/etc/hcp/keys]
```

```

PrivateKeyFile           : - [~/.hcp/id_rsa]
CACertificateFile        : - [/etc/hcp/cacert.pem]
CACertificatePath        : -
CARevocationFile        : - [/etc/hcp/crl.pem]
CARevocationPath        : -
ProtocolVersion         : 2
MaxConcurrentThread      : 0
UDPTransportExtensionBufferSize : 2147483648
MaxRequestFileEntryAtOnce : 50
MaxReceiveFileSize       : 107374182400
MaxSendFileSize          : 107374182400
MaxBufferSize           : 1073741824
MaxReceiveRate           : 100000000000
MaxSendRate              : 100000000000
InitHeaderBlockSize      : 51200
InitContentBlockSize     : 51200
MaxHeaderBlockSize       : 51200
MaxContentBlockSize      : 51200
TransportTimeout         : 180
FileLock                 : yes
FileLockRetries          : 0
FileLockRetryInterval    : 3
AcceptableCryptMethod    : AES256/CBC AES128/CBC
AcceptableDigestMethod   : SHA256 SHA160
DisableDataIntegrityChecking : no
AcceptDataIntegrityCheckingOnRejection : yes
TransportCharEncoding    : UTF8
HostEncoding              : UTF8
ApplicationLog            : INFO Rotation[size=no:0:0, pattern=no:]
o:]
--

```

1.7.6 h, help

```

=====
対応 OS : Linux / Windows
書式 : -h | --help
=====

```

hpwd コマンドのヘルプを表示します。

```

--
例 :
[user@localhost ~]$ hpwd -h
--

```

1.7.7 u, user

hcp コマンド参照

1.7.8 w, password

hcp コマンド参照

1.7.9 c, configfile

hcp コマンド参照

1.7.10 l, logfile

hcp コマンド参照

1.7.11 L, statlogfile

hcp コマンド参照

1.7.12 investigation

hcpd コマンド参照

1.7.13 multi-run

hcp コマンド参照

1.8 hmv

Usage: hmv [OPTION]... <HOST>[:<PORT>]:<SOURCE> <HOST>[:<PORT>]:<DEST>
or: hmv [OPTION]... <HOST>[:<PORT>]:<SOURCE>... <HOST>[:<PORT>]:<DIRECTORY>
or: hmv -H <HOST> [-P <PORT>] [OPTION]... <SOURCE> <DEST>
or: hmv -H <HOST> [-P <PORT>] [OPTION]... <SOURCE>... <DIRECTORY>

hmv コマンドは、リモート（サーバ、hcpd サーバ）上でファイルもしくはディレクトリを移動するコマンドです。

| 短縮名 | オプション名 | 概略 |
|-----|--------------|----------------|
| f | force | 移動先上書き |
| i | interactive | 移動先上書き確認 |
| N | no-overwrite | 移動先上書き禁止 |
| v | verbose | 実行記録を標準出力に出力する |

| | | |
|---|---------------|----------------------------|
| U | udp | UDP (HpFP)プロトコルを使用する |
| V | version | バージョンを表示する |
| t | configtest | 入力パラメータ及び設定ファイルのロード結果を表示する |
| h | help | ヘルプを表示する |
| u | user | ユーザ名 |
| w | password | パスワード |
| H | remotehost | リモートホスト指定 |
| P | remoteport | リモートポート指定 |
| c | configfile | 設定ファイル |
| l | logfile | ログファイル |
| L | statlogfile | 統計ログファイル |
| | investigation | 調査モード |
| | multi-run | 多重起動モード |

1.8.1 f, force

```
=====
対応 OS : Linux / Windows
書式 : -f | --force
=====
```

移動先にファイルもしくはディレクトリが存在する場合に上書きします。

```
--
例 :
[user@localhost ~]$ hmv -f ...
--
```

1.8.2 i, interactive

```
=====
対応 OS : Linux / Windows
書式 : -i | --interactive
=====
```

移動先にファイルもしくはディレクトリが存在する場合に上書するか確認します。

```
--
例 :
```

```
[user@localhost ~]$ hmv -i ...  
--
```

1.8.3 N, no-overwrite

```
=====
対応 OS : Linux / Windows
書式 : -N | --no-overwrite
=====
```

移動先にファイルもしくはディレクトリが存在する場合に上書せずにスキップします。

```
--
例 :
[user@localhost ~]$ hmv -N ...
--
```

1.8.4 v, verbose

```
=====
対応 OS : Linux / Windows
書式 : -v | --verbose
=====
```

ファイルもしくはディレクトリの移動の実行記録を標準出力へ出力します。

```
--
例 :
[user@localhost ~]$ hmv -v ...
SRC0 127.0.0.1:1874:/home/user/Desktop/hcp_hmv01.txt
SRC1 127.0.0.1:1874:/home/user/Desktop/hcp_hmv02.txt
DST 127.0.0.1:1874:/home/user/Desktop/hcp_hmv_dir
OK 0000 FM 00000001 /home/user/Desktop/hcp_hmv01.txt
OK 0000 FM 00000002 /home/user/Desktop/hcp_hmv02.txt
EXIT 0 REASON 0000
[user@localhost ~]$
--
```

```
--
例 :
[user@localhost ~]$ hmv -v ...
--
```

1.8.5 U, udp

hcp コマンド参照

1.8.6 V, version

=====
対応 OS : Linux / Windows

書式 : -V | --version
=====

hmv コマンドのバージョンを表示します。

--

例 :

```
[user@localhost ~]$ hmv -V  
hmv client (hmv) 1.2.4b_30 / Linux (HpFP2 2.0.0.85)
```

--

1.8.7 t, configtest

=====
対応 OS : Linux / Windows

書式 : -t | --configtest
=====

hmv コマンドの入力パラメータ及び設定情報を出力します。

--

例 :

```
[user@localhost ~]$ hmv -t  
...  
Number of Logical Processors : 4  
Number of Physical Processors : 4  
  
Command parameters  
force      : disable  
interactive : disable  
no-overwrite : disable  
verbose    : disable  
nodiskio    : disable (0, 0)  
remotehost  : disable  
remoteport  : disable  
udp         : disable (use TCP)  
version     : disable  
configtest  : enable  
help        : disable  
user        : disable  
password    : disable  
configfile  : disable  
logfile     : disable  
statlogfile : disable
```

HCP tools コマンドリファレンス

multi-run : disable

Target source:

Target destination:

Configuration parameters

| | |
|---|---|
| PubkeyAuthentication | : yes |
| WinLogonUserAuthentication | : yes |
| PAMAuthentication | : yes |
| LocalPasswordAuthentication | : yes |
| RequireServerCertificateSecurity | : yes |
| RejectFallbackServerCertificateSecurity | : yes |
| IgnoreCertificateCNInvalid | : yes |
| IgnoreCertificateDateInvalid | : no |
| IgnoreUnknownCA | : no |
| IgnoreRevocation | : no |
| CompressLevel | : -1 |
| HeaderCompress | : yes |
| ContentCompress | : yes |
| OCSPRevocationEnabled | : yes |
| StrictHostKeyChecking | : ask |
| ApplicationStatLog | : no Rotation[size=no:0:0, pattern=no:] |
| TransportStatLog | : no Rotation[size=no:0:0, pattern=no:] |
| PrivateKeySearchDir | : - [/etc/hcp/keys] |
| PrivateKeyFile | : - [~/.hcp/id_rsa] |
| CACertificateFile | : - [/etc/hcp/cacert.pem] |
| CACertificatePath | : - |
| CARevocationFile | : - [/etc/hcp/crl.pem] |
| CARevocationPath | : - |
| ProtocolVersion | : 2 |
| MaxConcurrentThread | : 0 |
| UDPTransportExtensionBufferSize | : 2147483648 |
| MaxRequestFileEntryAtOnce | : 50 |
| MaxReceiveFileSize | : 107374182400 |
| MaxSendFileSize | : 107374182400 |
| MaxBufferSize | : 1073741824 |
| MaxReceiveRate | : 10000000000 |
| MaxSendRate | : 10000000000 |
| InitHeaderBlockSize | : 51200 |
| InitContentBlockSize | : 51200 |
| MaxHeaderBlockSize | : 51200 |
| MaxContentBlockSize | : 51200 |
| TransportTimeout | : 180 |
| FileLock | : yes |
| FileLockRetries | : 0 |

```
FileLockRetryInterval      : 3
AcceptableCryptMethod      : AES256/CBC AES128/CBC
AcceptableDigestMethod     : SHA256 SHA160
DisableDataIntegrityChecking : no
AcceptDataIntegrityCheckingOnRejection : yes
TransportCharEncoding      : UTF8
HostEncoding               : UTF8
ApplicationLog              : INFO Rotation[size=no:0:0, pattern=no:]
--
```

1.8.8 h, help

```
=====
対応 OS : Linux / Windows
書式 : -h | --help
=====
```

hmv コマンドのヘルプを表示します。

```
--
例 :
[user@localhost ~]$ hmv -h
--
```

1.8.9 u, user

hcp コマンド参照

1.8.10 w, password

hcp コマンド参照

1.8.11 H, remotehost

```
=====
対応 OS : Linux / Windows
書式 : -H <remote-host> | --remotehost=<remote-host>
-----
```

```
remote-host
既定値 : なし
値の範囲 : IP アドレス もしくは ホスト名
=====
```

接続先のリモートホストを指定します。移動元と移動先を指定するパスに適用されます。このパラメータを指定した場合、このパスのホスト名を省略することができます。


```
--  
例 :  
[user@localhost ~]$ hmv -H 192.168.100.100 ...  
--
```

1.8.12 P, remoteport

```
=====
```

対応 OS : Linux / Windows
書式 : -P <remote-port> | --remoteport=<remote-port>

```
-----
```

remote-port
既定値 : なし
値の範囲 : ポート番号

```
=====
```

接続先のリモートホストのサービスポート番号を指定します。移動元と移動先を指定するパスに適用されます。

```
--  
例 :  
[user@localhost ~]$ hmv -P 1874 ...  
--
```

1.8.13 c, configfile

hcp コマンド参照

1.8.14 l, logfile

hcp コマンド参照

1.8.15 L, statlogfile

hcp コマンド参照

1.8.16 investigation

hcpd コマンド参照

1.8.17 multi-run

hcp コマンド参照

1.9 hln

Usage: hln [OPTION]... <HOST>[:<PORT>]:<TARGET> <HOST>[:<PORT>]:<LINK_NAME>
E>

```
or: hln [OPTION]... <HOST>[:<PORT>]:<TARGET>... <HOST>[:<PORT>]:<DIRECTORY>
```

```
or: hln -H <HOST> [-P <PORT>] [OPTION]... <TARGET> <LINK_NAME>
```

```
or: hln -H <HOST> [-P <PORT>] [OPTION]... <TARGET>... <DIRECTORY>
```

hln コマンドは、リモート（サーバ、hcpd サーバ）上でファイルもしくはディレクトリのリンクを作成するコマンドです。本コマンドは、Windows サービスではサポートしません。

| 短縮名 | オプション名 | 概略 |
|-----|----------------|-----------------------------------|
| f | force | リンク先上書き |
| i | interactive | リンク先上書き確認 |
| N | no-dereference | シンボリックリンク先ディレクトリの解決を無効にする |
| s | symbolic | シンボリックリンク作成 |
| I | logical | ターゲットパスを論理名として扱う（シンボリックリンクを解決する） |
| B | physical | ターゲットパスを物理名として扱う（シンボリックリンクを解決しない） |
| v | verbose | 実行記録を標準出力に出力する |
| U | udp | UDP (HpFP) プロトコルを使用する |
| V | version | バージョンを表示する |
| t | configtest | 入力パラメータ及び設定ファイルのロード結果を表示する |
| h | help | ヘルプを表示する |
| u | user | ユーザ名 |
| w | password | パスワード |
| H | remotehost | リモートホスト指定 |
| P | remoteport | リモートポート指定 |
| c | configfile | 設定ファイル |
| l | logfile | ログファイル |
| L | statlogfile | 統計ログファイル |
| | investigation | 調査モード |
| | multi-run | 多重起動モード |

1.9.1 f, force

```
=====
対応 OS : Linux / Windows
書式 : -f | --force
=====
```

リンクが既に存在する場合に上書きします（確認を行いません）。

```
--
例 :
[user@localhost ~]$ hln -f ...
--
```

1.9.2 i, interactive

```
=====
対応 OS : Linux / Windows
書式 : -i | --interactive
=====
```

リンクが既に存在する場合に、削除するか確認を行います。

```
--
例 :
[user@localhost ~]$ hln -i ...
--
```

1.9.3 N, no-dereference

```
=====
対応 OS : Linux / Windows
書式 : -N | --no-dereference
=====
```

リンクとして指定したパスがディレクトリを指すシンボリックリンクの場合に、このリンクを解決(Dereference)せずに処理します。このオプションを指定しない場合のリンク先のディレクトリにターゲットのファイル名でリンクが作成される動作を変更します。

```
--
例 :
[user@localhost ~]$ hln -N ...
--
```

1.9.4 s, symbolic

```
=====
対応 OS : Linux / Windows
```

書式 : -s | --symbolic

=====

シンボリックリンクを作成します。

--

例 :

```
[user@localhost ~]$ hln -s ...
```

--

1.9.5 I, logical

=====

対応 OS : Linux / Windows

書式 : -I | --logical

=====

ハードリンクを行う場合に、ターゲットのパスを論理名として扱います。パスがシンボリックリンクの場合は、解決したパスを使用します。

--

例 :

```
[user@localhost ~]$ hln -I ...
```

--

1.9.6 B, physical

=====

対応 OS : Linux / Windows

書式 : -B | --physical

=====

ハードリンクを行う場合に、ターゲットのパスを物理名として扱います。パスがシンボリックリンクの場合は、リンク先を解決せずに処理します。

--

例 :

```
[user@localhost ~]$ hln -B ...
```

--

1.9.7 v, verbose

=====

対応 OS : Linux / Windows

書式 : -v | --verbose

=====

リンク作成の実行記録を標準出力へ出力します。

--

例 :

```
[user@localhost ~]$ hln -v ...
TARGET0 127.0.0.1:1874:/home/user/Desktop/hcp_hln01.txt
TARGET1 127.0.0.1:1874:/home/user/Desktop/hcp_hln02.txt
LINK_NAME/DIR 127.0.0.1:1874:/home/user/Desktop/hcp_hln_dir
OK 0000 FL 00000001 /home/user/Desktop/hcp_hln01.txt
OK 0000 FL 00000002 /home/user/Desktop/hcp_hln02.txt
EXIT 0 REASON 0000
[user@localhost ~]$
```

--

--

例 :

```
[user@localhost ~]$ hln -v ...
```

--

1.9.8 U, udp

hcp コマンド参照

1.9.9 V, version

=====
対応 OS : Linux / Windows

書式 : -V | --version
=====

hln コマンドのバージョンを表示します。

--

例 :

```
[user@localhost ~]$ hln -V
hln client (hln) 1.2.4b_30 / Linux (HpFP2 2.0.0.85)
```

--

1.9.10 t, configtest

=====
対応 OS : Linux / Windows

書式 : -t | --configtest
=====

hln コマンドの入力パラメータ及び設定情報を出力します。

--

例 :

```
[user@localhost ~]$ hln -t
```

HCP tools コマンドリファレンス

...

Number of Logical Processors : 4

Number of Physical Processors : 4

Command parameters

force : disable
interactive : disable
no-dereference : disable
symbolic : disable
logical : disable
physical : enable
verbose : disable
nodiskio : disable (0, 0)
remotehost : disable
remoteport : disable
udp : disable (use TCP)
version : disable
configtest : enable
help : disable
user : disable
password : disable
configfile : disable
logfile : disable
statlogfile : disable
multi-run : disable

Target source:

Target destination:

Configuration parameters

PubkeyAuthentication : yes
WinLogonUserAuthentication : yes
PAMAuthentication : yes
LocalPasswordAuthentication : yes
RequireServerCertificateSecurity : yes
RejectFallbackServerCertificateSecurity : yes
IgnoreCertificateCNInvalid : yes
IgnoreCertificateDateInvalid : no
IgnoreUnknownCA : no
IgnoreRevocation : no
CompressLevel : -1
HeaderCompress : yes
ContentCompress : yes
OCSPRevocationEnabled : yes
StrictHostKeyChecking : ask

