Capability Check Tool for Networked File Function Manual

- Proper nouns, including product names, are the trademarks or registered trademarks of their respective manufacturers.

Attention
▪ Unauthorized reproduction of this manual, in full or in part, is prohibited.
▪ The contents of this manual and this product are subject to change without
  notice.

Preface

This manual describes how to check your environment for the capabilities to use the networked file function provided by HULFT products.

You can check the readiness for this function by executing a command, which varies according to the environment where HULFT products are to be installed.

Notations in this manual

・In this manual, the following products are collectively referred to as the "HULFT Products".

- HULFT7 for Windows-EX
- HULFT7 for Linux-EX
- HULFT7 for zLinux-EX
- HULFT7 for UNIX-EX
- HULFT8 for Windows
- HULFT8 for Linux
- HULFT8 for zLinux
- HULFT8 for UNIX
- HULFT-DataMagic for Windows Ver.2
- HULFT-DataMagic for UNIX Ver.2
- HULFT-DataMagic for Linux Ver.2
- HULFT-DataMagic for zLinux Ver.2
- DataMagic Server for Windows Ver.3
- DataMagic Server for UNIX Ver.3
- DataMagic Server for Linux Ver.3
- DataMagic Server for zLinux Ver.3
- DataMagic Desktop Ver.3

・Applications other than HULFT Products are referred to as "external applications".

・The functionality and operations are the same for the UNIX version and the Linux version. If you are using the Linux version, read "UNIX" as "Linux" when reviewing the information in this manual.

Meaning of the version, level, and revision numbers

In this manual, the product version information is referred in the following format:

Examples: Ver.1.0.0
          A B C

A: Version

B: Level

C: Revision

An increase in the A number indicates a version upgrade.

An increase in the B number indicates a level upgrade.

An increase in the C number indicates a revision upgrade.

Notations used in the description of commands

[ ]	Square brackets. Items enclosed by square brackets are optional.
{ }	Curly brackets. One of the items or sets of items enclosed in curly brackets must be selected.
......	Repetition mark. The repetition mark is used to indicate that the item before it must be repeatedly entered. This mark sometimes comes after a single word or a group of items enclosed by square or curly brackets. Because items enclosed by square or curly brackets are regarded as a single unit, they must be repeated as a unit.
|	Vertical line. This line is used to separate selections.
Italic text	Indicates a variable value (a value that varies according to the items to be processed or the situation). Example: *yyyymmdd*

The comma (,), equals sign (=), and other symbols must be entered as indicated.

Set values for commands

Uppercase letter	Indicates that uppercase letters (A-Z) can be set.
Lowercase letter	Indicates that lowercase letters (a-z) can be set.
Alphabet character	Indicates that uppercase letters (A-Z) and lowercase letters (a-z) can be set.
Alphanumeric character	Indicates that uppercase letters (A-Z), lowercase letters (a-z), and numbers (0-9) can be set.

Table of Contents

1.　Before You Start Checking

1.1　Requirements for using the networked file function

HULFT Products support for networked file. This function enables processing for files that are stored on a file server on the network, as well as for local files.

To use this functionality, the following requirements must be met.

In Windows

・The file server supports SMB or CIFS.

・Exclusive control by byte range locking is possible.

・Access permissions to the target file and the path are granted.

・Oplocks (opportunistic locks) are disabled.

> Remarks: You do not need to disable oplocks if a file is to be used as a shared file with an application that uses SMB or CIFS as the file sharing protocol.

In UNIX

・The file server supports NFS version 3 (NFS v3).

・Exclusive control by byte range locking is possible.

・Access permissions to the target file and the path are granted.

・Attribute caching for the target file and path is disabled.

HULFT Products provide a capability check tool for the networked file function. By using this command-based tool in an environment where HULFT Products are to be installed, you can check whether a file server satisfies the above requirements or not.

> Remarks: If you can use another method to check whether a file server satisfies the above requirements, you do not need to use this tool to check the file server.

> Note: Depending on the results of the check performed by using this tool, you might not be able to use the networked file function.
> For details, see *4.6 Support levels*.

1.2　Flow of a capability check

1	**Checking a case involving typical operations and command execution modes**

↓

2	**Executing commands**

↓

3	**Seeing the results of command execution**
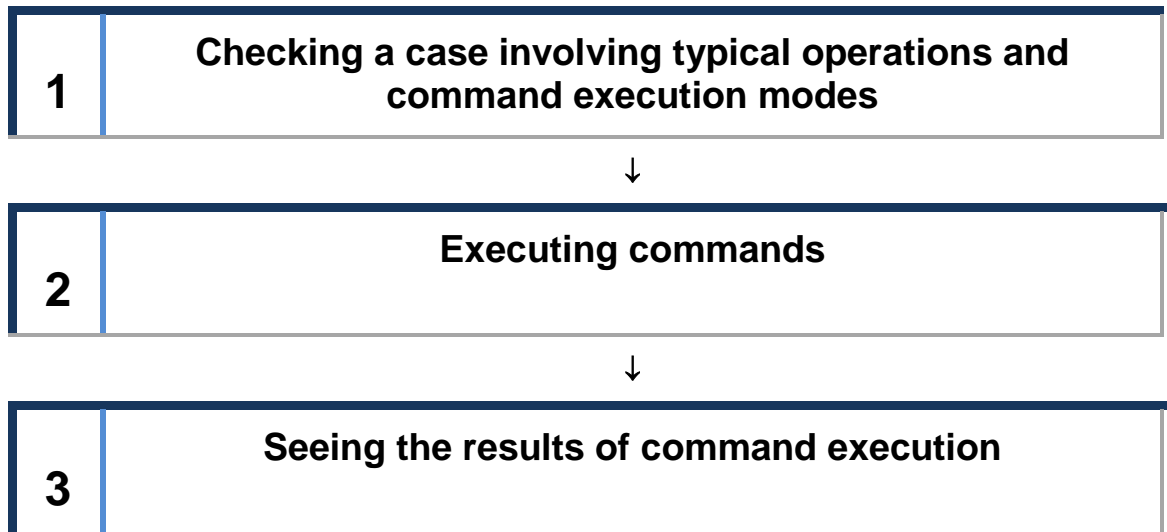
2. Checking a Case Involving Typical Operations and Command Execution Modes

HULFT products provide three command execution modes, one for each of the cases assumed to be typical of the networked file function. Decide which case is most appropriate to the projected system, and then select a suitable command execution mode.

2.1 Assumed operations

(1) Operational case 1: Accessing a networked file from a single host

This operational case assumes that processing for a file on a file server is performed by HULFT products installed on a different host.

The command execution mode in this case is "single environment check mode". In this mode, the capability for the networked file function is checked. When a host executes a command in this mode, the host creates a file in the specified directory. The file is then checked as to whether it can be locked, read, written, and deleted normally.
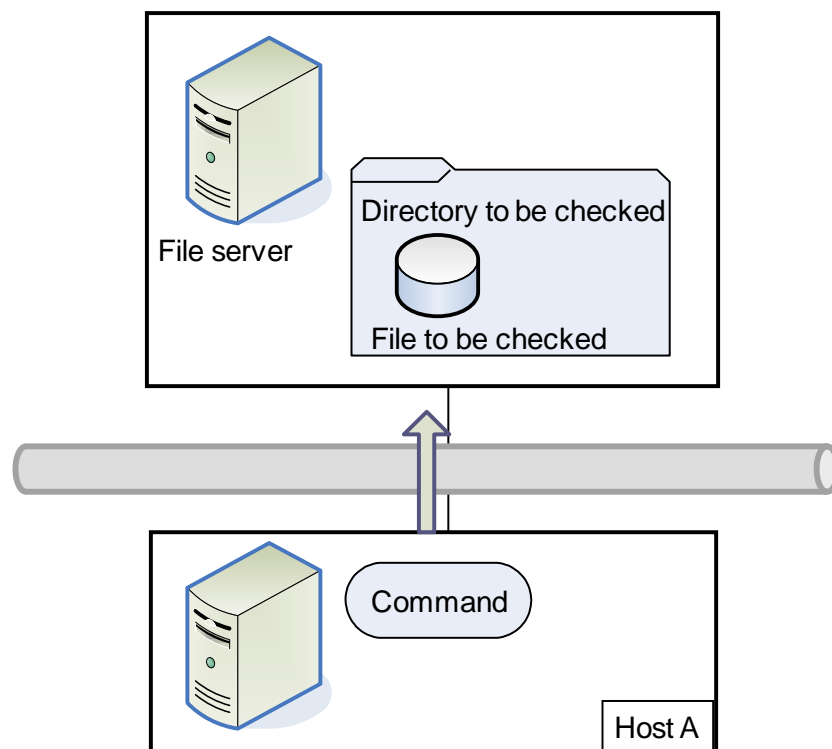
Figure 1. Single environment check mode

(2) Operational case 2: Accessing a networked file from multiple hosts

This operational case assumes that multiple HULFT products installed on different hosts perform processing for a file stored on a file server.

The command execution mode in this case is "mutual environment check mode". When each host executes a command in this mode, the same specified directory is checked as to whether a file can be created, locked, and deleted normally.

