RELIABLE FILE TRANSFER

HULFT7e

Windows

Administration Manual

Note

* Reproduction of this document by any means, in whole or in part, without the prior written consent of the manufacturer, is strictly prohibited by copyright regulations.

* The contents of this document and its corresponding product are subject to change without prior notice.

Preface

This document describes operational environment and system outline. In addition, it explains the environment settings required for the operation of HULFT on Windows. The document is for the individuals who are involved in designing and establishing applications or systems as well as for those who engage in the installation of HULFT.

The screen shots and operational procedure in this document are based on assumption that Microsoft Windows XP Professional is used.

The image and explanation may differ from your environment depending on the type or setting of OS in use.

Structure of This Document

This document is composed of the following chapters:

Chapter 1	HULFT Operational Environment
Chapter 2	Outline of HULFT System
Chapter 3	Environment Settings for Windows
Chapter 4	Points to be Noted for HULFT Operation
Appendix 1	I/O Format of Operation Log

Symbols and Notations

<Description of Product Name>

- In this document, HULFT7e for Windows is generally named as 'HULFT.'
- This document generically names following products as 'Nonstop':

  HULFT for Himalaya Ver.5
  HULFT7 for NSKJ
  HULFT7 for NSKH
  NonStop Kernel
  NonStop Server

- In the case the document indicates each product, relevant product name is provided.

<Version, Level, and Revision of HULFT>

Version information is displayed under following format:

Example: 7. 0. 0

a) b) c)

a): Version

b): Level

c): Revision

Upgrade of the number appeared in a)—Version Upgrade

Upgrade of the number appeared in b)—Level Upgrade

Upgrade of the number appeared in c)—Revision Upgrade

<Command or Control Card Explanation>

[ ]: Brackets indicate that the enclosed items are optional.

{}: Braces indicate that enclosed items are multiple options, from which one option must be selected.

.....: Repeat symbol indicates options, which should be repeated if necessary. The symbol may follow single word or a group of options enclosed within either brackets or braces.The part enclosed within either brackets or braces in a format is regarded as one unit. Repeat the whole part in between the symbols on a unit basis.

| : Vertical bar is used to set off options.

Italics: Italics indicate a variable. (a value that varies depending on the target or state)

Example: *yyyymmdd*

Type in comma (,) and equal sign (=) in the exact location as they are represented.

<Command or Management Information Settings>

Uppercase characters:	Uppercase characters indicate that uppercase alphabets (A-Z) can be used.
Lowercase characters:	Lowercase characters indicate that lowercase alphabets (a-z) can be used.
Alphabets:	Alphabets indicate that both uppercase (A-Z) and lowercase (a-z) alphabets can be used.
Alphanumeric characters:	Alphanumeric characters indicate uppercase and lowercase alphabets (A-Z, a-z) as well as numeral (0-9) can be used.

Where to Look up

Depending on the users and the purpose of usage, HULFT manuals are classified as follows. For file names and stored locations and so on, refer to the 'readme' file included in the installation CD.

HULFT7e Functions Manual

The manual describes the functions of HULFT. The explanation is for the first-time users of HULFT as well as those who are in charge of the introduction of HULFT.

HULFT7e New Functions and Incompatibility Manual

The manual explains the functions of new product and describes incompatibility with old versions. It is for the individuals who are involved in introduction of HULFT, upgrade of version, level, and revision, and product transition.

HULFT7e Windows Installation Manual

The manual explains all the procedures from product installation, environment settings, to file transfer (File Transfer Test). It is for new users of HULFT for Windows and for system administrators who implement version and/or revision upgrade.

HULFT7e Windows Administration Manual

The manual describes environment settings required for the operation of HULFT on Windows. The explanation is for the individuals who are involved in designing and establishing application system and for those who are in charge of the introduction of HULFT.

HULFT7e Windows Operation Manual

The manual describes the environment settings required for the operation of HULFT on Windows. The explanation is for the individuals who are involved in designing and establishing application systems and for those who engage in daily operation of the system.

HULFT7e Windows Error Codes and Messages

The manual describes the error codes and message contents of HULFT. It is for the individuals who are involved in designing and establishing application systems and for those who engage in daily operation of the system.

HULFT

Table of Contents

Chapter 1

HULFT Operational Environment

This chapter describes the operational environment of HULFT and the connection forms for using HULFT.

1.1 Connection Forms

HULFT enables file transfer among Mainframes, office computers, UNIX, Linux, Windows, and Nonstop using TCP/IP which is de facto standard protocol in the industry.

Figure 1.1 illustrates a connection example using HULFT.

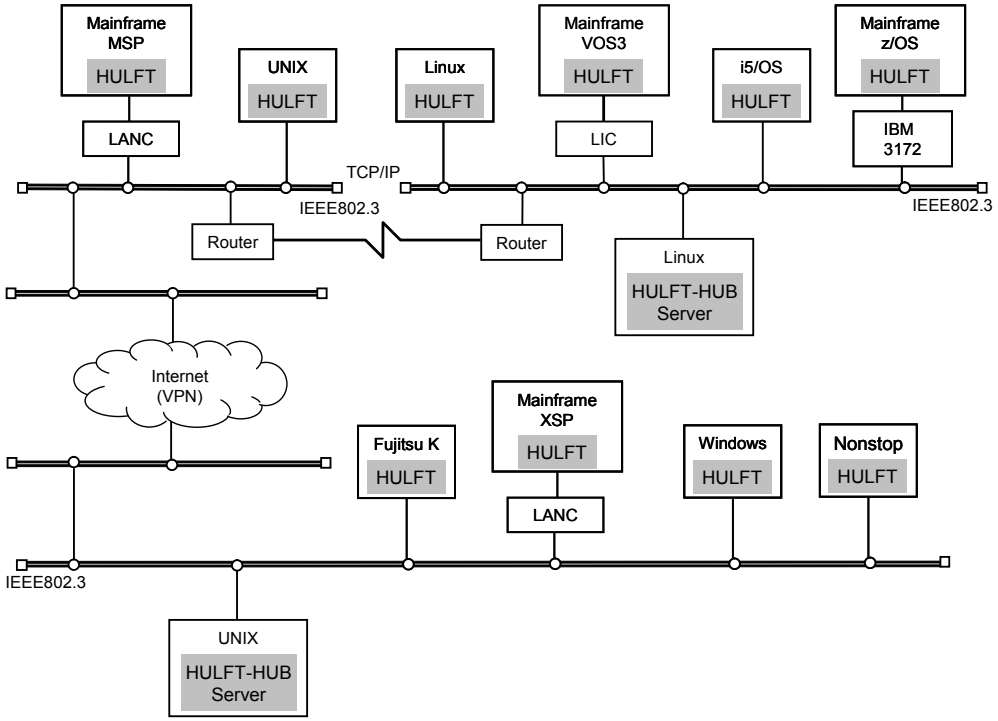


Figure 1.1 HULFT Connection Forms

[Remarks] HULFT-HUB is a middleware that realizes hub-and-spoke type data interaction by cooperating with HULFT. The application consists of HULFT-HUB Server, which enables users to carry out data transfer and accumulation, and HULFT-HUB Manager, on which users can set and operate HULFT-HUB Server.

1.2 Structure of HULFT Operation

HULFT for Windows consists of four communication systems—Service, Send, Receive, and Request Acknowledge—and the Management screen system. The Management screen system interactively controls the Send and Receive status. The four communication systems operate as resident programs.

In addition, a related product HULFT Manager enables users to inquire about the Send and Receive status, to register, update and delete the Send and the Receive Management Information from remote Windows-based personal computers.

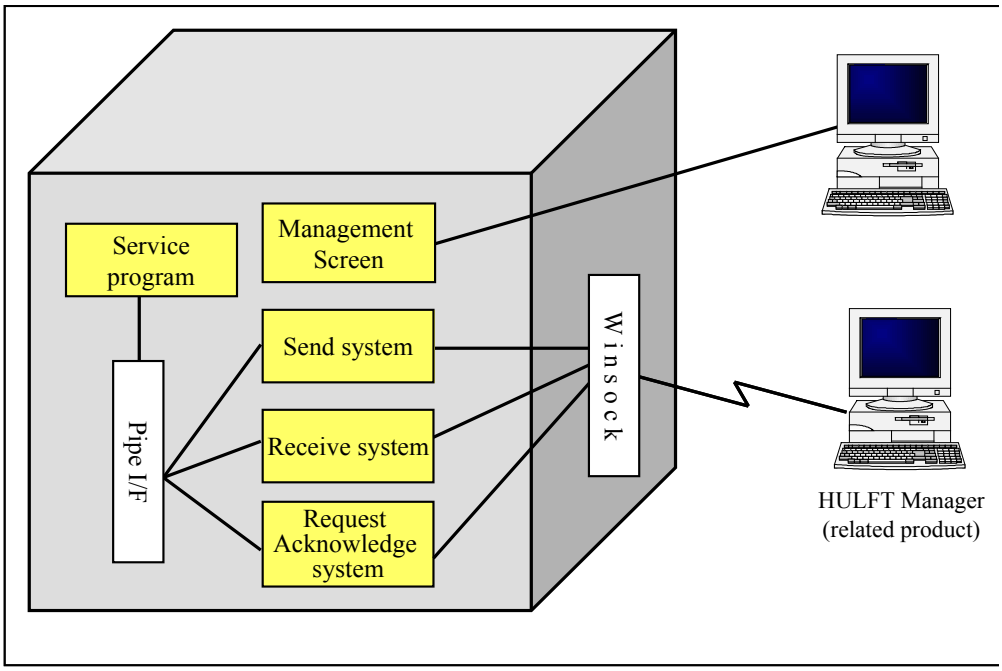


Figure 1.2 Structure of HULFT Operation

1.3 Software Requirements

For the software requirements, visit our website from the following URL:

http://www.hulft.com/

~~~~~~~~~~~~~~~~~~~~~~~~ Memo ~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~

Chapter 2

Outline of HULFT System

The mechanism of the Send, Receive and Request Acknowledge systems is explained in this chapter.

2.1 Transfer System

Send process is the process which sends files from the sending side to a specified host.

Receive process is the process which receives the data sent by the host on the sending side and creates a Receive file at the host on the receiving side.

Either of the hosts can start both Send process and Receive process.

2.1.1 Flow of Send Process

In the Send process started at the sending side, the Send file is transferred by issuing the command. This request is called the 'Send File' in HULFT.

This process is based on the assumption that both Send process on the sending side and Receive process on the receiving side have already started.

The flow of the Send process started from the sending side is explained in Figure 2.1.

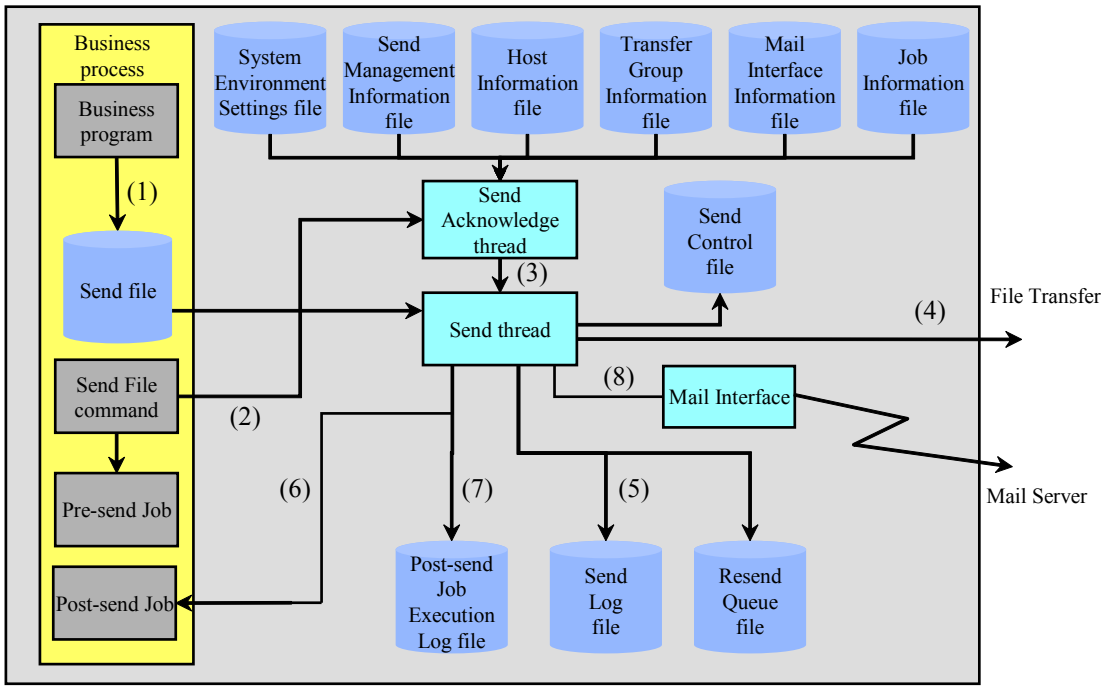


Figure 2.1 Send Process Started at Sending Side

(1) Creation of Send files

Send files are created in the course of general business operations.

(2) Issue of Send File command

Issueing Send File command to the Send Acknowledge thread can start the Send process from the sending side. Send File program can be executed automatically by incorporating it in the same operation as mentioned in step (1) Creation of Send files. Moreover, the Send process can also be started on the Management screen.

Setting a Pre-send Job in the Send Management Information enables HULFT to execute the job before the Send File command issues a request to the Send Acknowledge thread.

(3) Creation of Send thread

The Send process receives the Send File command and creates the Send thread within itself according to the criteria of each management information file. This is carried out at the time of each transfer.

(4) Execution of sending

The Send thread performs code conversion or file compression based on the settings of the Send Management Information and transmits the data of the Send file to the remote host.

(5) Registration of Send Log information

After completing the send operation, the Send thread writes the result of the send operation in the Send Log file. If the Send Process terminates unsuccessfully, Send thread registers the information on the failed transfer in the Resend Queue file.

(6) Start-up of Post-send Job

The Send thread activates the Post-send Job registered in the Job Information according to the condition registered in the Send Management Information. Regarding the job to be activated, HULFT activates either the job specified for successful termination of the sending processing or the one for unsuccessful termination of the processing, depending on the transfer result.

(7) Recording Job Execution Log information

After completing the sending operation, the Send thread writes the execution result of the job in the Job Execution Log file.

(8) Mail transmission

The Send thread transmits e-mail according to the condition registered in the Mail Interface Information.

[Note]

- **The term 'job' indicates the operation unit, such as application programs and batch files.**
- **The execution result of Pre-send Job is not written in log files.**
- **Pre-send Job is not executed in the case of the Resend File.**

2.1.2 Flow of Send Process Started at Receiving Side

The Send Process can also be started by a request from the receiving side. This request is known as the 'Send Request' in HULFT.

This process is based on the assumption that both Send process and Request Acknowledge process on the sending side as well as the Receive process on the receiving side have already started.

The flow of the Send process started at the receiving side is explained in Figure 2.2.

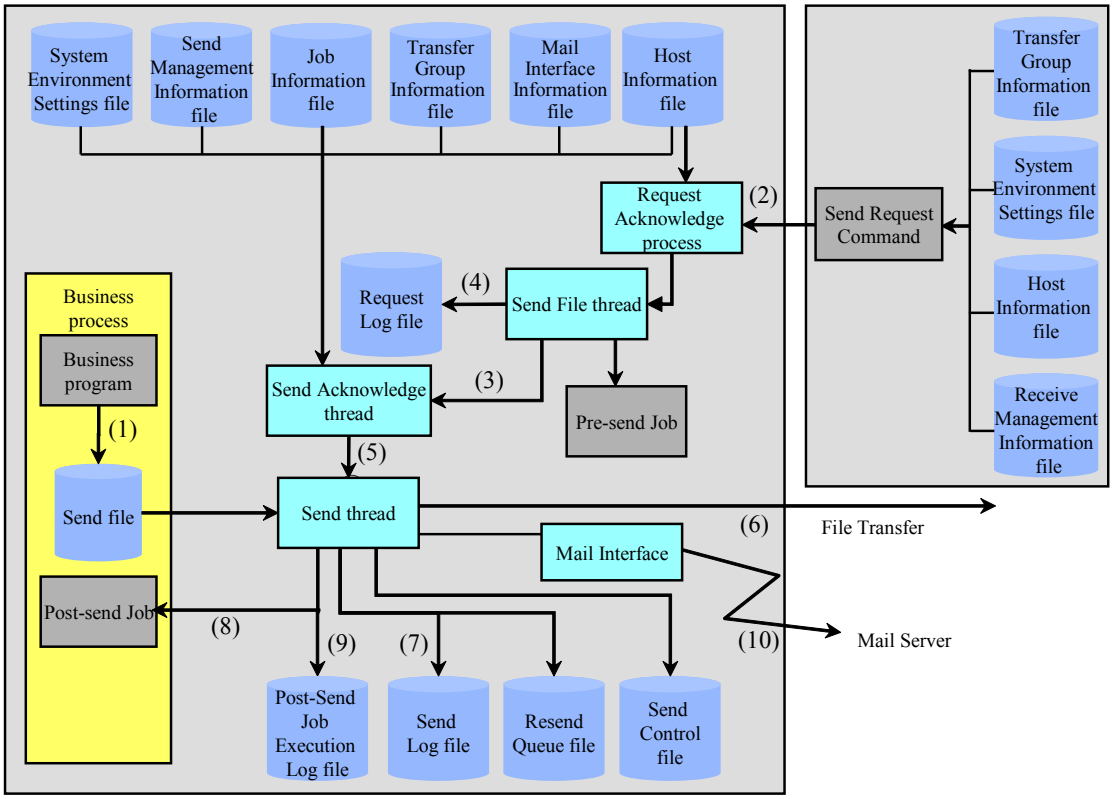


Figure 2.2 Send Process Started at Receiving Side

(1) Creation of Send files

Send files are created in the course of general business operations.

(2) Send Request queue from receiving side

Request Acknowledge process receives the Send Request from the receiving side. When the Send Request is received, a Send File thread is created and operations from (3) onwards are executed.

Setting a Pre-send Job in the Send Management Information enables HULFT to execute the job before the processing explained in above (3).

(3) Issue of Send File command

The Send File thread requests the send operation to the Send Acknowledge thread.

(4) Registration of Request Acknowledge Log information

The Send File thread writes the requests from the receiving side in the Request Acknowledge Log file.

(5) Creation of Send thread

The Send Acknowledge thread receives the Send File command and creates Send thread situated within itself according to the criteria in each management information file. This operation is carried out at the time of each transfer.

(6) Execution of sending

The Send thread carries out code conversion or file compression based on the settings of the Send Management Information and transmits the Send file data to the remote host.

(7) Registration of Send Log information

After completing the send operation, the Send thread writes the result of the send operation in the Send Log file. In addition, when the Send Process terminates unsuccessfully, Send thread puts the information on the transfer that terminates unsuccessfully into Resend Queue file.

(8) Startup of Post-send Job

The Send thread activates the Post-send Job registered in the Job Information according to the condition registered in the Send Management Information. The job started can be either the one specified for normal end or the one specified for unsuccessful termination of the Send process, depending on the transfer result.

(9) Registration of job execution log information

The Send thread writes the execution result of the job in the Job Execution Log file after the send operation.

(10) Mail transmission

The Send thread transmits email messages according to the conditions registered in the Mail Interface Information.

[Note]

- **The term 'job' indicates the operation unit, such as application programs and batch files.**
- **The execution result of Pre-send Job is not written in log files.**
- **Pre-send Job is not executed in the case of the Resend File.**

2.1.3 Flow of Receive Process

The flow of Receive Process explained below is the Receive process when the remote host of receive is in one location (single receive). The case of multiple remote hosts is explained in "2.1.17 Receive Process from Multiple Hosts."

This process is based on the assumption that both Send process on the sending side and Receive process on the receiving side have already started.

The flow of Receive Process is illustrated in Figure 2.3.

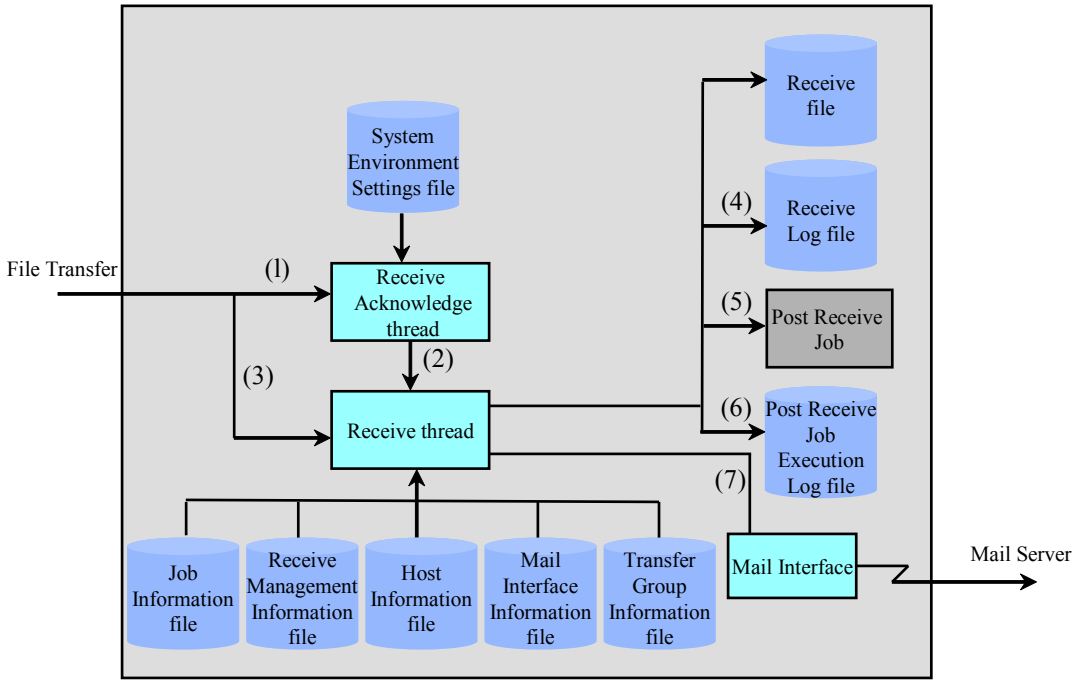


Figure 2.3 Flow of Receive Process

(1) Waiting of connection from sending side

The Receive Acknowledge thread waits for the connection from the sending side.

(2) Creation of Receive thread

The Receive Acknowledge thread creates the Receive thread when the thread is connected from the sending side. The thread carrys out this operation with respect to each transfer.

(3) Execution of receiving

The Receive thread receives the data according to the file information related to the transfer received from the sending side and the Receive Management Information set on the receiving side. After being given code conversion and/or decompression, the received data is written into a Receive file.

(4) Recording Receive Log information

After completing the Receive operation, the Receive thread writes the result of the Receive operation in the Receive Log file.

(5) Startup of Post-receive Job

The Receive thread activates the Post-receive Job registered in the Job Information according to the condition registered in the Receive Management Information. Regarding Job startup, a job for either successful or unsuccessful termination of the Receive process is activated in accordance with transfer result.

(6) Recording job execution log information

The Receive thread writes the execution result of the job in the Job Execution Log file after the Receive operation.

(7) Mail transmission

The Receive thread transmits e-mail according to the condition registered in the Mail Interface Information.

2.1.4 Outline of Send Management

The Send system is managed by using HULFT operation command. The Management screen also allows to manage the information necessary for the Send system.

Figure 2.4 illustrates the outline of these management functions.

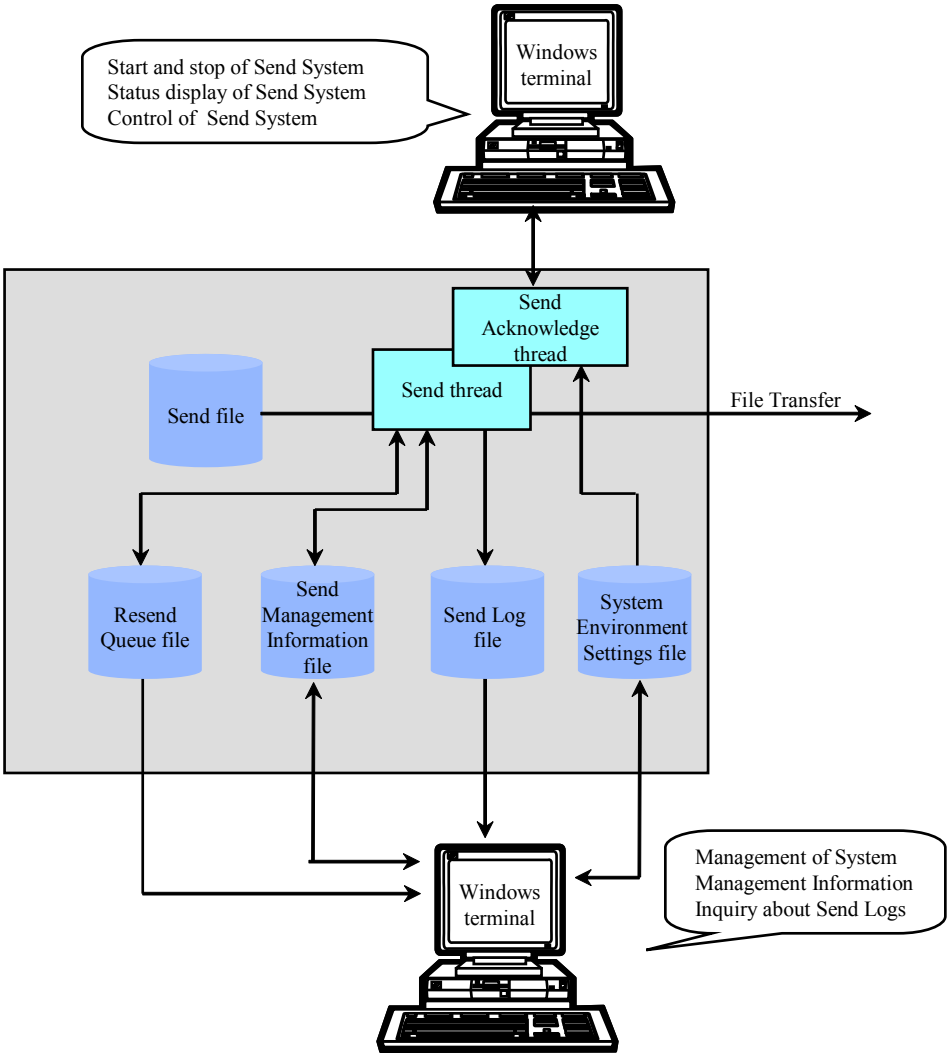


Figure 2.4 Outline of Send Management

2.1.5 Outline of Receive Management

The Receive system is managed by using HULFT operation command. The Management screen also allows manage the information necessary for the Receive system.

Figure 2.5 illustrates the outline of these management functions.

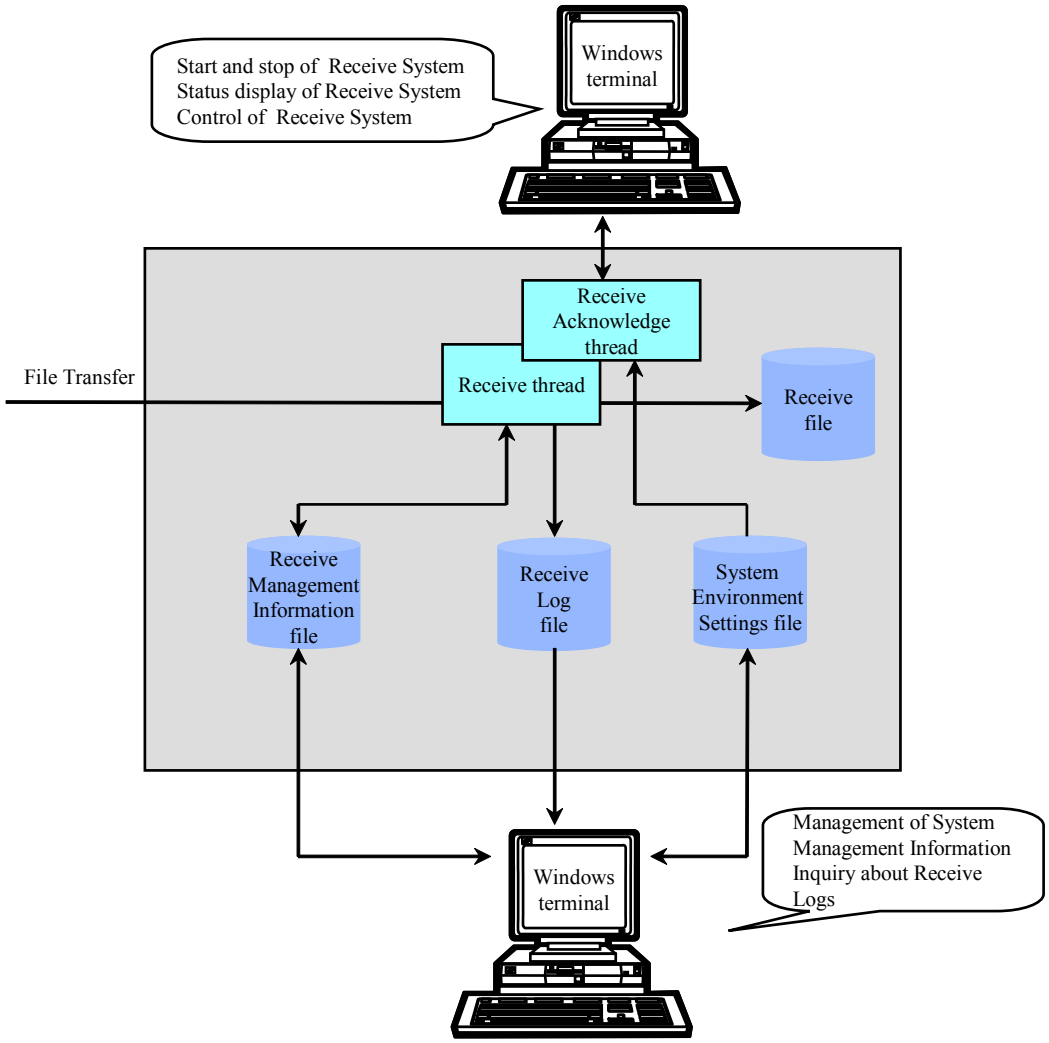


Figure 2.5 Outline of Receive Management

2.1.6 Registration Method of Send Operation

(1) Registration Information for Send Management

In the case of send operations, it is necessary to register management information beforehand in the system management information files. Refer to *Operation Manual* for the explanation of each field.

The outline of how the information is registered and the relationship of these information is explained in Figure 2.6 given below.

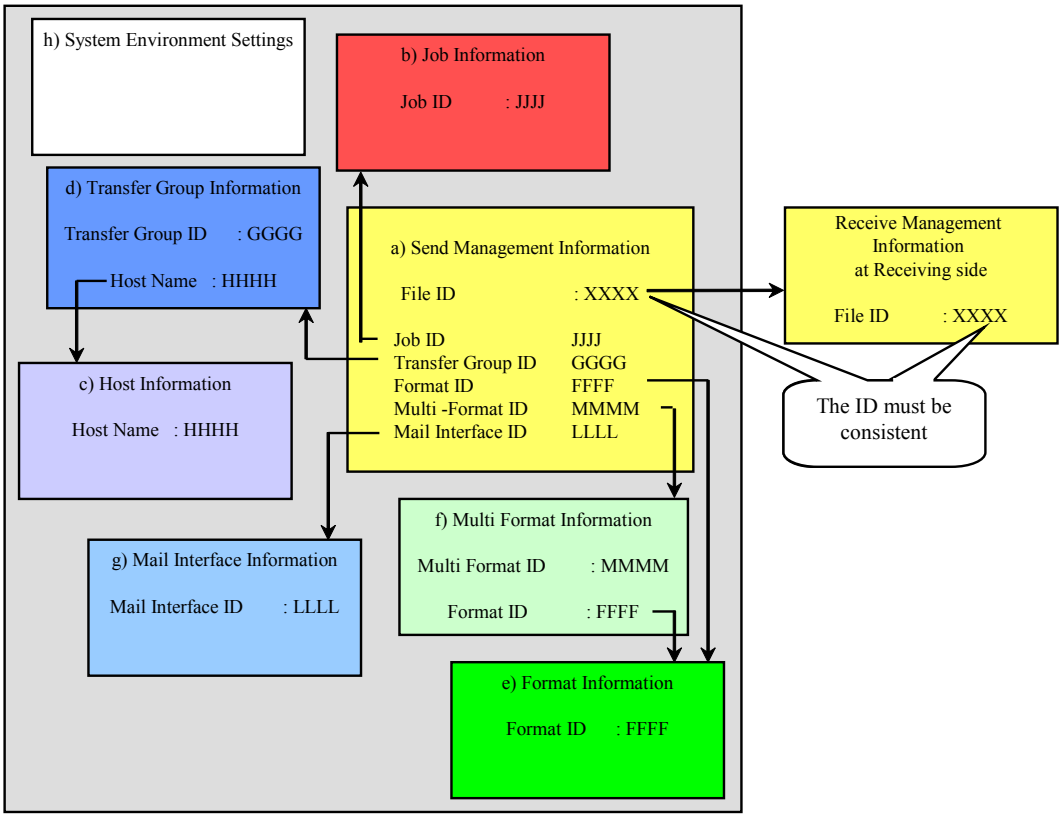


Figure 2.6 Registration Information for Send Management

a) Send Management Information

This is the core management information of send operations. It is necessary to register file IDs corresponding to the ones in the Receive Management Information of the receiving side. Fields such as Send file attributes and transfer conditions are registered.

b) Job Information

The Pre-send Job Information and Job Information executed upon the completion of the send operation are registered. In the jobs that are executed when the send operation is completed, there are jobs that are started when the send operation is completed successfully and there are jobs that are started when the send operation is completed unsuccessfully. If multiple jobs are registered in one Job ID, they are executed in the order of their registration. This is not necessary if no jobs are started.

c) Host Information

Information of the remote hosts that are the Send destinations is registered.

d) Transfer Group Information

Remote hosts of Send destinations (hosts that are registered in Host Information) are registered in groups. Multicasting is possible if multiple remote hosts are registered in one group.

e) Format Information

The Format Information of Send file is registered. Register the information in the case of the files that require field-by-field consideration of code conversion or receiving in CSV or XML format. On the other hand, the registration is not necessary in the case of Text Transfer or Binary Transfer without code conversion. Concerning Multi Format Transfer, it is necessary to register the Format Information that corresponds to each record.

f) Multi Format Information

The Multi Format Information of Send file is registered. Register the information in the case of the files that require record-by-record consideration of code conversion or receiving in CSV or XML format. On the other hand, the registration is not necessary in the case of the Text Transfer, the Binary Transfer without code conversion, or the Format Transfer.