```

ApplicationStatLog      : no Rotation[size=no:0:0, pattern=no:]
TransportStatLog        : no Rotation[size=no:0:0, pattern=no:]
PrivateKeySearchDir     : - [/etc/hcp/keys]
PrivateKeyFile           : - [~/.hcp/id_rsa]
CACertificateFile       : - [/etc/hcp/cacert.pem]
CACertificatePath       : -
CARevocationFile        : - [/etc/hcp/crl.pem]
CARevocationPath        : -
ProtocolVersion         : 2
MaxConcurrentThread     : 0
UDPTransportExtensionBufferSize : 2147483648
MaxRequestFileEntryAtOnce : 50
MaxReceiveFileSize      : 107374182400
MaxSendFileSize         : 107374182400
MaxBufferSize           : 1073741824
MaxReceiveRate          : 10000000000
MaxSendRate             : 10000000000
InitHeaderBlockSize     : 51200
InitContentBlockSize    : 51200
MaxHeaderBlockSize      : 51200
MaxContentBlockSize     : 51200
TransportTimeout        : 180
FileLock                : yes
FileLockRetries         : 0
FileLockRetryInterval   : 3
AcceptableCryptMethod    : AES256/CBC AES128/CBC
AcceptableDigestMethod   : SHA256 SHA160
DisableDataIntegrityChecking : no
AcceptDataIntegrityCheckingOnRejection : yes
TransportCharEncoding    : UTF8
HostEncoding            : UTF8
ApplicationLog           : INFO Rotation[size=no:0:0, pattern=no:]
o:]
--

```

1.9.11 h, help

hln コマンドのヘルプを表示します。

1.9.12 u, user

hcp コマンド参照

1.9.13 w, password

hcp コマンド参照

1.9.14 H, remotehost

=====
対応 OS : Linux / Windows

書式 : -H <remote-host> | --remotehost=<remote-host>

remote-host

既定値 : なし

値の範囲 : IP アドレス もしくは ホスト名
=====

接続先のリモートホストを指定します。リンクターゲット、リンク名もしくはディレクトリを指定するパスに適用されます。このパラメータを指定した場合、このパスのホスト名を省略することができます。

--

例 :

[user@localhost ~]\$ hln -H 192.168.100.100 ...

--

1.9.15 P, remoteport

=====
対応 OS : Linux / Windows

書式 : -P <remote-port> | --remoteport=<remote-port>

remote-port

既定値 : なし

値の範囲 : ポート番号
=====

接続先のリモートホストのサービスポート番号を指定します。リンクターゲット、リンク名もしくはディレクトリを指定するパスに適用されます。

--

例 :

[user@localhost ~]\$ hln -P 1874 ...

--

1.9.16 c, configfile

hcp コマンド参照

1.9.17 l, logfile

hcp コマンド参照

1.9.18 L, statlogfile

hcp コマンド参照

1.9.19 investigation

hcpd コマンド参照

1.9.20 multi-run

hcp コマンド参照

2 設定リファレンス

2.1 hcpd.conf

hcpd デーモンの設定を記述します。

| 項目名 | 説明 |
|----------------------------------|--|
| ProtocolVersion | プロトコルバージョン(2 固定) |
| TCPListenAddress | TCP サービス待機アドレス設定 |
| UDPListenAddress | HpFP サービス待機アドレス設定 |
| UDPServiceExtensionBufferSize | HpFP サービス拡張バッファサイズ設定 |
| UseServerCertificateSecurity | サーバ証明書セキュリティ機能を使用する |
| RequireServerCertificateSecurity | サーバ証明書セキュリティ機能を要求する |
| ServerKeyFile | サーバ鍵ファイル（“.pub”ファイルが存在する場合は、秘密鍵と公開鍵のペアにより通信セキュリティを確保します） |
| ServerCertificateFile | サーバ証明書ファイル |
| ServerCertificateChainFile | サーバ証明書証明書連鎖ファイル |
| LocalPasswordAuthentication | LPA 認証 |
| PAMAuthentication | PAM 認証 |

| | |
|--|------------------------|
| PubkeyAuthentication | RSA 認証 |
| WinLogonUserAuthentication | Windows 認証 |
| PerformSystemAuthenticationRegardlessUsers | システム認証実行制御 |
| UserDirectoryFallbackAvailable | ユーザディレクトリ後退制御 |
| RejectOnUserHomeDirectoryNotFound | ユーザホームディレクトリ検査 |
| UsePrivilegeSeparation | 特権分離 |
| PrivilegeSeparationUser | 特権分離時デフォルトユーザ |
| ApplyUserPermission | ユーザパーミッション適用 |
| AuthorizedKeySearchDir | 認証鍵探索ディレクトリ |
| AuthorizedKeyFile | 認証鍵ファイル |
| CACertificateFile | CA 証明書ファイル |
| CACertificatePath | CA 証明書探索ディレクトリ (予約) |
| CARevocationFile | CA CRL ファイル |
| CARevocationPath | CA CRL 探索ディレクトリ (予約) |
| OCSPRevocationEnabled | OCSP (オンライン証明書失効検査) 機能 |
| LocalUserFile | ユーザ設定ファイル |
| LocalPasswordFile | LPA 認証資格情報設定ファイル |
| AcceptableCryptMethod | 暗号方式 |
| AcceptableDigestMethod | ダイジェスト方式 |
| RequireDataIntegrityChecking | 暗号通信メッセージ検査要求 |
| TransportCharEncoding | トランスポート文字エンコード |
| HostEncoding | ホスト文字エンコード |
| HeaderCompress | ヘッダ圧縮 (予約) |
| ContentCompress | コンテンツ圧縮 (予約) |
| MaxConcurrentThread | 使用スレッド数制限(Linux) |
| MaxTotalConnection | 接続数制限 |
| MaxTcpConnection | TCP 接続数制限 |
| MaxUdpConnection | UDP (HpFP)接続数制限 |

| | |
|-----------------------------|----------------------------|
| MaxConnectionPerUser | ユーザ当たりの接続数制限 |
| MaxConnectionPerSec | 秒間接続数制限 |
| MaxReceiveFileSize | 受信ファイルサイズ制限 |
| MaxSendFileSize | 送信ファイルサイズ制限 |
| MaxRequestFileEntryAtOnce | 一括ファイル要求制限 |
| MaxTotalBufferSize | ペイロードバッファメモリ割当制限（システム全体） |
| MaxBufferSizePerConnection | ペイロードバッファメモリ割当制限（接続単位） |
| MaxTotalReceiveRate | 受信レート制限（システム全体） |
| MaxTotalSendRate | 送信レート制限（システム全体） |
| MaxReceiveRatePerConnection | 受信レート制限（接続単位） |
| MaxSendRatePerConnection | 送信レート制限（接続単位） |
| InitHeaderBlockSize | 初期ヘッダブロックサイズ |
| InitContentBlockSize | 初期コンテンツブロックサイズ |
| MaxHeaderBlockSize | 最大ヘッダブロックサイズ |
| MaxContentBlockSize | 最大コンテンツブロックサイズ |
| FileLock | ファイルロック |
| FileLockRetries | ファイルロック再試行回数 |
| FileLockRetryInterval | ファイルロック試行間隔（秒） |
| TransportTimeout | トランスポートタイムアウト |
| IdleTimeout | アイドルタイムアウト |
| DocPoint | ドキュメントポイント（ファイルシステムアクセス制限） |
| AccessList | アクセス制御リスト |
| SyslogOption | syslog オプション |
| SyslogFacility | syslog ファシリティ |
| SystemLog | ログ設定 |
| SystemLogLevel | ログレベル |
| ApplicationStatLog | アプリケーション統計 |

| | |
|-------------------------------------|--------------|
| TransportStatLog | トランスポート統計 |
| SystemStatLog | システム統計 |
| FileOperationLog | ファイル操作ログ |
| CallbackScript | アプリ連携コールバック |
| EnsureDestinationInFileTransfer | 宛先ディレクトリ保証制御 |
| StatLogPerUserInPrivilegeSeparation | 統計ログ出力先制御 |

2.1.1 TCPListenAddress

```
=====
対応 OS : Linux / Windows
書式 : TCPListenAddress <tcp_service_addr>:<tcp_service_port>[ <acl_name>
>]
```

```
-----
tcp_service_addr
既定値 : なし
値の範囲 : IP アドレス
```

```
-----
tcp_service_port
既定値 : なし
値の範囲 : 1 - 65535
```

```
-----
acl_name
既定値 : なし
値の範囲 : 設定ファイル中に記述されたアクセスコントロールリストの名前
=====
```

クライアントからの接続を受け付ける TCP サービスを定義します。

tcp_service_addr は、TCP サービスの IP アドレスを指定します。

tcp_service_port は、同サービスのポート番号を指定します。

acl_name は、オプションでこの TCP サービスに設定するアクセスコントロールリストを名前指定します。このオプションを省略した場合、もしくは指定された名前がいずれのアクセスコントロールリストの名前と一致しない場合は、無名のアクセスコントロールリストがサービスに設定されます。

```
--
例 :
TCPListenAddress 0.0.0.0:874
--
```

2.1.2 UDPListenAddress

=====
 対応 OS : Linux / Windows

書式 : UDPListenAddress <hpfp_service_addr>:<hpfp_service_port>[:<hpfp_udp_port>[:<hpfp_sndbuf>[:<hpfp_rcvbuf>[:<hpfp_mss>]]]] [<acl_name>]

hpfp_service_addr

既定値 : なし

値の範囲 : IP アドレス値

hpfp_service_port

既定値 : なし

値の範囲 : ポート番号

hpfp_udp_port

書式 : (DEFAULT | D | <decimal_number>)

既定値 : 65520

値の範囲 : decimal_number は、1 - 65535 の範囲。D は DEFAULT の略記。

hpfp_sndbuf

書式 : (DEFAULT | D | <decimal_number>[(T|G|M|K)]B)

既定値 : 100MB

値の範囲 : バイト換算で符号なし倍長整数の範囲。 D は DEFAULT の略記。

hpfp_rcvbuf

書式 : (DEFAULT | D | <decimal_number>[(T|G|M|K)]B)

既定値 : 200MB

値の範囲 : バイト換算で符号なし倍長整数の範囲。 D は DEFAULT の略記。

hpfp_mss

書式 : (DEFAULT | D | NONE | N | <decimal_number>[(T|G|M|K)]B)

既定値 : NONE

値の範囲 : バイト換算で符号なし整数の範囲。 D は DEFAULT の略記。 N は NONE の略記。

acl_name

既定値 : なし

値の範囲 : 設定ファイル中に記述されたアクセスコントロールリストの名前
 =====

クライアントからの接続を受け付ける HpFP(UDP)サービスを定義します。

hpfp_service_addr は、HpFP サービスの IP アドレスを指定します。

hpfp_service_port は、同サービスのポート番号を指定します。

`hfp_udp_port` は、オプションです。HpFP プロトコルで使用する UDP トランスポートのポート番号を指定します。D を指定すると既定値を使用します。

`hfp_sndbuf` は、オプションです。HpFP プロトコルで使用する送信バッファのサイズを指定します。D を指定すると既定値を使用します。

`hfp_rcvbuf` は、オプションです。HpFP プロトコルで使用する受信バッファのサイズを指定します。D を指定すると既定値を使用します。

`hfp_mss` は、オプションです。HpFP プロトコルで使用する MSS を指定します。D を指定すると既定値を使用します。N を指定した場合、HpFP プロトコルによる MTU 探索を行い値を決定します。

`acl_name` は、オプションです。TCP サービと同様。

--

例 :

`UDPListenAddress 0.0.0.0:874`

--

2.1.3 UDPServiceExtensionBufferSize

=====

対応 OS : Linux / Windows

書式 : `UDPServiceExtensionBufferSize <ext-buf-size>`

`ext-buf-size`

既定値 : 2GB

値の範囲 : 符号なし倍長整数 (バイト単位)

=====

HpFP(UDP)サービスで使用する拡張バッファのサイズを指定します。

HpFP セッションでは、遅延やロス及び通信量の増加などに伴い通信用のバッファを指定されたサイズ (`UDPListenAddress` の `hfp_sndbuf` 及び `hfp_rcvbuf`) に拡張します。この拡張するバッファの合計容量を指定された値で制限します。

0 を指定した場合は、この制限を行いません。

また、バッファの初期サイズ (拡張される前のバッファのサイズ) は 1MB です。

--

例 :

`UDPServiceExtensionBufferSize 4GB`

--

2.1.4 UseServerCertificateSecurity

=====
対応 OS : Linux / Windows

書式 : UseServerCertificateSecurity <flag-available>

flag-available

既定値 : yes

値の範囲 : yes, no
=====

サーバ証明書セキュリティ機能を設定します。yes を指定した場合、ServerCertificateFile で指定されたサーバ証明書もしくは、ServerKeyFile のパスから解決された公開鍵を使用して、クライアントとの間で PKI 技術を基礎とするセキュリティ通信を提供します。

--

例 :

UseServerCertificateSecurity no

--

2.1.5 RequireServerCertificateSecurity

=====
対応 OS : Linux / Windows

書式 : RequireServerCertificateSecurity <flag-available>

flag-available

既定値 : yes

値の範囲 : yes, no
=====

クライアントに対してサーバ証明書セキュリティ機能によるセキュリティ通信を要求するか設定します。yes を指定した場合、クライアントが同機能を使用しない通信（平文通信）を要求した場合、接続を拒否します。

--

例 :

RequireServerCertificateSecurity no

--

2.1.6 ServerKeyFile

=====
対応 OS : Linux / Windows

書式 : ServerKeyFile <file-path>

file-path

既定値 :

/etc/hcp/key/server.key

C:/ProgramData/Clealink/HCP Tools/key/server.key (Windows)

値の範囲 : ファイルシステムのパス文字列

=====

サーバ証明書セキュリティ機能で使用するサーバの秘密鍵のパスを指定します。
秘密鍵のパスに接尾辞“.pub”を付加したパスに公開鍵が存在する場合は、
ServerCertificateFile の設定を使用せずにこの鍵ペアを使用してセキュリティ通信
を提供します。

--

例 :

ServerKeyFile /etc/hcp/key/server.key

--

クライアントではサーバに初めてアクセスすると、下記の様に公開鍵を
known_hosts に登録するか確認されます。

```
A secure connection for host 127.0.0.1 can't be established.  
RSA key fingerprint is SHA256: 0fzb9DY4qxXWPm/L/4cBKkk+FQ9577NIRYxRquZ6eW  
A=.  
Are you sure you want to continue connecting [yes/no] ?
```

登録を許可すると、次のパスに確認済みのホストとして記録されます。次回以降、
同じホストの同じ公開鍵であれば確認を求められなくなります。

<ユーザホームディレクトリ>/_hcp/known_hosts

<ユーザホームディレクトリ>/_hcp/known_hosts (Windows)

2.1.7 ServerCertificateFile

=====

対応 OS : Linux / Windows

書式 : ServerCertificateFile <file-path>

file-path

既定値 :

/etc/hcp/cert/server.crt

C:/ProgramData/Clealink/HCP Tools/cert/server.crt (Windows)

値の範囲 : ファイルシステムのパス文字列

=====

サーバ証明書のパスを指定します。


```
--  
例 :  
ServerCertificateFile /etc/hcp/cert/server.crt  
--
```

2.1.8 ServerCertificateChainFile

```
=====
```

対応 OS : Linux / Windows
書式 : ServerCertificateChainFile <file-path>

```
-----
```

file-path

既定値 :

/etc/hcp/cert/chain.crt

C:/ProgramData/Clealink/HCP Tools/cert/chain.crt (Windows)

値の範囲 : ファイルシステムのパス文字列

```
=====
```

サーバ証明書の中間証明書を複数指定します。指定されたファイルに踏まれる中間証明書はクライアントに送信されるサーバ証明書に続けてファイルに保管されている順に中間証明書として送信されます。

```
--  
例 :  
ServerCertificateChainFile /etc/hcp/cert/chain.crt  
--
```

2.1.9 LocalPasswordAuthentication

```
=====
```

対応 OS : Linux / Windows
書式 : LocalPasswordAuthentication <flag-available>

```
-----
```

flag-available

既定値 : no

値の範囲 : yes, no

```
=====
```

LPA 認証を設定します。yes を指定すると LPA 認証を有効にします。本認証方式を含め、一つでも認証を有効にした場合、クライアントからアクセスするには認証が必要になります（匿名アクセス禁止）。

```
--  
例 :  
LocalPasswordAuthentication yes  
--
```

2.1.10 PAMAuthentication

```
=====
対応 OS : Linux
書式 : PAMAuthentication <flag-available>
-----
```

```
flag-available
既定値 : yes
値の範囲 : yes, no
=====
```

PAM 認証を設定します。yes を指定すると PAM 認証を有効にします。

```
--
例 :
PAMAuthentication no
--
```

PAM 認証は、Linux プラットホーム向けの RPM パッケージで配布されるソフトウェアで有効です。

PAM 認証は、システムの構成に依存します。サービスが動作するオペレーティングシステムの環境にあわせて、次の様な PAM 設定ファイルを適切に構成します。

/etc/pam.d/hcpd

2.1.11 PubkeyAuthentication

```
=====
対応 OS : Linux / Windows
書式 : PubkeyAuthentication <flag-available>
-----
```

```
flag-available
既定値 : no
値の範囲 : yes, no
=====
```

RSA 認証を設定します。yes を指定すると RSA 認証を有効にします。

```
--
例 :
PubkeyAuthentication yes
--
```

2.1.12 WinLogonUserAuthentication

```
=====
対応 OS : Windows
```

書式 : WinLogonUserAuthentication <flag-available>

flag-available

既定値 : yes

値の範囲 : yes, no
=====

Windows 認証を設定します。yes を指定すると Windows 認証を有効にします。

--

例 :

WinLogonUserAuthentication no

--

Windows 認証は、Windows プラットホーム向けに配布されるソフトウェアで有効です。

Windows 認証は、システム標準の API の認証機能へ入力されたユーザ ID（ドメイン名を含む）とパスワードをそのまま引き渡して行われます。

2.1.13 PerformSystemAuthenticationRegardlessUsers

=====

対応 OS : Linux / Windows

書式 : PerformSystemAuthenticationRegardlessUsers <flag-available>

flag-available

既定値 : no

値の範囲 : yes, no
=====

システムの認証(PAM 認証、Windows 認証)をユーザ定義の指定に関わらず実行するか設定します。

--

例 :

PerformSystemAuthenticationRegardlessUsers yes

--

2.1.14 UserDirectoryFallbackAvailable

=====

対応 OS : Linux / Windows

書式 : UserDirectoryFallbackAvailable <flag-available>

flag-available

既定値 : no

値の範囲 : yes, no

ユーザのホームディレクトリ、作業ディレクトリなどの後退 (Fallback) を許可するか設定します。認証の結果により認識されたホームディレクトリが存在しない場合などに、ルートドキュメントに記載されたディレクトリに後退する動作をするか制御します。本オプションは、クライアントが 1.1.0 以降のソフトウェアの場合は、常に無効(no)として動作します。

--

例 :

UserDirectoryFallbackAvailable yes

--

2.1.15 RejectOnUserHomeDirectoryNotFound

対応 OS : Linux / Windows

書式 : RejectOnUserHomeDirectoryNotFound <flag-available>

flag-available

既定値 : no

値の範囲 : yes, no

ユーザのホームディレクトリが見つからない場合にアクセスを拒否するか設定します。本オプションは、クライアントが 1.2.0 より以前のソフトウェアの場合は、常に有効(yes)として動作します。

--

例 :

RejectOnUserHomeDirectoryNotFound yes

--

2.1.16 UsePrivilegeSeparation

対応 OS : Linux / Windows

書式 : UsePrivilegeSeparation <flag-available>

flag-available

既定値 : yes

値の範囲 : yes, no

クライアントセッションに対する特権分離を設定します。**yes** を指定した場合、同セッションでの処理をサーバ待機プロセスとは別のプロセスで実行します。分離されたプロセス上では認証結果に基づくユーザの資格情報(UID/GID)が実効権限として設定されます。

--

例:

```
UsePrivilegeSeparation no
```

--

特権分離を無効にした場合は、クライアントセッション上の処理はサービスの実行権限に従って動作します。

Windows サービスでは、Windows ログオン認証以外で認証された場合（LPA 認証、もしくは RSA 認証）は、サービスの実行権限で動作します（特権分離は適用されません）。

2.1.17 PrivilegeSeparationMinimumUID

=====
対応 OS : Linux / Windows

書式 : PrivilegeSeparationMinimumUID <min-uid>

min-uid

既定値 : 1

値の範囲 : 符号なし整数値

=====
特権分離時にセッションを実行できる UID の最小値を設定します。特別な権限を所有するユーザによる実行などを抑制するために使用可能です。

--

例:

```
PrivilegeSeparationMinimumUID 1000
```

--

2.1.18 PrivilegeSeparationMinimumGID

=====
対応 OS : Linux / Windows

書式 : PrivilegeSeparationMinimumGID <min-gid>

min-gid

既定値 : 1

値の範囲 : 符号なし整数値

特権分離時にセッションを実行できる GID の最小値を設定します。特別な権限を所有するユーザによる実行などを抑制するために使用可能です。

```
--  
例 :  
PrivilegeSeparationMinimumGID 1000  
--
```

2.1.19 PrivilegeSeparationUser

```
=====
```

対応 OS : Linux / Windows
書式 : PrivilegeSeparationUser <username>

```
-----
```

username
既定値 : なし
値の範囲 : システムに存在するユーザ名

```
=====
```

特権分離を有効にした場合に、適用するユーザ権限が特定されなかった場合に適用するユーザの資格情報を設定します。

指定しない場合は、プラットフォームに依存したユーザ名（nobody など）に解決されます。

```
--  
例 :  
PrivilegeSeparationUser nobody  
--
```

2.1.20 ApplyUserPermission