File server

Directory to be checked

File to be checked

Command

Command

Host A

Host B
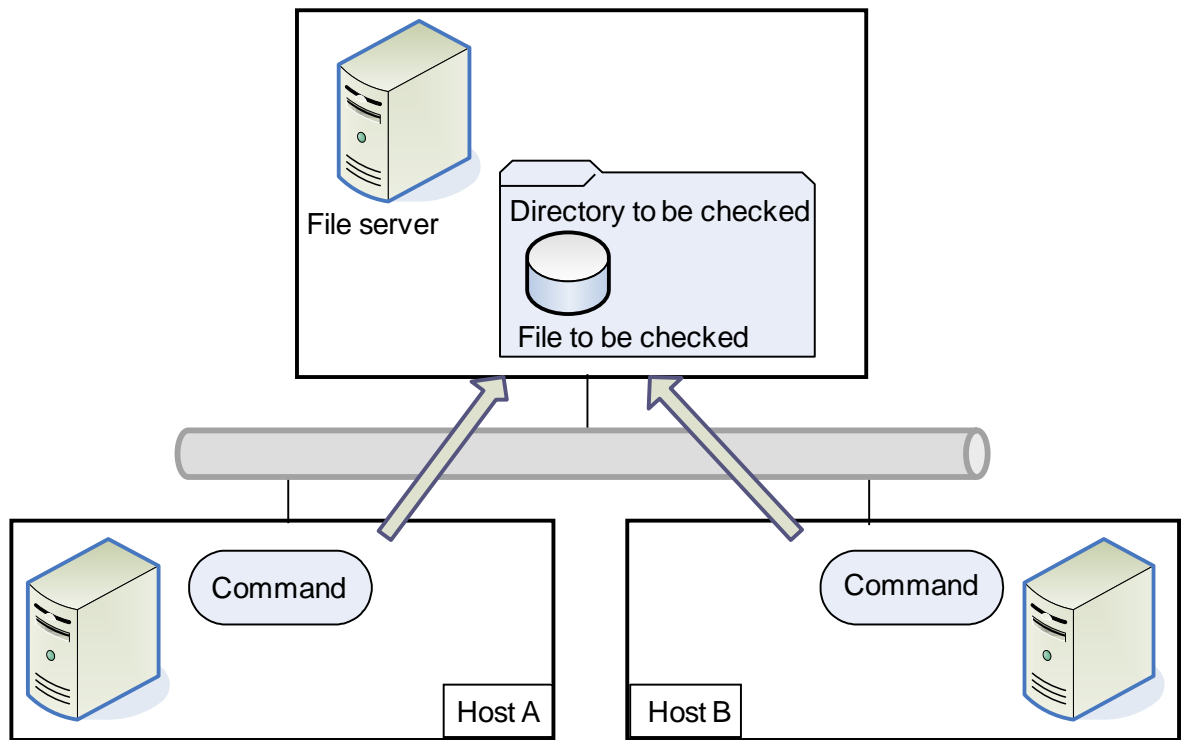
Figure 2. Mutual environment check mode

Note that the processing performed in single environment check mode is also executed in this mutual environment check mode. For details, see *3.1.5 Single environment check processing in mutual environment check mode*.

(3) Operational case 3: Accessing a networked file by linking with an external application

This operational case assumes that HULFT products link with an external application to perform processing for a file stored on a file server.

The command execution mode in this case is "external application linkage check mode". When a command is executed in this mode, the same exclusive control processing as that performed by HULFT products are performed for the specified file.
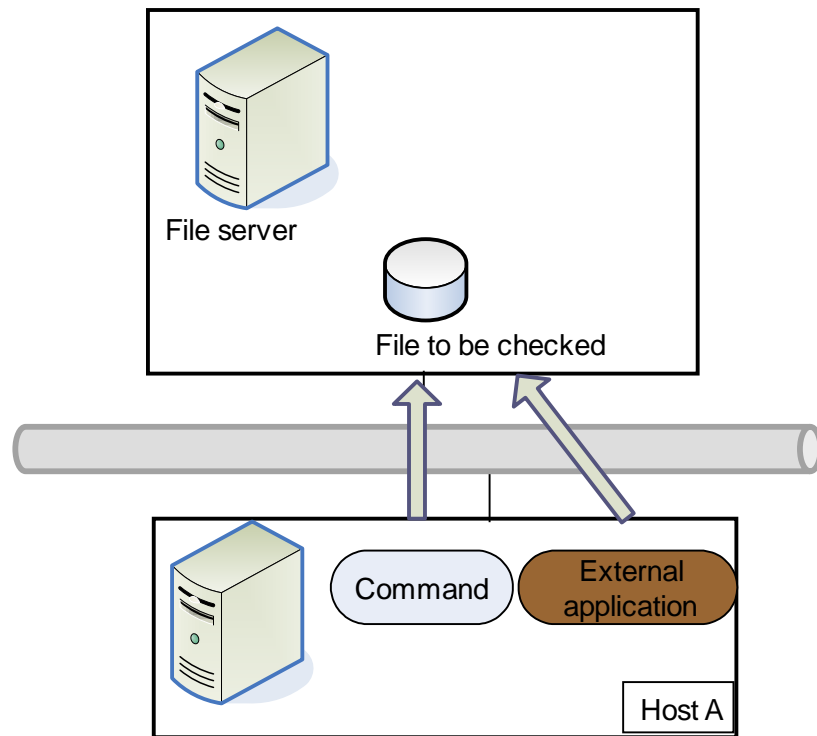
Figure 3. External application linkage check mode

2.2   Flow for determining the command execution mode

```
                        Start

                   Operational          No
                   case 3?
                                              Operational        No
                   Yes                        case 2?

              Execute the                     Yes
              hulntwchk
              command in mutual          Execute the
              environment check          hulntwchk
              mode.                      command in mutual
                                         environment check
                                         mode.

              Successful?    No
                                         Successful?    No
              Yes                                            Execute the
              Execute the                                    hulntwchk
              hulntwchk                  Yes                 command in single
              command in external                           environment check
              application linkage                           mode.
              check mode.

              Successful?    No
                                                        Successful?    No
              Yes
                                                        Yes

              The usage in       The usage in       The usage in
              operational case   operational case   operational case   Not supported.
              3 is supported.    2 is supported.    1 is supported.
```

For details on the syntax and parameters for each command execution mode, see *3.2 Command syntaxes and parameters (Windows)* or *3.3 Command syntaxes and parameters (UNIX)*.

3. Executing Commands

The Capability Check Tool for Networked File Function is a command-based tool. Run the tool by executing the `hulntwchk` command from the command prompt or terminal. Checking starts automatically. The parameters that can be specified for this command differ depending on the OS. First, check the contents of *3.1 Preparations and knowledge required before executing a command*. Then, if you want to execute the `hulntwchk` command on a host on Windows operating system, see *3.2 Command syntaxes and parameters (Windows)*. If you want to execute the `hulntwchk` command on a host on UNIX system, see *3.3 Command syntaxes and parameters (UNIX)*.

3.1 Preparations and knowledge required before executing a command

3.1.1 Setting the LANG environment variable (in UNIX)

To execute the command in UNIX, you must first set the LANG environment variable to prevent the command line from being garbled. Use the following table to configure the settings that correspond to your language and OS.

Table 1. LANG environment variable settings

Language	OS	Character encoding	LANG environment variable setting
Japanese	HP-UX	EUC	ja_JP.eucJP
		SJIS	ja_JP.SJIS
		UTF-8	ja_JP.utf8
	AIX	EUC	Specify either of the following: ・ ja_JP ・ ja_JP.IBM-eucJP
		SJIS	Specify one of the following: ・ Ja_JP ・ Ja_JP.IBM-932 ・ Ja_JP.IBM-943
		UTF-8	Specify either of the following: ・ JA_JP ・ JA_JP.UTF-8

Language	OS	Character encoding	LANG environment variable setting
	Solaris	EUC	Specify either of the following: ・ ja_JP.eucJP ・ ja
		SJIS	ja_JP.PCK
		UTF-8	ja_JP.UTF-8
	Linux	EUC	Specify one of the following: ・ ja_JP.eucjp ・ ja_JP ・ japanese ・ japanese.euc ・ ja_JP.ujis
		SJIS	ja_JP.sjis
		UTF-8	ja_JP.utf8
English	All OSs	--	C

The following are examples of this setting:

Example: Environment settings for Solaris (Japanese version)

When using sh:

LANG=ja

export LANG

When using csh:

setenv LANG ja

Example: Environment settings for Solaris (English version)

When using sh:

LANG=C

export LANG

When using csh:

setenv LANG C

3.1.2 Processing performed by the hulntwchk command

The `hulntwchk` command is used to check whether the following processing can be performed normally for a specified file and path.

・Opening the file

・Reading from and writing to the file

・Applying byte range locking to the file

・Deleting the file

3.1.3 OS-specific items to check

Some items checked by the `hulntwchk` command are specific to either Windows operating system or UNIX system. For details, see the attached *Check List for Networked File Capability*.