g) Mail Interface Information

Mail information to be transmitted after Send process is registered. The mail addresses and the body message for mail interface are registered. When no mail message is transmitted, this is not necessary.

h) System Environment Settings

Generation path of work file, Send Process Multiplex Level, etc. are are defined. This is registered only once for each system.

(2) Registration method

Each of the above information is registered by using the Management screen. There is another method of registering information other than the System Environment Settings by creating a parameter file (definition card) using an editor such as Notepad and registering the information with the system management commands.

If you have installed HULFT Manager, the registration on HULFT Manager is also available.

2.1.7 Registration Method of Receive Operation

(1) Registration Information for Receive Management

In the case of receive operations, it is necessary to register management information beforehand in the system management information files. Refer to *Operation Manual* for the explanation on each field. The outline of how the information is registered and the relation of these information is explained in Figure 2.7 given below.

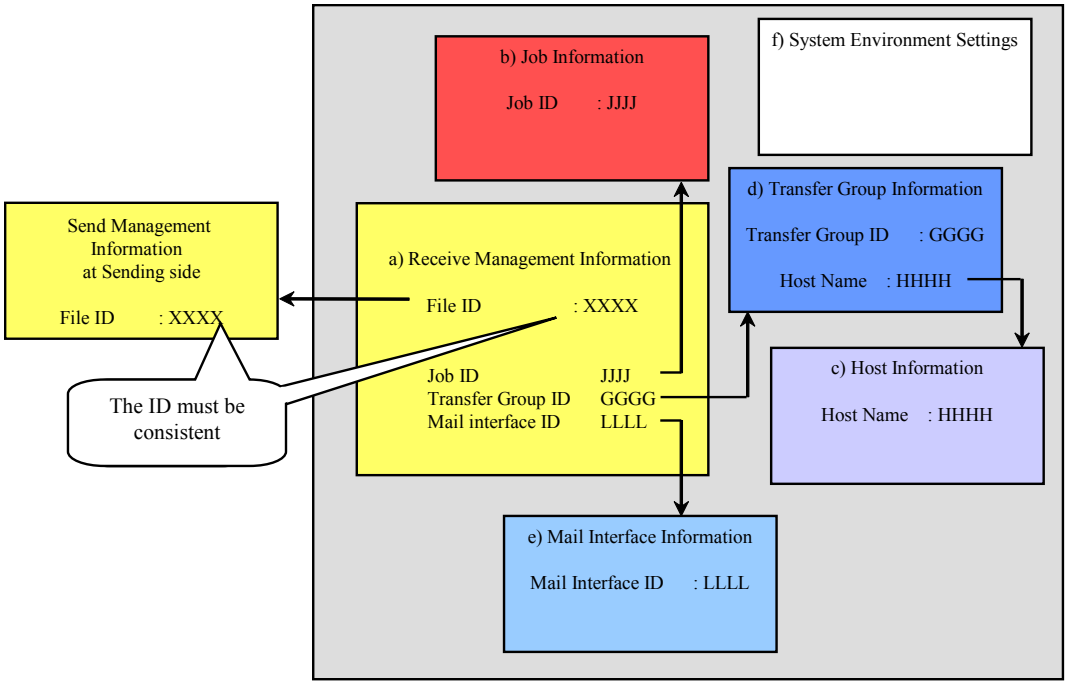


Figure 2.7 Registration Information for Receive Management

a) Receive Management Information

The registration for the Receive operation has to be done so that it corresponds to the File ID of the Send Management Information on the sending side without fail. The attribute of the Receive file and the transfer conditions are registered.

b) Job Information

The information of the job executed when the Receive operation is completed is registered. There are jobs that are executed when the Receive operation is terminated normally and jobs that are executed when the Receive operation is terminated abnormally. If multiple jobs are registered in one Job ID, they are executed in the order of their registration. This is not necessary if no jobs are started.

c) Host Information

Information of the remote hosts that are the sending host is registered.

d) Transfer Group Information

Remote host of the sending host (host registered in the Host Information) is registered. This is specified when the Send Request is made to the remote host. This is not necessary when no request is issued.

This must be registered if you wish to handle receiving from an non-registered host as error.

e) Mail Interface Information

Mail information to be transmitted after Receive process is registered. The mail addresses and the body message for mail interface are registered. When no mail message is transmitted, this is not necessary.

f) System Environment Settings

Creation path of work file, Receive Acknowledge Port No. etc. are defined. It is registered only once for each system.

(2) Registration Method

Each of the above information is registered by using the Management screen. There is another method of registering information other than the System Environment Settings by creating a parameter file (definition card) using an editor such as Notepad and registering the information with the system management commands.

If you have installed HULFT Manager, the registration on HULFT Manager is also available.

2.1.8 Message Transmission

It is possible to Send optional message simultaneously when file is transferred from the Send system to the Receive system. Message can be used for 'outputting to receive and send logs', 'changing various management information dynamically', (for instance, change receive file name dynamically), 'passing it to startup job as environment variable' etc.

(1) Message Transmission Function

To transmit a message, specify a parameter for message in the issued request command. Message should be specified in the Send File command when sending, while it should be specified in the Send Request command when executing Send Request.

To dynamically modify the setting values in each management information using messages, users need setting that the values to be modified are replaced with messages (from '$MSG0' to '$MSG5').

A maximum of 6 messages can be specified, and each will be replaced for the one where the message type of management information on the receiving side and the message type (0 to5) on the sending side match. When the message type specified in management information is not specified on the sending side, it is handled as 0 byte message on the receiving side.

To replace messages, 'Message Dynamic Parameter Specification' of the System Environment Settings file should be made effective. Refer to "3.4 System Environment Settings" for system environment settings.

[Note] When the message specified on the message sending side is replaced, an error occurs before the execution of file transfer in the following cases:

- **When the size which can be specified by management information exceeds**
- **When the file name cannot be applied upon Send file name replacement**

(2) Information Which Can be Specified in Messages

The setting values or ID in each management information, which can be specified by message replacement, are as follows:

[Send Management Information]

- Send File Name
- Transfer Group ID
- Pre-send Job ID
- Successful Job ID
- Multi Format ID
- Format ID
- Interface DBID
- Mail Interface ID

[Receive Management Information]

- Receive File Name
- Successful Job ID
- Transfer Group ID
- Mail Interface ID

[Job Information]

- Argument of startup jobs

[Mail Interface Information]

- Title
- Address
- CC
- Text

[Note] Each setting value or ID of HULFT for Mainframe and HULFT for i5OS is registered in uppercase alphabet. Specifying the messages in lowercase alphabet keeps the messages in lower case as they are. Set the messages in uppercase alphabet, when specifying the message that is to be replaced with IDs.

(3) Message Replacement Process of Sending

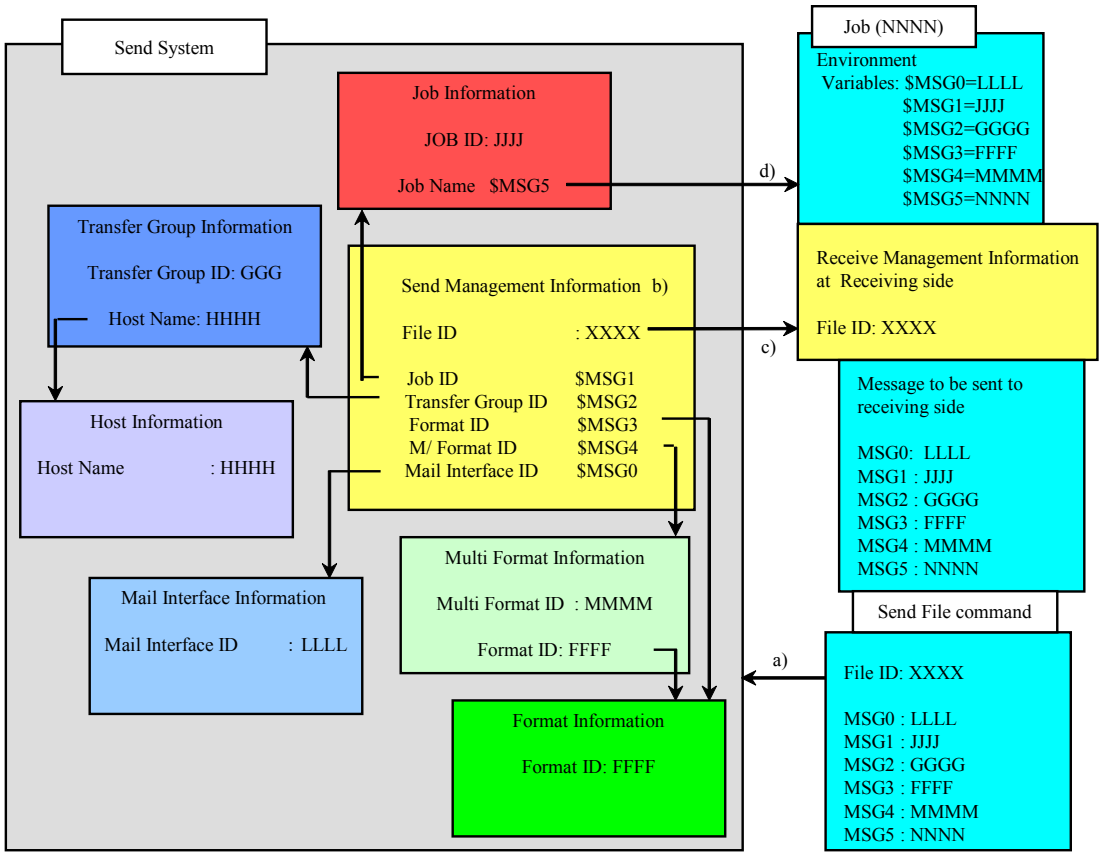


Figure 2.8 Message Transmission (of Sending)

a) Message transmission instruction from Send File command

The Send File is issued to the Send system from Send File command after specifying the File ID and the Message.

b) Modification to Send Management Information content

The Send system replaces the Message specified by the Send File command with $MSG0 to $MSG5 set in the Send Management Information of the specified 'File ID'. The Send process is executed according to the replaced value.

c) Message transmitssion to receiving side

The Send system notifies the Message specified in the Send File program, to the Receive program before the data transfer starts.

d) Environment parameter setting to subsequent job

The Send system sets the Message specified in the Send File program as environment variables ($MSG0 to $MSG5) so that Post-send Job can replace the setting values in the Job Information with the specified messages.

(4) Message Replacement Process of Receiving

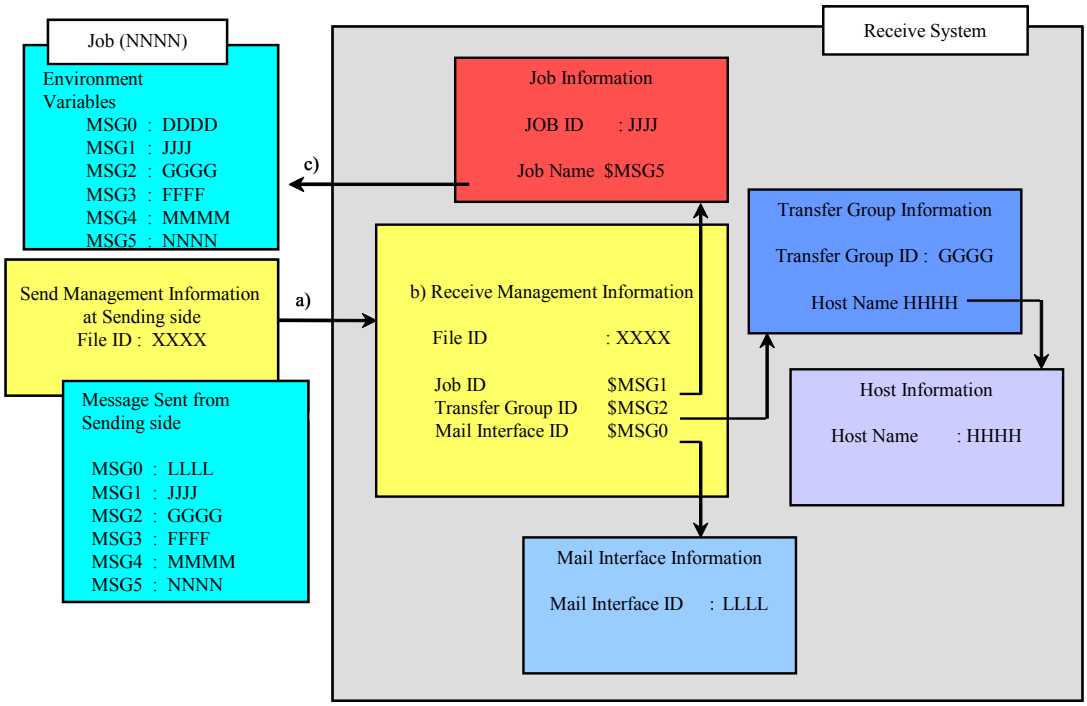


Figure 2.9 Message Transmission (of Receiving)

a) Acknowledge of message notified from Send system

The Receive system accepts the notification of the File ID and the Message from the Send system.

b) Modification to Receive Management Information contents

The Receive system replaces $MSG0 to $MSG5 set in Receive Management Information of the File ID which is notified from the Send system with the Message specified in the Send File command. The Receive process is executed according to the replaced value.

c) Environment variable setting for subsequent job

The Receive system sets the Message notified by the Send system as environment variables ($MSG0 to $MSG5) so that Post-receive Job can replace the setting values in the Job Information with the messages.

(5) Message Replacement Process of Send Request

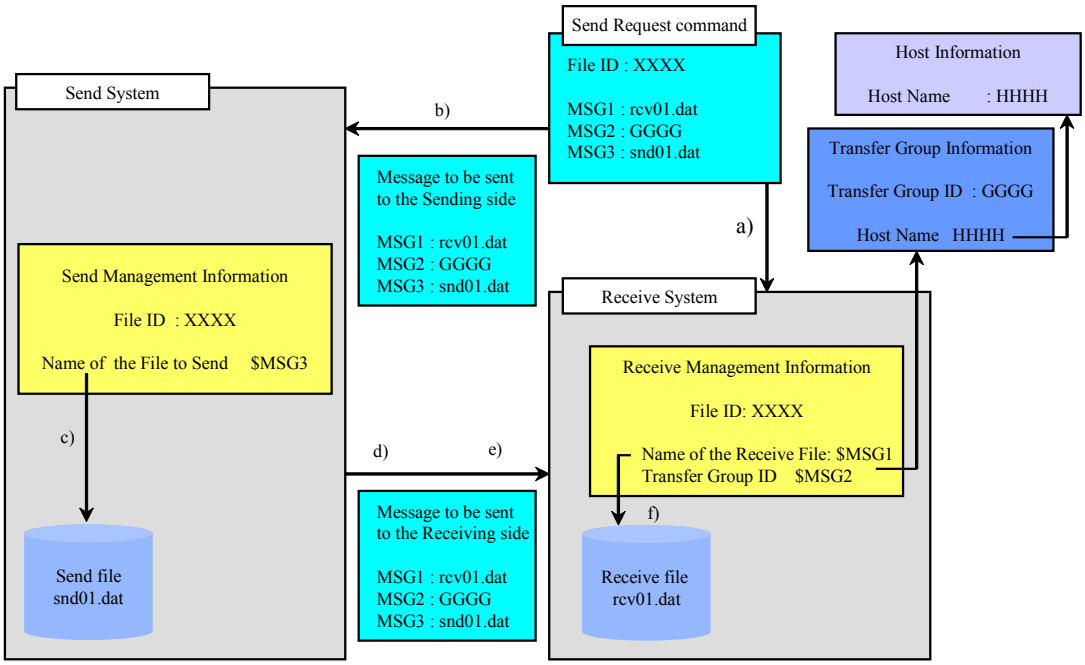


Figure 2.10 Message Transmission (of Send Request)

a) Message transmission instruction from Send Request command

The File ID and the Message are specified by the Send Request command. Send Request command replaces $MSG0 to $MSG5 set in the Receive Management Information with the specified Message.

b) Issue of Send Request

The Send Request command notifies specified the Message along with the request issue to the Send system.

c) Modification to Send Management Information contents

The Send system replaces $MSG0 to $MSG5 set in the Send Management Information of the File ID notified from Send Request command to notified Message. Send process is executed according to the replaced value.

d) Message transmission to receiving side

The Send system notifies the Message notified from the Send Request command to the Receive system before the data transfer.

e) Notification acceptance of message from the Send system

The Receive system receives the notification of the File ID and the Message from the Send system.

f) Change of content of Receive Management Information.

The Receive system replaces the notified message with $MSG0 to $MSG5 set in the Receive Management Information of the File ID notified from the Send system. The Receive process is executed according to the replaced value.

2.1.9 Synchronous Transfer and Asynchronous Transfer

Synchronous transfer and asynchronous transfer can be selected for the Send File and the Send Request.

(1) Flow of Synchronous Transfer

If transfer is made by synchronous transfer, the Send File and the Send Request do not end until transfer process completes. As a result, user job can carry out the subsequent process after confirming the status code of the Send and Receive process.

The flow of synchronous transfer process is explained in Figure 2.11.

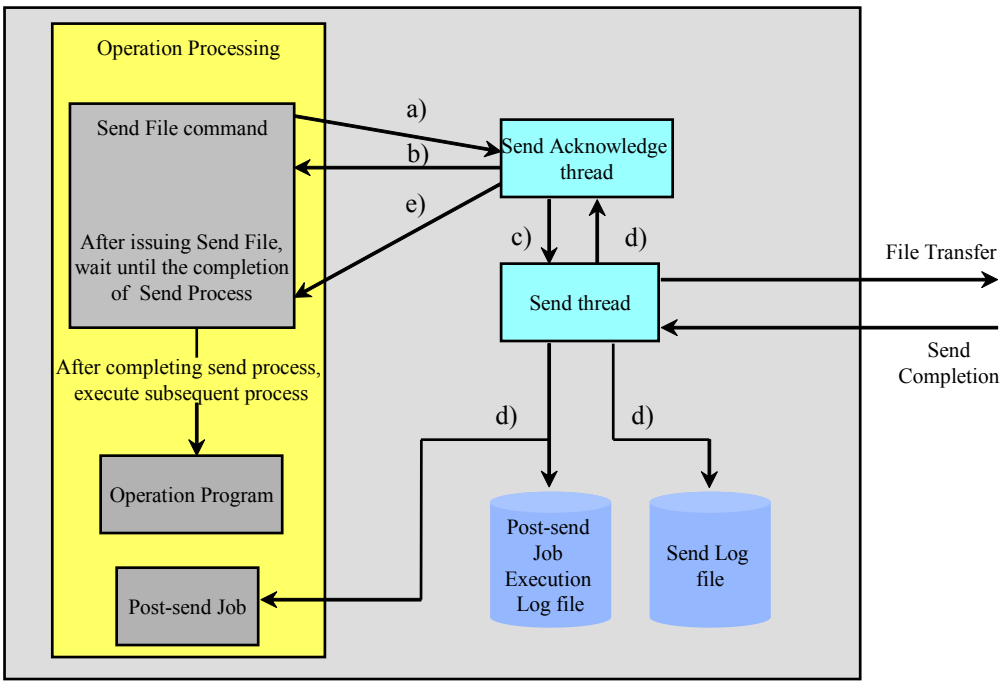


Figure 2.11 Synchronous Transfer

a) Startup of Send File command

Send File command issues a send command to the Send Acknowledge thread.

b) Notification of Send File

The Send Acknowledge thread notifies the Send File command of the acceptance of Send File.

c) Startup of Send thread

The Send thread is activated and file transfer is executed.

d) Notification of Send completion

After completing the sending operation, the Send thread notifies the Send Acknowledge Tread of the result of the file transfer and writes the result into the Send Log file. The job registered as the Post-send Job is started.

e) Completion of Send File command

The Send Acknowledge thread receives the Send completion notification from the Send thread and notifies the contents to the Send File command.

(2) Flow of Asynchronous Transfer

When the Send and Receive process is carried out in asynchronous transfer, the Send File or Send Request completes before the Send and Receive process ends. Actual Send and Receive Process is executed asynchronously with the user job that issued the request, and the status code is not notified to the user job.

The flow of asynchronous transfer is explained in Figure 2.12.

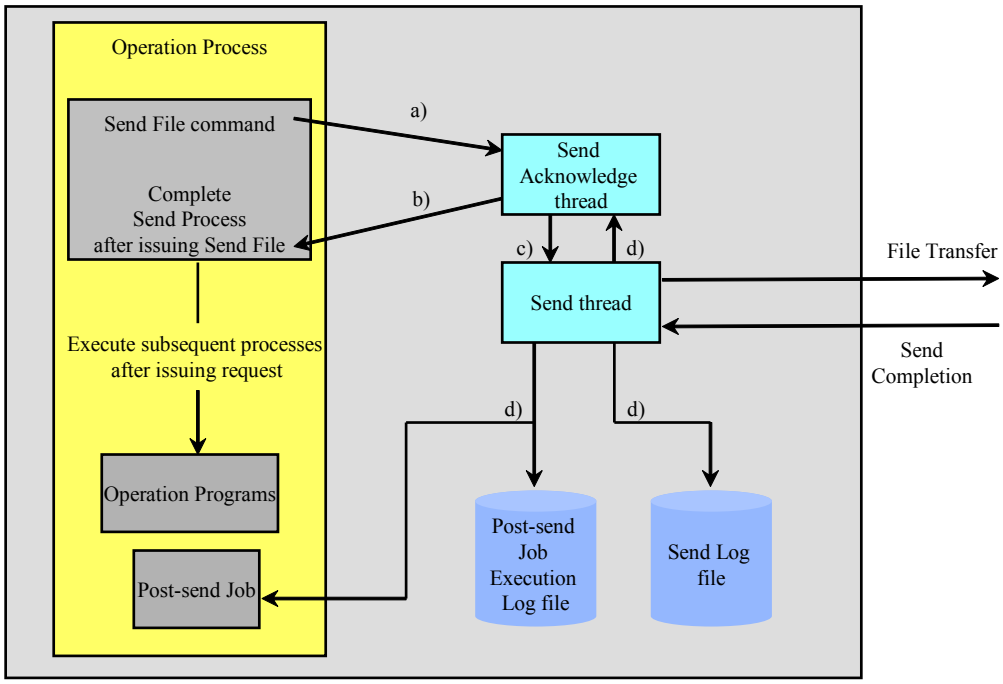


Figure 2.12 Asynchronous Transfer

a) Startup of Send File command

Send File command issues the send command to the Send Acknowledge thread.

b) Completion of Send File command

The Send Acknowledge thread notifies the Send File command of the acceptance of Send File. The Send File command ends after receiving notification.