```
=====
```

対応 OS : Linux / Windows
書式 : ApplyUserPermission <flag-available>

```
-----
```

flag-available
既定値 : yes
値の範囲 : yes, no

```
=====
```

特権分離を使用しない場合に、認証されたユーザの資格情報(UID/GID)を転送先のファイルパーミッションに適用するか設定します。

```
--  
例 :
```

ApplyUserPermission yes

--

2.1.21 AuthorizedKeySearchDir

=====
対応 OS : Linux / Windows

書式 : AuthorizedKeySearchDir <search-dir>

search-dir

既定値 :

/etc/hcp/authkeys

C:/ProgramData/Clealink/HCP Tools/authkeys (Windows)

値の範囲 : ファイルシステムのパス文字列
=====

RSA 認証の際にユーザの公開鍵を特定するために探索するディレクトリを指定します。

--

例 :

AuthorizedKeySearchDir /etc/hcp/authkeys

--

指定されたディレクトリで次の様なファイル名を公開鍵が保管されたファイルと見做して探索します。

<ユーザ名>.pub

2.1.22 AuthorizedKeyFile

=====
対応 OS : Linux / Windows

書式 : AuthorizedKeyFile <file-path>

file-path

既定値 :

~/.hcp/authorized_keys

~/_hcp/authorized_keys (Windows)

値の範囲 : ユーザディレクトリを表すパス表記(~、チルダ)を含むファイルパス
=====

RSA 認証の際にユーザの公開鍵を特定するために探索するユーザホームディレクトリ内の鍵保管ファイルの書式を指定します。

2.1.23 CACertificateFile

=====
対応 OS : Linux / Windows

書式 : CACertificateFile <file-path>

file-path

既定値 :

/etc/hcp/cacert.pem

C:/ProgramData/Clealink/HCP Tools/cacert.pem (Windows)

値の範囲 : ファイルシステムのパス文字列
=====

クライアント認証に使用する CA 証明書及び中間証明書が保管されているファイルを指定します。

--

例 :

CACertificateFile /etc/hcp/cacert.pem

--

PEM 形式の証明書をサポートします。

2.1.24 CACertificatePath

=====
対応 OS : Linux / Windows

書式 : CACertificatePath <dir-path>

dir-path

既定値 :

/etc/ssl

C:/ProgramData/Clealink/HCP Tools/ssl (Windows)

値の範囲 : ファイルシステムのパス文字列
=====

クライアント認証に使用する CA 証明書及び中間証明書が保管されているディレクトリを指定します。

--

例 :

CACertificatePath /etc/ssl

--

PEM 形式の証明書をサポートします。

2.1.25 CARevocationFile

=====
対応 OS : Linux / Windows

書式 : CARevocationFile <file-path>

file-path

既定値 :

/etc/hcp/crl.pem

C:/ProgramData/Clealink/HCP Tools/crl.pem (Windows)

値の範囲 : ファイルシステムのパス文字列
=====

クライアント認証に使用する CRL が保管されているファイルを指定します。

--

例 :

CARevocationFile /etc/hcp/crl.pem

--

PEM 形式の失効リスト(CRL)をサポートします。

2.1.26 CARevocationPath

=====
対応 OS : Linux / Windows

書式 : CARevocationPath <dir-path>

dir-path

既定値 :

/etc/ssl

C:/ProgramData/Clealink/HCP Tools/ssl (Windows)

値の範囲 : ファイルシステムのパス文字列
=====

クライアント認証に使用する CRL が保管されているディレクトリを指定します。

--

例 :

CARevocationPath /etc/ssl

--

PEM 形式の失効リスト(CRL)をサポートします。

2.1.27 OCSPRevocationEnabled

=====
対応 OS : Linux / Windows

書式 : OCSPRevocationEnabled <flag-available>

flag-available

既定値 : yes

値の範囲 : yes, no

クライアント認証で使用する OSCP（オンライン証明書失効検査プロトコル）を設定します。yes を指定すると有効にします。

--

例 :

OCSPRevocationEnabled no

--

2.1.28 LocalUserFile

対応 OS : Linux / Windows

書式 : LocalUserFile <file-path>

file-path

既定値 :

/etc/hcp/users

C:/ProgramData/Clealink/HCP Tools/users (Windows)

値の範囲 : ファイルシステムのパス文字列

ユーザ情報を定義したファイルを指定します。

--

例 :

LocalUserFile /etc/hcp/users

--

HCP tools の認証では、このファイルに定義されたユーザを認証において認識します（定義されないユーザで認証を試行した場合は失敗します）。

2.1.29 LocalPasswordFile

対応 OS : Linux / Windows

書式 : LocalPasswordFile <file-path>

file-path

既定値 :

/etc/hcp/passwd

C:/ProgramData/Clealink/HCP Tools/passwd (Windows)

値の範囲 : ファイルシステムのパス文字列

=====

LPA 認証で使用するユーザの資格情報（パスワードハッシュ）を定義したファイルを指定します。

--
例 :
LocalPasswordFile /etc/hcp/passwd
--

2.1.30 AcceptableCryptMethod

=====

対応 OS : Linux / Windows
書式 : AcceptableCryptMethod <method-names>

method-names
書式 : <method-name>[...]
既定値 : AES256/CBC AES128/CBC

method-name
値の範囲 : PLAIN, AES128/CBC, AES192/CBC, AES256/CBC

=====

暗号アルゴリズムを設定します。

--
例 :
AcceptableCryptMethod AES256/CBC PLAIN
--

クライアントとの間で通信するメッセージの暗号化に使用されます。どのアルゴリズムが使用されるかは、クライアントの設定と併せて評価し、一致したアルゴリズムを使用します。

2.1.31 AcceptableDigestMethod

=====

対応 OS : Linux / Windows
書式 : AcceptableDigestMethod <method-names>

method-names
書式 : <method-name>[...]
既定値 : SHA256 SHA160

method-name

値の範囲 : NONE, SHA160, SHA224, SHA256, SHA384, SHA512, MD5, MM32, MM128

ダイジェストアルゴリズムを設定します。

```
--  
例 :  
AcceptableDigestMethod SHA256 MM128 NONE  
--
```

クライアントとの間で通信するメッセージ、ファイルやそのデータブロックの検証に使用されます。どのアルゴリズムが使用されるかは、クライアントの設定と併せて評価し、一致したアルゴリズムを使用します。

また、暗号化による通信を行う場合、セキュリティダイジェストではないアルゴリズム (MD5, MM32, MM128) は、指定されなかったものとして扱われます。

2.1.32 RequireDataIntegrityChecking

対応 OS : Linux / Windows

書式 : RequireDataIntegrityChecking <flag-available>

flag-available

既定値 : yes

値の範囲 : yes, no

クライアントとの間で暗号化通信を行う場合に、サーバが MAC(Message Authentication Code)による通信メッセージの検査 (データ完全性検査) を要求するか設定します。

no を指定した場合、クライアントが通信メッセージの検査を行わないことを要求した場合に、その要求を受け入れて通信を継続します。

yes を指定した場合、この要求を拒否します。

通常は yes (デフォルト) で使用してください。前述の通信メッセージの検査が省略されることを理解したうえでご利用ください。通常、暗号通信のパフォーマンス向上の目的で使用します。

```
--  
例 :  
RequireDataIntegrityChecking no  
--
```

2.1.33 TransportCharEncoding

=====
対応 OS : Linux / Windows

書式 : TransportCharEncoding <encodings>

encodings

書式 : <encoding>[...]

既定値 : UTF8

encoding

値の範囲 : US-ASCII, UTF8, UTF16, UTF32
=====

トランスポートで使用する文字列のエンコーディング方式を指定します。

--

例 :

TransportCharEncoding UTF8 UTF16 US-ASCII

--

クライアントとの間で交換するファイルパスなどの文字列に適用されます。どのエンコーディングが使用されるかは、クライアントの設定と併せて評価し、一致したエンコーディングを使用します。

2.1.34 HostEncoding

=====
対応 OS : Linux / Windows

書式 : HostEncoding <encoding>

encoding

既定値 :

UTF-8

CP932 (Windows)

値の範囲 : システムかつエンコーディング変換ライブラリでサポートされるエンコーディング名 (プラットフォーム依存)
=====

ホストで使用されている文字列エンコーディングを指定します。

--

例 :

HostEncoding EUC-JP

--

ファイルパスなどをソフトウェアの内部文字列表現へ変換する際に使用されます。

2.1.35 HeaderCompress (予約)

2.1.36 ContentCompress (予約)

2.1.37 MaxConcurrentThread

```
=====
対応 OS : Linux
書式 : MaxConcurrentThread <max-threads>
-----
```

```
max-threads
既定値 : 0
値の範囲 : 符号付整数
=====
```

スレッド数制限を設定します。

2.1.38 MaxTotalConnection

```
=====
対応 OS : Linux / Windows
書式 : MaxTotalConnection <max-total-con>
-----
```

```
max-total-con
既定値 : 10
値の範囲 : 符号付整数
=====
```

接続数制限を設定します。

```
--
例 :
MaxTotalConnection 5
--
```

2.1.39 MaxTcpConnection

```
=====
対応 OS : Linux / Windows
書式 : MaxTcpConnection <max-tcp-con>
-----
```

```
max-tcp-con
既定値 : 5
値の範囲 : 符号付整数
=====
```

TCP 接続数制限を設定します。

--

例 :

MaxTcpConnection 5

--

2.1.40 MaxUdpConnection

HpFP(UDP)接続数制限を設定します。

=====

対応 OS : Linux / Windows

書式 : MaxUdpConnection <max-udp-con>

max-udp-con

既定値 : 5

値の範囲 : 符号付き整数

=====

--

例 :

MaxUdpConnection 5

--

2.1.41 MaxConnectionPerUser

=====

対応 OS : Linux / Windows

書式 : MaxConnectionPerUser <max-con-per-user>

max-con-per-user

既定値 : 1

値の範囲 : 符号付き整数

=====

ユーザ毎の接続数制限を設定します。

--

例 :

MaxConnectionPerUser 1

--

2.1.42 MaxConnectionPerSec

=====

対応 OS : Linux / Windows

書式 : MaxConnectionPerSec <max-con-per-sec>

max-con-per-sec

既定値 : 1

値の範囲 : 符号付き整数

=====

秒間の接続数制限を設定します。

--

例 :

MaxConnectionPerSec 10

--

2.1.43 MaxReceiveFileSize

=====

対応 OS : Linux / Windows

書式 : MaxReceiveFileSize <file-size>

file-size

既定値 : 100GB

値の範囲 : 符号付き倍長整数

=====

受信を許可する最大のファイルサイズを設定します。

--

例 :

MaxReceiveFileSize 1GB

--

2.1.44 MaxSendFileSize

=====

対応 OS : Linux / Windows

書式 : MaxSendFileSize <file-size>

file-size

既定値 : 100GB

値の範囲 : 符号付き倍長整数

=====

送信を許可する最大のファイルサイズを設定します。

--

例 :

MaxSendFileSize 1GB

--

2.1.45 MaxRequestFileEntryAtOnce

=====
対応 OS : Linux / Windows

書式 : MaxRequestFileEntryAtOnce <max-file-req-at-once>

max-file-req-at-once

既定値 : 50

値の範囲 : 符号付き整数
=====

ファイル要求の一括送信を許可する最大の数を設定します。

--

例 :

MaxRequestFileEntryAtOnce 1000

--

2.1.46 MaxTotalBufferSize

=====
対応 OS : Linux / Windows

書式 : MaxTotalBufferSize <max-total-buf-size>

max-total-buf-size

既定値 : 2GB

値の範囲 : 符号付き倍長整数
=====

ファイルのデータ処理に使用できる最大のメモリバッファサイズを設定します。

--

例 :

MaxTotalBufferSize 4GB

--

2.1.47 MaxBufferSizePerConnection

=====
対応 OS : Linux / Windows

書式 : MaxBufferSizePerConnection <max-buf-size-per-con>

max-buf-size-per-con

既定値 : 100MB

値の範囲 : 符号なし倍長整数
=====

クライアントセッション毎のファイルのデータ処理に使用できる最大のメモリバッファサイズを設定します。

--

例：

```
MaxBufferSizePerConnection 512MB
```

--

2.1.48 MaxTotalReceiveRate

=====
対応 OS : Linux / Windows

書式 : MaxTotalReceiveRate <bandwidth>

bandwidth

既定値 : 10Gbit

値の範囲 : 符号なし倍長整数
=====

トランスポートの受信帯域のシェーピング（制限）を設定します。

--

例：

```
MaxTotalReceiveRate 1Gbit
```

--

本機能は、TCP/HpFP(UDP)層とアプリケーション層の間で帯域制限を行います。指定値が 5Gbps を超える場合は、無制限（シェーピングなし）に設定されます（他の帯域シェーピングオプションも同様）。

2.1.49 MaxTotalSendRate

=====
対応 OS : Linux / Windows

書式 : MaxTotalSendRate <bandwidth>

bandwidth

既定値 : 10Gbit

値の範囲 : 符号なし倍長整数
=====

トランスポートの送信帯域のシェーピング（制限）を設定します。

--

例：

```
MaxTotalSendRate 1Gbit
```

--

2.1.50 MaxReceiveRatePerConnection

=====
対応 OS : Linux / Windows

書式 : MaxReceiveRatePerConnection <bandwidth>

bandwidth

既定値 : 10Gbit

値の範囲 : 符号なし倍長整数
=====

クライアントセッション毎のトランスポート受信帯域のシェーピング（制限）を設定します。

--

例 :

MaxReceiveRatePerConnection 100Mbit

--

2.1.51 MaxSendRatePerConnection

=====
対応 OS : Linux / Windows

書式 : MaxSendRatePerConnection <bandwidth>

bandwidth

既定値 : 10Gbit

値の範囲 : 符号なし倍長整数
=====

クライアントセッション毎のトランスポート送信帯域のシェーピング（制限）を設定します。

--

例 :

MaxSendRatePerConnection 100Mbit

--

2.1.52 InitHeaderBlockSize

=====
対応 OS : Linux / Windows

書式 : InitHeaderBlockSize <block-size>

block-size

既定値 : 4KB

値の範囲 : 符号なし倍長整数
=====

初期ヘッダブロックのサイズを設定します。

```
--  
例 :  
InitHeaderBlockSize 10KB  
--
```

ファイル要求などのメッセージを複数含むヘッダブロックが形成可能な上限のサイズを与えます。このオプションは通信開始直後に適用されます。

2.1.53 InitContentSize

```
=====
```

対応 OS : Linux / Windows
書式 : InitContentSize <block-size>

```
-----
```

block-size
既定値 : 512KB
値の範囲 : 符号なし倍長整数

```
=====
```

初期コンテンツブロックのサイズを設定します。

```
--  
例 :  
InitContentSize 100KB  
--
```

ファイルのデータを複数含むコンテンツブロックが形成可能な上限のサイズを与えます。このオプションは通信開始直後に適用されます。

2.1.54 MaxHeaderBlockSize

```
=====
```

対応 OS : Linux / Windows
書式 : MaxHeaderBlockSize <block-size>

```
-----
```

block-size
既定値 : 10MB
値の範囲 : 符号なし倍長整数

```
=====
```

ヘッダブロックのサイズを拡張する最大のサイズを設定します。

```
--  
例 :  
MaxHeaderBlockSize 100KB  
--
```

通信を開始すると消費帯域を評価して形成可能なヘッダブロックのサイズを増減させます。本オプションは、このサイズを増加させることができる上限値を与えます。

2.1.55 MaxContentSize

```
=====
対応 OS : Linux / Windows
書式 : MaxContentSize <block-size>
```

```
-----
block-size
既定値 : 100MB
値の範囲 : 符号なし倍長整数
=====
```

コンテンツブロックのサイズを拡張する最大のサイズを設定します。

```
--
例 :
MaxContentSize 1MB
--
```

通信を開始すると消費帯域を評価して形成可能なコンテンツブロックのサイズを増減させます。本オプションは、このサイズを増加させることができる上限値を与えます。

2.1.56 FileLock

```
=====
対応 OS : Linux / Windows
書式 : FileLock <flag-available>
```

```
-----
flag-available
既定値 : yes
値の範囲 : yes, no
=====
```

ファイルの読み書きにファイルロックを使用するか設定します。

NFS などのネットワークファイルシステムでハングアップが発生した際に、同現象を回避するなどの目的で **no** に設定します。ロック要求の処理は適切に動作していて、単にロックの獲得を待機している場合は、次の様なログが記録されるか (FileLockRetries が 0) 、

```
2018/07/05 16:34:10 00007f638bd97740:INFO :Acquiring a lock to the file w
as rejected at the first trial.
```

```
2018/07/05 16:34:13 00007f638bd97740:INFO :Acquiring a lock to the file c
ontinues to be rejected about few seconds.
```

試行回数の上限に達したことを検出して処理を中断します（FileLockRetries が 0 以上）。

no に設定した場合は、ファイルのロックを獲得せずにファイルデータの読み書きを実行します。

```
--
例 :
FileLock no
--
```

2.1.57 FileLockRetries

```
=====
対応 OS : Linux / Windows
書式 : FileLockRetries <num-retries>
```

```
-----
num-retries
既定値 : 0
値の範囲 : 符号なし整数
=====
```

ファイルロックの獲得を試行する回数を設定します。0 を指定するとロックを獲得するまで待機（ブロック）します。

```
--
例 :
FileLockRetries 5
--
```

2.1.58 FileLockRetryInterval

```
=====
対応 OS : Linux / Windows
書式 : FileLockRetryInterval <retry-interval>
```

```
-----
retry-interval
既定値 : 3
値の範囲 : 符号なし整数
=====
```

ファイルロックの要求間隔を設定します（秒単位）。