3.1.4 Path to Shared Environment by HULFT Products

The Path to Shared Environment by HULFT Products is a path that is set to exclusively control a networked file. This path is set in the System Environment Settings of HULFT products. To exclusively control the same file from multiple HULFT products, set the same path on all of the hosts.

For normal operations, the exclusive control management file is stored in the directory indicated by this path. However, when you execute the `hulntwchk` command, a synchronization file is created in this directory. The synchronization file is used during processing in mutual environment check mode to control the operating status of the commands that were executed from the local host and another host.

3.1.5 Single environment check processing in mutual environment check mode

As a requirement to use the networked file function in mutual environment check mode, check processing in single environment check mode must be successful.

In mutual environment check mode, the same check processing as that performed in single environment check mode is performed first. This enables you to proceed without executing a command in each mode. After this check processing is complete, the check processing specific to mutual environment check mode is performed.

3.1.6 Synchronization processing in mutual environment check mode

In mutual environment check mode, commands that were executed in different environments perform processing for the same file. However, the information on what processing is currently performed by each command and the information on what processing should be performed next

by each command are necessary for both commands to correctly process the file. The `hulntwchk` commands use a method for sequentially synchronizing processing by coordinating the locking of a byte range of the same file.

As shown in the following figure, a synchronization file is created in the directory specified for the Path to Shared Environment by HULFT Products, and the commands synchronize processing by coordinating the locking of a byte range of the same file.
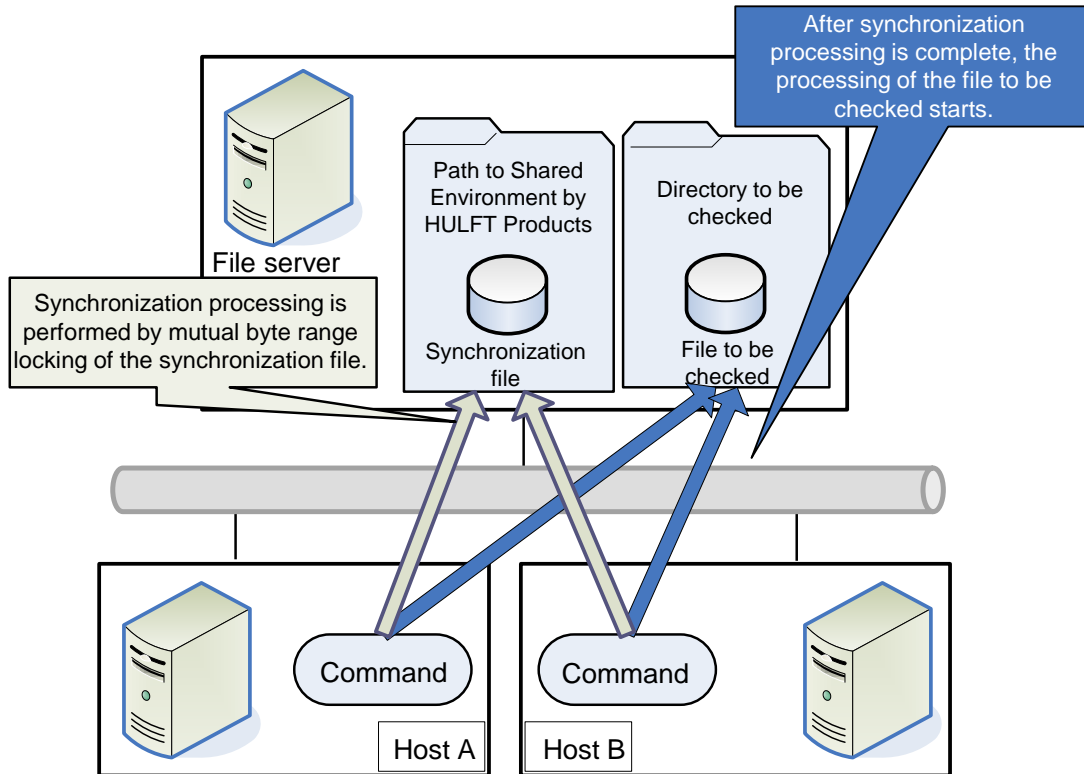
Figure 4. Synchronization processing in mutual environment check mode

Note: To execute the `hulntwchk` command in this mode, specify the paths of the newly created directories for the following parameters:
  `-p` parameter (the path to be checked)
  `-h` parameter (the Path to Shared Environment by HULFT Products)
If you plan to execute multiple `hulntwchk` commands in succession in this mode, before you execute each subsequent `hulntwchk` command, delete the path to be checked and the path that is specified for the Path to Shared Environment by HULFT Products, and then re-create these paths. If you do not delete and re-create these directory paths and you then execute the `hulntwchk` command for the same directory paths as those specified when you previously executed the `hulntwchk` command, an error might occur.

3.1.7 Synchronization processing in external application linkage check mode

In external application linkage check mode, an external application and the `hulntwchk` command mutually perform operations on a file to check the operation of that file. The overview of these operations is as follows:

1. The user executes the `hulntwchk` command in this mode.
   A file to be checked is created. The command is then placed in the processing waiting state.

2. The user checks the contents of the output log, and then processes the file to be checked from an external application with the processing that is indicated in the output log.
   After the file to be checked is processed, an external application synchronization file is created.
   An external application synchronization file named `HULSTOP.stop` is created at the same level as the file being checked.

3. The user deletes the external application synchronization file.
   The `hulntwchk` command detects that the external application synchronization file was deleted, and then performs the following processing:
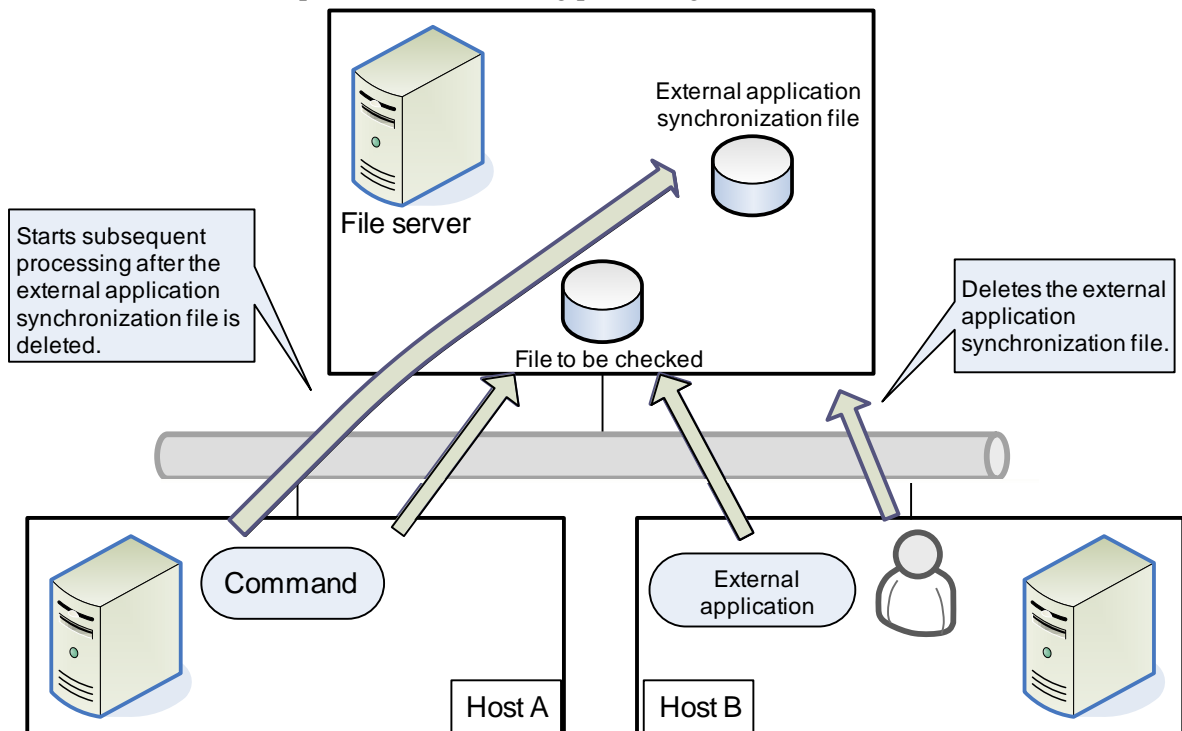
Figure 5. Processing to synchronize with an external application

A log output example in the processing waiting state (in Windows) is as follows:

Test Processing "Open, from an external application, a file created by the command" will now start.

A file to be checked will now be created. (\\hostname\path\a.txt)

Check whether the file to be checked, which was created by the command, can be opened from the external application. (\\hostname\path\a.txt)

After you check the target file, close the file.

An external application synchronization file will now be created. (\\hostname\path\HULSTOP.stop)

A synchronization thread will now be generated.

Processing waits until the external application synchronization file is deleted.

In this example, the user implements a test to check whether the file to be checked (`\\hostname\path\a.txt`) can be opened from an external application.

After the test is complete, the user deletes the external application synchronization file (`\\hostname\path\HULSTOP.stop`). Then, the next test starts.