c) Creation of Send thread

The Send thread is activated and file transfer is executed.

d) Notification of send completion

After completing the sending operation, the Send thread notifies the result of the file transfer to the Send Acknowledge thread and writes the result into the Send Log file. In addition, the Send thread starts the job registered as a Post-send Job.

2.1.10 Multicasting

The same Send file can be sent to multiple hosts. This can be implemented by registering multiple hosts in the Transfer Group Information. The Post-send Job is activated each time HULFT executes sending to a host. A maximum of 1000 hosts can be registered in a Transfer Group.

The flow of multicasting process is explained in Figure 2.13.

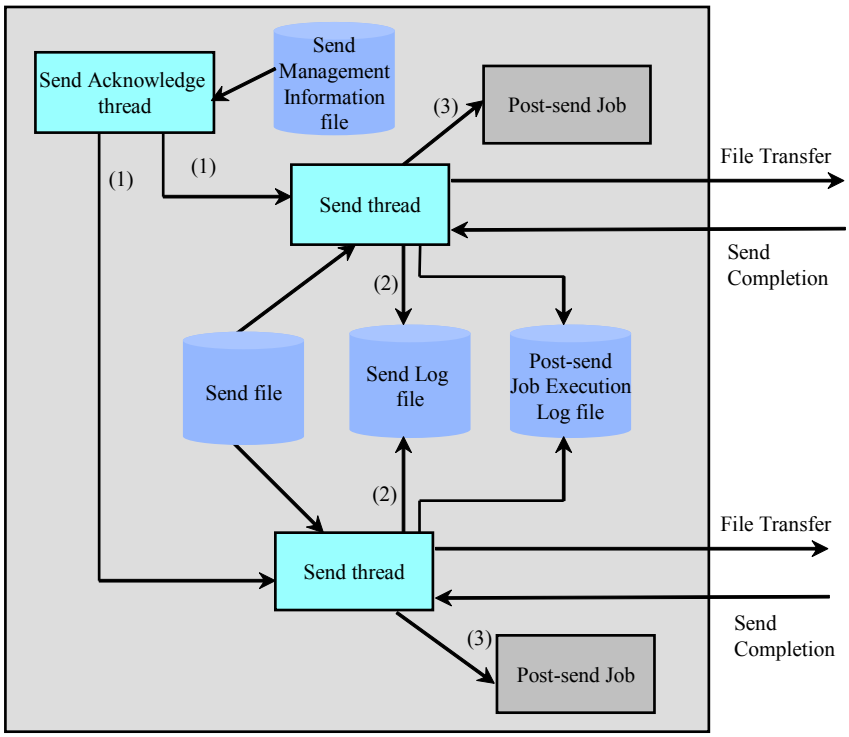


Figure 2.13 Multicasting

(1) Startup of Send thread

Send Acknowledge thread starts the Send thread to all the hosts registered in Transfer Group Information of the Send file. Each Send thread transfers the Send file.

(2) Registration of Send Log information

The Send thread writes the results of sending by hosts into the Send Log file.

(3) Startup of Post-send Job

The Send thread starts the job registered as a Post-send Job.

2.1.11 Dynamic Specification

It is necessary to set management information in HULFT before file transfer. However, the settings of the management information can be dynamically changed.

You can use one File ID throughout the operation when the name of a Send file and/or sending destination changes each time you carry out transfer, because HULFT can dynamically change the name of Send file and/or sending destination.

Moreover, it is possible to make the Receive file name identical with the Send file name by setting environment variable in the Receive Management Information.

(1) Dynamic Specification of Send Operation

'Transfer Group ID' and 'Send File Name' in the Send Management Information can be changed dynamically at the time of Send File. The contents to be changed should be specified as the parameters of the Send File command.

When issuing Send File command, you can dynamically specify the Host Names. Dynamic specification of the Host Names disables the Transfer Group ID in the Send Management Information.

To perform dynamic specification at the time of sending, the 'Dynamic Parameter Specification' in the System Environment Settings file should be enabled. Refer to "3.4 System Environment Settings" for the System Environment Settings file.

(2) Dynamic Specification of Receive Operation

The same Send file name can be used by setting 'Receive File Name' in the Receive Management Information to '$SNDFILE'. When the host types are same both at the sending side and the receiving side, it is possible to make the 'path name ($SNDPATH) ' same as the sending side.

[Note]

- **When using $SNDPATH, specify $SNDFILE at the same time. It is not possible to specify only $SNDPATH.**
- **If the Receive file exceeds 200 bytes when both $SNDPATH and $SNDFILE or $SNDFILE only is specified, an error occurs.**

2.1.12 Flow of Resend File Process

Transfer can be done again by issuing the Resend File command when transfer is interrupted after the start due to some troubles. This request is called the 'Resend File' in HULFT. The Resend File makes the Send system carry out transfer from checkpoint, where the send interrupted position is considered as the checkpoint. This transfer is called the 'Checkpoint Resend File.' It is effective when the volume of data is large, because this function does not retransmit the data which has already been transmitted. Data is transferred from the beginning when 'Checkpoint Resend File' is not specified at the time of issuance of Resend File. Use this option when it is necessary to transfer from the beginning, such as some changes are made in the Send file after the unsuccessful transfer.

These processings are based on the assumption that both Send process on the sending side and Receive process on the receiving side have already started.

The flow of Checkpoint Resend File process is explained in Figure 2.14.

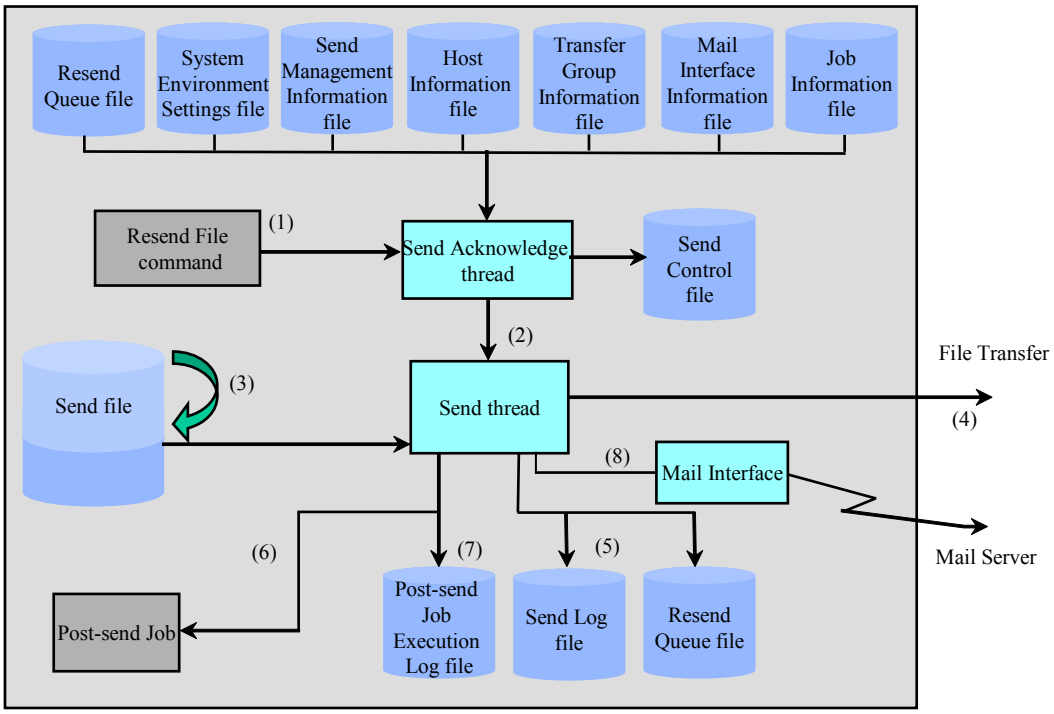


Figure 2.14 Resend File Process

(1) Issue of Resend File command

Issueing Resend File command to the Send Acknowledge thread can start Resend Process. Even when Pre-send Job has been set in Send Management Information, Pre-send Job will not be started.

(2) Creation of Send thread

The Send Acknowledge thread receives the Resend File command and creates Send thread according to the criteria from Resend Queue file and each management information file.

(3) Skip reading of Send file

In the case of Checkpoint Resend File, the Send thread determines the sent size and the sent record count from the Resend Queue file and skips the reading of data that has already been transferred.

(4) Execution of sending

Based on the settings of the transferred Send Management Information that was previously interrupted, the Send thread converts codes and compresses the file, then transfers the unsent data of the Send file to the remote host. At the time of executing Resend File command, the data is transferred from the beginning of the Send file if 'no Checkpoint Resend File' is specified.

(5) Registration of Send Log information

After completion of Send process, the Send thread writes the results in the Send Log file. In addition, when the Send Process terminates unsuccessfully, Send thread puts the information on the transfer that terminates unsuccessfully into Resend Queue file.

(6) Startup of Post-send Job

The Send thread activates the Post-send Job registered in the Job Information according to the condition registered in the Send Management Information. Regarding the job to be activated, HULFT activates either the job specified for successful termination of the resending processing or the one for unsuccessful termination of the processing, depending on the transfer result.

(7) Recording job execution log information

The Send thread writes the execution results of the job in the Post-send Job Execution Log file.

(8) Transmission of mail

The Send thread transmits e-mail according to the condition registered in the Mail Interface Information.

[Remarks] When HULFT performs Send File or Send Request, the application deletes the Resend Queue Record, of which criteria are identical to that of the Send file, from Resend Queue. Set criteria to delete in Criteria to Delete Resend Queue (resenddel) of the System Environment Settings.

2.1.13 Flow of Resend Request Process

The request from the receiving side carries out Resend process by issuing a Resend Request command from the receiving side. This is called the 'Resend Request' in HULFT. The Resend Request makes the Send system to carry out transfer from checkpoint, where the send interrupted place is considered as the checkpoint. This transferring is called 'Checkpoint Resend Request.' This function is effective when the data volume of Send files is large so that the data that has already been sent is not transferred again. Moreover, data is transferred from the beginning when 'Checkpoint Resend Request' is not specified at the time of Resend Request issuance. Use this option when it is necessary to transfer from the beginning, such as some changes are made in the Send file after the unsuccessful transfer.

These processings are based on the assumption that both Send process on the sending side and Receive process on the receiving side have already started.

The flow of Checkpoint Resend Request process is explained in Figure 2.15.

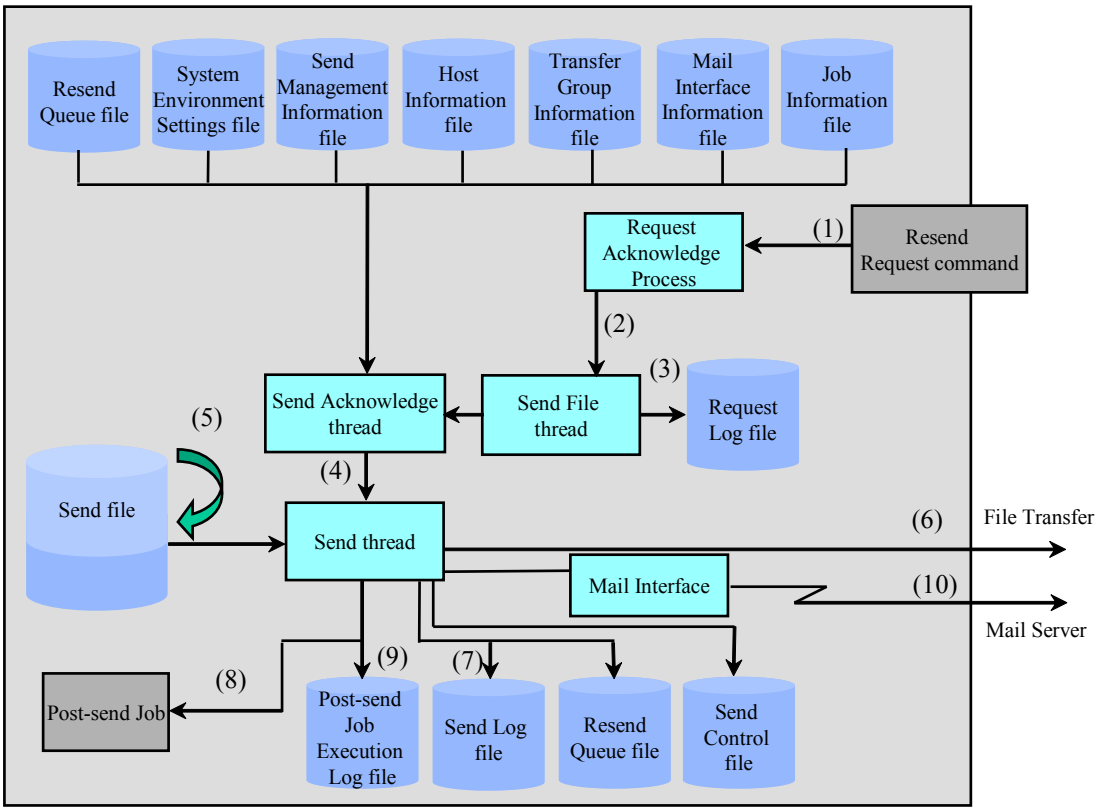


Figure 2.15 Resend Request Process

(1) Issue of Resend Request command

The Request Acknowledge process acknowledges the Resend Request from the receiving side. In the case of resend, the Pre-send Job is not started even when it is set in the Send Management Information.

(2) Issue of Resend File command

The Send File thread requests resend to the Send Acknowledge thread.

(3) Registration of Request Acknowledge Log information

The Send File thread writes the request from the receiving side to the Request Acknowledge Log file.

(4) Creation of Send thread

The Send Acknowledge thread receives the Resend File command and starts Send thread according to the criteria from Resend Queue file and each management information file.

(5) Skip reading of Send file

The Send thread determines the sent size and the sent record count from the Resend Queue file and skips the reading of data that has already been transferred.

(6) Execution of sending

Based on the settings of the transferred Send Management Information, the Send thread converts codes and compresses the file, then transfers the unsent data of the Send file to the remote host. At the time of executing Resend File command, the data is transferred from the beginning of the Send file if 'Checkpoint Resend Request' is disabled.

(7) Registration of Send Log information

After completion of Send process, the Send thread writes the result in Send Log file. In addition, when the Send Process terminates unsuccessfully, Send thread puts the information on the transfer that terminates unsuccessfully into Resend Queue file.

(8) Startup of Post-send Job

The Send thread activates the Post-send Job registered in the Job Information according to the condition registered in the Send Management Information. The job started can be either the one specified for normal end or the one specified for unsuccessful termination of the Send process, depending on the transfer result.

(9) Registration of job execution log information

The Send thread writes the execution result of job in the Post-send Job Execution Log file.

(10) Transmission of mail

The Send thread transmits email messages according to the conditions registered in the Mail Interface Information.

[Remarks] When HULFT performs Send File or Send Request, the application deletes the Resend Queue Record, of which criteria are identical to that of the Send file, from Resend Queue. Set criteria to delete in Criteria to Delete Resend Queue (resenddel) of the System Environment Settings.

2.1.14 Auto Resend

Among errors during file transfer, in the event of errors related to network or errors at the time of forced termination of HULFT for UNIX/Linux on the receiving side due to failover, HULFT can execute resending automatically. At the time of an unsuccessful transfer, it is possible to transfer the data from the point where the error has occurred (Checkpoint Resend File) by carrying out resending.

(1) Flow of Auto Resend

The flow of auto resend process is explained in Figure 2.16.

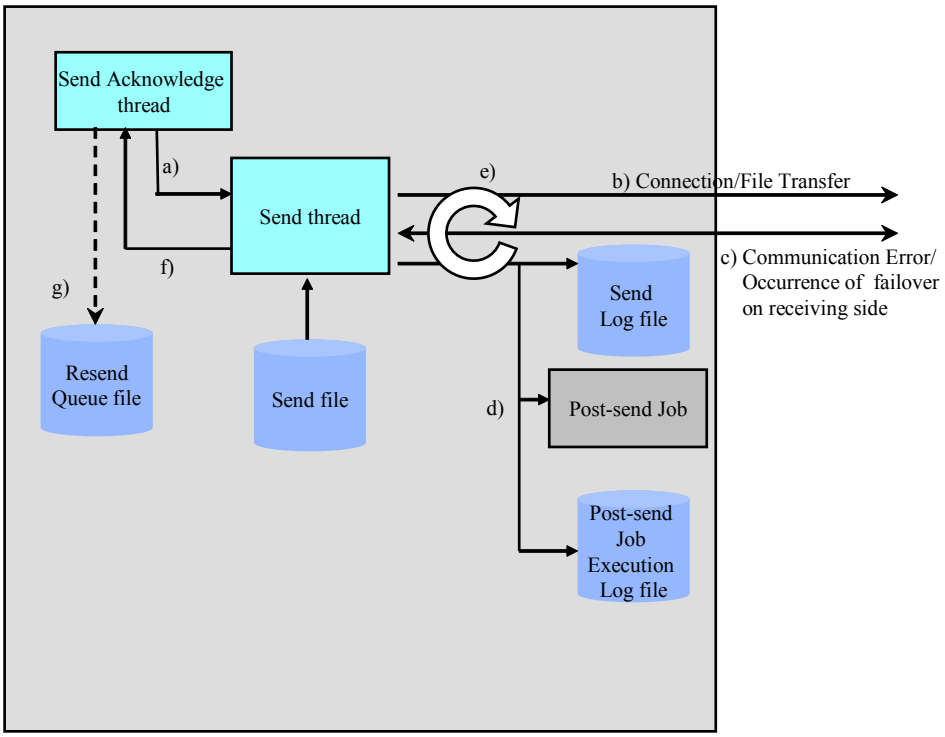


Figure 2.16 Auto Resend

a) Startup of Send thread

The Send Acknowledge thread starts the Send thread.

b) Connection to Receive Acknowledge thread

The Send thread connects to the Receive Acknowledge thread, and transfers files. If the Send thread cannot connect to the Receive Acknowledge thread, it repeats the connecting process the number of times specified in Connection Retry Count.

c) Communication error/occurrence of failover on receiving side

When an error occurred in communication line during file transfer or a forced termination of HULFT for UNIX/Linux on the receiving side due to failover occurred, the Send thread detects the error and discontinues processes.