```
--
例 :
```

FileLockRetryInterval 10

--

2.1.59 TransportTimeout

=====
対応 OS : Linux / Windows

書式 : TransportTimeout <timeout>

timeout

既定値 : 180

値の範囲 : 符号なし整数

=====
トランスポートのタイムアウトを秒単位で設定します。

--

例 :

TransportTimeout 60

--

2.1.60 IdleTimeout

=====
対応 OS : Linux / Windows

書式 : IdleTimeout <timeout>

timeout

既定値 : 600

値の範囲 : 符号なし整数

=====
セッションのアイドルタイムアウトを秒単位で設定します。接続上でオペレーション（コマンド実行を単位とする処理）が指定時間実行されない場合に、サーバから接続を切断します。クライアントがターミナル機能（複数のコマンドを任意の時点で実行可能。API ライブラリで提供されます）を使用している場合に適用されます。

--

例 :

IdleTimeout 180

--

2.1.61 DocPoint

=====
対応 OS : Linux / Windows

書式 : DocPoint <doc_point_name>

doc_point_name

既定値 : なし

値の範囲 : 文字列

ドキュメントポイントを定義します。doc_point_name は、このドキュメントポイントの名称を記述します。

最初に定義されたドキュメントポイントは、ユーザのホームディレクトリが解決できなかった場合の既定のホームディレクトリとして使用されます。

--

例 :

DocPoint /home

--

2.1.62 DocPath

対応 OS : Linux / Windows

書式 : DocPath <doc_path>

doc_path

既定値 : なし

値の範囲 : ファイルシステムのパス文字列

ドキュメントポイントが指すディレクトリパスを指定します。

--

例 :

DocPath /home

--

この項目で指定されたディレクトリパスのファイルやディレクトリへのアクセスを許可します。

2.1.63 PermitAccessRead

対応 OS : Linux / Windows

書式 : PermitAccessRead <flag-available>

flag-available

既定値 : no

値の範囲 : yes, no

ドキュメントポイントでの読み込みアクセスを設定します。

--

例 :

PermitAccessRead yes

--

no を指定した場合、このドキュメントポイント内のファイルの読み出しは禁止されます。ファイル転送を行う場合は、転送元のファイルの読み出しなどが読み込みエラーとなります。

2.1.64 PermitAccessWrite

対応 OS : Linux / Windows

書式 : PermitAccessWrite <flag-available>

flag-available

既定値 : no

値の範囲 : yes, no

ドキュメントポイントでの書き込みアクセスを設定します。

--

例 :

PermitAccessWrite yes

--

no を指定した場合、このドキュメントポイント内のファイルへの書出しは禁止されます。ファイル転送を行う場合は、転送先のファイルへの書出しなどが書き込みエラーとなります。

2.1.65 PermitAccessOverwrite

対応 OS : Linux / Windows

書式 : PermitAccessOverwrite <flag-available>

flag-available

既定値 : no

値の範囲 : yes, no

ドキュメントポイントでの上書きアクセスを設定します。

```
--
例：
    PermitAccessOverwrite yes
--
```

no を指定した場合、このドキュメントポイント内のファイルへの上書きは禁止されます。ファイル転送を行う場合は、転送先のファイルへ上書き（既に存在するファイルへのデータ書込み）などが書込みエラーとなります。

2.1.66 PermitAccessDelete

```
=====
対応 OS : Linux / Windows
書式 : PermitAccessDelete <flag-available>
```

```
-----
flag-available
既定値 : no
値の範囲 : yes, no
=====
```

ドキュメントポイントでの削除アクセスを設定します。

```
--
例：
    PermitAccessDelete yes
--
```

no を指定した場合、このドキュメントポイント内のファイルの削除は禁止されます。

2.1.67 PermitAccessRandomRead (予約)

```
=====
対応 OS : Linux / Windows
書式 : PermitAccessRandomRead <flag-available>
```

```
-----
flag-available
既定値 : no
値の範囲 : yes, no
=====
```

ドキュメントポイントでのランダム読込みアクセスを設定します。

```
--
例：
```

```
PermitAccessRandomRead yes
--
```

2.1.68 PermitAccessRandomWrite (予約)

```
=====
対応 OS : Linux / Windows
書式 : PermitAccessRandomWrite <flag-available>
-----
```

```
flag-available
既定値 : no
値の範囲 : yes, no
=====
```

ドキュメントポイントでのランダム書込みアクセスを設定します。

```
--
例 :
    PermitAccessRandomRead yes
--
```

2.1.69 AccessList

```
=====
対応 OS : Linux / Windows
書式 : AccessList <acl_name>
-----
```

```
acl_name
既定値 : なし
値の範囲 : 文字列
=====
```

アクセスコントロールリストを定義します。 **acl_name** はオプションです。アクセスコントロールリストの名前を指定します。

```
--
例 :
AccessList acl1
--
```

acl-name を指定しない場合は、無名のアクセスコントロールリストとして扱われます。この無名のアクセスコントロールリストは、1 つだけ定義できます。

2.1.70 Allow

```
=====
対応 OS : Linux / Windows
書式 : Allow (<ip_addr> <net_mask>|any)[ <hpf Cong_mode_modifier>]
```

ip_addr

既定値 : なし

値の範囲 : IP アドレス

net_mask

既定値 : なし

値の範囲 : サブネットマスク表記

hpfp_cong_mode_modifier

書式 : <modifier>[...]

modifier := (+|-)(M|S|A)[...]

既定値 : なし
=====

アクセスコントロールリスト内に、許可アクセスを定義します。

ip_addr は、IP アドレスを指定します。

net_mask は、ネットマスクを指定します。

キーワード **any** は、全てのネットワークを表します。

hpfp_cong_mode_modifier は、HpFP の輻輳制御モードの上書きを指定します。

+は、記述したルールで接続を受け付けた場合に、直後に記載した輻輳制御モードの許可を追加します。

-は、直後に記載した輻輳制御モードの許可を取り消します。

M、**S** および **A** は、それぞれ HpFP の輻輳制御モードの **MODEST**、**FAIR_FAST_START** および **AGGRESSIVE** を表します。

--

例 :

Allow 192.168.1.0 255.255.255.0 -A+M

--

接続元の IP アドレスが指定されたネットワークに含まれる場合、そのアクセスを許可します。

2.1.71 Deny

=====

対応 OS : Linux / Windows

書式 : Deny (<ip_addr> <net_mask>|any)

ip_addr

既定値 : なし

値の範囲 : IP アドレス

net_mask

既定値 : なし

値の範囲 : サブネットマスク表記

アクセスコントロールリスト内に、拒否アクセスを定義します。

--

例 :

Deny 192.168.1.0 255.255.255.0

--

接続元の IP アドレスが指定されたネットワークに含まれる場合、そのアクセスを拒否します。

2.1.72 SyslogOption

対応 OS : Linux / Windows

書式 : SyslogOption <syslog-options>

syslog-options

書式 : syslog-option[...]

既定値 : CONS PID

syslog-option

値の範囲 : CONS, NDELAY, NOWAIT, ODELAY, PERROR, PID

syslog のオプションを設定します（複数指定可）。

各オプションは、接頭辞“LOG_”を付加した syslog のオプションに対応します。

--

例 :

SyslogOption PID

--

2.1.73 SyslogFacility

対応 OS : Linux / Windows

書式 : SyslogFacility <syslog-facility>

syslog-facility

既定値 : DAEMON

値の範囲 : AUTH, CRON, DAEMON, FTP, LOCAL0 - LOCAL7, LPR, MAIL, NEWS, USER, UUCP

=====
syslog のファシリティを設定します。

各ファシリティは、接頭辞“LOG_”を付加した **syslog** のファシリティに対応します。

--

例 :

SyslogFacility FTP

--

2.1.74 SystemLog

=====
対応 OS : Linux / Windows

書式 : SystemLog <log-level>[<log-rotation-conf>]

log-level

既定値 : INFO

値の範囲 : EMERG, ALERT, CRIT, ERR, WARNING, INFO, DEBUG

log-rotation-conf

書式 : (FileSize <file-size> <backups> | DatePattern <date-pattern>)

file-size

既定値 : なし

値の範囲 : 符号あり倍長整数

backups

既定値 : なし

値の範囲 : 符号なし整数

date-pattern

既定値 : なし

値の範囲 : yyyy-MM, yyyy-MM-dd, yyyy-MM-dd-HH, yyyy-MM-dd-HH-mm

=====
システムのログ設定を行います。

log-level は、ログのレベルを指定します。

log-rotation-conf は、ログのローテーションを指定します。指定しない場合は、ローテーションしません。**FileSize** を指定するとファイルサイズを閾値としたローテーションを行い、**DatePattern** を指定すると日時にもとづくローテーションを行います。

file-size は、ローテーションを行いファイルサイズの閾値をバイト単位で指定します。

backups は、**FileSize** を指定してローテーションを行う場合にローテーションされたファイルを保持する世代数の上限を指定します。

date-pattern は、日時でローテーションするパターンを指定します。

FileSize を指定した場合は、ログのパス名に次の様に世代数を接尾辞として付加してファイルをローテーションします。

```
<指定されたログのパス> // 現在書込み中のログのパス
<指定されたログのパス>.1
<指定されたログのパス>.2
...
<指定されたログのパス>.n // backups に n を指定
```

backups で指定された世代数を超えるファイルは、ローテーション実行時に削除されます。

DatePattern を指定した場合は、ログのパス名に次の様に指定されたパターンに従って日時を接尾辞として付加してファイルをローテーションします。

```
// yyyy-MM-dd の場合
<指定されたログのパス> // 現在書込み中のログのパス
<指定されたログのパス>.2019-12-10 // 2019/12/10 のログ
<指定されたログのパス>.2019-12-09
...
<指定されたログのパス>.2019-11-30
...
```

ローテーションは、指定されたパターンで定まる日時の区間を単位に行います。月単位の場合は、その月の 1 日 0 時 0 分 0 秒から翌月の同時刻が到来する直前までになります。

例：

2019/11/01 00:00:00 から 2019/12/01 00:00:00 まで
(但し、2019/12/01 00:00:00 を含まない)

分単位の場合は、その分の 0 秒から次の分の 0 の直前までになります。

例：

2019/11/01 10:30:00 から 2019/11/01 10:31:00 まで
(但し、2019/11/01 10:31:00 を含まない)

ローテーションは、各区間を経過後にログの書込みが要求されるとその書込みの前に実行されます。ファイル名は、該当する区間の接尾辞が付加された名前に変更されます。

// yyyy-MM-dd-HH-mm の場合のローテーションの事例

2019/12/10 00:00 サーバ起動

2019/12/10 00:05 サーバ停止

2019/12/10 00:07 サーバ再起動

...
--

<指定されたログのパス>

<指定されたログのパス>.2019-12-10-00-11

<指定されたログのパス>.2019-12-10-00-10 // 00:09 から 00:10 の間は書込みがなかった

<指定されたログのパス>.2019-12-10-00-08

<指定されたログのパス>.2019-12-10-00-07 // 再起動後の 00:07 のログ

<指定されたログのパス>.2019-12-10-00-05 // サーバ停止までの 00:05 のログ (再起動時にファイルの更新日時から判断してローテーション実施)

...

<指定されたログのパス>.2019-12-10-00-00

但し、特権分離を使用しているサーバでは定期的に (128 ミリ秒程度の間隔で) ローテーションが実行されます。ローテーションが遅延して実行される場合があるため、通常は次の区間に出力されるログが現在の区間に出力されることがあります。

--

例 1：

SystemLog WARNING FileSize 10MB 10

例 2：

SystemLog WARNING DatePattern yyyy-MM-dd

例 3：

SystemLog WARNING // SystemLogLevel に同じ

--

2.1.75 SystemLogLevel

=====

対応 OS : Linux / Windows

書式 : SystemLogLevel <log-level>

log-level

既定値 : INFO

値の範囲 : EMERG, ALERT, CRIT, ERR, WARNING, INFO, DEBUG

システムのログレベルを設定します。syslog の機能には影響しません。

SystemLog と共に記述した場合は、ログレベルの値のみ上書きされます。

--

例 :

SystemLogLevel WARNING

--

2.1.76 ApplicationStatLog

対応 OS : Linux / Windows

書式 : ApplicationStatLog <flag-available>[<log-rotation-conf>]

flag-available

既定値 : no

値の範囲 : yes, no

log-rotation-conf

書式 : (FileSize <file-size> <backups> | DatePattern <date-pattern>)

file-size

既定値 : なし

値の範囲 : 符号あり倍長整数

backups

既定値 : なし

値の範囲 : 符号なし整数

date-pattern

既定値 : なし

値の範囲 : yyyy-MM, yyyy-MM-dd, yyyy-MM-dd-HH, yyyy-MM-dd-HH-mm

アプリケーション統計を設定します。

flag-available に yes を指定するとアプリケーションの統計情報を出力します。

log-rotation-conf は、出力のローテーションを指定します。**SystemLog** で設定するローテーションと同様の動作を行います。但し、特権分離を使用している場合に実行される定期的なローテーションは行われません。

FileSize と **DatePattern** でそれぞれ統計ログの基準パスとして指定されたパスをもとに次の様にローテーションが行われます。

```
// FileSize の場合
<指定されたパス>.application
<指定されたパス>.application.1
<指定されたパス>.application.2
...
<指定されたパス>.application.n
// DatePattern の場合
<指定されたパス>.application
<指定されたパス>.application.2019-12-10
<指定されたパス>.application.2019-12-09
...
```

ローテーションされたファイルに統計情報のヘッダは含まれません。

```
--
例 1 :
ApplicationStatLog yes
例 2 :
ApplicationStatLog yes FileSize 10MB 10
例 3 :
ApplicationStatLog yes DatePattern yyyy-MM-dd
--
```

2.1.77 TransportStatLog

=====
対応 OS : Linux / Windows

書式 : TransportStatLog <flag-available>[<log-rotation-conf>]

flag-available

既定値 : no

値の範囲 : yes, no

log-rotation-conf

書式 : (FileSize <file-size> <backups> | DatePattern <date-pattern>)

file-size

既定値 : なし

値の範囲 : 符号あり倍長整数

backups

既定値 : なし

値の範囲 : 符号なし整数

date-pattern

既定値 : なし

値の範囲 : yyyy-MM, yyyy-MM-dd, yyyy-MM-dd-HH, yyyy-MM-dd-HH-mm

トランスポート統計を設定します。

flag-available に **yes** を指定するとトランスポートの統計情報を出力します。

log-rotation-conf は、出力のローテーションを指定します。**SystemLog** で設定するローテーションと同様の動作を行います。**SystemLog** と同様に特権分離を使用している場合は、定期的なローテーションが実行されます。

FileSize と **DatePattern** でそれぞれ統計ログの基準パスとして指定されたパスをもとに次の様にローテーションが行われます。

// **FileSize** の場合

```
<指定されたパス>.transport.tcp.service_<サービス番号>.<サービスポート番号>.th
read_<スレッド番号>
```

```
<指定されたパス>.transport.tcp.service_<サービス番号>.<サービスポート番号>.th
read_<スレッド番号>.1
```

```
<指定されたパス>.transport.tcp.service_<サービス番号>.<サービスポート番号>.th
read_<スレッド番号>.2
```

```
...
```

```
<指定されたパス>.transport.tcp.service_<サービス番号>.<サービスポート番号>.th
read_<スレッド番号>.n
```

// **DatePattern** の場合

```
<指定されたパス>.transport.tcp.service_<サービス番号>.<サービスポート番号>.th
read_<スレッド番号>
```

```
<指定されたパス>.transport.tcp.service_<サービス番号>.<サービスポート番号>.th
read_<スレッド番号>.2019-12-10
```

```
<指定されたパス>.transport.tcp.service_<サービス番号>.<サービスポート番号>.th
read_<スレッド番号>.2019-12-09
```

```
...
```

ローテーションされたファイルに統計情報のヘッダは含まれません。

--

例 1 :

TransportStatLog yes

例 2 :

TransportStatLog yes FileSize 10MB 10

例 3 :

TransportStatLog yes DatePattern yyyy-MM-dd

--

2.1.78 SystemStatLog

=====
対応 OS : Linux / Windows

書式 : SystemStatLog <flag-available>[<log-rotation-conf>]

flag-available

既定値 : yes

値の範囲 : yes, no

log-rotation-conf

書式 : (FileSize <file-size> <backups> | DatePattern <date-pattern>)

file-size

既定値 : なし

値の範囲 : 符号あり倍長整数

backups

既定値 : なし

値の範囲 : 符号なし整数

date-pattern

既定値 : なし

値の範囲 : yyyy-MM, yyyy-MM-dd, yyyy-MM-dd-HH, yyyy-MM-dd-HH-mm
=====

システム統計を設定します。

flag-available に **yes** を指定するとシステムの統計情報を出力します。

log-rotation-conf は、出力のローテーションを指定します。**SystemLog** で設定するローテーションと同様の動作を行います。但し、特権分離を使用している場合に実行される定期的なローテーションは行われません。

FileSize と **DatePattern** でそれぞれ統計ログの基準パスとして指定されたパスをもとに次の様にローテーションが行われます。

// **FileSize** の場合

<指定されたパス>.system

```
<指定されたパス>.system.1
<指定されたパス>.system.2
...
<指定されたパス>.system.n
// DatePattern の場合
<指定されたパス>.system
<指定されたパス>.system.2019-12-10
<指定されたパス>.system.2019-12-09
...
```

ローテーションされたファイルに統計情報のヘッダは含まれません。

```
--
例 1 :
SystemStatLog yes
例 2 :
SystemStatLog yes FileSize 10MB 10
例 3 :
SystemStatLog yes DatePattern yyyy-MM-dd
--
```

2.1.79 FileOperationLog

=====

対応 OS : Linux / Windows

書式 : FileOperationLog <flag-available>[<log-rotation-conf>[<log-path>]]

flag-available

既定値 : no

値の範囲 : yes, no

log-rotation-conf

書式 : (FileSize <file-size> <backups> | DatePattern <date-pattern>)

file-size

既定値 : なし

値の範囲 : 符号あり倍長整数

backups

既定値 : なし

値の範囲 : 符号なし整数

date-pattern

既定値 : なし

値の範囲 : yyyy-MM, yyyy-MM-dd, yyyy-MM-dd-HH, yyyy-MM-dd-HH-mm

log-path

既定値 : なし

値の範囲 : ファイルシステムのパス文字列

ファイル操作をロギングする機能を設定します。

flag-available に yes を指定するとファイル操作のログを出力します。

log-rotation-conf は、出力のローテーションを指定します。SystemLog で設定するローテーションと同様の動作を行います。

FileSize と DatePattern でそれぞれ統計ログの基準パスとして指定されたパスをもとに次の様にローテーションが行われます。

// FileSize の場合

<ログのパス>

<ログのパス>.1

<ログのパス>.2

...

<ログのパス>.n

// DatePattern の場合

<ログのパス>

<ログのパス>.2019-12-10

<ログのパス>.2019-12-09

...

ローテーションされたファイルに統計情報のヘッダは含まれません。

log-path は、ログを出力するファイルのパスを指定します。指定しない場合は、次の値が使用されます。

/var/log/hcpd.file.operation.log

C:/ProgramData/Clealink/HCP Tools/hcpd.file.operation.log (Windows)

ファイルの操作ログでは、次のファイル I/O 処理を記録します。

- ファイル読み込み終了
- ファイル書き込み終了
- ファイル削除
- ディレクトリ作成
- ファイルリネーム
- ハードリンク作成

HCP tools コマンドリファレンス

- シンボリックリンク作成
- ファイルリスト参照

また、アプリケーションの送達確認を含めた記録として次の様な情報も記録されます。

- ファイル転送アップロード完了
- ファイル転送ダウンロード完了
- ファイル削除完了
- ディレクトリ作成完了
- ファイルリネーム完了
- ハードリンク作成完了
- シンボリックリンク作成完了

共通する情報として、次の項目が記録されます。

- 日時
- アクセス元 IP 及びポート番号
- ユーザ名

ファイルパスは、各処理毎に固有の内容が記録されます。

ログは次の書式で出力されます。