For details on the processing procedure and details on processing performed by an external application, see the attached *Check List for Networked File Capability*.

Remarks: If you execute the `hulntwchk` command with the `-o logfile` argument specified to output logs to a file, because logs are not output to the standard output, open the specified file and then check the contents of the logs.

3.2   Command syntaxes and parameters (Windows)

The `hulntwchk` command provides three operation modes. This subsection describes the syntax and parameters in HULFT products for Windows for each mode.

3.2.1   Single environment check mode

Syntax

`hulntwchk -m single -p` *targetpath* `-h` *hulsharepath* [`-o` *logfile*] [`-t` *timeout*]

Parameters

`-m single`

> Single environment check mode.

> If you specify this parameter, the command checks whether the networked file function can be used on the target path from an execution environment.

`-p` *targetpath*

> A path to be checked.

> Use up to 200 bytes to specify the path of the directory in which the target file for processing performed by HULFT products are stored.

`-h` *hulsharepath*

> The Path to Shared Environment by HULFT Products.

> In single environment check mode, use up to 200 bytes to specify the path of the location in which HULFT products are to be installed.

`-o` *logfile*

> A log file name.

> Use up to 200 bytes to specify the path of the file to which log information, such as information on checked items and error messages, is output. Specify a local host file name as the log file name.

> If you omit this parameter, logs are output to the standard output.

`-t` *timeout*

> A timeout time (in seconds).

Specify a synchronization period that is taken until processing results are received when the command is in the processing waiting state. The specifiable range is from 10 to 600.

If you omit this parameter, the value is set to 30 seconds.

3.2.2   Mutual environment check mode

Syntax

```
hulntwchk -m double -p targetpath -h hulsharepath [-o logfile] [-t timeout]
```

Parameters

```
-m double
```

Mutual environment check mode.

If you specify this parameter, the command checks whether a command executed in one environment and a command executed in another environment can mutually perform operations on a networked file.

-p *targetpath*

A path to be checked.

Use up to 200 bytes to specify the path of the directory in which the target file of HULFT products processing are stored.

-h *hulsharepath*

The Path to Shared Environment by HULFT Products.

In mutual environment check mode, use up to 200 bytes to specify a network path that can be accessed from both environments.

-o *logfile*

A log file name.

Use up to 200 bytes to specify the path of the file to which log information, such as information on checked items and error messages, is output. Specify a local host file name as the log file name.

If you omit this parameter, logs are output to the standard output.

-t *timeout*

A timeout time (in seconds).

Specify a synchronization period that is taken until processing results are received when the command is in the processing waiting state. The specifiable range is from 10 to 600.

If you omit this parameter, the value is set to 30 seconds.

3.2.3   External application linkage check mode

Syntax

```
hulntwchk -m userapl -f targetfile [-o logfile] [-t timeout]
```

Parameters

`-m userapl`

External application linkage check mode.

If you specify this parameter, the command checks whether HULFT products and an external application can mutually perform operations on a networked file.

`-f` *targetfile*

The name of a file to be checked.

Use up to 200 bytes to specify a file to be mutually used by HULFT products and an external application.

`-o` *logfile*

A log file name.

Use up to 200 bytes to specify the path of the file to which log information, such as information on checked items and error messages, is output. Specify a local host file name as the log file name.

If you omit this parameter, logs are output to the standard output.

`-t` *timeout*

A timeout time (in seconds).

Specify a synchronization period that is taken until processing results are received when the command is in the processing waiting state. The specifiable range is from 10 to 600.

If you omit this parameter, the value is set to 30 seconds.

3.3 Command syntaxes and parameters (UNIX)

The `hulntwchk` command provides three operation modes. This subsection describes the syntax and parameters in UNIX for each mode.

3.3.1 Single environment check mode

Syntax

`hulntwchk -m single -p` *targetpath* `-h` *hulsharepath* [`-O` *OWNER* [`-G` *GROUP*]] [`-o` *logfile*] [`-t` *timeout*]

Parameters

`-m single`

> Single environment check mode.

> If you specify this parameter, the command checks whether the networked file function can be used on the target path from an execution environment.

`-p` *targetpath*

> A path to be checked.

> Use up to 200 bytes to specify the path of the directory in which the target file for processing performed by HULFT products are stored.

`-h` *hulsharepath*

> The Path to Shared Environment by HULFT Products.

> In single environment check mode, use up to 200 bytes to specify the path of the location in which HULFT products are to be installed.

`-O` *OWNER*

> A file owner.

> Use up to 14 bytes to specify the owner of the file to be created in the location indicated by the path to be checked.

> If you omit this parameter, the user executing the command is set.

`-G` *GROUP*

> A file owner group.

> Use up to 14 bytes to specify the owner group of the file to be created in the location indicated by the path to be checked.

You cannot specify only -G *GROUP*. If you specify -G *GROUP*, you must also specify -O *OWNER*.

If you do not specify -G *GROUP*, the group to which the owner specified for -O *OWNER* belongs is set for -G *GROUP*. If you specify neither -O *OWNER* nor -G *GROUP*, the group to which the user executing the command belongs is set for -G *GROUP*.

-o *logfile*

A log file name.

Use up to 200 bytes to specify the path of the file to which log information, such as information on checked items and error messages, is output. Specify a local host file name as the log file name.

If you omit this parameter, logs are output to the standard output.

-t *timeout*

A timeout time (in seconds).

Specify a synchronization period that is taken until processing results are received when the command is in the processing waiting state. The specifiable range is from 10 to 600.

If you omit this parameter, the value is set to 30 seconds.

3.3.2 Mutual environment check mode

Syntax

```
hulntwchk -m double -p targetpath -h hulsharepath [-O OWNER [-G GROUP]]
[-o logfile] [-t timeout]
```

Parameters

-m double

Mutual environment check mode.

If you specify this parameter, the command checks whether a command executed in one environment and a command executed in another environment can mutually perform operations on a networked file.

-p *targetpath*

A path to be checked.

Use up to 200 bytes to specify the path of the directory in which the target file for processing performed by HULFT products are stored.

`-h` *hulsharepath*

The Path to Shared Environment by HULFT Products.

In mutual environment check mode, use up to 200 bytes to specify a network path that can be accessed from both environments.

`-O` *OWNER*

A file owner.

Use up to 14 bytes to specify the owner of the file to be created in the location indicated by the path to be checked.

If you omit this parameter, the user executing the command is set.

`-G` *GROUP*

A file owner group.

Use up to 14 bytes to specify the owner group of the file to be created in the location indicated by the path to be checked.

You cannot specify only `-G` *GROUP*. If you specify `-G` *GROUP*, you must also specify `-O` *OWNER*.

If you do not specify `-G` *GROUP*, the group to which the owner specified for `-O` *OWNER* belongs is set for `-G` *GROUP*. If you specify neither `-O` *OWNER* nor `-G` *GROUP*, the group to which the user executing the command belongs is set for `-G` *GROUP*.

`-o` *logfile*

A log file name.

Use up to 200 bytes to specify the path of the file to which log information, such as information on checked items and error messages, is output. Specify a local host file name as the log file name.

If you omit this parameter, logs are output to the standard output.

`-t` *timeout*

A timeout time (in seconds).

Specify a synchronization period that is taken until processing results are received when the command is in the processing waiting state. The specifiable range is from 10 to 600.

If you omit this parameter, the value is set to 30 seconds.

3.3.3   External application linkage check mode

Syntax

```
hulntwchk -m userapl -f targetfile [-o logfile] [-t timeout]
```

Parameters

`-m userapl`

External application linkage check mode.

If you specify this parameter, the command checks whether HULFT products and an external application can mutually perform operations on a networked file.

`-f` *targetfile*

The name of a file to be checked.

Use up to 200 bytes to specify a file to be mutually used by HULFT products and an external application.

`-o` *logfile*

A log file name.

Use up to 200 bytes to specify the path of the file to which log information, such as information on checked items and error messages, is output. Specify a local host file name as the log file name.

If you omit this parameter, logs are output to the standard output.

`-t` *timeout*

A timeout time (in seconds).

Specify a synchronization period that is taken until processing results are received when the command is in the processing waiting state. The specifiable range is from 10 to 600.

If you omit this parameter, the value is set to 30 seconds.

4. Analyzing the Execution Results

If the check processing performed by executing the `hulntwchk` command ends successfully, the command has verified that the checked environment supports the networked file function. If the check ends unsuccessfully, the checked environment does not support the networked file function. In this case, you need to take appropriate actions.

This section describes how to verify the execution results of the `hulntwchk` command.