d) Registration of log information, startup of Post-send Job

The Send thread writes the result of communication error in the Send Log file and starts a job registered as the unsuccessful job.

e) Auto resend

If a failure occurs in c) , the Send File waits until the specified wait time expires and then repeats the procedure from b) to d) specified number of times.

The above 'specified wait time' indicates 'waiting state for Auto Resend' and you can configure it in the Connect Retry Interval in the System Environment Settings.

Regarding the 'specified number of times' stated above, you can configure it in the Auto Resend Retry Count in the System Environment Settings.

f) Notification of send result

The Send thread notifies the send result to the Send Acknowledge thread.

g) Registration of Resend Queue file

The Send Acknowledge thread writes the send result received from the Send thread in the Resend Queue file when the result is unsuccessful.

[Remarks]

- Executing the Resend File command can resend the Resend Queue written in the Resend Queue file manually.
- If a communication error occurs in c) , the receiving side executes termination of the process to terminate. Reconnection from the process on the sending side by Auto Resend re-generates Receive Acknowledge thread.

(2) Auto Resend Error Code

The target error codes which brings about Auto Resend on HULFT for Windows are the Status codes related to systems. The shown below is a list of these codes:

a) When a communication error occurs

731	Failed in Send process.
732	Failed in Receive process.
735	Timeout occurred in data Receive process.

b) When the host on receiving side is HULFT for UNIX/Linux

591	Fail over occurred.

[Remarks] For error codes, refer to the *Error Codes and Messages* of the remote host.

2.1.15 Send Multiplex Level

In the sending system, you can specify the number of times that HULFT executes sending at a time. One method to specify is the Send Process Multiplex Level, which sets the multiplex level of sending in the entire sending system. The other method is to specify the Send Process Multiplex Level by Host, which configures the multiplex level of sending by the unit of host. The details are explained in this section.

(1) Send Multiplex Level of Entire System

You can set the multiplex level of sending in the entire system by specifying the Send Process Multiplex Level in the System Environment Settings.

When HULFT issues a number of the Send File commands more than the specified value in the Send Process Multiplex Level, the application places the excess on the Resend Queue and waits until the ongoing process ends and fresh transfer becomes available.

(2) Send Multiplex Level by Host

You can specify the multiplex level of sending by the unit of host, by configuring the Send Process Multiplex Level by Host in the Host Information. If HULFT acknowledges a number of the Send File commands more than the specified value in the Send Process Multiplex Level by Host, the application places the excess on the Resend Queue and waits until the ongoing process ends and fresh transfer becomes available.

You can change the setting of the Send Process Multiplex Level by Host on the Management screen of the Host Information or with a relevant command. HULFT applies the modified setting value of the Send Process Multiplex Level by Host starting from the Send File commands acknowledged after the change.

When HULFT issues the requests more than the value specified in the Send Process Multiplex Level, the application treats the requests as waiting state, even if the number of request does not exceed the value specified in the Send Process Multiplex Level by Host. Make the specified value of the Send Process Multiplex Level by Host not greater than that of the Send Process Multiplex Level.

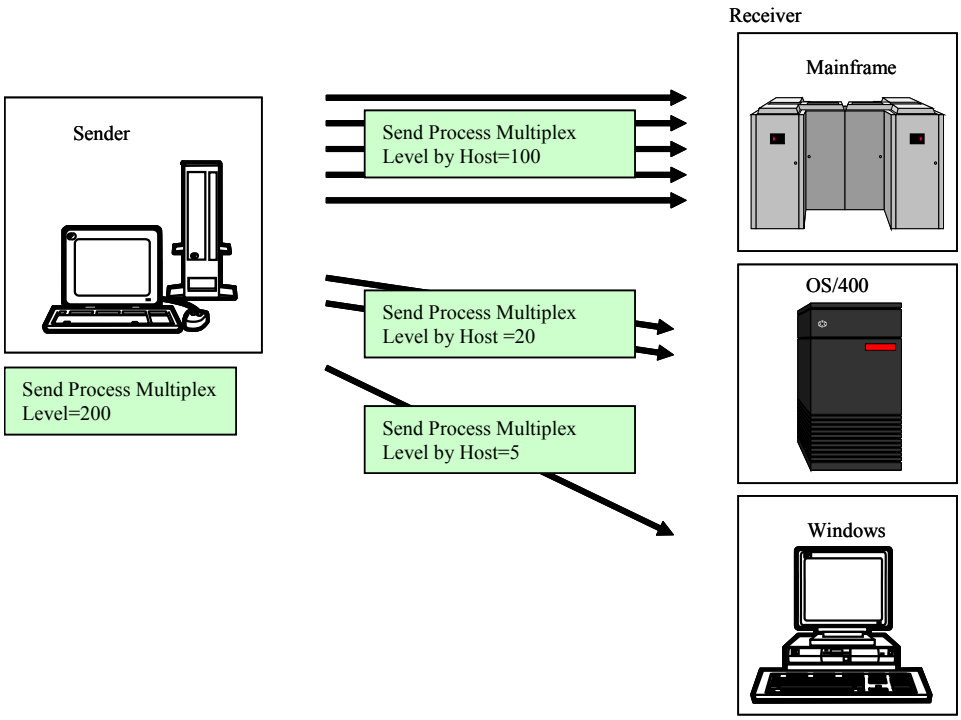


Figure 2.17 Send Process Multiplex Level and Send Process Multiplex Level by Host

2.1.16 Receive Multiplex Level

In the receiving system, you can specify the number of times that HULFT executes receiving at a time. The multiplex level of receiving can be set for the entire receiving system. In the receiving system, everytime the system receives a file, it starts one thread.

In the receiving system. if HULFT acknowledges a number of connections from sending side more than the specified value in the Receive Process Multiplex Level, the application treats it as an error in the Receive Process Multiplex Level and cuts off the connection with sending side.

[Remarks] In the sending system, if the error in the Receive Multiplex Level is returned, setting the Receive Multiplex Level Over Retry in the System Environment Settings file enables the system to reconnect remote host automatically. For the details of the System EnvironmentSettings file, refer to "3.4 System Environment Settings."

2.1.17 Receive Process from Multiple Hosts

Data of the same type can be received from multiple hosts, and the data from multiple places can be summarized and processed after receiving. This is called the 'Multiple Receive' in HULFT.

In addition, you can extract only the required data and process it since the Multiple Receive Information file manages information about the data from which host is stored in which part of the Receive file.

The flow of Receive process from multiple hosts is explained in Figure 2.18.

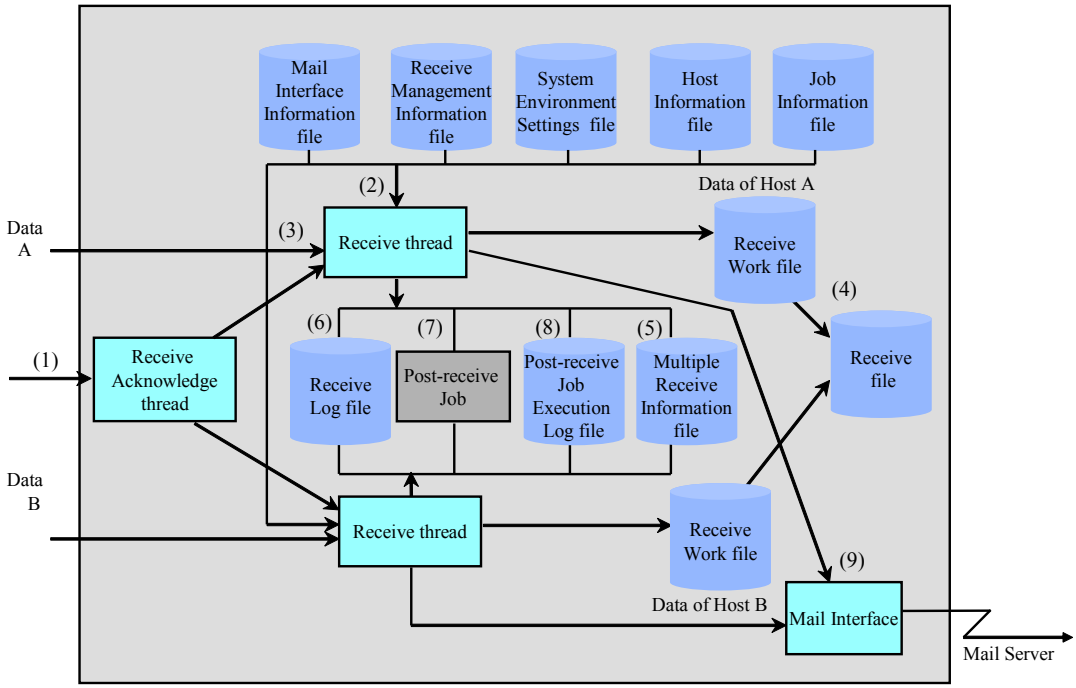


Figure 2.18 Receive Process from Multiple Hosts

(1) Waiting of connection from sending side

The Receive Acknowledge thread waits for the connection from the sending side.

(2) Creation of Receive thread

The Receive thread is created if the Receive Acknowledge thread is connected from the sending side. Receive thread will be created for each connection from the remote host.

(3) Execution of receiving

When registered as 'Multiple Receive' in the Receive Management Information, the Receive thread receives the data in Receive work file as per the Receive Management Information set in the receiving side and the file related to transfer received from the sending side. At the time of receiving from multiple hosts, Receive work files are created separately and transmission progresses parallelly.

(4) Accumulation of receive data in the Receive file

The Receive thread writes the data of receive work file to the Receive file. If an error occurs during the receiving to receive work file, HULFT does not write the received data up to the point to the Receive file.

(5) Registration of Multiple Receive Information file

The Receive thread registers information such as from which remote host the data received and where it is saved in the Receive File in the Multiple Receive Information file. This information will not be written when the Receive process terminates unsuccessfully.

(6) Registration of Receive Log information

After Receive Process ends, the Receive Acknowledge thread writes the result of the process onto the Receive Log file.

(7) Startup of Post-receive Job

According to the conditions registered in the Receive Management Information, the Receive thread starts subsequent jobs registered in the Job Information after transfer. The job can be started either upon successful or unsuccessful termination.

The job is started for each receive completion from the remote host.

(8) Registration of Job Execution Log information

The Receive thread writes the execution result of the job in the Post-receive Job Execution Log file.

(9) Transmission of mail

The Receive thread sends email messages according to the conditions registered in the Mail Interface Information. Email is sent on completion of each receive from the respective remote host.

2.1.18 Job Execution

(1) Pre-send Job

The job can be started at the point when the Send File or the Send Request is acknowledged.

When the started job ends normally (when Exit Code = 0), the Send process starts. Send process cannot be done at the time of unsuccessful termination. For example, you can start a user job to create sending data at the time the Send File program or the Send Request program is acknowledged, and utilize the data for sending process at the time the sending data has been created.

Set the Job ID which is registered in the Job Information as the Pre-send Job ID of Send Management Information in order to use this Pre-send Job function.

(2) Post-send Job and Post-receive Job

The job can be started at the point when the Send process or the Receive process ends. A different job can be started depending on whether the Send processing or the Receive processing terminates successfully or unsuccessfully.

Set the Job ID which is registered in the Job Information to the Successful Job ID of the Send Management Information or the Receive Management Information to start the job when the Send processing or the Receive processing terminates successfully.

Set the Job ID which is registered in the Job Information to the Unsuccessful Job ID of the Send Management Information or the Receive Management Information to start the job when the Send processing or the Receive processing terminates unsuccessfully.

Execution result of the Post-send or Post-receive Job is written in the Job Execution Log file after sending or receiving.

(3) Conversion to Environment Variables

HULFT adds environment variables to the startup job when a Pre-send Job, Post-send Job or Post-receive Job starts. The list of environment variables to be added is as follows.

Table 2.1 List of Environment Variables

Variable Name	Conversion Rules	Pre-send	Post-send	Post-receive
$HOST	Convert to the Send/Receive Host Name		✓	✓
$FILEID	Convert to the File ID	✓	✓	✓
$COUNT	Convert to the Send/Receive Record Count		✓	✓
$FILENM	Convert to the Send/Receive File Name	✓	✓	✓
$STATUS	Send/Receive Status code (Format: 9999) For Status code, refer to *Error Codes and Messages*		✓	✓
$DETAIL	Send/Receive Detail code (Format: 9999) For Detail code, refer to *Error Codes and Messages*		✓	✓
$SDATE	Convert to Send/Receive Start Date (format: YYMMDD)		✓	✓
$EDATE	Convert to Send/Receive End Date (format: YYMMDD)		✓	✓
$STIME	Convert to Send/Receive Start Time (format: HHMMSS)		✓	✓
$ETIME	Convert to Send/Receive End Time (format: HHMMSS)		✓	✓
$SDATE2	Convert to Send/Receive Start Date (format: YYYYMMDD)		✓	✓
$EDATE2	Convert to Send and Receive End Date (format: YYYYMMDD)		✓	✓
$MSG0	Convert to Message 0	✓	✓	✓
$MSG1	Convert to Message 1	✓	✓	✓
$MSG2	Convert to Message 2	✓	✓	✓
$MSG3	Convert to Message 3	✓	✓	✓
$MSG4	Convert to Message 4	✓	✓	✓
$MSG5	Convert to Message 5	✓	✓	✓

[Note] When the Notification in the Receive Management Information is set to the 'Successful Job Completion,' HULFT converts $EDATE, $EDATE2, and $ETIME into the date of transfer completion, not into the date of job completion.

Also, when the startup job of Job Information is specified and the above-mentioned environment variable is specified as the parameter of the startup application, the value will automatically be replaced by the environment variable and the application will start.

<Conversion example>

Table 2.2 Conversion Example

Field to be converted	Values to Convert
Send Host	HOST0001
File ID	FILEID01
Send Start Date	20030321
Send Start Time	155400
Send End Date	20030321
Send End Time	160232
Send Record Count	12000
Send Filename	d:\hulft\rcv\fileid01.dat
Send Status Code	0
Send Detailed Code	0

<Example of using environment variables in a batch file>