```
=====
書式:
yyyy/mm/dd HH:MM:SS.<usec> <remote-ip> <username> <hcp-operation-name>[ <
sub-operation-label>] <path>...
yyyy/mm/dd HH:MM:SS.<usec> <remote-ip> <username> <file-io-operation-name
>\[<hcp-operation-name>\] <path>...
-----
usec
値の範囲 : マイクロ秒(000000 - 999999)
-----
remote-ip
値の範囲 : アクセス元 IP 及びポート番号
-----
username
値の範囲 : ユーザ名 (/etc/hcp/users に定義のユーザ名) 及び認証方式
-----
hcp-operation-name
値の範囲 : FT, FR, LR, DC, FM, FL
-----
```

sub-operation-label

値の範囲 : U, D, H, S

file-io-operation-name

値の範囲 : FileRead, FileWritten, FileDeleted, DirectoryCreated,
FileRenamed, LinkCreated, SymbolicLinkCreated, ListFilesRawFormat
=====

hcp-operation-name は、次のアプリケーションの種別を区別するラベルを表します。

- FT (ファイル転送)
- FR (ファイル削除)
- LR (ファイル一覧 ls, dir 出力)
- DC (ディレクトリ作成)
- FM (ファイル移動)
- FL (リンク作成)

sub-operation-label は、特定のアプリケーションの種別で処理の種類を分類するためのラベルを表します。

- U (アップロード。FT で表示)
- D (ダウンロード。FT で表示)
- H (ハードリンク作成。FL で表示)
- S (シンボリックリンク作成。FL で表示)

file-io-operation-name は、ファイルの I/O 操作の種別を区別するラベルを表します。

- FileRead (ファイル読み込み終了)
- FileWritten (ファイル書き込み終了)
- FileDeleted (ファイル削除実施済み)
- DirectoryCreated (ディレクトリ作成済み)
- FileRenamed (ファイル名変更済み)
- LinkCreated (ハードリンク作成済み)
- SymbolicLinkCreated (シンボリックリンク作成済み)
- ListFilesRawFormat (ls もしくは dir 実行予定)

--

出力例:

```
2020/01/31 10:34:52.277120 127.0.0.1:51660 user[PAM] FileWritten[FT] /home/user/file_nodiskio_0
```



```

2020/01/31 10:34:52.277175 127.0.0.1:51660 user[PAM] FT U /home/user/file
_nodiskio_0
2020/01/31 10:35:14.946750 127.0.0.1:51662 user[PAM] FileRead[FT] /home/u
ser/file_nodiskio_0
2020/01/31 10:35:15.002770 127.0.0.1:51662 user[PAM] FT D /home/user/file
_nodiskio_0
2020/01/31 10:35:47.713558 127.0.0.1:51664 user[PAM] FileDeleted[FR] /hom
e/user/stat.3.log
2020/01/31 10:35:47.765413 127.0.0.1:51664 user[PAM] FR /home/user/stat.3.
log
2020/01/31 10:38:45.686206 127.0.0.1:51670 user[PAM] DirectoryCreated[DC]
/home/user/hmkdir13
2020/01/31 10:38:45.789370 127.0.0.1:51670 user[PAM] DC /home/user/hmkdir
13
2020/01/31 10:39:22.411968 127.0.0.1:51674 user[PAM] FileRenamed[FM] /hom
e/user/stat.log /home/user/stat2.log
2020/01/31 10:39:22.463710 127.0.0.1:51674 user[PAM] FM /home/user/stat.1
og /home/user/stat2.log
2020/01/31 10:40:00.087660 127.0.0.1:51678 user[PAM] SymbolicLinkCreated
[FL] /home/user/stat2.log /home/user/stat.log
2020/01/31 10:40:00.165831 127.0.0.1:51678 user[PAM] FL S /home/user/stat
2.log /home/user/stat.log
2020/01/31 10:40:13.693415 127.0.0.1:51680 user[PAM] LinkCreated[FL] /hom
e/user/stat2.log /home/user/stat.h.log
2020/01/31 10:40:13.746160 127.0.0.1:51680 user[PAM] FL H /home/user/stat
2.log /home/user/stat.h.log
2020/02/06 13:54:21.282066 127.0.0.1:50186 user[PAM] ListFilesRawFormat[L
R] /home/user/hmkdir4
2020/02/06 13:54:21.282104 127.0.0.1:50186 user[PAM] ListFilesRawFormat[L
R] /home/user/hmkdir5
--

```

--

例 1 :

FileOperationLog yes

例 2 :

FileOperationLog yes FileSize 10MB 10

例 3 :

FileOperationLog yes DatePattern yyyy-MM-dd

例 4 :

FileOperationLog yes FileSize 10MB 10 /var/tmp/hcpd.file.operation.log

--

2.1.80 CallbackScript

=====

対応 OS : Linux / Windows

書式 : CallbackScript <flag-available>[<script-path>[<data-store-path>]]

flag-available

既定値 : no

値の範囲 : yes, no

script-path

既定値 : なし

値の範囲 : ファイルシステムのパス文字列

data-store-path

既定値 : なし

値の範囲 : ファイルシステムのパス文字列

=====

アプリケーションの実行が終了したタイミングで呼び出すスクリプト（プログラム）を設定します。

flag-available に yes を指定するとスクリプトの呼出を有効にします。

script-path は、スクリプトのパスを指定します。チルダ(~)を先頭に使用すると、アプリケーションを実行しているユーザ（認証されたユーザ）のホームディレクトリを展開して使用します。

data-store-path は、スクリプトの実行時に与えられるデータ（パラメータ情報や実行結果）を保存するパスを指定します。チルダ(~)を先頭に使用すると、アプリケーションを実行しているユーザ（認証されたユーザ）のホームディレクトリを展開して使用します。

全てのアプリケーションの処理が終了すると、その直後に次の様な書式でコマンドを実行します。

=====

コマンド書式 : <script-path> <exit-code> <start-date-and-time> <end-date-and-time> <remote-ip> <username> <hcp-operation-name> <param-saved-path> <output-saved-path>

script-path

値の範囲 : ファイルシステムのパス文字列（チルダ展開済み）

exit-code

値の範囲 : 終了コード

start-date-and-time

値の範囲 : 次の書式の日時 YYYY/MM/DD hh:mm:ss

end-date-and-time

値の範囲 : 次の書式の日時 YYYY/MM/DD hh:mm:ss

remote-ip

値の範囲 : アクセス元 IP アドレス及びポート番号

username

値の範囲 : ユーザ名 (/etc/hcp/users に定義のユーザ名) 及び認証方式

hcp-operation-name

値の範囲 : hcp, hrm, hcp-ls, hmkdir, hpwd, hmv, hln, transfer, remove,
listraw, mkd, pwd, move, link, cwd

param-saved-path

値の範囲 : ファイルシステムのパス文字列

output-saved-path

値の範囲 : ファイルシステムのパス文字列

=====

script-path には、hcpd.conf で設定されたスクリプトパスにチルダ(~)展開が実施されたパスが使用されます。

exit-code には、実行したアプリケーションの終了結果を表す理由コードが渡されます。アプリケーション統計に記録される理由コードと同じものが渡されます。

start-date-and-time 及び **end-date-and-time** には、それぞれアプリケーションの処理を開始した日時と終了した日時が渡されます。

hcp-operation-name には、次のアプリケーション (API オペレーション) の種別を区別するラベルが渡されます。

- hcp (ファイル転送コマンド)
- hrm (ファイル削除コマンド)
- hcp-ls (ファイル一覧コマンド)
- hmkdir (ディレクトリ作成コマンド)
- hpwd (ワーキングディレクトリ出力コマンド)
- hmv (ファイル移動コマンド)
- hln (リンク作成コマンド)
- transfer (API ファイル転送処理)
- remove (API ファイル削除処理)

- listraw (API ファイル一覧処理)
- mkd (API ディレクトリ作成処理)
- pwd (API ワーキングディレクトリ取得処理)
- move (API ファイル移動処理)
- link (API リンク作成処理)

param-saved-path には、入力パラメータの情報が保存されたファイルのパスが渡されます。このファイルには、クライアントで確認できる実行結果記録(.hcp.out)のうちサーバ上で特定された指定されたパス情報と処理のオプションが記録されます。

--

出力例:

```
[user@localhost ~]$ cat .hcp/callback/hcp.cb.20200206_152252_468.13521.param
OPT copy_mode ALLCOPY
OPT overwrite_mode FORCE
OPT fail_action_mode HALT
OPT preserve_permission no
OPT recursive yes
OPT any_dirs no
OPT regex no
OPT verify_payload no
OPT copy_symlink no
OPT follow_symlink no
OPT no_copy_empty_file no
OPT no_copy_empty_dir no
OPT no_copy_dot_file no
OPT no_copy_dot_dir no
OPT copy_hidden no
OPT check_archive no
OPT resuming no
OPT no_app_io yes num_files 1 file_size 1024
OPT no_sess_io no
SRC /home/user
--
```

output-saved-path には、実行結果の情報が保存されたファイルのパスが渡されます。このファイルには、クライアントで確認できる実行結果記録(.hcp.out)のうち各ファイルの実行結果に相当する内容が記録されます。

--

出力例:

```
[user@localhost ~]$ cat .hcp/callback/hcp.cb.20200206_152252_468.13521.out
```

```
OK 0000 FT 00000001 /home/user/file_nodiskio_0
```

```
--
```

```
--
```

例 :

```
CallbackScript /var/tmp/hcp_callback.sh /var/tmp/hcp_callback
```

```
--
```

2.1.81 EnsureDestinationInFileTransfer

```
=====
```

対応 OS : Linux / Windows

書式 : EnsureDestinationInFileTransfer <flag-available>

```
-----
```

flag-available

既定値 : yes

値の範囲 : yes, no

```
=====
```

ファイル転送で宛先のディレクトリが存在しない場合に、作成を試みるか制御します。本オプションは、1.1.0 以降のソフトウェアでは常に無効(no)として動作します。

```
--
```

例 :

```
EnsureDestinationInFileTransfer no
```

```
--
```

2.1.82 StatLogPerUserInPrivilegeSeparation

```
=====
```

対応 OS : Linux / Windows

書式 : StatLogPerUserInPrivilegeSeparation <flag-available>

```
-----
```

flag-available

既定値 : no

値の範囲 : yes, no

```
=====
```

特権分離を使用している場合の統計ログをユーザ毎に記録するか設定します。

```
--
```

例 :

```
StatLogPerUserInPrivilegeSeparation yes
```

```
--
```

2.2 クライアントコマンド共通設定項目

クライアントコマンドに共通して指定できる設定を記述します。

| 項目名 | 説明 |
|---|------------------------|
| ProtocolVersion | プロトコルバージョン(2 固定) |
| RequireServerCertificateSecurity | サーバ証明書セキュリティを要求 |
| RejectFallbackServerCertificateSecurity | サーバ証明書セキュリティダウングレード禁止 |
| IgnoreCertificateCNInvalid | 共通名(Common name)を検査しない |
| IgnoreCertificateDateInvalid | 証明書の期限を検査しない |
| IgnoreUnknownCA | CA 及び証明書連鎖を検査しない |
| IgnoreRevocation | 証明書の失効を検査しない |
| CACertificateFile | CA 証明書ファイル |
| CACertificatePath | CA 証明書探索ディレクトリ (予約) |
| CARevocationFile | CRL ファイル |
| CARevocationPath | CRL 探索ディレクトリ (予約) |
| OCSPRevocationEnabled | OCSP (オンライン証明書失効検査) 機能 |
| StrictHostKeyChecking | サーバホスト鍵受入ポリシー設定 |
| LocalPasswordAuthentication | LPA 認証 |
| PAMAuthentication | PAM 認証 |
| PubkeyAuthentication | RSA 認証 |
| WinLogonUserAuthentication | Windows 認証 |
| PrivateKeySearchDir | 秘密鍵探索ディレクトリ |
| PrivateKeyFile | 秘密鍵ファイル |
| AcceptableCryptMethod | 暗号方式 |
| AcceptableDigestMethod | ダイジェスト方式 |
| DisableDataIntegrityChecking | 暗号通信メッセージ検査無効設定 |
| AcceptDataIntegrityCheckingOnRejection | 暗号通信メッセージ検査無効拒否受入設定 |
| TransportCharEncoding | トランスポート文字エンコーディング |

| | |
|---------------------------------|-------------------------|
| HostEncoding | ホスト文字エンコーディング |
| CompressLevel | 圧縮レベル |
| HeaderCompress | ヘッダ圧縮 |
| ContentCompress | コンテンツ圧縮 |
| MaxConcurrentThread | 使用スレッド数制限(Linux) |
| MaxReceiveFileSize | 受信ファイルサイズ制限 |
| MaxSendFileSize | 送信ファイルサイズ制限 |
| MaxRequestFileEntryAtOnce | 一括ファイル要求制限 |
| MaxBufferSize | ペイロードバッファメモリ割当制限 |
| MaxReceiveRate | 受信レート制限 (接続単位) |
| MaxSendRate | 送信レート制限 (接続単位) |
| InitHeaderBlockSize | 初期ヘッダブロックサイズ |
| InitContentBlockSize | 初期コンテンツブロックサイズ |
| MaxHeaderBlockSize | 最大ヘッダブロックサイズ |
| MaxContentBlockSize | 最大コンテンツブロックサイズ |
| FileLock | ファイルロック |
| FileLockRetries | ファイルロック再試行回数 |
| FileLockRetryInterval | ファイルロック試行間隔 (秒) |
| TransportTimeout | トランスポートタイムアウト |
| ApplicationLog | ログ設定 |
| ApplicationLogLevel | ログレベル |
| ApplicationStatLog | アプリケーション統計 |
| TransportStatLog | トランスポート統計 |
| UDPTransportExtensionBufferSize | HpFP トランスポート拡張バッファサイズ設定 |

2.2.1 RequireServerCertificateSecurity

=====

対応 OS : Linux / Windows

書式 : RequireServerCertificateSecurity <flag-available>

flag-available

既定値 : yes

値の範囲 : yes, no

=====

サーバに対してサーバ証明書セキュリティ機能によるセキュリティ通信を要求するか設定します。yes を指定した場合、サーバが同機能を使用する通信を設定していない場合、接続を拒否します。

--

例 :

RequireServerCertificateSecurity no

--

2.2.2 RejectFallbackServerCertificateSecurity

=====

対応 OS : Linux / Windows

書式 : RejectFallbackServerCertificateSecurity <flag-available>

flag-available

既定値 : yes

値の範囲 : yes, no

=====

サーバ証明書セキュリティ機能を使用したセキュリティ通信を確立できない場合に、安全でない通信（平文通信）にフォールバックする動作を禁止します。

--

例 :

RejectFallbackServerCertificateSecurity no

--

2.2.3 IgnoreCertificateCNInvalid

=====

対応 OS : Linux / Windows

書式 : IgnoreCertificateCNInvalid <flag-available>

flag-available

既定値 : yes

値の範囲 : yes, no

=====

サーバ証明書の共通名(Common Name)の検査を無効にする設定を行います。yes を指定すると、サーバ証明書の検証を行う際に証明書の共通名が接続に指定したアドレス（もしくはホスト名）と一致するかどうかの確認を行いません。


```
--  
例 :  
IgnoreCertificateCNInvalid no  
--
```

2.2.4 IgnoreCertificateDateInvalid

```
=====
```

対応 OS : Linux / Windows
書式 : IgnoreCertificateDateInvalid <flag-available>

```
-----
```

flag-available
既定値 : no
値の範囲 : yes, no

```
=====
```

証明書の有効期間の検査を無効にする設定を行います。yes を指定すると、証明書の検証を行う際に証明書の有効期間(NotBefore 及び NotAfter)の検査を行いません。

```
--  
例 :  
IgnoreCertificateDateInvalid yes  
--
```

2.2.5 IgnoreUnknownCA

```
=====
```

対応 OS : Linux / Windows
書式 : IgnoreUnknownCA <flag-available>

```
-----
```

flag-available
既定値 : no
値の範囲 : yes, no

```
=====
```

CA 証明書の検査を無効にする設定を行います。yes を指定すると、サーバ証明書の中間証明書およびルート証明書に関する検証を行いません。

```
--  
例 :  
IgnoreUnknownCA yes  
--
```

2.2.6 IgnoreRevocation

```
=====
```

対応 OS : Linux / Windows
書式 : IgnoreRevocation <flag-available>

flag-available

既定値 : no

値の範囲 : yes, no
=====

証明書失効検査（CRL 検査）を無効にする設定を行います。yes を指定すると、証明書の失効状態の確認を行いません。

--

例 :

IgnoreRevocation yes

--

2.2.7 CACertificateFile

=====

対応 OS : Linux / Windows

書式 : CACertificateFile <file-path>

file-path

既定値 :

/etc/hcp/cacert.pem

C:/ProgramData/Clealink/HCP Tools/cacert.pem (Windows)

値の範囲 : ファイルシステムのパス文字列
=====

サーバ証明書の検証に使用する CA 証明書が保管されているファイルのパスを指定します。

--

例 :

CACertificateFile /etc/hcp/cacert.pem

--

PEM 形式の証明書をサポートします。

2.2.8 CACertificatePath (予約)

=====

対応 OS : Linux / Windows

書式 : CACertificatePath <dir-path>

dir-path

既定値 : yes

/etc/ssl

C:/ProgramData/Clealink/HCP Tools/ssl (Windows)

値の範囲 : ファイルシステムのパス文字列

サーバ証明書の検証に使用する CA 証明書が保管されているディレクトリのパスを指定します。

--

例 :

CACertificatePath /etc/ssl

--

PEM 形式の証明書をサポートします。

2.2.9 CARevocationFile

対応 OS : Linux / Windows

書式 : CACRevocationFile <file-path>

file-path

既定値 : yes

/etc/hcp/crl.pem

C:/ProgramData/Clealink/HCP Tools/crl.pem (Windows)

値の範囲 : ファイルシステムのパス文字列

サーバ証明書の検証に使用する失効リスト(CRL)が保管されているファイルを指定します。

--

例 :

CACRevocationFile /etc/hcp/crl.pem

--

PEM 形式の失効リスト(CRL)をサポートします。

2.2.10 CACRevocationPath (予約)

対応 OS : Linux / Windows

書式 : CACRevocationPath <dir-path>

dir-path

既定値 : yes

/etc/ssl

C:/ProgramData/Clealink/HCP Tools/ssl (Windows)

値の範囲 : ファイルシステムのパス文字列

サーバ証明書の検証に使用する失効リスト(CRL)が保管されているディレクトリを指定します。

--

例 :

CARevocationPath /etc/ssl

--

PEM 形式の失効リスト(CRL)をサポートします。

2.2.11 OCSPRevocationEnabled

対応 OS : Linux / Windows

書式 : OCSPRevocationEnabled <flag-available>

flag-available

既定値 : yes

値の範囲 : yes, no

サーバ証明書の検証に使用する OCSP (オンライン証明書失効検査プロトコル) を設定します。yes を指定すると有効にします。

--

例 :

OCSPRevocationEnabled no

--

2.2.12 StrictHostKeyChecking

対応 OS : Linux / Windows

書式 : StrictHostKeyChecking <switch>

switch

既定値 : ask

値の範囲 : ask, yes, no

サーバホスト鍵の受け入れポリシーを指定します。ask を指定すると、未知の鍵を受信した場合に受け入れるかどうか確認を行います。yes を指定すると、受信した鍵が未知の場合は処理を中断します。no を指定すると、受信した鍵が未知の場合に確認を行わずに受けれて処理を継続します。