4.1 Flow of results analysis

The following figure shows the verification flow:

```
┌─────────────────┐                      ┌─────┐
│  Verification   │                      │  1  │
│     starts.     │                      └──┬──┘
└────────┬────────┘                         │
         │                                  ▼
         ▼                      ┌────────────────────────────┐
┌─────────────────┐            │ See 4.4.1 Error symptoms and│
│  Execute the    │            │ solutions based on the      │
│  command.       │            │ available information.      │
└────────┬────────┘            └──────────────┬─────────────┘
         │                                     │
         ▼                                     ▼
    ◇ Is the return ◇──── Yes          ◇ Can you find the item ◇──── No
    ◇ value "0"?    ◇                  ◇ that corresponds to the ◇
         │                             ◇ symptom?              ◇
         No                                   │
         │                                   Yes
         ▼                                    ▼
┌─────────────────────┐            ◇ Can you implement the ◇──── No
│ Check the logs to   │            ◇ solution?             ◇
│ find the lines on   │                      │
│ which NG was output │                     Yes
│ in the check process│                      │
│ end message.        │                      ▼
└──────────┬──────────┘         ┌──────────────────┐  ┌──────────────────────┐
           │                    │ Implement the    │  │ Ask the server       │
           ▼                    │ solution for the │  │ administrator to     │
┌─────────────────────┐         │ symptom.         │  │ implement the        │
│ In the messages on  │         └────────┬─────────┘  │ solution.            │
│ which NG was output,│                  │            └──────────┬───────────┘
│ find the operation  │                  │                       │
│ identifier, check-  │                  ▼
│ target identifier,  │            ◇ Is the problem solved? ◇──── Yes
│ processing type     │                  │
│ identifier, and     │                  No
│ processing          │                  │
│ identifier.         │                  ▼
└──────────┬──────────┘         ┌─────────────────────┐
           │                    │ Collect information │
           ▼                    │ such as the error   │
┌─────────────────────┐         │ message, and the    │
│ See the list of     │         │ used function and   │
│ check items, and    │         │ arguments.          │
│ find the check item │         └──────────┬──────────┘
│ and processing      │                    │
│ number that         │                    ▼
│ correspond to each  │         ┌─────────────────────┐
│ identifier.         │         │ Contact the support │
└──────────┬──────────┘         │ center or vendor for│
           │                    │ the file server.    │
           ▼                    └──────────┬──────────┘
┌─────────────────────┐                    │
│ See the list of used│                    ▼
│ functions, and find │            ◇ Is the problem solved? ◇──── Yes
│ the used function   │                    │
│ and arguments based │                    No
│ on the processing   │                    │
│ number.             │                    ▼
└──────────┬──────────┘         ┌──────────────────┐
           │                    │  The mode being  │
           ▼                    │  verified is not │
┌─────────────────────┐         │  supported.      │
│ From the messages   │         └──────────────────┘
│ between the check   │
│ process start       │         ┌──────────────────┐
│ message and the     │         │  Re-execute the  │
│ check process end   │         │  command.        │
│ message, identify   │         └──────────────────┘
│ the messages that   │
│ start with an       │
│ exclamation mark(!).│
└──────────┬──────────┘
           │
           ▼
      ┌─────┐        ┌──────────────────┐
      │  1  │        │  The mode being  │
      └─────┘        │  verified is     │
                     │  supported.      │
                     └──────────────────┘
```

4.2 Checking the return value of the hulntwchk command

If all check processing is successful, processing ends by returning a value of 0. In this case, the verified environment supports the networked file function.

If at least one error occurs during check processing, processing ends by returning a value other than 0. In this case, the next step is to check the contents of the output log.

4.3 Checking the contents of the output log

If an error occurs during check processing, check the contents of the output log. The log is output to the file specified in the relevant parameter or to the standard output.

4.3.1 Type of log output messages

The following four types of messages are output to logs.

Table 2. List of message types

Message type	Description
Check process start message	This message indicates that check processing starts. The identifier type and identifier of the applicable item are output.
Check process end message	This message indicates that check processing ended. If check processing is successful, [OK] is output at the end of the line. If an error occurred, [NG] is output.
Error message	This message contains the description of an error that occurred during check processing. Depending on the error, an error code that is defined by the OS is output at the end of an error message.
Operation status message	This message contains information about the check processing that is currently being executed. The message is output for each operation.

(1) The check process start message and the check process end message

In the check process start message and the check process end message, information indicating the start and end of check processing is output for each identifier listed in the attached *Check List for Networked File Capability*.

Identifiers are used to uniquely classify the check items and processing to which the `hulntwchk` command processing applies. Identifiers can be one of the four types described below.

Table 3. List of identifiers

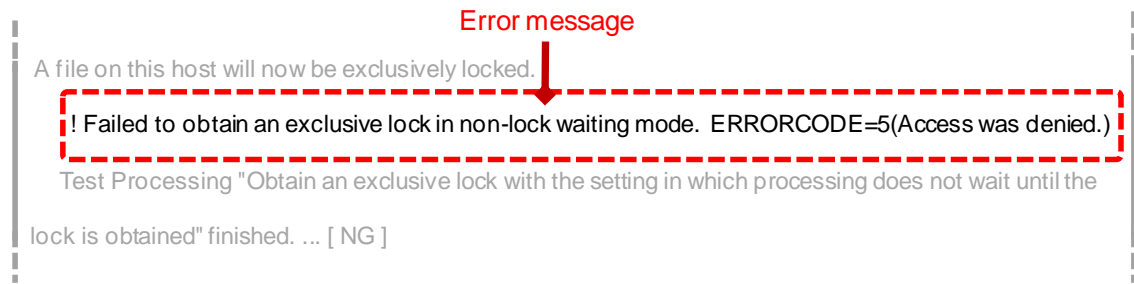| Identifier type	Description
Operation identifier	Environment check mode in which check processing is to be performed
Check-target identifier	A path or file to be checked
Processing type identifier	Processing classification of the executed check processing
Processing identifier	Description of the executed check processing

[OK] or [NG] is output at the end of a check process end message. This indicates whether the check item was successful or failed. Find the messages for which [NG] is output and then check their identifiers to determine the processing in which a failure occurred.
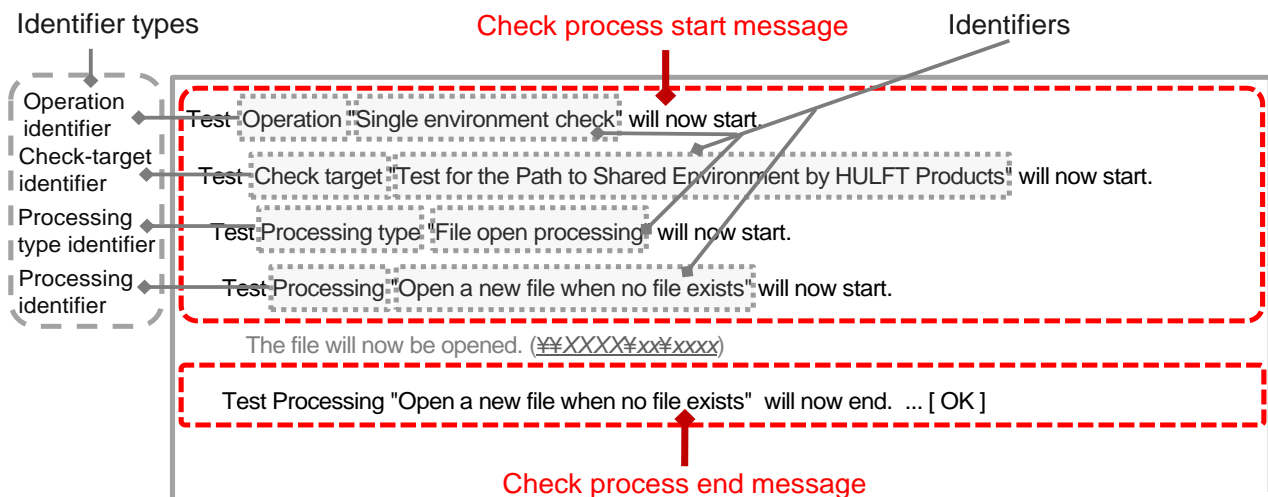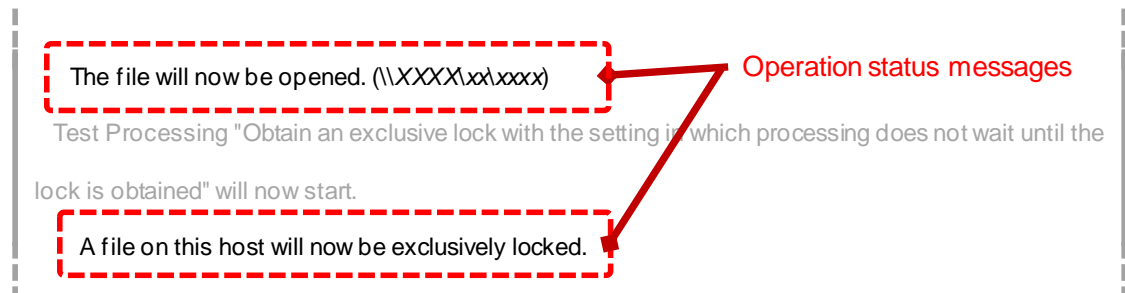

(2) Error message

If an error occurred, the description of the error is output as an error message. An exclamation mark (!) is added at the beginning of this message.

Depending on the error type, an error code that is defined by the OS is output in the format of ERRORCODE=*error-code*（*message-that-corresponds-to-the-error-code*）at the end of an error message.