```
@echo off
echo %$HOST% %$FILEID% %$SDATE2% %$STIME% %$EDATE2% %$ETIME%
echo count=%$COUNT% status=%$STATUS%
if not %$STATUS% == 0 goto end
echo %$FILENM%
del %$FILENM%
:end
```

<Execution result>

```
HOST0001 FILEID01 20030321 155400 20030321 160232
count=12000 status=0
d:\hulft\rcv\fileid01.dat
```

<Example of using in Job Information>

```
notepad.exe $FILENM
```

[Actual startup command]

```
notepad.exe d:\hulft\rcv\fileid01.dat
```

(4) Points to be Noted for Job Execution

When the job cannot be executed, check the following points.

a) Permissions

The job may not start because of the types of permissions. The execution permission of each job is as follows:

Pre-send Job: User permission that issued Send File command (utlsend)
Post-send Job: User permission that activated the Send system
Post-receive Job: User permission that activated the Receive system

When Account Name in System Environment Settings is specified, make sure that Account Name is correct.

b) Setting of Job Information

Make sure that the setting of startup job is correct.

c) Existence of Job Information

Check whether the command described in the startup job exists or not.

d) Command

Execute the command registered in Job Information in the command line and check whether the command works or not.

[Note] Because $STARTTRANSFERID and $NEWTRANSFERID are the environment variables that are specified as reserved words, you cannot use these terms in a job.

2.1.19 Mail Interface

(1) Post-send and Post-receive Mail Interface

Email messages can be transmitted when a Send process or a Receive process completes.

The subject line (Title), addresses (Address and CC) and body message (Text) of an email can be specified. The Send file, Receive file, and any file (file size up to 20KB) can be attached to the email. When a file is attached, the contents of 'Text' registered in Mail Interface Information is also transmitted as an attached file.

When the mail interface function is used, set 'Mail Interface ID' of the Send Management Information or the Receive Management Information as the Mail Interface ID that is registered in the Mail Interface Information.

[Note] When you execute Mail Interface processing, set the environment of Mail Interface in advance.

For the details, refer to "3.4 System Environment Settings."

(2) Conversion to Environment Variables

HULFT adds environment variables to the email when When mail interface process starts after send or receive. The list of environment variables to be added is as follows.

Table 2.3 List of Environment Variables

Variable Name	Conversion Rule	Post-send	Post receive
$HOST	Convert to Send/Receive Host Name	✓	✓
$FILEID	Convert to File ID	✓	✓
$COUNT	Convert to Send /Receive Record Count	✓	✓
$FILENM	Convert to Send/Receive File Name	✓	✓
$STATUS	Send/Receive Status code(Format: 9999) Refers to *Error Codes and Messages* for Status code	✓	✓
$DETAIL	Send/Receive Detail code (Format : 9999) Refers to *Error Codes and Messages* for Detail code	✓	✓
$SDATE	Convert to Send/Receive Start Date (Format: YYMMDD)	✓	✓
$EDATE	Convert to Send/Receive End Date (Format: YYMMDD)	✓	✓
$STIME	Convert to Send/Receive Start Time (Format: HHMMSS)	✓	✓
$ETIME	Convert to Send/Receive End Time (Format: HHMMSS)	✓	✓
$SDATE2	Convert to Send/Receive Start Date (Format: YYYYMMDD)	✓	✓
$EDATE2	Convert to Send/Receive End Date(Format: YYYYMMDD)	✓	✓
$MSG0	Convert to Message 0	✓	✓
$MSG1	Convert to Message 1	✓	✓
$MSG2	Convert to Message 2	✓	✓
$MSG3	Convert to Message 3	✓	✓
$MSG4	Convert to Message 4	✓	✓
$MSG5	Convert to Message 5	✓	✓

<Conversion example>

Table 2.4 Example of Conversion

Conversion Items	Conversion value
Transfer destination host name	HOST0001
File ID	FILEID01
Status code	0

<Example of using in the Text>

The transfer result to $HOST of File ID ($FILEID) is $STATUS.

<Text after conversion>

The transfer result to HOST0001 of File ID (FILEID01) is 0.

[Note]

- **E-mails are not sent when the sending cancellation on the sending side or the receiving cancellation on the receiving side is carried out.**
- **Emails are not sent when receive process multiplex level error is generated at the receiving side.**
- **Use a mail software that supports MIME to read emails sent by HULFT.**

2.1.20 Receive Completion Notification

In the Receive process, you can select a timing to notify the completion of receiving to the sending side out of two options—'Receive Completion' or 'Successful Job Completion.' The option selection is registered in Receive Management Information. When the Receive process terminates unsuccessfully, completion is notified at the time of unsuccessful termination, irrespective of the selection.

(1) Receive Completion

When 'Receive Completion' is selected as the Notification, the receive completion is notified to the sending side immediately after all data is received and before execution of Post-send Job. The sending side is notified irrespective of the status of the subsequent job when the subsequent job is specified.

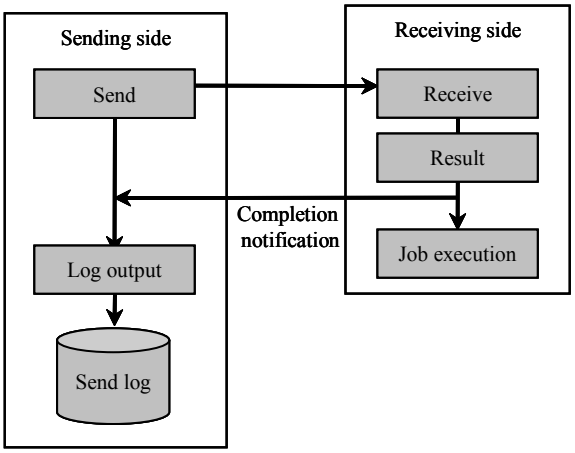


Figure 2.19 Notification Timing (Receive Completion)

(2) Successful Job Completion

When 'Successful Job Completion' is selected as the Notification, receive completion is not notified to the sending side until the end of relevant job, if a Successful Job ID is specified in Receive Management Information. The sending side is notified of the unsuccessful termination when the job terminates unsuccessfully. The receive completion is notified to the sending side at that time when job operating time exceeds 'Job Timeout' of the system environment settings.

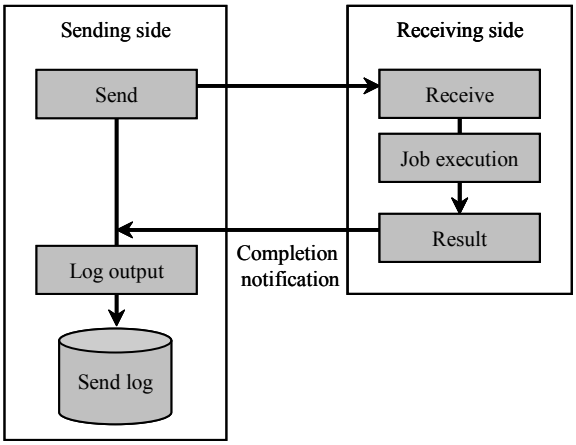


Figure 2.20 Notification Timing (Successful Job Completion)

2.1.21 Receive in CSV Format

When the file is received on UNIX, Linux, Windows and Nonstop, HULFT can convert the receiving file into CSV format. This is called 'CSV format receive' in HULFT.

CSV format receive can be done at the time of format transfer or multi format transfer. To do this, specify 'CSV' in the Interface DBID of Send Management Information.

In the CSV format file, extension (.csv) is applied to the file name registered in the Receive Management Information and this is created as a separate Receive file.

The specification of CSV format on the receiving side and information such as delimiter, enclosed character and whether space is deleted or not can be set for each File ID in the CSV Environment Settings file.

The flow of CSV format receive is explained in Figure 2.21.

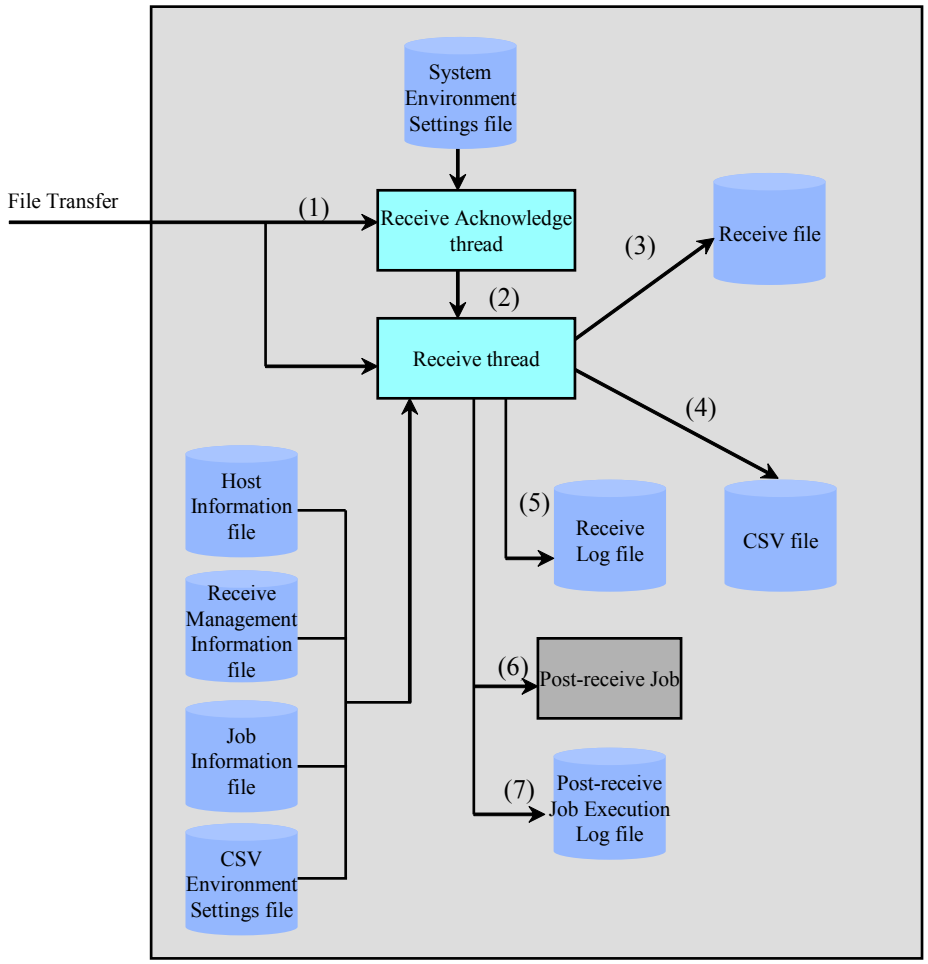


Figure 2.21 CSV Format Receive

(1) Waiting of connection from sending side

The Receive Acknowledge thread waits for the connection from the sending side.

(2) Creation of Receive thread

The Receive Acknowledge thread creates the Receive thread when connected from the sending side.

(3) Execution of receiving

The Receive thread receives data according to the transfer-related file information received from the sending side and the contents of the Receive Management Information at the receiving side.

(4) CSV format file creation

The Receive thread creates the CSV file from the Receive file.

(5) Registration of Receive Log information

After conversion process to CSV format, the Receive thread writes the result of the Receive operation in the Receive Log file.

(6) Startup of Post-receive Job

After the transfer, the Receive thread activates the subsequent job registered in the Job Information according to the conditions registered in the Receive Management Information. The job can be started either when the process ends normally or when it terminates unsuccessfully. The job is started for each receive completion from the remote host.

(7) Registeration of job execution log information

The Receive thread writes the execution result of the job in the Post-receive Job Execution Log file.

2.1.22 Receive in XML Format

When the file is received on UNIX, Linux, and Windows, HULFT can receive data using the XML format. This is called 'XML format receive' in HULFT.

XML format receive can be done at the time of format transfer or multi format transfer. To do this, specify 'XML' in the Interface DBID of the Send Management Information.

In the XML format file, extension (.xml) is applied to the file name registered in the Receive Management Information and this is created as separate Receive file.

To receive in XML format, you should use HULFT Manager to connect to the sending side and the receiving to configure XML Environment Settings. For HULFT Manager, refer to "2.2.5 Manager Connection."

The flow of XML format receive is explained in Figure 2.22.

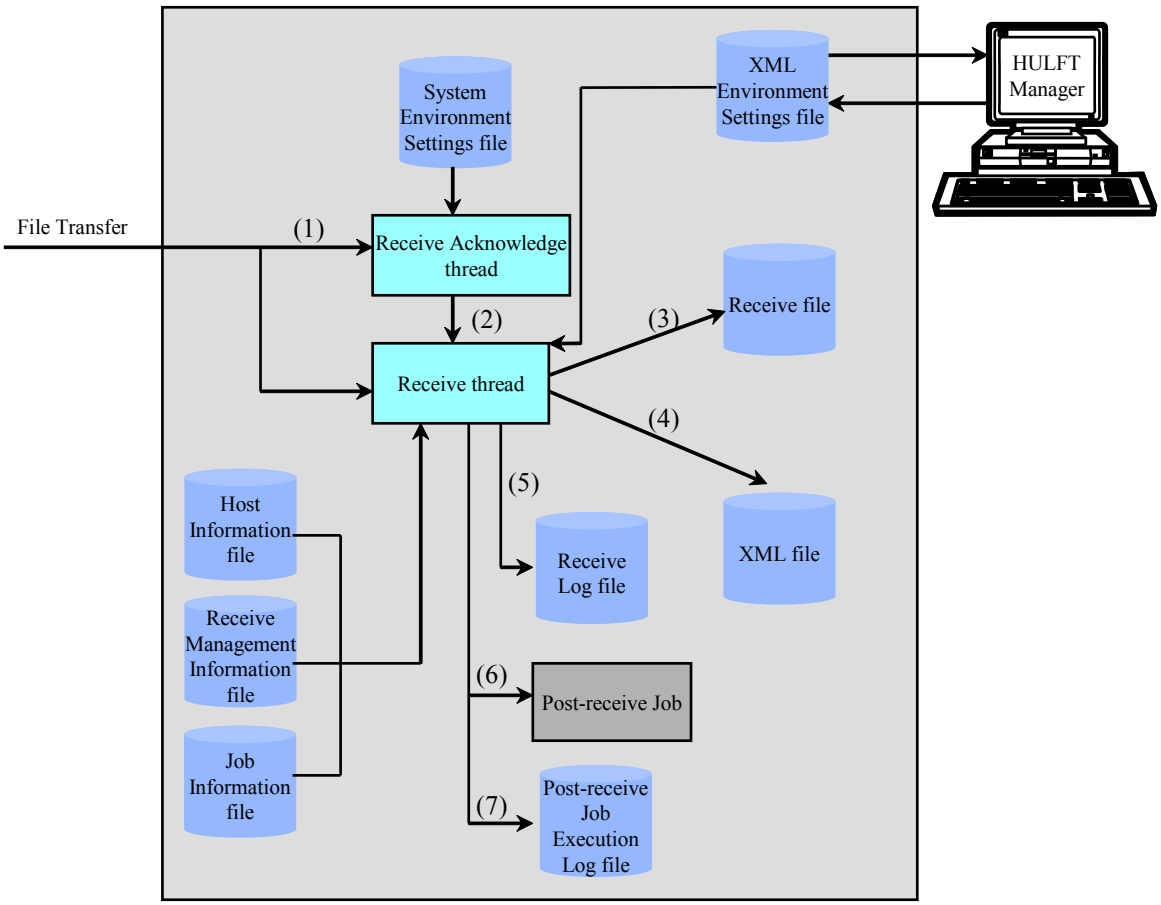


Figure 2.22 XML Format Receive

(1) Waiting of connection from sending side

The Receive Acknowledge thread waits for the connection from the sending side.

(2) Creation of Receive thread

The Receive Acknowledge thread creates the Receive thread when connected from the sending side.

(3) Execution of receiving

The Receive thread receives data according to the transfer-related file information received from the sending side and the contents of the Receive Management Information at the receiving side.

(4) XML format file creation

The Receive thread creates the XML file from the Receive file based on information in XML Environment Settings file.

(5) Registration of Receive Log information

After conversion process to XML format, the Receive thread writes the result in the Receive Log file.

(6) Startup of Post-receive Job

After the transfer, the Receive thread activates the subsequent job registered in the Job Information according to the conditions registered in the Receive Management Information.The job can be started either when the process ends normally or when it terminates unsuccessfully. The job is started for each receive completion from the remote host.

(7) Registration of job execution log information

The receive thread writes the execution result of the job in the Post-receive Job Execution Log file.

2.1.23 Generation File

HULFT can manage generations of Receive files. This is useful when the same file is received and saved many times at irregular intervals in one day.

While managing Receive files in generation files, set 'Enabled' in the Generation File of the Receive Management Information. Receive thread creates a Receive file by adding an extension to the Receive file name registered in the Receive Management Information. Details of each Receive file are recorded in the Generation File Information file.

Maximum generation file count can be set in the Receive Management Information. The number circulates from 1 when the number exceeds this count. You can set up to '9999' at a maximum for the Generation File Count.

(1) Receive Files Managed in Generation Files

In the case of Receive files managed in generation files, a generation file number is added to the Receive File Name registered in Receive Management Information. The extension is allocated sequentially in the order from '.0001' until receive completion.

<Example>

- Receive File Name registered in Receive Management Information: c:\gyoumu\data\hulft.rcv
- Receive File Name that completes the receive firstly: c:\gyoumu\data\hulft.rcv.0001
- Receive File Name that completes the receive secondly: c:\gyoumu\data\hulft.rcv.0002

[Remarks]

- When the Generation File is enabled, the file name of environment variable $FILENM of Post-receive Job becomes the one that the generation file number is added to the extension.
- Even when the Generation File is enabled, the Receive file name registered in the Receive Management Information will be displayed in the message at receive start.
- When the Generation File is enabled, specify the Receive file name within 190 bytes since an extension will be added to the Receive file name.

(2) Generation File Information File

Information about each Receive file managed in generation file can be confirmed with the Generation File Information file. The name of the Generation File Information file is 'File ID.info' and it will be created in the path where the system files exist (HULPATH).

[Remarks] When the Generation File Information file is deleted, the management number starts from '0001' again.

2.1.24 Data Transfer Method (Framed Message Transfer)

HULFT Ver.7 or higher allows you to select Transfer Speed Priority Mode or the Error Detection Priority Mode. In the Transfer Speed Priority Mode, HULFT transfers data without waiting for the receiving result notification from remote host.

Using the Transfer Speed Priority Mode improves transfer speed in high-speed network environment and large data transfer. The Error Detection Priority Mode, which is the conventional transfer method of HULFT Ver.6 or lower, waits for the confirmation of block transfer from remote host and then proceeds to the transfer of next block. Therefore, the transfer speed in the Error Detection Priority Mode is slower than the Transfer Speed Priority Mode, yet the former performs error detection swiftly.