```
--  
例 :  
StrictHostKeyChecking no  
--
```

2.2.13 LocalPasswordAuthentication

```
=====
```

対応 OS : Linux / Windows
書式 : LocalPasswordAuthentication <flag-available>

```
-----
```

flag-available
既定値 : yes
値の範囲 : yes, no

```
=====
```

LPA 認証の設定を行います。no を指定した場合、サーバから認証を要求された際に LPA 認証を行いません。

```
--  
例 :  
LocalPasswordAuthentication no  
--
```

2.2.14 PAMAuthentication

```
=====
```

対応 OS : Linux / Windows
書式 : PAMAuthentication <flag-available>

```
-----
```

flag-available
既定値 : yes
値の範囲 : yes, no

```
=====
```

PAM 認証の設定を行います。no を指定した場合、サーバから認証を要求された際に PAM 認証を行いません。

```
--  
例 :  
PAMAuthentication no  
--
```

2.2.15 PubkeyAuthentication

```
=====
```

対応 OS : Linux / Windows
書式 : PubkeyAuthentication <flag-available>

flag-available

既定値 : yes

値の範囲 : yes, no
=====

RSA 認証の設定を行います。no を指定した場合、サーバから認証を要求された際に RSA 認証を行いません。

--

例 :

PubkeyAuthentication no

--

2.2.16 WinLogonUserAuthentication

=====

対応 OS : Linux / Windows

書式 : WinLogonUserAuthentication <flag-available>

flag-available

既定値 : yes

値の範囲 : yes, no
=====

Windows 認証の設定を行います。no を指定した場合、サーバから認証を要求された際に Windows 認証を行いません。

--

例 :

WinLogonUserAuthentication no

--

2.2.17 PrivateKeySearchDir

=====

対応 OS : Linux / Windows

書式 : PrivateKeySearchDir <flag-available>

flag-available

既定値 :

/etc/hcp/keys

C:/ProgramData/Clealink/HCP Tools/keys (Windows)

値の範囲 : ファイルシステムのパス文字列
=====

RSA 認証で使用するユーザの秘密鍵を特定するために探索するディレクトリを指定します。

--

例 :

```
PrivateKeySearchDir /etc/hcp/keys
```

--

指定されたディレクトリで次の様なファイル名を秘密鍵が保管されたファイルと見做して探索します。

<ユーザ名>.key

このユーザ名は、接続先のサーバのユーザ名 (-u オプションで指定するユーザ名もしくはログインプロンプトで入力するユーザ名) ではなく、ローカルコンピュータのユーザ名が使用されます。ローカルコンピュータのユーザ名がサーバのユーザ名と異なる場合はご注意ください。

秘密鍵と同じディレクトリに次の拡張子を持つ同じユーザ名のファイルが存在する場合は、クライアント証明書と見做して認証を行います。

- crt

- cer

PEM 形式の秘密鍵及び証明書をサポートします。

2.2.18 PrivateKeyFile

=====
対応 OS : Linux / Windows

書式 : PrivateKeyFile <file-path>

file-path

既定値 :

~/.hcp/id_rsa

~/_hcp/id_rsa (Windows)

値の範囲 : ユーザディレクトリを表すパス表記(~、チルダ)を含むファイルパス

=====
RSA 認証で使用するユーザの秘密鍵を特定するために探索するユーザホームディレクトリ内の鍵保管ファイルのパスを指定します。

--

例 :

```
PrivateKeySearchDir ~/.hcp/id_rsa
```

--

特定したファイルに次の拡張子が付加されたファイルが存在する場合は、クライアント証明書と見做して認証を行います。

- crt
- cer

PEM 形式の秘密鍵及び証明書をサポートします。

2.2.19 AcceptableCryptMethod

=====
対応 OS : Linux / Windows

書式 : AcceptableCryptMethod <method-names>

method-names

書式 : <method-name>[...]

既定値 : AES256/CBC AES128/CBC

method-name

値の範囲 : PLAIN, AES128/CBC, AES192/CBC, AES256/CBC
=====

暗号アルゴリズムを設定します。

--

例 :

AcceptableCryptMethod AES256/CBC PLAIN

--

サーバとの間で通信するメッセージの暗号化に使用されます。どのアルゴリズムが使用されるかは、サーバの設定と併せて評価し、一致したアルゴリズムを使用します。

2.2.20 AcceptableDigestMethod

=====
対応 OS : Linux / Windows

書式 : AcceptableDigestMethod <method-names>

method-names

書式 : <method-name>[...]

既定値 : SHA256 SHA160

method-name

値の範囲 : NONE, SHA160, SHA224, SHA256, SHA384, SHA512, MD5, MM32, MM128
=====

ダイジェストアルゴリズムを設定します。

```
--
例 :
AcceptableDigestMethod SHA256 MM128 NONE
--
```

サーバとの間で通信するメッセージ、ファイルやそのデータブロックの検証に使用されます。どのアルゴリズムが使用されるかは、サーバの設定と併せて評価し、一致したアルゴリズムを使用します。

また、暗号化による通信を行う場合、セキュリティダイジェストではないアルゴリズム (MD5, MM32, MM128) は、指定されなかったものとして扱われます。

2.2.21 DisableDataIntegrityChecking

```
=====
対応 OS : Linux / Windows
書式 : DisableDataIntegrityChecking <flag-available>
```

```
-----
flag-available
既定値 : no
値の範囲 : yes, no
=====
```

サーバとの間で暗号化通信を行う場合に、MAC による通信メッセージの検査 (データ完全性検査) を無効にする (省略する) ことをサーバに要求するか設定します。

yes を指定すると、サーバが許可した場合に通信メッセージの検査を行わない暗号化通信を行います。サーバが拒否した場合は、後述の設定項目 `AcceptDataIntegrityCheckingOnRejection` の設定に従って動作します。

通常は no (デフォルト) で使用してください。前述の通信メッセージの検査が省略されることを理解したうえでご利用ください。通常、暗号通信のパフォーマンス向上の目的で使用します。

```
--
例 :
DisableDataIntegrityChecking yes
--
```

2.2.22 AcceptDataIntegrityCheckingOnRejection

```
=====
対応 OS : Linux / Windows
```

書式 : AcceptDataIntegrityCheckingOnRejection <flag-available>

flag-available

既定値 : yes

値の範囲 : yes, no
=====

サーバとの間で暗号化通信を行う場合に、MAC による通信メッセージの検査を無効にする（省略する）要求が拒否された場合に、通信を継続か中断するか設定します。

yes を指定すると、通信メッセージの検査を行って通信を継続します。no を指定すると、通信を中断してアプリケーションを終了します。

--

例 :

AcceptDataIntegrityCheckingOnRejection no

--

2.2.23 TransportCharEncoding

=====

対応 OS : Linux / Windows

書式 : TransportCharEncoding <encodings>

encodings

書式 : <encoding>[...]

既定値 : UTF8

encoding

値の範囲 : US-ASCII, UTF8, UTF16, UTF32
=====

トランスポートで使用する文字列のエンコーディング方式を指定します。

--

例 :

TransportCharEncoding UTF8 UTF16 US-ASCII

--

サーバとの間で交換するファイルパスなどの文字列に適用されます。どのエンコーディングが使用されるかは、サーバの設定と併せて評価し、一致したエンコーディングを使用します。

2.2.24 HostEncoding

=====
対応 OS : Linux / Windows

書式 : HostEncoding <encoding>

encoding

既定値 :

UTF-8

CP932 (Windows)

値の範囲 : システムかつエンコーディング変換ライブラリでサポートされるエンコーディング名 (プラットフォーム依存)

=====
ホストで使用されている文字列エンコーディングを指定します。

--

例 :

HostEncoding EUC-JP

--

2.2.25 CompressLevel

=====
対応 OS : Linux / Windows

書式 : CompressLevel <compress-level>

compress-level

既定値 : -1

値の範囲 : -1, 0 - 9
=====

通信メッセージを圧縮するレベルを指定します。

--

例 :

CompressLevel 9

--

-1 を指定した場合、内部的にレベル 6 が選択されます。

0 を指定した場合、圧縮は行いません。

2.2.26 HeaderCompress

=====
対応 OS : Linux / Windows

書式 : HeaderCompress <flag-available>

flag-available

既定値 : yes

値の範囲 : yes, no
=====

ファイル要求などのヘッダ情報を圧縮する設定を行います。

--

例 :

HeaderCompress no

--

2.2.27 ContentCompress

=====

対応 OS : Linux / Windows

書式 : ContentCompress <flag-available>

flag-available

既定値 : yes

値の範囲 : yes, no
=====

ファイルのデータを圧縮する設定を行います。

--

例 :

ContentCompress no

--

2.2.28 MaxConcurrentThread

=====

対応 OS : Linux

書式 : MaxConcurrentThread <max-threads>

max-threads

既定値 : 0

値の範囲 : 符号付整数
=====

スレッド数制限を設定します。

2.2.29 MaxReceiveFileSize

=====

対応 OS : Linux / Windows

書式 : MaxReceiveFileSize <file-size>

file-size

既定値 : 100GB

値の範囲 : 符号付き倍長整数
=====

受信を許可する最大のファイルサイズを設定します。

--

例 :

MaxReceiveFileSize 1GB

--

2.2.30 MaxSendFileSize

=====

対応 OS : Linux / Windows

書式 : MaxSendFileSize <file-size>

file-size

既定値 : 100GB

値の範囲 : 符号付き倍長整数
=====

送信を許可する最大のファイルサイズを設定します。

--

例 :

MaxSendFileSize 1GB

--

2.2.31 MaxRequestFileEntryAtOnce

=====

対応 OS : Linux / Windows

書式 : MaxRequestFileEntryAtOnce <max-file-req-at-once>

max-file-req-at-once

既定値 : 50

値の範囲 : 符号付き整数
=====

ファイル要求の一括送信を許可する最大の数を設定します。

--

例 :

MaxRequestFileEntryAtOnce 1000

--

2.2.32 MaxBufferSize

```
=====
対応 OS : Linux / Windows
書式 : MaxBufferSize <max-buf-size>
```

```
-----
max-buf-size
既定値 : 1GB
値の範囲 : 符号なし倍長整数
=====
```

ファイルのデータ処理に使用できる最大のメモリバッファサイズを設定します。

```
--
例 :
MaxBufferSize 1GB
--
```

2.2.33 MaxReceiveRate

```
=====
対応 OS : Linux / Windows
書式 : MaxReceiveRate <bandwidth>
```

```
-----
bandwidth
既定値 : 10Gbit
値の範囲 : 符号なし倍長整数
=====
```

トランスポートの受信帯域のシェーピング（制限）を設定します。

```
--
例 :
MaxReceiveRate 1Gbit
--
```

2.2.34 MaxSendRate

```
=====
対応 OS : Linux / Windows
書式 : MaxSendRate <bandwidth>
```

```
-----
bandwidth
既定値 : 10Gbit
値の範囲 : 符号なし倍長整数
=====
```

トランスポートの送信帯域のシェーピング（制限）を設定します。

```
--  
例 :  
MaxSendRate 1Gbit  
--
```

2.2.35 InitHeaderBlockSize

```
=====
```

対応 OS : Linux / Windows
書式 : InitHeaderBlockSize <block-size>

```
-----
```

block-size
既定値 : 4KB
値の範囲 : 符号なし倍長整数

```
=====
```

初期ヘッダブロックのサイズを設定します。

```
--  
例 :  
InitHeaderBlockSize 10KB  
--
```

ファイル要求などのメッセージを複数含むヘッダブロックが形成可能な上限のサイズを与えます。このオプションは通信開始直後に適用されます。

2.2.36 InitContentBlockSize

```
=====
```

対応 OS : Linux / Windows
書式 : InitContentBlockSize <block-size>

```
-----
```

block-size
既定値 : 512KB
値の範囲 : 符号なし倍長整数

```
=====
```

初期コンテンツブロックのサイズを設定します。

```
--  
例 :  
InitContentBlockSize 100KB  
--
```

ファイルのデータを複数含むコンテンツブロックが形成可能な上限のサイズを与えます。このオプションは通信開始直後に適用されます。

2.2.37 MaxHeaderBlockSize

=====
対応 OS : Linux / Windows

書式 : MaxHeaderBlockSize <block-size>

block-size

既定値 : 10MB

値の範囲 : 符号なし倍長整数
=====

ヘッダブロックのサイズを拡張する最大のサイズを設定します。

--

例 :

MaxHeaderBlockSize 100KB

--

通信を開始すると消費帯域を評価して形成可能なヘッダブロックのサイズを増減させます。本オプションは、このサイズを増加させることができる上限値を与えます。

2.2.38 MaxContentBlockSize

=====
対応 OS : Linux / Windows

書式 : MaxContentBlockSize <block-size>

block-size

既定値 : 100MB

値の範囲 : 符号なし倍長整数
=====

コンテンツブロックのサイズを拡張する最大のサイズを設定します。

--

例 :

MaxContentBlockSize 1MB

--

通信を開始すると消費帯域を評価して形成可能なコンテンツブロックのサイズを増減させます。本オプションは、このサイズを増加させることができる上限値を与えます。

2.2.39 FileLock

hcpd 設定項目参照

2.2.40 FileLockRetries

hcpd 設定項目参照

2.2.41 FileLockRetryInterval

hcpd 設定項目参照

2.2.42 TransportTimeout

hcpd 設定項目参照

2.2.43 ApplicationLog

=====
対応 OS : Linux / Windows

書式 : ApplicationLog <log-level>[<log-rotation-conf>]

log-level

既定値 : INFO

値の範囲 : EMERG, ALERT, CRIT, ERR, WARNING, INFO, DEBUG

log-rotation-conf

書式 : (FileSize <file-size> <backups> | DatePattern <date-pattern>)

file-size

既定値 : なし

値の範囲 : 符号あり倍長整数

backups

既定値 : なし

値の範囲 : 符号なし整数

date-pattern

既定値 : なし

値の範囲 : yyyy-MM, yyyy-MM-dd, yyyy-MM-dd-HH, yyyy-MM-dd-HH-mm
=====

アプリケーションのログ設定を行います。

log-level は、ログのレベルを指定します。

log-rotation-conf は、ログのローテーションを指定します。SystemLog で設定するローテーションと同様の動作を行います。但し、定期的なローテーションは行いません。

```
--
例 1 :
ApplicationLog WARNING FileSize 10MB 10
例 2 :
ApplicationLog WARNING DatePattern yyyy-MM-dd
例 3 :
ApplicationLog WARNING // ApplicationLogLevel に同じ
--
```

2.2.44 ApplicationLogLevel

```
=====
対応 OS : Linux / Windows
書式 : ApplicationLogLevel <log-level>
-----
log-level
既定値 : INFO
値の範囲 : EMERG, ALERT, CRIT, ERR, WARNING, INFO, DEBUG
=====
```

アプリケーションのログレベルを設定します。

ApplicationLog と共に記述した場合は、ログレベルの値のみ上書きされます。

```
--
例 :
ApplicationLogLevel WARNING
--
```

2.2.45 ApplicationStatLog

```
=====
対応 OS : Linux / Windows
書式 : ApplicationStatLog <flag-available>[ <log-rotation-conf>]
-----
flag-available
既定値 : no
値の範囲 : yes, no
-----
log-rotation-conf
書式 : ( FileSize <file-size> <backups> | DatePattern <date-pattern> )
-----
file-size
既定値 : なし
値の範囲 : 符号あり倍長整数
-----
backups
```

既定値 : なし
値の範囲 : 符号なし整数

date-pattern

既定値 : なし
値の範囲 : yyyy-MM, yyyy-MM-dd, yyyy-MM-dd-HH, yyyy-MM-dd-HH-mm

=====

アプリケーション統計を設定します。

flag-available に yes を指定するとアプリケーションの統計情報を出力します。

log-rotation-conf は、出力のローテーションを指定します。hcpd の設定項目 ApplicationStatLog と同様の動作を行います。

--
例 1 :
ApplicationStatLog yes
例 2 :
ApplicationStatLog yes FileSize 10MB 10
例 3 :
ApplicationStatLog yes DatePattern yyyy-MM-dd
--

2.2.46 TransportStatLog

=====

対応 OS : Linux / Windows
書式 : TransportStatLog <flag-available>[<log-rotation-conf>]

flag-available

既定値 : no
値の範囲 : yes, no

log-rotation-conf

書式 : (FileSize <file-size> <backups> | DatePattern <date-pattern>)

file-size

既定値 : なし
値の範囲 : 符号あり倍長整数

backups

既定値 : なし
値の範囲 : 符号なし整数

date-pattern

既定値 : なし

値の範囲 : yyyy-MM, yyyy-MM-dd, yyyy-MM-dd-HH, yyyy-MM-dd-HH-mm

=====

トランスポート統計を設定します。

flag-available に yes を指定するとトランスポートの統計情報を出力します。

log-rotation-conf は、出力のローテーションを指定します。hcpd の設定項目 TransportStatLog と同様の動作を行います。

FileSize と DatePattern でそれぞれ統計ログの基準パスとして指定されたパスをもとに次の様にローテーションが行われます。

// FileSize の場合

<指定されたパス>.transport.tcp

<指定されたパス>.transport.tcp.1

<指定されたパス>.transport.tcp.2

...

<指定されたパス>.transport.tcp.n

// DatePattern の場合

<指定されたパス>.transport.tcp

<指定されたパス>.transport.tcp.2019-12-10

<指定されたパス>.transport.tcp.2019-12-09

...

ローテーションされたファイルに統計情報のヘッダは含まれません。

--

例 1 :

TransportStatLog yes

例 2 :

TransportStatLog yes FileSize 10MB 10

例 3 :

TransportStatLog yes DatePattern yyyy-MM-dd

--

2.2.47 UDPTransportExtensionBufferSize

=====

対応 OS : Linux / Windows

書式 : UDPTransportExtensionBufferSize <ext-buf-size>

ext-buf-size

既定値 : 2GB

値の範囲 : 符号なし倍長整数 (バイト単位)

=====

HpFP(UDP)トランスポートで使用する拡張バッファのサイズを指定します。

HpFP セッションでは、遅延やロス及び通信量の増加などに伴い通信用のバッファを指定されたサイズ (UDPListenAddress の `hpf_sndbuf` 及び `hpf_rcvbuf`) に拡張します。この拡張するバッファの合計容量を指定された値で制限します。

0 を指定した場合は、この制限を行いません。

また、バッファの初期サイズ (拡張される前のバッファのサイズ) は 1MB です。

--

例 :

UDPTransportExtensionBufferSize 4GB

--

2.3 hcp.conf

| 項目名 | 説明 |
|---------|-----------------|
| Include | 外部設定ファイルのインクルード |

2.3.1 Include

=====

対応 OS : Linux / Windows

書式 : Include <file-path>

file-path

既定値 : なし

値の範囲 : ファイルシステムのパス文字列

=====

クライアントコマンドに共通して指定できる設定を記述したファイルをインクルードします。インクルードは、`hcp.conf` の任意の位置に記述できます。インクルードの記述の前に設定された同じ項目の値は上書きされます。インクルードはネストできません。

--

例 :

Include /etc/hcp/hcp-common.conf

--

2.4 hrm.conf

hrm コマンドの設定を記述します。設定内容は、hcp.conf に同じです。

2.5 hcp-ls.conf

hcp-ls コマンドの設定を記述します。設定内容は、hcp.conf に同じです。

2.6 hmkdir.conf

hmkdir コマンドの設定を記述します。設定内容は、hcp.conf に同じです。

2.7 hpwd.conf

hpwd コマンドの設定を記述します。設定内容は、hcp.conf に同じです。

2.8 hmv.conf

hmv コマンドの設定を記述します。設定内容は、hcp.conf に同じです。

2.9 hln.conf

hln コマンドの設定を記述します。設定内容は、hcp.conf に同じです。

2.10 users

hcpd デーモンが認証で認識するユーザを定義します。各行に次の書式でユーザを定義します。

書式 <username>:<auth_methods>:[<mapping_uid>]:[<mapping_gid>]:[<hpfp_config_modes>]:[<home_dir>]

username は、ユーザ名を記述します。セパレータを除く文字が使用可能で、システム上にユーザとして定義されない文字列（HCP tools 固有に定めたユーザ名）を指定することもできます。

auth_methods は、このユーザで利用を許可する認証方法をカンマ(,)区切りで列挙します。使用できる値は、次の通りです。

- LPA
- PAM
- RSA
- WLU

`mapping_uid` は、このユーザに指定したい UID を記述します。省略された場合は、ユーザ名からシステムへ問い合わせを行います。

`mapping_gid` は、このユーザに指定したい GID を記述します。

`hpfp_cong_modes` は、このユーザで許可する HpFP トランスポートの輻輳制御モードを 0 個以上指定します。MODEST、FAIR_FAST_START 及び AGGRESSIVE の三つのモードの省略表記（それぞれ、M、S 及び A）を指定します。FAIR モードはデフォルトで含まれます。

`home_dir` は、このユーザのホームディレクトリを記述します。省略された場合は、ユーザ名からシステムへ問い合わせを行います。