Error message

A file on this host will now be exclusively locked.

! Failed to obtain an exclusive lock in non-lock waiting mode. ERRORCODE=5(Access was denied.)

Test Processing "Obtain an exclusive lock with the setting in which processing does not wait until the lock is obtained" finished. ... [NG]

(3)   Operation status message

The check processing that is being performed by the `hulntwchk` command is output.

The file will now be opened. (*XXXX**xx**xxxx*)

Operation status messages

Test Processing "Obtain an exclusive lock with the setting in which processing does not wait until the lock is obtained" will now start.

A file on this host will now be exclusively locked.

4.3.2   Example of checking logs

This subsection describes how to check logs assuming output in Windows.

This example assumes the following output log:

Test Operation "Single environment check" will now start.

 Test Check target "Test for the Path to Shared Environment by HULFT Products" will now start.

  Test Processing type "File open processing" will now start.

   Test Processing "Open a new file when no file exists" will now start.

    The file will now be opened. (*XXXX\xx\xxxx*)

   Test Processing "Open a new file when no file exists" finished. ... [OK]

    The file was deleted. (*XXXX\xx\xxxx*)

  Test Processing type "File open processing" finished. ... [OK]

  Test Processing type "Byte range locking" will now start.

    The file will now be opened. (*XXXX\xx\xxxx*)

   Test Processing "Obtain an exclusive lock with the setting in which processing does not wait until the lock is obtained" will now start.

     A file on this host will now be exclusively locked.

     ! Failed to obtain an exclusive lock in non-lock waiting mode. ERRORCODE=5(Access was denied.)

   Test Processing "Obtain an exclusive lock with the setting in which processing does not wait until the lock is obtained" finished. ... [NG]

   Test Processing "Obtain a shared lock with the setting in which processing waits until the lock is obtained" will now start.

     A file on this host will now be shared locked.

   Test Processing "Obtain a shared lock with the setting in which processing waits until the lock is obtained" finished. ... [OK]

(omitted)

  Test Processing type "Byte range locking" finished. ... [NG]

 Test Check target "Test for the Path to Shared Environment by HULFT Products" finished. ... [NG]

 Test Check target "Test for the path to be checked" will now start.

(omitted)

 Test Check target "Test for the path to be checked" finished. ... [OK]

Test Operation "Single environment check" finished. ... [NG]

Figure 6. A log output example (when byte range locking has failed)

(1)   Checking lines on which NG was output

First, find the check process end message lines on which `[NG]` was output.

In this example, `[NG]` was output to, counting from the top, lines 14, 19, 20, and 24.

1	Test Operation "Single environment check" will now start.
2	Test Check target "Test for the Path to Shared Environment by HULFT Products" will now start.
3	Test Processing type "File open processing" will now start.
4	Test Processing "Open a new file when no file exists" will now start.
5	The file will now be opened. (¥¥*XXXX*¥*xx*¥*xxxx*)
6	Test Processing "Open a new file when no file exists" finished. ... [OK]
7	The file was deleted. (¥¥*XXXX*¥*xx*¥*xxxx*)
8	Test Processing type "File open processing" finished. ... [OK]
9	Test Processing type "Byte range locking" will now start.
10	The file will now be opened. (¥¥*XXXX*¥*xx*¥*xxxx*)
11	Test Processing "Obtain an exclusive lock with the setting in which processing does not wait until the lock is obtained" will now start.
12	A file on this host will now be exclusively locked.
13	! Failed to obtain an exclusive lock in non-lock waiting mode. ERRORCODE=5(Access was denied.)
14	Test Processing "Obtain an exclusive lock with the setting in which processing does not wait until the lock is obtained" finished. ... [NG]
15	Test Processing "Obtain a shared lock with the setting in which processing waits until the lock is obtained" will now start.
16	A file on this host will now be shared locked.
17	Test Processing "Obtain a shared lock with the setting in which processing waits until the lock is obtained" finished. ... [OK]
18	(omitted)
19	Test Processing type "Byte range locking" finished. ... [NG]
20	Test Check target "Test for the Path to Shared Environment by HULFT Products" finished. ... [NG]
21	Test Check target "Test for the path to be checked" will now start.
22	(omitted)
23	Test Check target "Test for the path to be checked" finished. ... [OK]
24	Test Operation "Single environment check" finished. ... [NG]

(2) Checking identifiers

Check the corresponding identifiers.

The identifier that corresponds to each line in this example is as follows: the processing identifier of line 14 is "Obtain an exclusive lock with the setting in which processing does not wait until the lock is obtained", the processing type identifier of line 19 is "Byte range locking", the check-target identifier of line 20 is "Test for the Path to Shared Environment by HULFT Products", and the operation identifier of line 24 is "Single environment check".

13	! Failed to obtain an exclusive lock in non-lock waiting mode. ERRORCODE=5(Access was denied.)
14	Test Processing "Obtain an exclusive lock with the setting in which processing does not wait until the lock is obtained" finished. ... [NG]
15	Test Processing "Obtain a shared lock with the setting in which processing waits until the lock is obtained" will now start.
16	A file on this host will now be shared locked.
17	Test Processing "Obtain a shared lock with the setting in which processing waits until the lock is obtained" finished. ... [OK]
18	(omitted)
19	Test Processing type "Byte range locking" finished. ... [NG]
20	Test Check target "Test for the Path to Shared Environment by HULFT Products" finished. ... [NG]
21	Test Check target "Test for the path to be checked" will now start.
22	(omitted)
23	Test Check target "Test for the path to be checked" finished. ... [OK]
24	Test Operation "Single environment check" finished. ... [NG]

(3) Check of the status before check, processing number, and information to be checked

Compare these identifiers with the identifiers listed in *Check List for Networked File Capability*. Then, from the corresponding check items, find the status before check, the processing number, and the information to be checked.

In this example, the status before check is "The file is not locked.", the processing number is "7", and the information to be checked is "A lock can be obtained."

Attached document: List of network file functionality installation check items

1. Check Items for Windows

Table 1. Check item list (Windows)

No.	Identifier				Status before check	Processing number	Information to be checked
	Operation identifier	Check-target identifier	Processing type identifier	Processing identifier			
1	Single environment check	Test for the path of the environment settings shared by HULFT products	File open processing	Open a new file when no file exists	No file exists.	1	A file can be created and opened.
2			Byte range lock processing	Obtain an exclusive lock with the setting in which processing does not wait until obtaining the lock	The file is not locked.	7	A lock can be obtained.
3				Obtain a shared lock with the setting in which processing waits until the lock is obtained	The file is not locked.	8	A lock can be obtained.

(4) Checking a function name and the values set for its arguments

On the list of used functions in the attached *Check List for Networked File Capability*, check the processing and the function that are executed in the corresponding processing number, and the values set for its arguments.

In this example, the function `LockFileEx` is used in the processing "Obtain an exclusive lock with the setting in which processing does not wait until the lock is obtained". The argument of the lock option (dwFlags) is `LOCKFILE_FAIL_IMMEDIATELY | LOCKFILE_EXCLUSIVE_LOCK`.
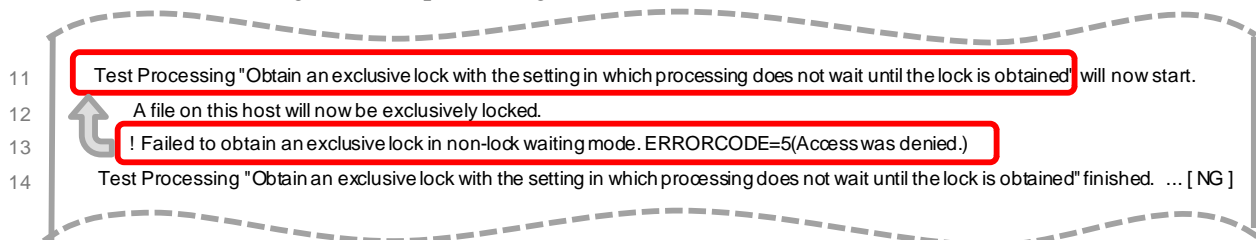
Table 2. List of used functions (Windows)

Processing performed by the command		Processing number	Used function	Argument related to the operation to be performed		
File open processing				Access mode (dwDesiredAccess)	Sharing mode (dwShareMode)	Creation method (dwCreationDisposition)
	Create and open a file if no file exists	1	CreateFile	GENERIC_READ | GENERIC_WRITE	FILE_SHARE_READ | FILE_SHARE_WRITE	CREATE_NEW
	Open a file if a file exists	2		GENERIC_READ | GENERIC_WRITE	FILE_SHARE_READ | FILE_SHARE_WRITE	OPEN_EXISTING
	Open a file with read rights	3		GENERIC_READ	FILE_SHARE_READ | FILE_SHARE_WRITE	OPEN_EXISTING
	Open a file with read and write rights	4		GENERIC_READ | GENERIC_WRITE	FILE_SHARE_READ | FILE_SHARE_WRITE	OPEN_EXISTING
	Open a file with the setting in which the file is not shared	5		GENERIC_READ | GENERIC_WRITE	0	OPEN_EXISTING
	Other file open processing	6		GENERIC_READ | GENERIC_WRITE	FILE_SHARE_READ | FILE_SHARE_WRITE	OPEN_ALWAYS
Byte range lock acquisition processing				Lock option (dwFlags)		
	Obtain an exclusive lock with the setting in which processing does not wait until the lock is obtained	7	LockFileEx	LOCKFILE_FAIL_IMMEDIATELY | LOCKFILE_EXCLUSIVE_LOCK		

(5) Checking error messages

Find the message beginning with an exclamation mark (!) in the log information. Check the lines above this line to check the processing during which this error message was output.