(1) Transfer Speed Priority Mode and Error Detection Priority Mode

Shown below is an illustration of the difference between the Transfer Speed Priority Mode and the Error Detection Priority Mode.

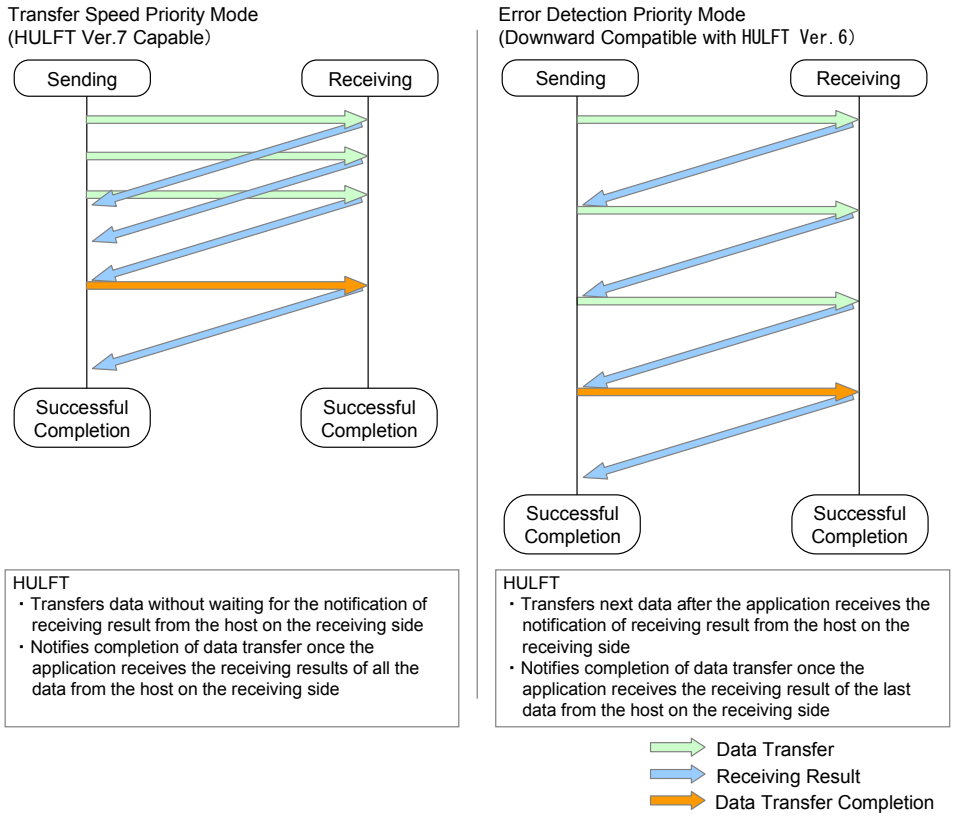


Figure 2.23 Difference between Transfer Speed Priority Mode and Error Detection Priority Mode

To set data transfer method, both sending host and receiving host should use HULFT Ver.7 or higher. In addition, HUB-Server to be routed through should be Ver.2.2 or higher, if these hosts use HULFT-HUB. If you use HULFT of lower version (namely, HULFT Ver.6 or lower), data transfer is performed in the Error Detection Priority Mode.

Further, using the Transfer Speed Priority Mode requires both hosts on the sending side and the receiving side to set the Transfer Speed Priority Mode. Should either of the host has specified the Error Detection Priority Mode, HULFT transfers data in the Error Detection Priority Mode.

(2) Setting of Framed Message Transfer Type

Data transfer method can be set by selecting the Framed Message Transfer Type in the System Environment Settings. There are four options in the Framed Message Transfer Type:

N (Error Detection Priority Mode)

This option transfers data in the Error Detection Priority Mode, at the time both of sending and receiving

S (Send Speed Priority Mode)

This option transfers data in the Transfer Speed Priority Mode at the time of sending, yet it transfers data in the Error Detection Priority Mode at the time of receiving.

R (Receive Speed Priority Mode)

This option transfers data in the Transfer Speed Priority Mode at the time of receiving, yet it transfers data in the Error Detection Priority Mode at the time of sending.

A (Send and Receive Speed Priority Mode)

This option transfers data in the Transfer Speed Priority Mode at the time both of sending and receiving.

Combination of the Framed Message Transfer types of the sending side and the receiving side determines which data transfer method to be performed. See the following table:

Table 2.5 Combination of Framed Message Transfer Types of Sending and Receiving

Sending Side \ Receiving Side	N	S	R	A
N				
S			✓	✓
R				
A			✓	✓

✓: Operation in the Transfer Speed Priority Mode
Blank: Operation in the Error Detection Priority Mode

Framed Message Transfer Type should be set based upon user's circumstances of sending and receiving. Shown below is an illustration of setting examples of the Framed Message Type based upon user's circumstances:

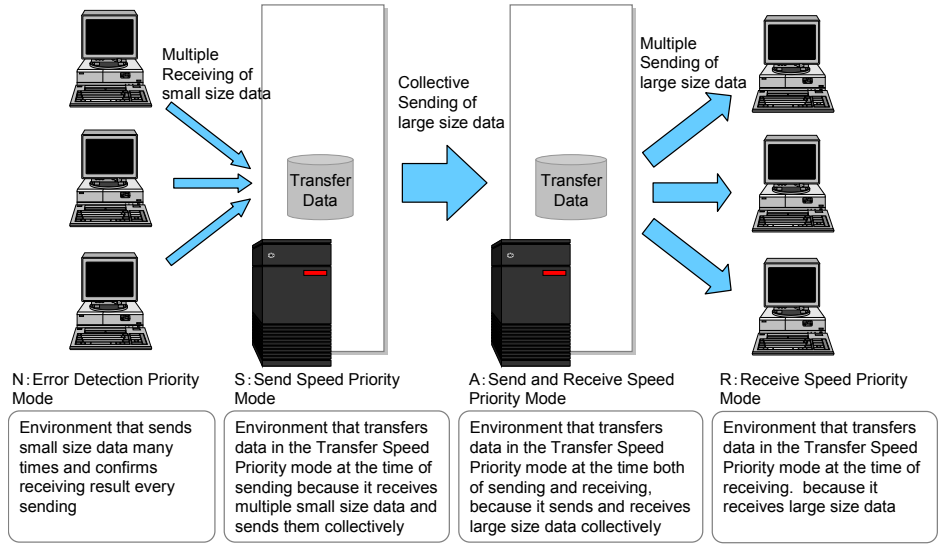
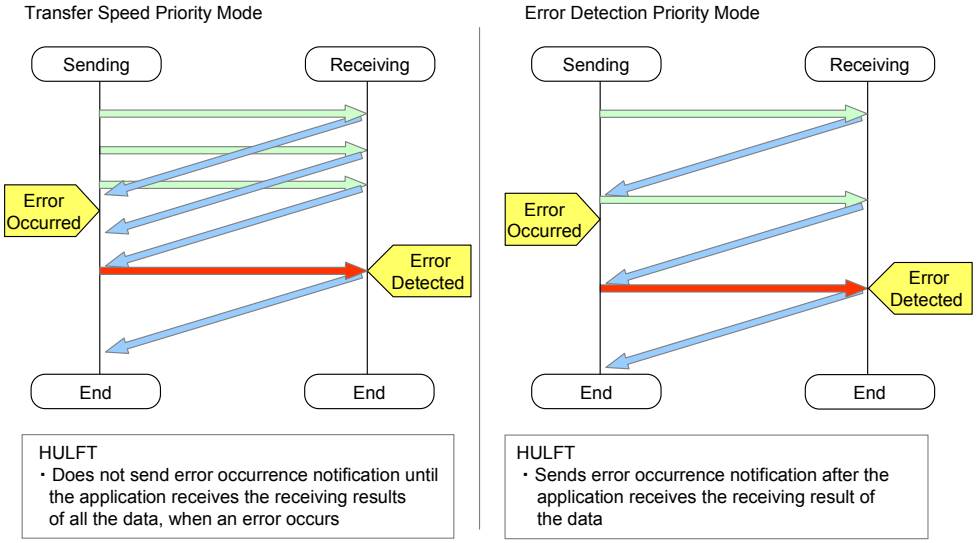


Figure 2.24 Setting Examples of Framed Message Transfer Types

(3) Timing of Error Detection

The timing of error detection differs depending on the specified data transfer method. This section explains the difference in the timing of error detection between the Error Detection Priority Mode and the Transfer Speed Priority Mode.

Where an error occurs on the sending side



Where an error occurs on the receiving side

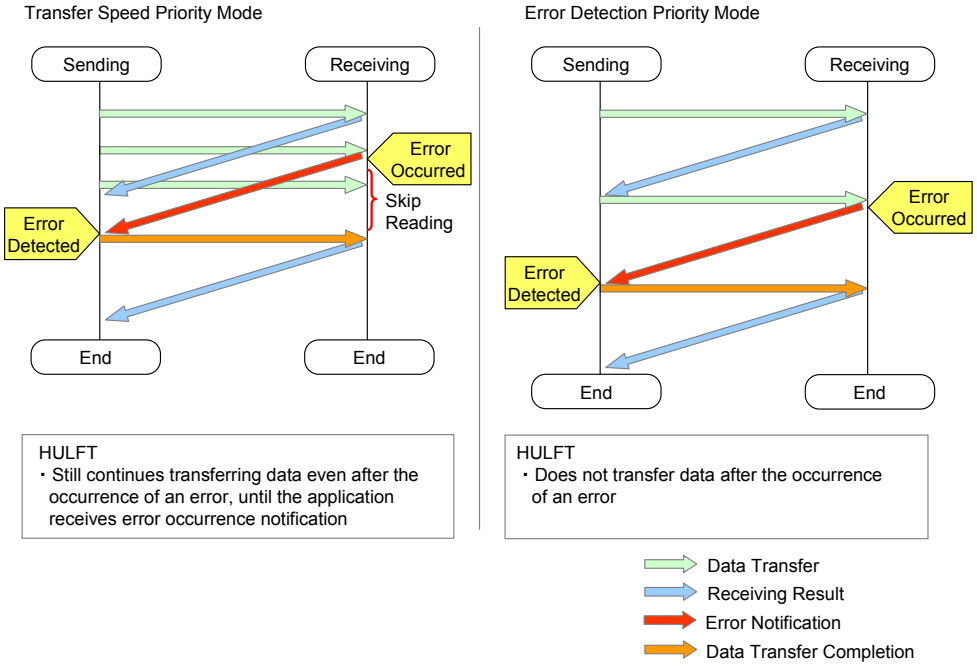


Figure 2.25 Error Detection Timing in Transfer Speed Priority Mode and Error Detection Priority Mode

In the Transfer Speed Priority Mode, utilization of resources varies more widely than the Error Detection Priority Mode, which may cause delay in the error detection timing of transfer and the acknowledgement timing of cancellation.

(4) Points to be Noted

If there is no improvement in transfer speed despite the specification of the Transfer Speed Priority Mode, following factors may be considered.

Solve your problem by consulting with following information:

- **Partner host specifies the Error Detection Priority Mode**

  Set the Framed Message Transfer Type to any one of the Transfer Speed Priority Mode options both for sending and receiving hosts.

- **Partner host uses the product version that does not support the Transfer Speed Priority Mode**

  Upgrade HULFT to the product version that supports the Transfer Speed Priority Mode both for sending and receiving hosts.

- **Insufficient network band**

  The Transfer Speed Priority Mode requires more network traffic than the Error Detection Priority Mode. Insufficient network band causes mild effect of the Transfer Speed Priority Mode.

- **Small Socket Buffer Size in sending and/or receiving hosts**

  Modify the setting of the Socket Buffer Size in the System Environment Settings on HULFT.

  Usually the larger the Socket Buffer Size is, the faster the transfer speed becomes, yet optimum value varies depending on network environment.

- **The Transfer Block Size on the sending side is not the optimum value of network**

  Modify the setting of the Transfer Block Size in the Send Management Information on HULFT.

  Usually, the greater Transfer Block Size is, the faster the transfer speed becomes, yet optimum value varies depending on network environment.

2.2 Request Acknowledge System

The Request Acknowledge system is a system that accepts a request of the remote host and executes the processing based on the requested service.

In the Request Acknowledge system, the Send Request process, the Resend Request process, the Job Execution Result Notification, the Remote Job Execution, the Manager Connection, and the Post-receive Job Result Inquiry are executed through the Request Acknowledge process.

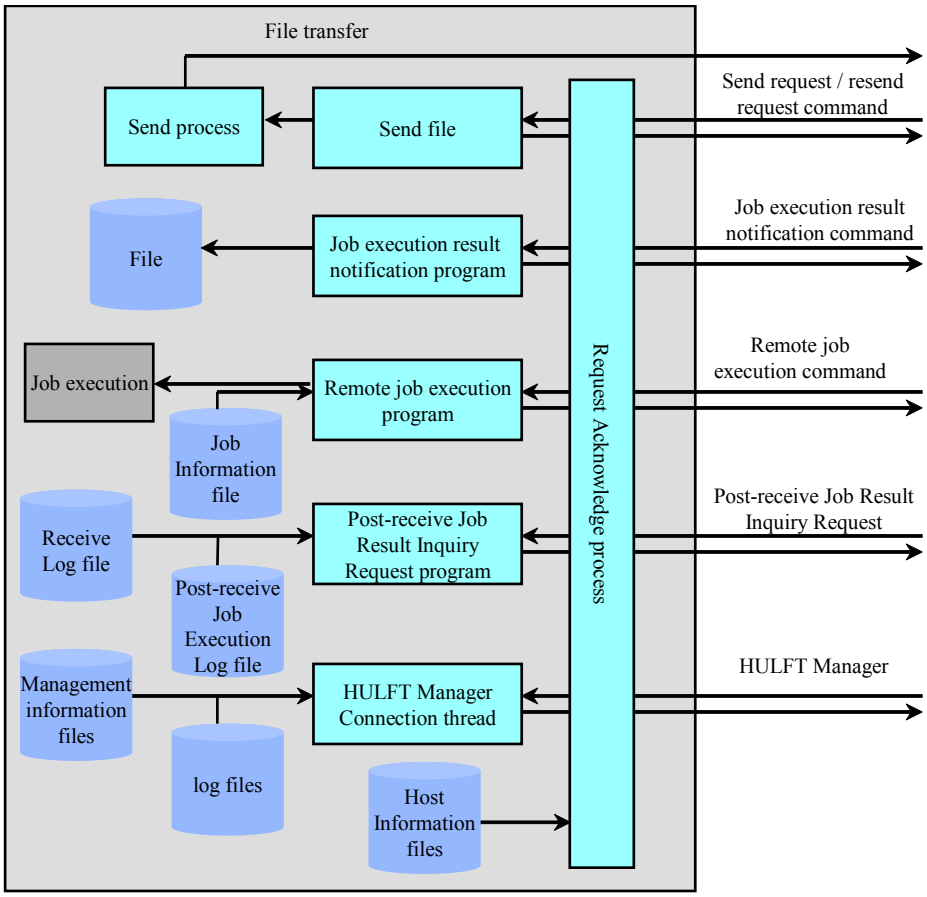


Figure 2.26 Outline of Request Acknowledge System

2.2.1 Send Request Process

If a Send Request is received from the host that carries out the Receive process, the Request Acknowledge program generates a Send File thread. The generated Send File thread called an API (utlsendex) and issues Send File to the Send program.

Similarly, the Resend File is issued when a Resend Request is acknowledged. Refer to "2.1.2 Flow of Send Process Started at Receiving Side."

2.2.2 Job Execution Result Notification

Job result can be notified after send and receive by the remote host that started the job. It is necessary to include the job execution result notification command within a job of the remote host in order to use this function. The notified job result will be output to any file.

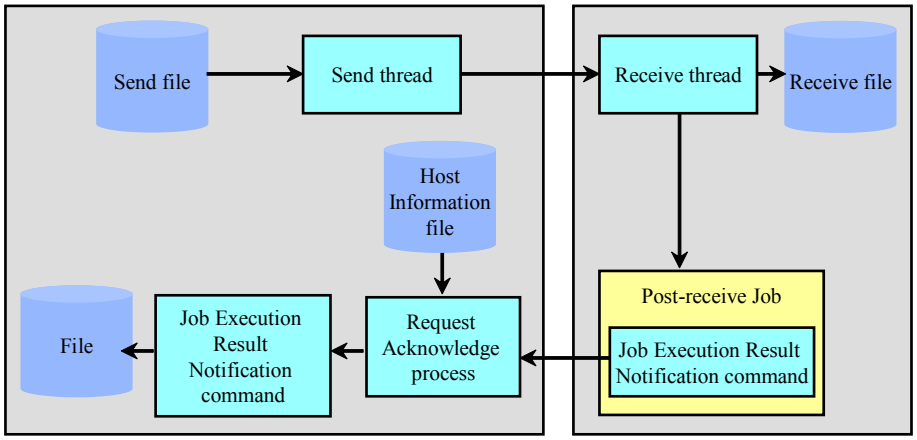


Figure 2.27 Job Execution Result Notification Function

2.2.3 Remote Job Execution

Execution of the job of local host can be carried out from other host. When the remote job execution request is acknowledged, the Request Acknowledge program generates a process thread. The process thread executes the job according to the content registered in the Job Information based on the Job ID specified by the remote job execution command. It is necessary to register the executed job in the Job Information beforehand.

The result of the executed job can be notified to the host, which conducted the execution request. The timeout period until notification can also be specified.

When multiple jobs are registered in the Job ID and if there are any jobs with status code other than 0, the subsequent job will not be executed.

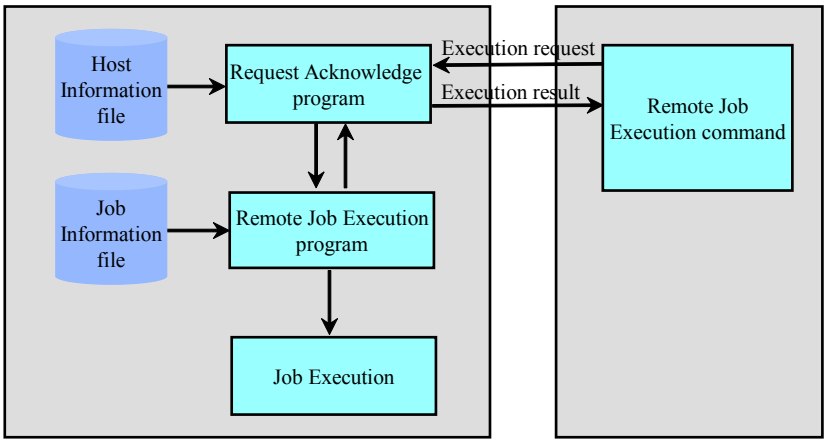


Figure 2.28 Remote Job Execution Function

2.2.4 Post-receive Job Result Inquiry

You can view the Job Execution Log of the receiving side from the Send Detail Information Inquiry screen of HULFT Management screen at the sending side.

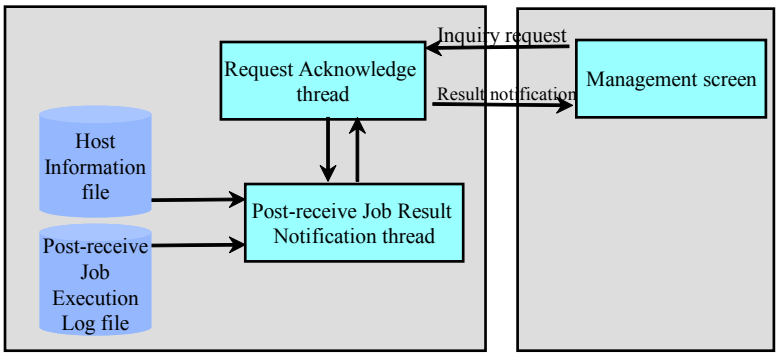


Figure 2.29 Post-receive Job Result Inquiry Function

2.2.5 Manager Connection

This program carries out the connection process from HULFT Manager. When update and deletion of management information, send and receive request, confirmation and deletion of Send and Receive Log are carried out from a remote personal computer by using HULFT Manager, requests are executed by the manager connection program. For details, refer to the manual of HULFT Manager.

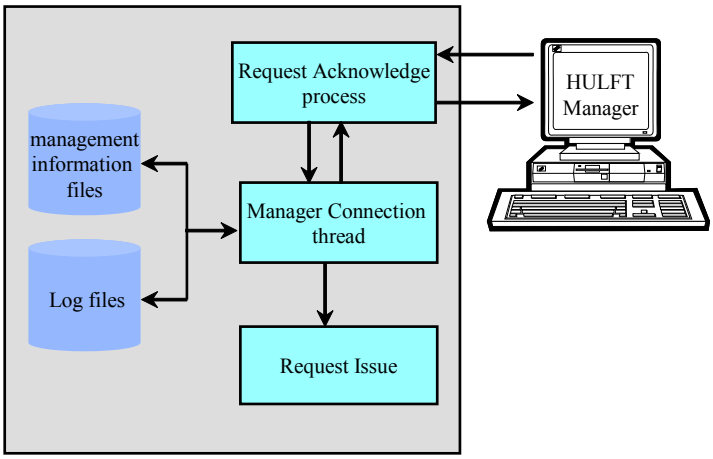


Figure 2.30 Manager Connection Function

2.3 Output of Log

In HULFT, when a process such as file transfer is executed, the result will be output to the log file.

Log records are accumulated in the file. If you continue to use the system without deleting the log records for a long time, the disk capacity becomes full. Therefore, it is necessary either to delete them manually or automatically, or to operate by log file switch.

Processes of which log record will be output are as follows.

(1) Send Process (one output for each send)

(2) Receive Process (one output for each receive)

(3) Post-send and Receive Job Execution (one output for each job execution)

(4) Request Acknowledge Process (one output for each acknowledge)

2.3.1 Log Deletion by Manual Operation

The log file can be deleted by the respective log deletion command.

The deletion of log can be regularly executed by incorporating this command into the scheduler and the like.

Refer to *Operation Manual* for details of each log deletion command.

2.3.2 Operation by Switching Log File

When a certain number of records are accumulated in each log file, the log file can be switched. As a result, it is possible to operate log files so that they do not exceed a certain size.

Each process checks the number of log records before outputting logs, renames the log file to a log switch file at the timing of log switching, and then creates a fresh log file.

By using the Management screen and utilities, you can refer to the logs without ever noticing the existence of such log files and log switch files.

This switch timing is set in the Log Switch Count of the System Environment Settings file. Refer to "3.4 System Environment Settings" for the System environment Settings file.

The flow of log switch process is explained in Figure 2.31.

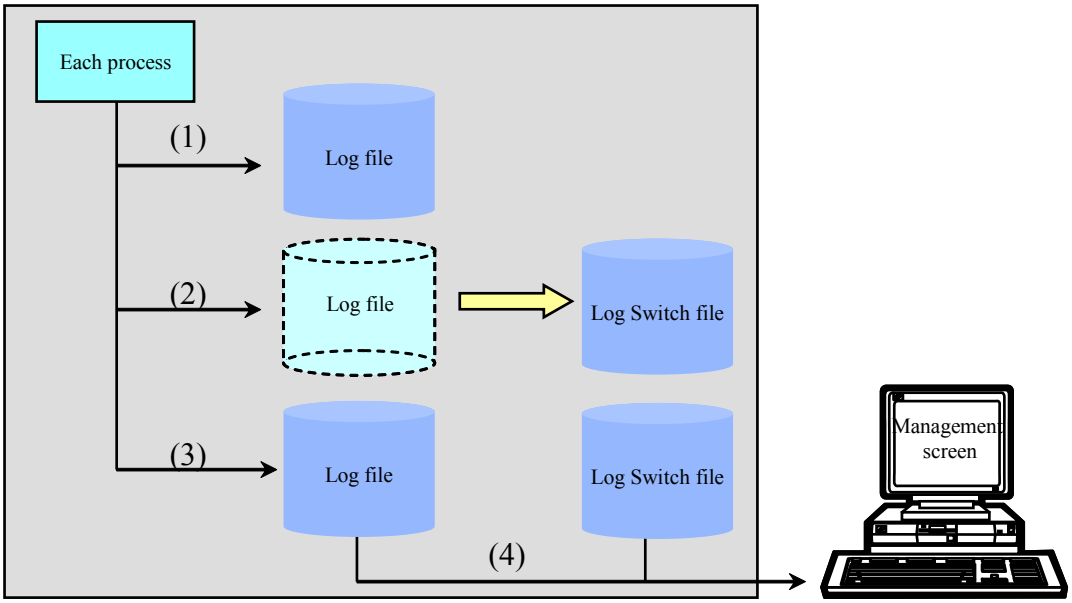


Figure 2.31 Flow of Log Switch

(1) Count check

Before the log output, each process checks the number of log files. When the number has not reached the maximum, output to the log file and process is ended.

(2) Switch

In each process, when the number of log files reaches the switch timing, log file is renamed to the log switch file.

In such cases, when the log switch file exists already, renaming is done after deleting that file.

(3) Log output

Each process newly creates individual log files and outputs log records. A process keeps outputting log records into its log file until the application reaches the specified timing to switch the log file.

(4) Log confirmation in Management screen

The content of the log can be confirmed on the Management screen and log file and the content of log switch file can be referred.

2.4 Management Screen

In HULFT, it is possible to register each management information file for send and receive and to view the Send and Receive Log files that stores the status of send and receive on the screen.

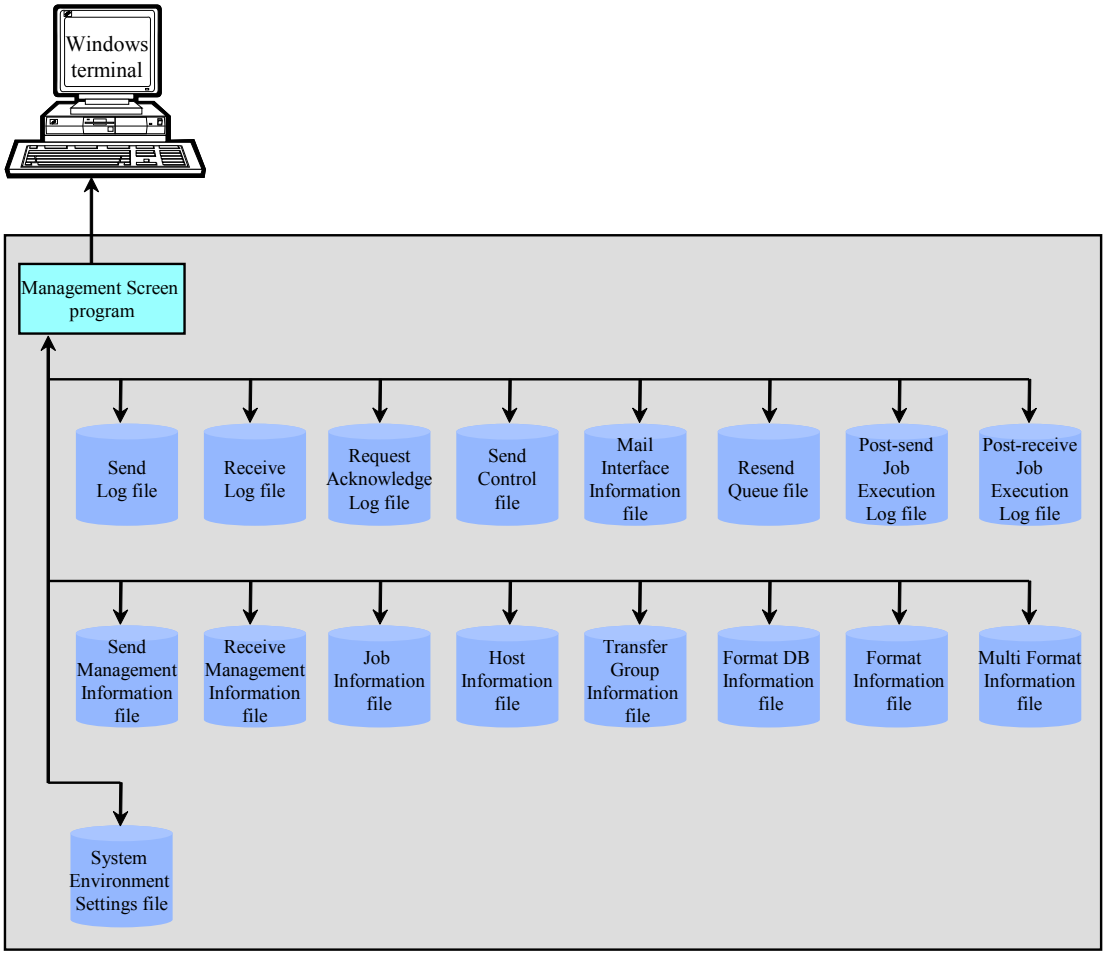


Figure 2.32 Outline of the Screen

2.5 Code Conversion

2.5.1 Data Transfer

HULFT enables to transfer single byte code, double byte code, numerical data as well as data where they are mixed, and to convert code into the data format which is used by each host.

The following data transfer types are provided in HULFT. It is possible to select a transfer type that is suitable for the data type to send.

Table 2.6 Data Transfer Type

Transfer type	Transfer data type
Text Transfer	The text data of one record of the file composed of the alphanumeric character and kanji that is delimited by line feed is transferred while carrying out code conversion. 1 byte code conversion and 2 byte code conversion is possible.
Binary Transfer	Transfer the data that has no concept of record without code conversion.
Format Trasnsfer	The data format of one record is of fixed length, and by defining that format, the file for which code conversion needs to be done for each field is transferred. 1 byte code conversion, 2 byte code conversion, numeric data conversion is possible.
Multi Format Transfer	The record that has multiple format within 1 file is transferred while converting to a format that matches with each record by defining the format and record key. 1 byte code conversion, 2 byte code conversion, numeric data conversion is possible.

2.5.2 Code Conversion

HULFT converts send and receive data to the code system of the machine which carries out Send and Receive process at the time of data transfer process. This conversion process is called 'code conversion'.

The code conversion of HULFT includes following three types of code conversions.

(1) Single Byte Code Conversion

When transferring with EBCDIC-type host, conversion between ASCII code and EBCDIC code is carried out.

Users can select EBCDIC codes from seven code types—'Kana Characters,' 'Lowercase,' 'ASCII,' 'ASPEN,' 'IBM Standard,' 'IBM Standard Extension,' and 'NEC Kana.' Also, three types of user-defined tables can be created optionally.