```
--  
例：  
hcp_user:LPA,RSA:1000:1000::/home/user  
user:PAM:::  
--
```

2.11 passwd

hcpd デーモンが LPA 認証で使用するユーザの資格情報（ハッシュ化されたパスワード）を定義します。各行に次の書式で資格情報を定義します。

書式 <username>:<hash_method>:<hash_value>

`username` は、ユーザ名を記述します。

`hash_method` は、資格情報の生成方法を次の何れかを記述します。

- md5
- sha
- sha224
- sha256
- sha384
- sha512

`hash_value` は、次の式に従って変換されたハッシュ化された文字列を記述します。

`hex_string (hash_function (<ユーザ入力パスワード>))`

`hex_string` は、バイナリデータを 16 進数文字列表記に変換する関数です。

`hash_function` は、`hash_method` で指定したダイジェスト関数です。

通常この文字列は、下記の様にオペレーティングシステムのツールを使用して作成することができます。

```
$ echo my_password | openssl dgst -md5
(stdin)= 07a9cef2...871ef
```

```
--
```

例：

```
# password is "password01"
hcp_user:sha256:4b8f353889d9a05d17946e26d014efe99407cba8bd9d0102d4aab10ce
6229043
```

```
--
```

3 統計リファレンス

3.1 アプリケーション統計

ファイル転送(hcp)やファイル削除(hrm)等の各アプリケーション実行後に、クライアントおよびサーバ上でその実行結果の情報を所定のファイルに記録します。

| 列名 | 説明 |
|--------------------|-------------------|
| Start Date_Time | 開始日時 |
| End Date_Time | 終了日時 |
| Exit | 終了コード |
| Remote Host | リモートホスト |
| User | ユーザ名 |
| Application | アプリケーション名 |
| Total | トータルデータ伝送量(バイト) |
| Payload | ファイルデータ伝送量(バイト) |
| Files | ファイル数 |
| Directories | ディレクトリ数 |
| Average Throughput | 平均スループット(bps) |
| Encoding | トランスポート文字エンコーディング |
| Security | トランスポートセキュリティ |
| Compression | 圧縮 |
| Proto | トランスポートプロトコル |

| | |
|---------|-----------------|
| Local | ローカルアドレス及びポート番号 |
| Foreign | リモートアドレス及びポート番号 |
| PID | プロセス ID |

Start Date_Time は、アプリケーションの実行開始日時が記録されます。次の書式で出力されます。

yyyy/mm/dd HH:MM::SS

End Date_Time は、アプリケーションの実行終了日時が記録されます。次の書式で出力されます。

yyyy/mm/dd HH:MM::SS

Exit は、アプリケーションの終了コードとしてファイル処理理由コード（後述）が記録されます。

Remote Host は、リモート（対向）のホストのホスト名もしくは IP アドレスが記録されます。

User は、アプリケーションを実行したユーザ及び認証方法(PAM など)が記録されます。認証を行わなかった場合は、anonymous が出力されます。

Application は、この記録を生成したアプリケーションの名前が記録されます。次の何れかが出力されます。

- hcp
- hrm
- hcp-ls
- hmkdir
- hpwd
- hmv
- hln

また、アプリケーションを実行したプロトコル（通信手順）のバージョンが括弧内に下記の様に記載されます（順にプロトコル番号、プロトコル改訂番号）。

..., hcp [1.2], ...

Total は、アプリケーション層でのデータ転送のために必要となった全バイト数（ファイル要求等の制御に必要なデータ量を含む）が記録されます。

Payload は、アプリケーション層のデータを伝送した量（ファイル転送の場合、ファイルのデータサイズの合計）が記録されます。

Files は、アプリケーション層で処理されたファイル数が記録されます（ディレクトリは含みません）。

Directories は、アプリケーション層で処理されたディレクトリ数が記録されます。

Average Throughput は、アプリケーション層のデータの平均伝送スループットが記録されます。

Encoding は、使用されたトランスポート文字エンコーディングが記録されます。

Security は、使用されたトランスポートセキュリティが記録されます。ダイジェストアルゴリズムの用途が次の名称で付記されます。

- **MAC** (セッション暗号時のメッセージデータの検査)
- **Chunk** (ファイルのデータチャンクの検査)
- **File** (ファイルの検査)

セッション暗号時において、次の設定によりメッセージデータの検査が無効にされた場合は、“MAC”は付記されません。

- **RequireDataIntegrityChecking** (hcpd.conf)
- **DisableDataIntegrityChecking** (hcp.conf)

Compression は、使用された圧縮モード（圧縮レベル、ヘッダ圧縮およびコンテンツ圧縮の ON/OFF）が記録されます。

Proto は、使用されたトランスポートプロトコルが記録されます。現在は、次の何れかが出力されます。

- **TCP**
- **HpFP**

HpFP が記録される場合は、選択された輻輳制御モードが次の名称で付記されます。

- **F** (Fair モード)
- **S** (Fair Fast モード)
- **A** (Aggressive モード)
- **M** (Modest モード)
- **N** (指定なし)

Local は、トランスポートの通信セッションのローカル側の端点（エンドポイント）の情報（IP アドレス及びポート番号）が記録されます。

Foreign は、トランスポートの通信セッションのリモート側の端点（エンドポイント）の情報（IP アドレス及びポート番号）が記録されます。

PID は、アプリケーションのプロセス ID が記録されます。

--

例：

クライアント

```
Start Date_Time, End Date_Time, Exit, Remote Host, User, Application, Total (bytes), Payload (bytes), Files, Directories, Average Throughput (bps), Encoding, Security, Compression, Proto, Local, Foreign, PID
2018/07/05 18:14:31, 2018/07/05 18:14:31, 0000, 127.0.0.1, user, hcp [1.2], 4904, 108, 9, 2, 4765.582031, UTF8, AES/CBC/PKCS5Padding SHA256 [MAC], None, TCP, 127.0.0.1:35974, 127.0.0.1:874, 21483
2018/07/05 18:14:35, 2018/07/05 18:14:36, 0000, 127.0.0.1, user, hcp [1.2], 4904, 108, 9, 2, 3870.100830, UTF8, AES/CBC/PKCS5Padding SHA256 [MAC File], None, HpFP[N], 0.0.0.0:47342, 127.0.0.1:884, 21501
```

サーバ

```
Start Date_Time, End Date_Time, Exit, Remote Host, User, Application, Total (bytes), Payload (bytes), Files, Directories, Average Throughput (bps), Encoding, Security, Compression, Proto, Local, Foreign, PID
2018/07/05 18:17:46, 2018/07/05 18:17:46, 0000, 127.0.0.1:35978, user [PAM], hcp [1.2], 4904, 108, 9, 2, 2225.312988, UTF8, AES/CBC/PKCS5Padding SHA256 [MAC], None, TCP, 127.0.0.1:874, 127.0.0.1:35978, 21621
2018/07/05 18:17:51, 2018/07/05 18:17:51, 0000, 127.0.0.1:58642, user [PAM], hcp [1.2], 4904, 108, 9, 2, 2656.646729, UTF8, AES/CBC/PKCS5Padding SHA256 [MAC File], None, HpFP [N], 0.0.0.0:884, 127.0.0.1:58642, 21641
```

--

3.2 トランスポート統計

3.2.1 TCP 統計

TCP による通信を実行中にサーバおよびクライアントで通信量やトランスポート統計情報を定期的（約 1 秒毎）に記録します。

| 列名 | 説明 |
|-----------|------------|
| Date_Time | 記録日時 |
| Tx | 送信サイズ(バイト) |

| | |
|---------|-----------------|
| Rx | 受信サイズ(バイト) |
| AcqTx | 送信帯域獲得サイズ(バイト) |
| AcqRx | 受信帯域獲得サイズ(バイト) |
| Cwnd | 輻輳ウィンドウサイズ(バイト) |
| RTT | RTT (マイクロ秒) |
| Retrans | 再送回数 |
| TpTx | 送信スループット(bps) |
| TpRx | 受信スループット(bps) |
| AvgTpTx | 送信平均スループット(bps) |
| AvgTpRx | 受信平均スループット(bps) |

Date_Time は、この記録が作成された日時が記録されます。次の書式で出力されます。

yyyy/mm/dd HH:MM::SS.aaaaaa (マイクロ秒まで記載)

Tx は、この記録が作成されるまでにトランスポート層で送信されたバイト数が記録されます (差分)。

Rx は、この記録が作成されるまでにトランスポート層で受信したバイト数が記録されます (差分)。

AcqTx は、この記録が作成されるまでにトランスポート層で獲得された送信帯域のバイト数が記録されます (差分)。

AcqRx は、この記録が作成されるまでにトランスポート層で獲得された受信帯域のバイト数が記録されます (差分)。

Cwnd は、この記録が作成された時点で TCP の統計情報から取得した輻輳ウィンドウのサイズが記録されます。

RTT は、この記録が作成された時点で TCP の統計情報から取得した RTT が記録されます。

Retrans は、この記録が作成された時点で TCP の統計情報から取得した再送カウントが記録されます。

TpTx は、この記録のトランスポート層の送信スループットが記録されます。

TpRx は、この記録のトランスポート層の受信スループットが記録されます。

AvgTpTx は、この記録が作成された時点でのトランスポート層の平均送信スループットが記録されます。

AvgTpRx は、この記録が作成された時点でのトランスポート層の平均受信スループットが記録されます。

```
--
例：
--
PID 11168, Local 127.0.0.1:58374, Foreign 127.0.0.1:11111
Date_Time, Tx (bytes), Rx (bytes), AcqTx (bytes), AcqRx (bytes), Cwnd (bytes),
RTT (usec), Retrans, TpTx (bps), TpRx (bps), AvgTpTx (bps), AvgTpRx (bps)
2018/05/25 15:05:50.142156, 0, 0, 0, 0, 218450, 17, 0, 0.000000, 0.000000,
0.000000, 0.000000
2018/05/25 15:05:52.868790, 2288, 1760, 2288, 1764, 654830, 21, 0, 6712.9
42871, 5163.802246, 6712.854004, 5163.731934
2018/05/25 15:05:53.321617, 4436, 2920, 4436, 2920, 654830, 12764, 0, 783
69.882812, 51587.031250, 16918.314453, 11775.388672
--
```

3.2.2 HpFP 統計

HpFP による通信を実行中にサーバおよびクライアントで通信量やトランスポート統計情報を定期的（約 1 秒毎）に記録します。

| 列名 | 説明 |
|-------------|-----------------|
| Date_Time | 記録日時 |
| Tx | 送信サイズ(バイト) |
| Rx | 受信サイズ(バイト) |
| AcqTx | 送信帯域獲得サイズ(バイト) |
| AcqRx | 受信帯域獲得サイズ(バイト) |
| Cwnd | 輻輳ウィンドウサイズ(バイト) |
| RTT | RTT (マイクロ秒) |
| Retrans | 再送データサイズ(バイト) |
| Congestion | 輻輳検出カウント |
| MSS | MSS (バイト) |
| MaxTp | 最大スループット(bps) |
| LongCtxWait | コンテキスト長時間待機カウント |

| | |
|---------|-----------------|
| TpTx | 送信スループット(bps) |
| TpRx | 受信スループット(bps) |
| AvgTpTx | 送信平均スループット(bps) |
| AvgTpRx | 受信平均スループット(bps) |

Date_Time は、この記録が作成された日時が記録されます。次の書式で出力されます。

yyyy/mm/dd HH:MM::SS.uuuuuu (マイクロ秒まで記載)

Tx は、この記録が作成されるまでにトランスポート層で送信されたバイト数が記録されます (差分)。

Rx は、この記録が作成されるまでにトランスポート層で受信したバイト数が記録されます (差分)。

AcqTx は、この記録が作成されるまでにトランスポート層で獲得された送信帯域のバイト数が記録されます (差分)。

AcqRx は、この記録が作成されるまでにトランスポート層で獲得された受信帯域のバイト数が記録されます (差分)。

Cwnd は、この記録が作成された時点で HpFP の統計情報から取得した輻輳ウィンドウのサイズが記録されます。

RTT は、この記録が作成された時点で HpFP の統計情報から取得した RTT が記録されます。

Retrans は、この記録が作成された時点で HpFP の統計情報から取得した再送カウントが記録されます。

Congestion は、この記録が作成された時点で HpFP の統計情報から取得した輻輳検出カウント (累積値) が記録されます。

MSS は、この記録が作成された時点で HpFP の統計情報から取得した MSS が記録されます。

MaxTp は、この記録が作成された時点で HpFP の統計情報から取得した最大スループットが記録されます。

LongCtxWait は、この記録が作成された時点で HpFP の統計情報から取得したコンテキスト長時間待機カウントが記録されます。このカウントが計上されている場合は、コンテキストスイッチによる性能への影響が発生している場合があります。

TpTx は、この記録のトランスポート層の送信スループットが記録されます。

TpRx は、この記録のトランスポート層の受信スループットが記録されます。

AvgTpTx は、この記録が作成された時点でのトランスポート層の平均送信スループットが記録されます。

AvgTpRx は、この記録が作成された時点でのトランスポート層の平均受信スループットが記録されます。

--

例：

--

```
PID 2848, Local 127.0.0.1:63304, Foreign 127.0.0.1:884
Date_Time, Tx (bytes), Rx (bytes), TxAcq (bytes), RxAcq (bytes), Cwnd (bytes),
RTT (usec), Retrans (bytes), Congestion, MSS (bytes), MaxTp (bps),
LongCtxWait, TpTx (bps), TpRx (bps), AvgTpTx (bps), AvgTpRx (bps)
2019/10/30 09:34:17.520056, 0, 0, 0, 0, 89560, 382, 0, 0, 1236, 0, 0, 0.0
00000, 0.000000, 0.000000, 0.000000
2019/10/30 09:34:19.915566, 1052, 1124, 1052, 66536, 89560, 382, 0, 0, 12
36, 5536, 0, 3511.808594, 3752.160400, 3511.798340, 3752.149414
2019/10/30 09:34:20.300687, 1708, 620, 1708, 744, 89560, 43, 0, 0, 1236,
85832, 0, 35479.757812, 12879.069336, 7937.694824, 5015.702637
--
```

3.3 システム統計

サーバ(hcpd デーモン)実行中に、通信量およびアクセス状況及び状態を定期的（約 10 秒毎）に記録します。

| 列名 | 説明 |
|-----------|----------------|
| Date_Time | 記録日時 |
| Tx | 送信サイズ(バイト) |
| Rx | 受信サイズ(バイト) |
| AcqTx | 送信帯域獲得サイズ(バイト) |
| AcqRx | 受信帯域獲得サイズ(バイト) |
| Conn | 着信接続数 |
| TCPConn | 着信 TCP 接続数 |
| HpFPConn | 着信 HpFP 接続数 |
| UserConn | 着信ユーザ数 |

| | |
|--------------|--------------|
| WorkConn | 動作中接続数 |
| WorkTCPConn | 動作中 TCP 接続数 |
| WorkHpFPConn | 動作中 HpFP 接続数 |

Date_Time は、この記録が作成された日時が記録されます。次の書式で出力されます。

yyyy/mm/dd HH:MM::SS.aaaaaaa (マイクロ秒まで記載)

Tx は、この記録が作成されるまでにトランスポート層で送信されたバイト数が記録されます (差分)。

Rx は、この記録が作成されるまでにトランスポート層で受信したバイト数が記録されます (差分)。

AcqTx は、この記録が作成されるまでにトランスポート層で獲得された送信帯域のバイト数が記録されます (差分)。

AcqRx は、この記録が作成されるまでにトランスポート層で獲得された受信帯域のバイト数が記録されます (差分)。

Conn は、この記録が作成されるまでに着信した接続数が記録されます (差分)。

TCPConn は、この記録が作成されるまでに着信した TCP 接続数が記録されます (差分)。

HpFPConn は、この記録が作成されるまでに着信した HpFP 接続数が記録されます (差分)。

UserConn は、この記録が作成されるまでに着信したユーザ数が記録されます (差分)。

WorkConn は、この記録が作成される時点で動作中の接続数が記録されます (現在値)。

WorkTCPConn は、この記録が作成される時点で動作中の TCP 接続数が記録されます (現在値)。

WorkHpFPConn は、この記録が作成される時点で動作中の HpFP 接続数が記録されます (現在値)。

--
例 :
--