In this example, you can find a line that indicates that the processing "Obtain an exclusive lock with the setting in which processing does not wait until the lock is obtained" failed.

11	Test Processing "Obtain an exclusive lock with the setting in which processing does not wait until the lock is obtained" will now start.
12	A file on this host will now be exclusively locked.
13	! Failed to obtain an exclusive lock in non-lock waiting mode. ERRORCODE=5(Access was denied.)
14	Test Processing "Obtain an exclusive lock with the setting in which processing does not wait until the lock is obtained" finished. ... [NG]

Remarks: Depending on the error, multiple error messages might be output. In this case, resolve the problem by working from the top line to the bottom line.

4.4 Implementing solutions to problems

After you check the contents of the output log, implement solutions to problems.

See *4.4.1 Error symptoms and solutions* and check whether any error symptoms apply.

If you can find any applicable error symptoms, implement the corresponding solution.

> Note: If the file server is managed by another administrator or division and you need to check or change the file server settings, contact the administrator or division.

4.4.1　Error symptoms and solutions

Error symptom	Solution
The following messages are output: ・ A folder is specified for the file to be checked. ・ The file to be checked already exists. ・ The folder specified for the file to be checked is invalid. ・ The specified Path to Shared Environment by HULFT Products is invalid. ・ The specified path to be checked is invalid. ・ The log file could not be opened.	Check whether the path and file that were specified for arguments are valid. Next, check whether the user who executed the `hulntwchk` command has access permissions to the path and the file to be checked. If the user does not have these permissions, change settings so that the user can create, update, and delete files. If you specify a network drive path in Windows, specify the path in UNC format.
An error occurs during file open processing. An error occurs during read and write processing for the file. An error occurs during change processing of the file owner. An error occurs during file deletion processing.	Even if the executing user is Administrator (in Windows) or root (in UNIX), if no access permissions to the path and the file are granted to the executing user on the server, normal processing might not be performed. Check whether the user who executed the `hulntwchk` command has access permissions to the path and the file to be checked. If the user does not have the permissions, change settings so that the user can create, update, and delete files and change the owner.
An error occurs during link creation processing.	Check whether the file system on the file server supports hard links.
An error occurs during byte range locking.	Check whether the user who executed the `hulntwchk` The processing might not end until name resolution processing ends because one of the following problems occurred during networked file access processing: If byte range locking is disabled on the file server, enable byte range locking. If the problem cannot be solved, the target file server might not support byte range locking. See the documentation of the file server, or contact the manufacturer.
Processing times out.	Change the timeout time to a large enough value, and then retry the processing. If the problem cannot be solved, the byte range locking status might not be detected during synchronization processing or byte range locking. See the description of the error symptom "An error occurs during byte range locking" and take the appropriate actions.

Error symptom	Solution
Processing does not end even if the timeout time elapses.	The processing might not end until name resolution processing ends because one of the following problems occurred during networked file access processing: ・The specified host does not exist. ・The communication was disconnected. Execute `ping` for the target host, and then check whether the command ends normally. Execute `ping` on the command prompt or the terminal as follows: >ping *remote-host-name* If the command does not end normally, take the following actions: ・Set the IP address and subnet mask of the local host. ・Set the default gateway. ・Define the remote host name in the hosts file. ・Check whether the remote host name does not include an invalid character, such as a hash mark (`#`). ・Check whether the remote host runs.
Command processing is aborted and cannot end.	If a supported protocol is not used, change settings so that a supported protocol is used.
	A problem might have occurred on the file server.

4.5   Contacting the support center or vendor for the file server

If a problem that is not listed in *4.4.1 Error symptoms and solutions* occurs or if you implemented a solution but the problem was not solved, there might be a problem with the file server.

Based on the following collected information, contact the support center or vendor for the file server:

・Log information collected when an error occurs

・The name of the used function and the values set for its arguments (see the list of used functions in the attached *Check List for Networked File Capability*)

4.6   Support levels

HULFT products provide the support levels described below.

Table 4. Description of support levels

Level	Support level	Supported specifications
3	External application linkage support	Networked files can be used by linking HULFT products and an external application.
2	Mutual environment support	Networked files can be used from multiple HULFT products.
1	Single environment support	Networked files can be used from a single HULFT product.
0	Not supported	No networked files can be used by HULFT products.

A higher support level (that is, a level with a higher number) must meet all of the requirements for the lower support levels. For example, the support level "External application linkage support" is satisfied when execution results in all modes are normal. If the execution result in external application linkage check mode is normal but the execution results in both single environment check mode and mutual environment check mode are abnormal, the "External application linkage support" level is not satisfied.

The following table shows the correspondence between execution results and satisfied support levels.

Table 5. Execution results and satisfied support levels

Execution result			Support level
Single environment check mode	Mutual environment check mode	External application linkage check mode	
Y	Y	Y	External application linkage support
Y	Y	N	Mutual environment support
Y	N	Y	Single environment support
Y	N	N	Single environment support
N	N	Y	Not supported
N	N	N	Not supported

Y: Check processing ends normally.

N: Check processing ends abnormally or is not executed.

From the above results, you can find that the mutual environment support covers the single environment support, and the external application linkage support covers the mutual environment support. The following table shows the other support levels that are covered by each support level.

Table 6. List of other support levels that are covered by each support level

Support level	Covered support level		
	Single environment support	Mutual environment support	External application linkage support
External application linkage support	Y	Y	Y
Mutual environment support	Y	Y	N
Single environment support	Y	N	N

Y: Covered

N: Not covered

5. Notes

5.1 If the same file is checked from three or more hosts

If you want to check the same file from three or more hosts, execute check processing by using the `hulntwchk` command by rotating through the different available combinations of host pairs.

For example, to check the same file from three hosts, you would need to execute the `hulntwchk` command three times, each using a different pair of hosts. (The paired hosts share the same arrow color in the following figure.)
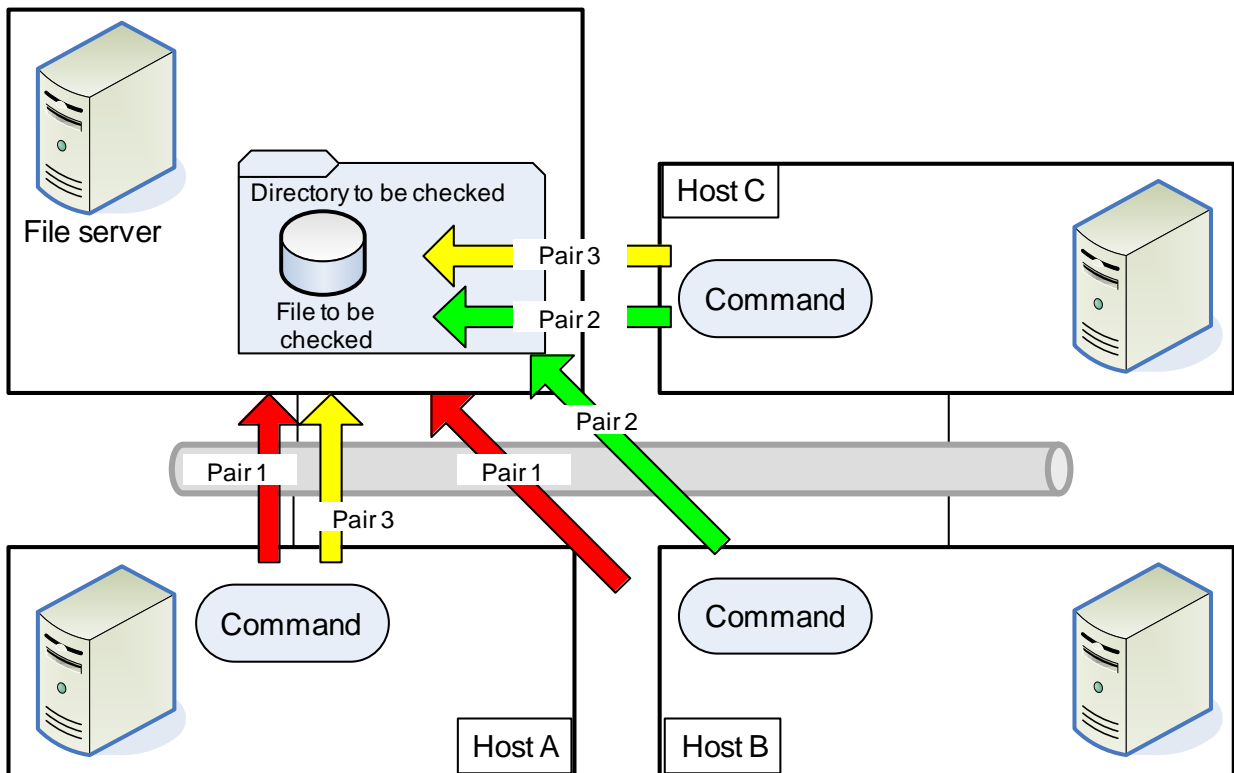

Figure 7. Checking process when three hosts perform operations on the same file

5.2 If files remain in the path to be checked

If this command is forcibly terminated or if the network is disconnected, files created by the `hulntwchk` command might remain in the directory to be checked. In this case, manually delete the files.