```

Date_Time, Tx (bytes), Rx (bytes), AcqTx (bytes), AcqRx (bytes), Conn, TC
PConn, HpFPConn, UserConn, WorkConn, WorkTCPConn, WorkHpFPConn
2018/07/05 18:17:40.657653, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
2018/07/05 18:17:50.666458, 4624, 5688, 4624, 5696, 2, 1, 1, 1, 1, 0, 1
2018/07/05 18:17:52.910125, 2624, 4120, 2624, 4124, 0, 0, 0, 1, 0, 0, 0
--

```

4 エラーコードリファレンス

4.1 コマンド終了ステータス

各コマンドの実行時に、0 から 255 の範囲で下表に記載の終了コードを実行結果として返します。

| 終了コード | 名前 | 説明 |
|-------|-------------------------------|--|
| 0 | EXIT_SUCCESS | 成功 |
| 1 | EXIT_FAILURE | 失敗(汎用) |
| 21 | EXIT_NOTICE | 注意付き終了 |
| 40 | EXIT_PENDING | 保留(予約) |
| 50 | EXIT_WARN | 警告(予約) |
| 71 | EXIT_ERROR_CONNECT_FAILED | 接続失敗 |
| 72 | EXIT_ERROR_AUTH_FAILED | 認証失敗 |
| 73 | EXIT_ERROR_CONNECTION_FAILURE | 通信エラー |
| 79 | EXIT_ERROR_NEG_FAILED | ネゴシエーションエラー |
| 81 | EXIT_ERROR_IO_FAILURE | アプリケーション I/O エラー (ディスク、ファイルアクセス等起因) |
| 90 | EXIT_ERROR_APP_FAILURE | アプリケーション処理エラー |
| 91 | EXIT_ERROR_APP_SETUP_FAILED | アプリケーション起動エラー |
| 99 | EXIT_ERROR_APP_ABORTED | アプリケーション中断 |
| 181 | EXIT_ERROR_LOAD_ARG | コマンド引数エラー |
| 182 | EXIT_ERROR_LOAD_CONF | コマンド設定エラー |
| 189 | EXIT_ERROR_CLI_SETUP_FAILED | CLI 起動エラー |

| | | |
|-----|----------------------------------|--------------|
| 191 | EXIT_ERROR_SERVICE_SETUP_FAILED | サービス起動エラー |
| 200 | EXIT_FATAL | 障害(汎用) |
| 201 | EXIT_FATAL_RUNTIME_SETUP_FAILURE | 実行環境起動エラー |
| 202 | EXIT_FATAL_MUTEX_SETUP_FAILURE | 実行プログラム同期エラー |

4.2 設定ファイルエラーコード

設定ファイルに記述エラーを検出した場合、コマンドは終了ステータス (EXIT_ERROR_LOAD_CONF) を実行結果として返します。その際、ユーザインタフェース（シェルの場合は標準エラー出力）には、検出したエラーに関するメッセージと原因の種別を表すエラーコード（下記表に記載）を表示します。エラーコードは、0 から 255 の範囲(8bit)で表されます。

このエラーは、下記の書式で表示されます。

書式 <config_path>: [<error_code>] <message>

config_path は、エラーが発生した設定ファイルのパスを表します。

error_code は、発生したエラーの種別を表すエラーコードがを表します。

message は、発生したエラーメッセージを表します。

--

例:

```
/etc/hcp/hcp.conf: [052] Configuration file parse error in line 3 : PPubk  
eyAuthentication yes
```

--

| エラー コード | 名前 | 説明 |
|------------|--------------------------|--------------|
| 0 | CONF_ERROR_OK | 成功(通常表示されない) |
| 50 | CONF_ERROR_NULL_NAME | 設定項目名の取得に失敗 |
| 51 | CONF_ERROR_EMPTY_NAME | 設定項目名が空文字 |
| 52 | CONF_ERROR_UNKNOWN_NAME | 設定項目名が不明 |
| 53 | CONF_ERROR_INVALID_NAME | 設定項目名が不正 |
| 54 | CONF_ERROR_INVALID_VALUE | 設定項目の値が不正 |

| | | |
|-----|---------------------------------------|--------------------------|
| 60 | CONF_ERROR_QUOTE_NOT_CLOSED | 設定値のクオートが閉じられていない |
| 61 | CONF_ERROR_QUOTE_IN_QUOTE | 設定値のクオート中にクオートが発見された |
| 70 | CONF_ERROR_NO_VALUES | 設定値がなかった |
| 71 | CONF_ERROR_TOO_MANY_VALUES | 設定項目の値の数が多すぎる |
| 79 | CONF_ERROR_OVER_LIMIT_NUM_OF_VALUES | 設定可能な値の数の上限を超えた |
| 80 | CONF_ERROR_UNEXPECTED_NESTED_STRUCT | ネストされた構造化設定項目が検出された |
| 81 | CONF_ERROR_UNEXPECTED_ENDING_STRUCT | 予期しない構造化設定項目の終了が検出された |
| 82 | CONF_ERROR_UNEXPECTED_VALUE_OF_STRUCT | 予期しない構造化設定項目内の設定項目が検出された |
| 84 | CONF_ERROR_ENDING_STRUCT_NOT_FOUND | 構造化設定項目の終了が見つからなかった |
| 85 | CONF_ERROR_OVER_LIMIT_NUM_OF_STRUCTS | 構造化設定項目の設定可能な上限の数を超えた |
| 99 | CONF_ERROR_LINE_LEN_EXCEEDED | 設定行の上限文字数を超えた |
| 200 | CONF_ERROR_CONF_NOT_FOUND | 設定ファイルが見つからなかった |
| 201 | CONF_ERROR_FILE_OPEN_FAILED | 設定ファイルを開くのに失敗した |
| 255 | CONF_ERROR_FAILURE | 失敗 |

4.3 ファイル処理理由コード

`verbose` オプションを指定した場合、ファイル転送などの処理結果が標準出力に下記表記載の理由コード(Reason code)と伴に出力される場合があります。理由コードは、0 から 65536 の範囲(16bit)で 16 進数で表示されます。

| 理由コード(16 進数) | 名称 | 説明 |
|-----------------|-------------------------|-----------------|
| 0000 | NO_ERROR | 成功 |
| 0001 | ALREADY_DONE | 実行済み |
| 2C01 | NOT_MODIFIED | ファイル未更新 |
| 2C02 | NOT_DIFFERENT | ファイル差分なし |
| 2C03 | NOT_PERMITTED | 処理不許可 |
| 2C04 | NOT_CONFIRMED_OVERWRITE | 上書き許可されなかった |
| 2C05 | NOT_CREATED | 作成されなかった |
| 9001 | UNKNOWN_ENCODING | 不明な文字エンコーディング方式 |
| 9002 | UNKNOWN_BCIPH | 不明な暗号方式 |
| 9003 | UNKNOWN_COMPR | 不明な圧縮方式 |
| 9004 | AUTH_REQUIRED | 認証が必要 |
| 9005 | FILE_ENTRY_NOT_FOUND | ファイル情報が見つからなかった |
| 9006 | FILE_ENTRY_ERROR | ファイル情報エラー |
| 9007 | FIND_FILE_ERROR | ファイル探索エラー |
| 9008 | AUTH_FAILED | 認証失敗 |
| 9009 | NEG_FAILED | ネゴシエーション失敗 |
| 900A | KEY_EXHG_REQUIRED | セキュリティ鍵の交換が必要 |
| 900B | APP_NOT_SUPPORTED | アプリ（コマンド）非対応 |
| 9FFB | SESS_IDLE_TIMEOUT | セッションアイドルタイムアウト |
| 9FFC | UNEXP_TERM | 予期しない終了 |
| 9FFD | APP_ABORT | アプリによる中断 |
| 9FFE | SESS_ABORT | セッションによる中断 |
| 9FFF | FILE_REQQ_FULL | ファイルキュー不足 |
| A001 | FILE_NOT_FOUND | ファイルが見つからなかった |

| | | |
|------|----------------------------|--------------------|
| A002 | FILE_ALREADY_EXISTS | ファイルが既に存在する |
| A003 | FILE_ACCESS_DENIED | ファイルアクセス不許可 |
| A004 | DIGEST_ERROR | データ完全性検査エラー |
| A005 | WRITE_ERROR | ファイル書込みエラー |
| A006 | CREATE_DIR_ERROR | ディレクトリ作成失敗 |
| A007 | CREATE_LINK_ERROR | リンク作成失敗 |
| A008 | READ_ERROR | ファイル読込みエラー |
| A009 | IS_DIR | ディレクトリ検出 |
| A00A | IS_NOT_DIR | ディレクトリ非検出 |
| A00B | RENAME_ERROR | ファイル名変更エラー |
| A00C | DELETE_ERROR | ファイル削除エラー |
| A00D | FILE_IDENT | ファイルが同一 |
| A00E | OP_NOT_PERMITTED | 操作不許可 |
| AFFD | MISS_LAST_MODIFIED | ファイル更新確認要求欠損 |
| AFFE | FILE_OUT_OF_DOC_ROOT | ドキュメントルート外アクセス |
| AFFF | FILE_SIZE_OVER_LIMIT | ファイルサイズ上限を超えた |
| B001 | BUSY | サーバビジー |
| B002 | CONNECT_FAILED | 接続失敗 |
| B003 | REFUSED | サーバ拒否 |
| BFFA | INCOMPATIBLE_PROTO_VERSION | プロトコル非互換 |
| BFFB | SETUP_FAILURE | セットアップ失敗 |
| BFFC | NET_UNAVAIL | 通信非有効状態 |
| BFFD | DISK_FULL | ディスク不足 |
| BFFE | SESS_NOT_FOUND | セッションが見つからなかった |
| BFFF | PROTO_ERROR | プロトコルエラー |
| FFFD | GC_ABORT | GC（プロセス停止時など）による中断 |
| FFFE | INTERNAL_ERROR | 内部エラー |

| | | |
|------|---------------|----|
| FFFF | GENERAL_ERROR | 失敗 |
|------|---------------|----|

5 コマンド実行モード

HCP tools では、クライアントのコマンド実行モードを次の二つを提供します。

- 単一起動モード (Single running mode)
- 多重起動モード (Multiple running mode)

多重起動モードは、コマンドの `-multi-run` オプションを指定すると動作します。本オプションを指定しない場合は、単一起動モードとなります。多重起動モードは、作業ディレクトリを複数のユーザで共有する場合は、バックグラウンド実行を行う場合などに利用します。

以下、それぞれのモードでの実行方法、注意事項などを説明します。

5.1 単一起動モード

本モードは、特定の作業ディレクトリで単一のユーザがシーケンシャルにコマンドを実行する様な利用形態で使します。コマンドの終了ステータスは、通常のコマンドと同様の方法で取得します。

```
--
例 :
[user@localhost ~]$ hcp src.txt dest.txt
...
[user@localhost ~]$ echo $?
0
(Windows)
C:\Users\user> hcp src.txt dest.txt
...
C:\Users\user> echo %ERRORLEVEL%
0
--
```

コマンドの実行記録は、`-v` オプションで標準出力に出力しない場合は、カレントディレクトリに次の名前で実行毎に初期化（トランケート）が行われて記録されます。

```
.hcp.out
_hcp.out (Windows)
```

アプリケーションのログは、デフォルトでは標準出力に下記の様に出力されますが、-l オプションでファイルパスを指定すると、そのファイルに出力されます。

```
2019/11/01 13:33:35 00007fae521a3b80:INFO :Set bandwidth outbound to unlimited since the specified value is over 5Gbps (given limit : 10000000000 bps).
2019/11/01 13:33:35 00007fae521a3b80:INFO :Set bandwidth inbound to unlimited since the specified value is over 5Gbps (given limit : 10000000000 bps).
2019/11/01 13:33:38 00007fae4aa4a700:INFO :A transport timeout monitor was setup (sock=5, _recv). Monitoring the timeout ...
2019/11/01 13:33:38 00007fae4aa4a700:INFO :A transport timeout monitor becomes disabled (sock=5, ret=0, errno=0, _recv).
```

統計記録（アプリケーション統計、トランスポート統計）は、保存先を指定しない場合はカレントディレクトリに次の名前で保存されます。これらの記録は、トランケートされず追記されます。

```
.hcp.statistics.application
.hcp.statistics.transport.tcp
.hcp.statistics.transport.hpfp
_hcp.statistics.application (Windows)
_hcp.statistics.transport.tcp (Windows)
_hcp.statistics.transport.hpfp (Windows)
```

統計記録は、-L オプションを使用すると、出力先のディレクトリとファイル名のプレフィックス（接頭辞）を指定して出力できます。

```
--
例 :
[root@localhost ~]# hcp -L /var/tmp/.hcp.statistics2 ...
// 出力先ディレクトリ : /var/tmp
// ファイル名プレフィックス : .hcp.statistics2
--
```

5.2 多重起動モード

本モードは、単一のディレクトリを複数のユーザで共有して作業を行う場合や、コマンドをバックグラウンドで起動して単一のユーザで複数コマンドを同時に実行する場合などに利用します。コマンドの終了ステータスは、単一起動モードと同様の方法で取得するか、結果出力に記録される EXIT パラメータなどを参照します。

```
--
例 :
```

```
[user@localhost ~]$ hcp --multi-run=/var/tmp ...
2019/11/01 14:02:06 00007f35da531b80:INFO :A configuration file (/home/user/.hcp/hcp.conf) was not found. So it was skipped.
2019/11/01 14:02:06 00007f35da531b80:      :Recording on multi-running is configured (ID=hcp.mr.20191101_140206_442.15279, dir=/var/tmp, out=ATR).
2019/11/01 14:02:06 00007f35da531b80:INFO :Set bandwidth outbound to unlimited since the specified value is over 5Gbps (given limit : 10000000000 bps).
2019/11/01 14:02:06 00007f35da531b80:INFO :Set bandwidth inbound to unlimited since the specified value is over 5Gbps (given limit : 10000000000 bps).
Login as:user
Password:
2019/11/01 14:02:09 00007f35d2dd8700:INFO :A transport timeout monitor was setup (sock=5, _recv). Monitoring the timeout ...
2019/11/01 14:02:09 00007f35d2dd8700:INFO :A transport timeout monitor becomes disabled (sock=5, ret=0, errno=0, _recv).

[user@localhost ~]$ hcp -l hcp.log --multi-run=/var/tmp ...
2019/11/01 14:04:23 00007fb04eac3b80:INFO :A configuration file (/home/user/.hcp/hcp.conf) was not found. So it was skipped.
2019/11/01 14:04:23 00007fb04eac3b80:      :Logging on multi-running is configured (ID=hcp.mr.20191101_140423_458.15318, dir=/var/tmp, path=/var/tmp/hcp.mr.20191101_140423_458.15318.log).
Login as:user
Password:
[user@localhost ~]$ less /var/tmp/hcp.mr.20191101_140423_458.15318.log
2019/11/01 14:04:23 00007fb04eac3b80:      :Recording on multi-running is configured (ID=hcp.mr.20191101_140423_458.15318, dir=/var/tmp, out=ATR).
2019/11/01 14:04:23 00007fb04eac3b80:INFO :Set bandwidth outbound to unlimited since the specified value is over 5Gbps (given limit : 10000000000 bps).
2019/11/01 14:04:23 00007fb04eac3b80:INFO :Set bandwidth inbound to unlimited since the specified value is over 5Gbps (given limit : 10000000000 bps).
2019/11/01 14:04:26 00007fb04736a700:INFO :A transport timeout monitor was setup (sock=6, _recv). Monitoring the timeout ...

[user@localhost ~]$ cat /var/tmp/hcp.mr.20191101_140423_458.15318.out | grep -e EXIT
EXIT 0 REASON 0000
--
```

本モードでは、コマンド実行開始時に—multi-run オプションに指定されたディレクトリに次の様なロックファイルを作成します。

```
hcp.mr.<YYYYMMDD_hhmmss_msec>.<pid>.lock
```


コマンド毎にこのロックファイルのプレフィックス (“`.lock`”より前の文字列) に文字列を付加したファイル名で次の記録を保存します。

- アプリケーション統計
- トランスポート統計
- 結果出力
- アプリケーションログ

また、本モードでは、次のエラーステータスで終了する場合は、これらの記録は提供されません。単一起動モードと同様にコマンドラインから終了ステータスの確認を行います。

- `EXIT_ERROR_LOAD_ARG` (181)
- `EXIT_ERROR_LOAD_CONF` (182)
- `EXIT_ERROR_CLI_SETUP_FAILED` (189)

バックグラウンドモードを使用する場合は、上記ケースで停止した場合終了ステータス等の確認が困難となりますので、事前に設定構成を担保するなどご注意ください。

コマンドの実行記録は、`—multi-run` オプションに従ってファイルに出力される場合は、次のファイル名で記録されます。

`hcp.mr.<YYYYMMDD_hhmmss_msec>.<pid>.out`

コマンドの`-v` オプションが適用される場合は、ファイルに保存されず標準出力に出力されます。

アプリケーションのログは、`—multi-run` オプションに従ってファイルに出力される場合は、次のファイル名で記録されます。

`hcp.mr.<YYYYMMDD_hhmmss_msec>.<pid>.log`

統計記録（アプリケーション統計、トランスポート統計）は、`—multi-run` オプションに従ってファイルに出力される場合は、次のファイル名で記録されます。

`hcp.mr.<YYYYMMDD_hhmmss_msec>.<pid>.statistics.application`
`hcp.mr.<YYYYMMDD_hhmmss_msec>.<pid>.statistics.transport.tcp`
`hcp.mr.<YYYYMMDD_hhmmss_msec>.<pid>.statistics.transport.hfpf`

6 HCP tools 制約事項一覧

| 名称 | 制約 |
|----|----|
|----|----|

| | |
|-----------------|----------------------|
| 設定項目の値の数 | 8 個まで |
| 設定項目の行文字数 | 256 |
| 設定項目のエラーメッセージ長 | 1024 |
| DocPoint 定義数 | 10 個まで |
| users の行文字数 | 256 |
| passwd の行文字数 | 256 |
| ファイルサイズ | 8EiB 符号付き倍長整数の最大値 |
| ファイルパス長 | 2048 文字 |
| ファイルパスに使用できない文字 | 非印字文字、OS 非対応文字、改行、など |
| ファイルキャッシュエントリ数 | 100000 個 |
| 処理待機中ファイルの結果出力数 | 1000 件まで |

※ファイルパス長は、プラットフォームのファイルシステムごとに実際のパス長の制限が異なります。Windows では、概ねドライブレターを含め 260 文字まで使用できます。

7 改訂履歴

| 編集日 | 変更点 |
|------------|-----------------------------|
| 2020/04/23 | MAC 無効化説明追加 |
| 2020/02/10 | FileOperationLog 説明追加 |
| 2019/12/16 | ホームディレクトリ検査設定項目説明追加 |
| 2019/12/11 | ログローテーション設定項目説明追加、ファイルパス長注記 |
| 2019/11/12 | 誤記訂正、秘密鍵探索規則注記 |
| 2019/11/05 | 項番出力訂正 |
| 2019/11/01 | 誤記訂正、記載改訂、設定項目説明追加 |
| 2019/06/07 | コマンド記述追記・編集、理由コード追記 |
| 2019/04/26 | users サンプル記述訂正 |
| 2019/04/25 | オプション説明追加、記載改訂 |
| 2019/02/01 | オプション説明追加 |
| 2019/01/30 | 誤記訂正、Include 設定項目説明追加 |
| 2019/01/29 | 誤記訂正 |

HCP tools コマンドリファレンス

| | |
|------------|--------------------|
| 2019/01/21 | サーバ鍵ファイルパス仕様訂正 |
| 2019/01/20 | 追加コマンド、Win 版対応記述追加 |
| 2018/07/30 | アプリ統計記述編集及び出力例改訂 |
| 2018/07/30 | 終了ステータス及び理由コード改訂 |