The following table shows the list of files that are created by the `hulntwchk` command.

Table 7. List of files that are created during check processing performed by the hulntwchk command

Path where files are created	File name
Path to Shared Environment by HULFT Products (-h *hulsharepath*)	`.#HULLOCK.`*host-name*`.`*process-ID*`.master`
	`.#HULLOCK.master`
	host-name`.`*process-ID*`.endwaitfile`
	host-name`.`*process-ID*`.endwaitfile.EXCL`
	`endwaitfile`
	`endwaitfile.EXCL`
	`.#HULLOCK.`*host-name*`.`*process-ID*`.0`
	`.#HULLOCK.`*host-name*`.`*process-ID*`.1`
	`.#HULLOCK.0`
	`.#HULLOCK.1`
Path to be checked (-p *targetpath*)	`.#HULLOCK.`*host-name*`.`*process-ID*`.0`
	`.#HULLOCK.`*host-name*`.`*process-ID*`.1`
	`.#HULLOCK.0`
	`.#HULLOCK.1`
	`.#HULLOCK.0.EXCL`
	`.#HULLOCK.1.EXCL`
Name of the file to be checked (-f *targetfile*)	`HULSTOP.stop`

5.3 If multiple users in the same environment perform operations on the same file

Depending on the environment settings, some users might not be able to perform operations on a file because of their user rights, even if file operations are performed from the same host.

Because of this, if you attempt to install multiple HULFT products in the same environment and to operate each HULFT product by using different user rights, execute the `hulntwchk` command by using each of the user rights in mutual environment check mode.
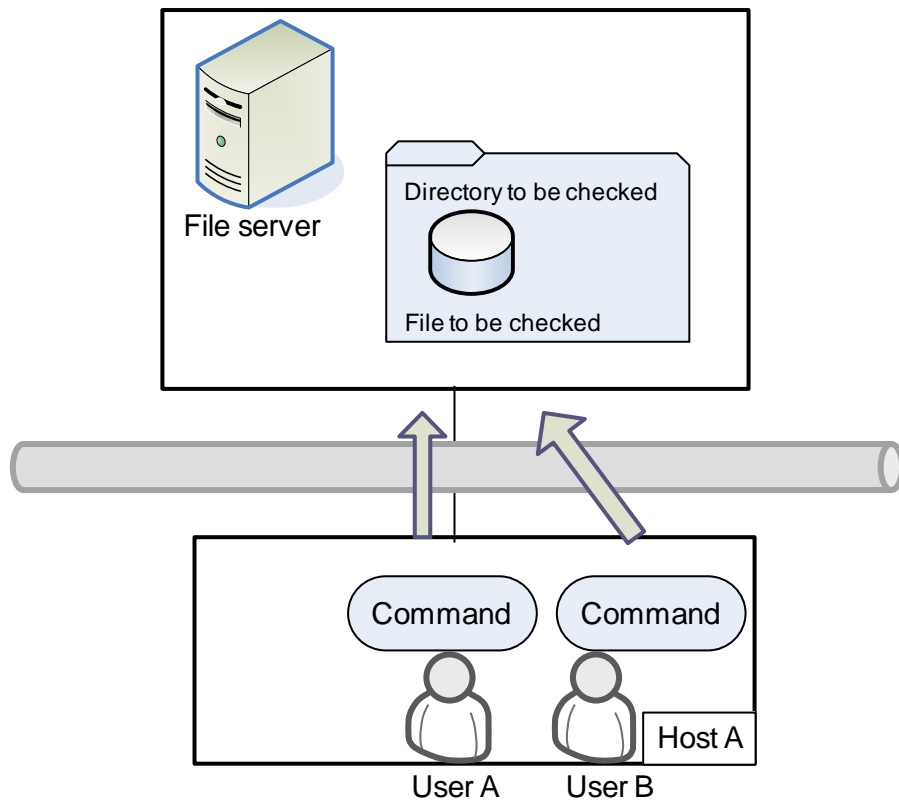

Figure 8. Checking a networked file when multiple users in the same environment perform operations on the same file

5.4   If multiple directories on the file server are checked

Depending on the file server settings, different access permissions might be granted for each directory. Because of this, if you attempt to use multiple directories on the same file server, execute the `hulntwchk` command for each directory.

You need to execute this command for the subdirectories of the directories that were successfully checked.
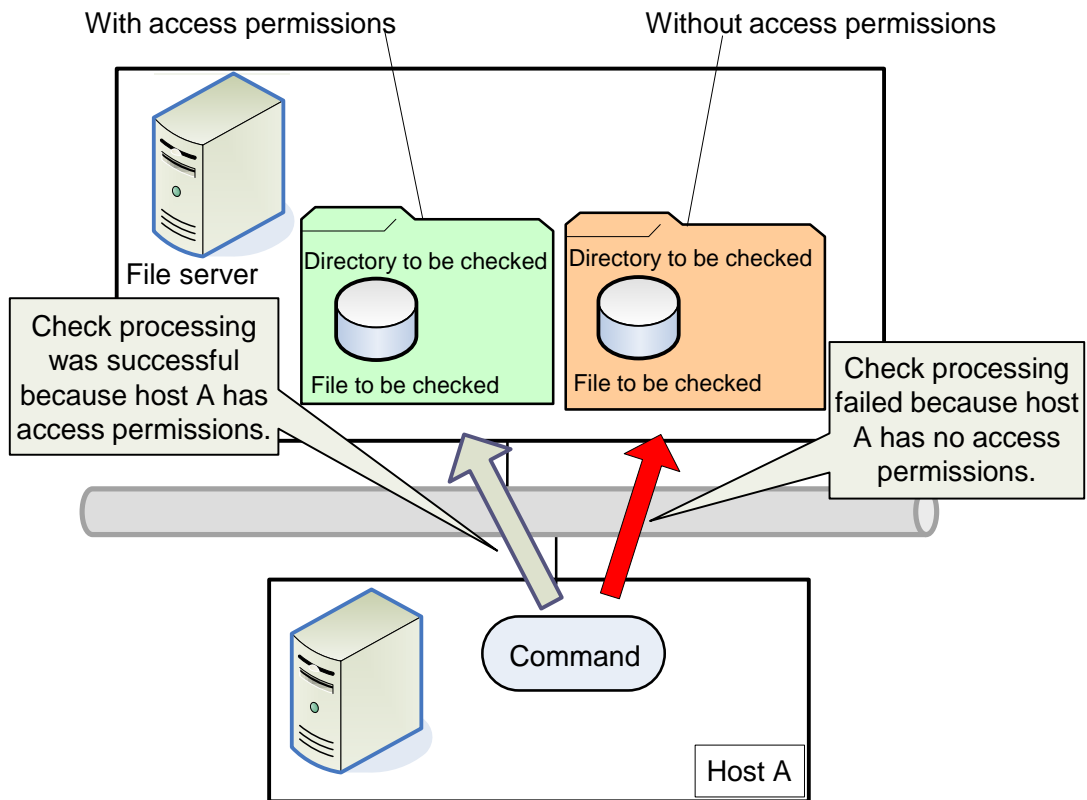
Figure 8. If different permissions are granted for each directory on a file server

5.5   Procedure for executing the hulntwchk command with Local System account (in Windows)

The service of HULFT products for Windows starts up with the Local System account by default. If you perform operations by using the service of HULFT products for Windows without changing the settings, execute the `hulntwchk` command with the Local System account.

If you use the `at` command to execute the `hulntwchk` command from the task scheduler, you can perform check processing with the Local System account.

The following shows an execution example:

1. Check the current time.

```
C:\Users\Administrator>time /T

12:30
```

2. Use the `at` command to register a schedule so that the `hulntwchk` command will start three minutes after the current time.

```
C:\Users\Administrator>at 12:33 "C:\HULFT Family\hulntwchk" -p
"\\nas\work" -h "C:\HULFT Family" -o "C:\HULFT Family\logfile.txt"
```

> Remarks: For details on the task scheduler and the `at` command, see the Windows documentation.

5.6 User accounts used when the hulntwchk command is executed in a UNIX environment

When you execute the `hulntwchk` command in a UNIX environment, make sure to do so from the user account that will be used to perform HULFT products operation.

Capability Check Tool for Networked File Function

First edition: December 1, 2014

Third edition: May 1, 2016

Saison Information Systems Co., Ltd.