Refer to "2.5.4 Single Byte Code" for details on single byte code conversion.

[Remarks] 14 types of code tables corresponding to the seven types of code types are provided as templates, based on which EBCDIC user tables can be created.

(2) Double Byte Code Conversion

HULFT converts the character of primary level and the secondary level of JIS by standard double byte codes. The code conversion can be done by treating the rest as External character code and by using external character table.

Refer to "2.5.5 Double Byte Code" for details of double byte code conversion.

[Remarks] The external character table is provided as a template for NEC special characters. It is possible to create external character table based on it.

(3) Numeric Data Conversion

In HULFT, conversion of numeric data of signed internal decimal data (pack data), signed external decimal data (zone data), floating-point data, unsigned external decimal data, binary data can be carried out. Numeric data conversion is carried out at the time of format transfer and multi format transfer.

Refer to "2.5.8 Code Conversion Rules for Format Transfer" for details of numeric data conversion.

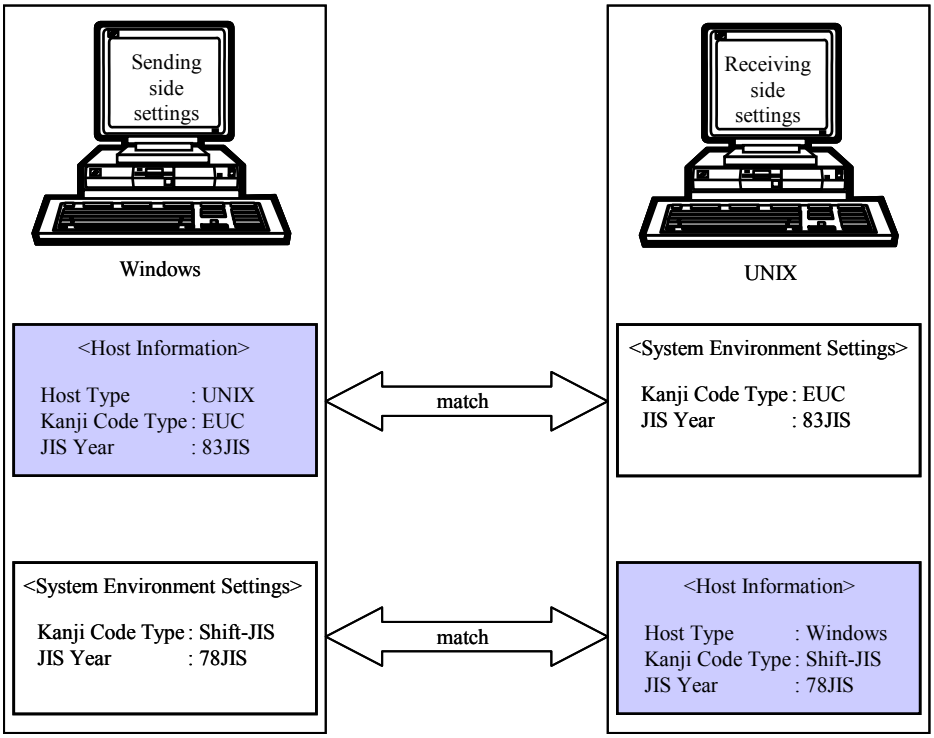
2.5.3 Code Conversion Destination

At the time of text transfer, format transfer and multi format transfer, it is possible to select whether to carry out code conversion process at the sending side or receiving side. Conversion at the sending side is called 'sending side conversion' and conversion at the receiving side is called 'receiving side conversion'. Moreover, it is possible to select 'no conversion' if conversion is not carried out.

In code conversion, to carry out normal conversion at the Sending Side Conversion and Receiving Side Conversion, set accurately the information in the System Environment Settings and Host Information. Example of code conversion setting contents is shown in the figure below.

<Setting example>

Windows
(fixed) Kanji Code Type : Shift-JIS
System Environment Settings JIS Year : 78JIS
UNIX
System Environment Settings Kanji Code Type : EUC
JIS Year : 83JIS



*1: Set the Host Type of the Host Information as the remote host type. For the Host Information, refer to *Operation Manual*.

*2: The Kanji Code Type of the System Environment Settings is set only in UNIX /Linux. It is fixed for other host types.

*3: Specification of the JIS Year is only available in HULFT for i5OS and HULFT for Mainframe. (Except for HULFT for VOS). The setting option is fixed to '78JIS' on HULFT for ACOS, while it is fixed to '83JIS' on others.

Figure 2.33 Code Conversion Settings Contents

2.5.4 Single Byte Code

There are two types of single byte codes (ASCII code and EBCDIC) used in HULFT.

Windows, UNIX, and Linux, use the ASCII code, while Mainframe and the office computer use EBCDIC.

The EBCDIC code system consists of multiple codes, and different EBCDIC codes may be used depending on host types and/or users.

In HULFT, conversion between different single byte codes are carried out during file transfer and the code can be selected depending on the environment of the remote host. The EBCDIC code can be set in the Send Management Information and the Receive Management Information. Refer to *Operation Manual* for details on Send Management Information and Receive Management Information.

HULFT supports one ASCII code and the following EBCDIC codes.

Table 2.7 EBCDIC Code Types

EBCDIC set	Remarks
A: Kana	EBCDIC that uses katakana
B: Lowercase	EBCDIC that uses lower case alphabet
C: ASCII	EBCDIC used on Mainframe (MSP, XSP)
D: ASPEN	EBCDIC used on Mainframe(VOS3)
E: IBM Standard	EBCDIC of lower case alphabet used on Mainframe(z/OS) and i5/OS
F: IBM Standard Extension	EBCDIC which includes katakana in addition to lower case alphabet used on Mainframe(z/OS) and i5/OS
G: NEC Kana	EBCDIC used on Mainframe(ACOS)

[Note] **Half width Kana characters of EUC code and some characters of UTF-8 are actually expressed in multiple bytes. However HULFT treats them as single byte codes. When code conversion of the characters of which size is more than one byte is carried out, the size of codes is increased or decreased.**

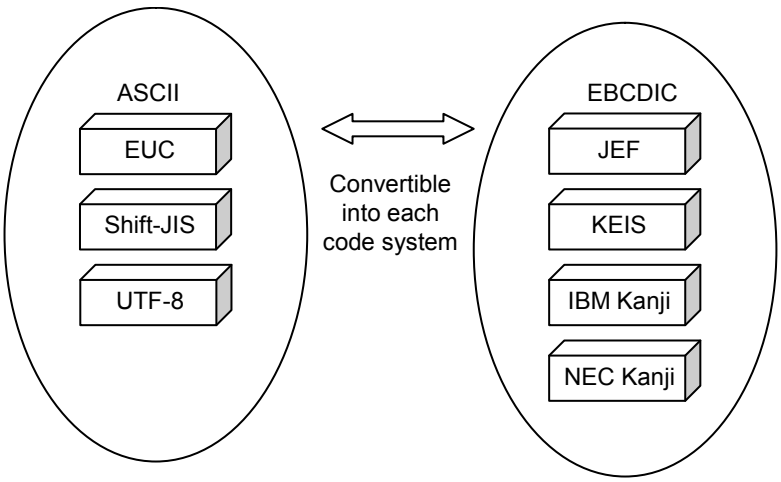


Figure 2.34 Single Byte Code Conversion

(1) Conversion between EBCDIC Code and ASCII Code

Code conversions can be carried out between ASCII code and seven types of EBCDIC codes shown in the EBCDIC code type in Table 2.6. The code conversion is done by selecting the code used as EBCDIC code from seven types of code.

When an EBCDIC code other than the above mentioned seven types is used, the user registers a conversion pattern with which the conversion can be carried out for the use environment (user table). The EBCDIC code template is provided, hence it is possible to update the code easily. Refer to "3.8 EBCDIC User Table Settings" for details on user table settings.

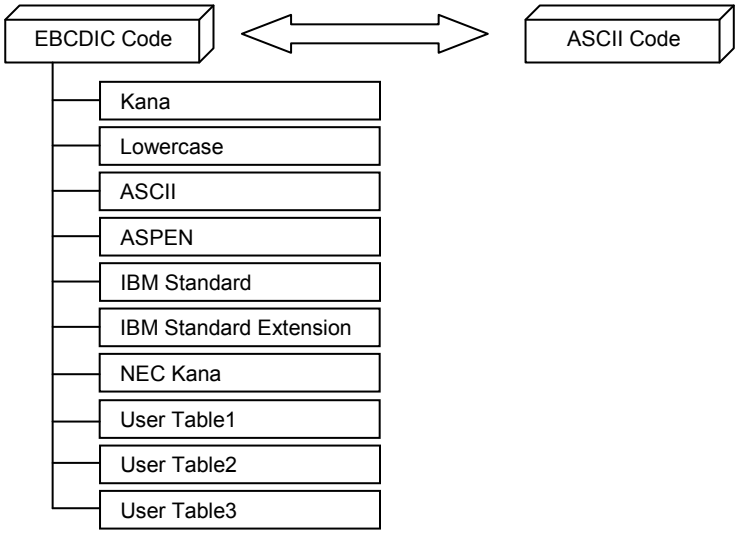


Figure 2.35 Conversion between EBCDIC Code and ASCII Code

(2) Conversion between ASCII Codes

HULFT does not carry out code conversion between ASCII codes. Therefore, the application treats them as No Conversion.

However, if the transfer falls under any of the following conversion patterns, namely, between SHFT-JIS and EUC, between EUC and UTF-8, and between SHIFT-JIS and UTF-8, HULFT converts only halfwidth Katakana characters.

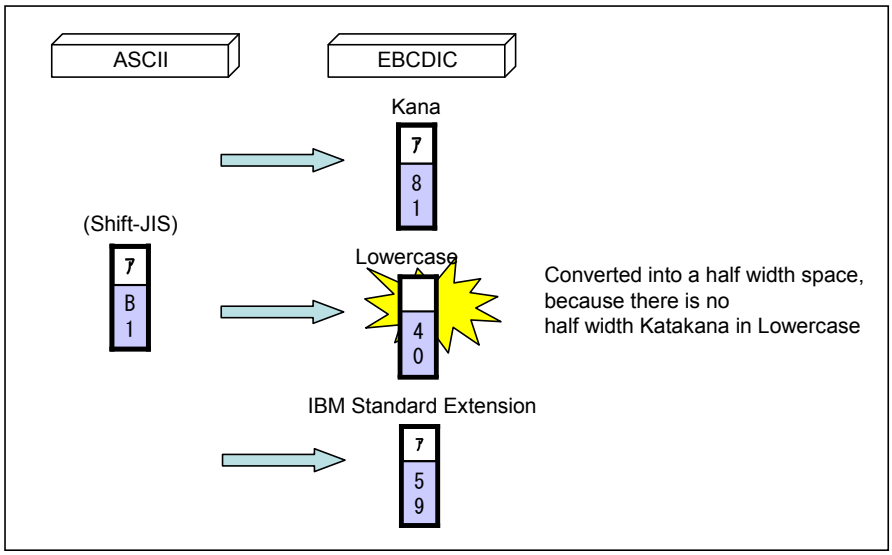
(3) Example of Single Byte Code Conversion

Single byte code conversion example is explained in this section. Refer to *Operation Manual* for conversion of each EBCDIC code.

a) Conversion of half width Katakana

When half width Katakana data is converted to the EBCDIC code type that does not have half width Katakana, it is converted into a half width space.

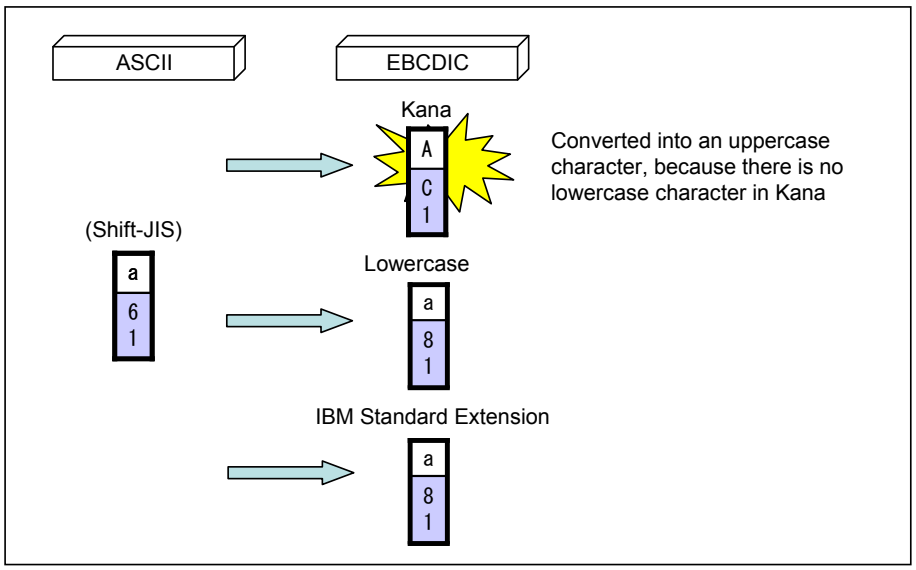
<Example> Conversion of half width Katakana



b) Conversion of lowercase characters

When a lowercase character is converted to the EBCDIC code type that does not have lowercase characters, it is converted into an uppercase characters.

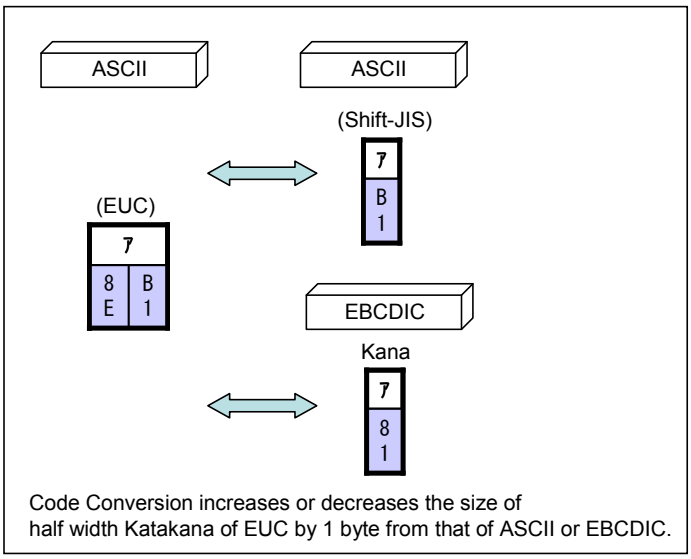
<Example> Conversion of lower case characters



c) Conversion of EUC half width Katakana

When converting to or from EUC half width Katakana characters, the data size is increased or decreased by 1 byte.

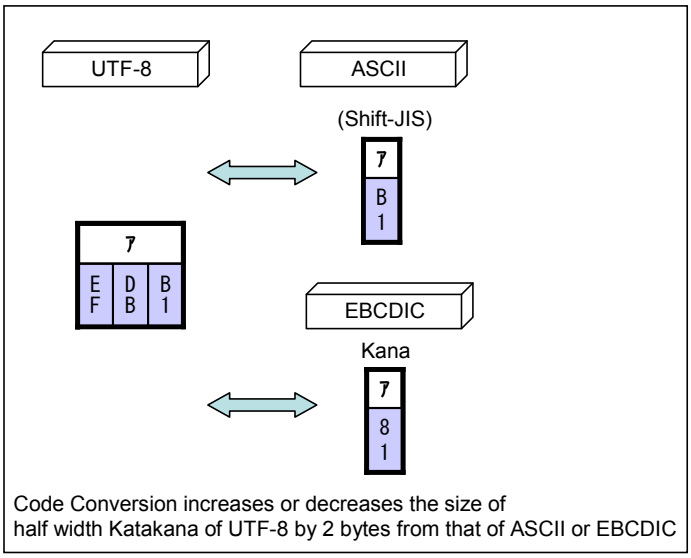
<Example> Conversion of EUC half width Katakana



d) Conversion of UTF-8

When converting to and from UTF-8 half width Katakana, the data size is increased or decreased by 2 bytes.

<Example> Conversion of UTF-8



2.5.5 Double Byte Code

There are multiple types of double byte codes used in HULFT—Shift-JIS, EUC, JEF (Fujitsu Kanji), IBM Kanji, KEIS (Hitachi Kanji), and NEC Kanji. In addition, UTF-8, which is a multbyte code type, is available as well.

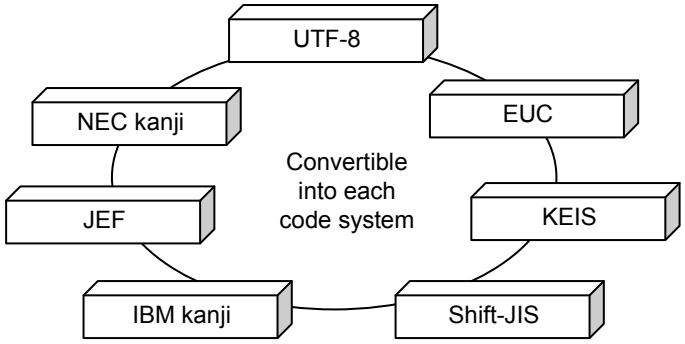


Figure 2.36 Double Byte Code Conversion

The code used in Windows is Shift-JIS. Selecting the code that the remote host is using from six kinds the code conversion is done between Shift-JIS and other five kinds of double byte codes.

As for JIS year, 83JIS contains 90JIS. When JIS year is different between Shift-JIS codes, code conversion will be carried out.

Table 2.8 Kanji Code Type

Kanji Code Type	Remarks
J: JEF	Code used on MSP, XSP, and K
I: IBM Kanji	Code used on z/OS and i5/OS
K: KEIS	Code used on VOS3
N: NEC Kanji	Code used on ACOS
S: Shift-JIS	Code used on Windows, UNIX, and Linux
E: EUC	Code used on UNIX and Linux
8: UTF-8	Code used on UNIX and Linux

(1) About External Character Code

HULFT treats characters other than level 1 and level 2 of JIS as external character code. During the external code conversion, the user registers the conversion pattern and the conversion becomes possible.

Refer to "3.9 External Character Table" for external character code settings.

Moreover, from the set value of the system environment setting 'External Character Table Mode', it is possible to select the conversion process of the external character code. It is possible to give preference to the conversion pattern of the user over the conversion pattern of the character in the first and the second levels of JIS that are provided by HULFT.

With a set value for the system environment settings Process when External character has not been Registered, settings for a non-registered conversion pattern can be configured, thus making it an error.

[Note]

- **When transfer between the same code type is carried out, neither the external character conversion nor the code conversion is executed.**
- **In the EUC code, HULFT can handle three-byte external characters.**

(2) Points to be noted regarding UTF-8

Conversion of N Type

When executing conversion from 'UTF-8' to 'N (Kanji)' code types other than 'UTF-8,' presence of single byte code in input data causes failure in code conversion.

Codes cannot be used as the first character

You cannot use the external character starting with 0x00, because UTF-8 supports only variable length.

System Environment Settings

When you specify UTF-8 as source or destination of code conversion for the field related to code conversion in the System Environment Settings, the setting may bring different behaviors of HULFT.

Code Conversion Mode (codechangemode)

0: Mode 0

In code conversion, this mode treats the codes of conversion destination as fixed length and executes conversion to external character. (except for UTF-8)

1: Mode 1

In code conversion, this mode treats the codes of conversion destination as variable length and executes conversion to external character.

When UTF-8 is source or destination code of conversion, HULFT always behaves as if the field is set to codechangemode=1.

Regarding the code type other than UTF-8, the behavior of HULFT varies in accordance with the setting, 0 or 1.

Space Character Mode (spcode)

This field specifies the conversion rule of space code.

0: Mode 0
1: Mode 1

When UTF-8 is source or destination code type of conversion, HULFT always behaves as if the field is set to spcode=1.

Regarding the code type other than UTF-8, the behavior of HULFT varies in accordance with the setting, 0 or 1.

For details, refer to "2.5.9 Rules of Space Code Conversion."

KEIS Em-size Space Mode (keisspmode)

This field specifies the conversion rules of em-size (full width) space code.

This tag is valid only when the Space Character Mode (spcode) is set to 1.

0: Convert the em-size space code into 0x4040.
1: Convert the em-size space code into 0xA1A1.

When UTF-8 is source or destination code type of conversion, HULFT always behaves as if the field is set to keisspcode=1.

Regarding the code type other than UTF-8, the behavior of HULFT varies in accordance with the setting, 0 or 1.

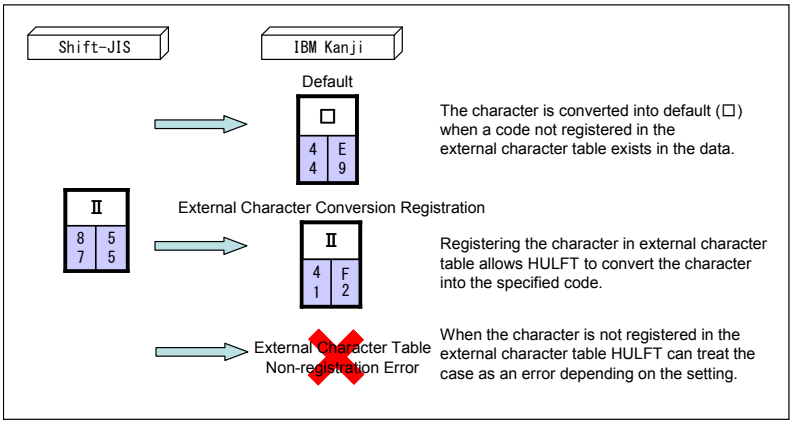
(3) Example of Double Byte Code Conversion

This section explains an example of double byte code conversion. Code conversions vary based on whether external characters are registered, how non-registered external characters and processed and the proority of external characters.

a) When non-registered external character code exists in the data

When the external character table registration code exists in the data, it is converted into a default value or a pertinent code as per the value of the External Character Table Mode. An error may occur while processing when non-registered external character and non-registered external character table code exits.

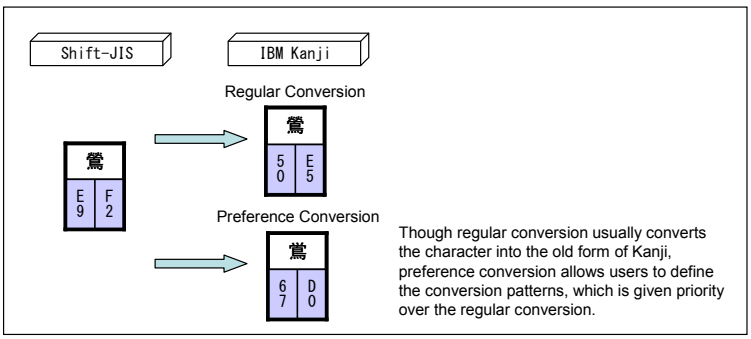
<Example> When the process differs based on the value of the Process when External character has not been Registered



b) When the JIS level 1, level 2 code is to be converted to user-defined code

When the JIS level 1, level 2 code is converted as the user defined, it is possible to convert those in the external character table with priority based on the value of the External Character Table Mode.

<Example> When the process differs based on the value of the External Character Table Mode



2.5.6 Text Transfer

Text transfer is a format of transferring text data where single byte data and double byte data coexist. It is possible to convert single byte code and double bytes code.

In the case of Windows, Unix, and Linux, the record of the text data is separated by line feed code. When you carry out text transfer from Windows to UNIX or Linux, a line feed code '0x0d0a' and '0x0a.' will become '0x0a.' Windows. When you carry out text transfer from Windows to Windows, a Line feed code '0x0d0a' and '0x0a' become '0x0d0a.' However, while transferring to Mainframe and office computers, the line feed code will be deleted.

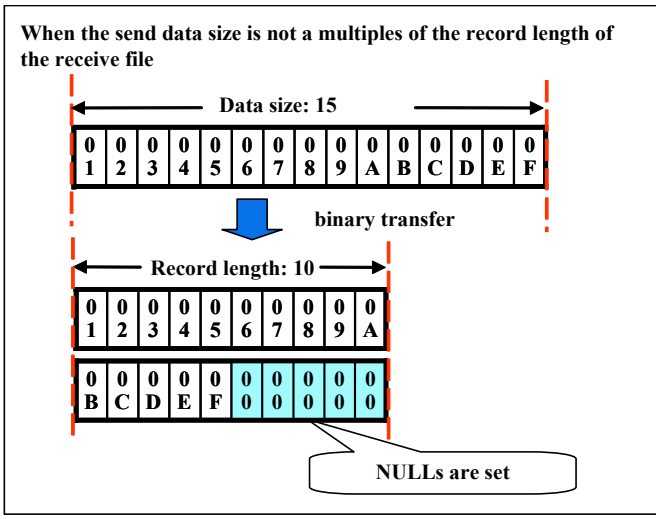
As for the tab code (0x09), when transferred from the sending side to Mainframe and office computers, conversion is carried out according to the conversion rules of EBCDIC user table and the settings of 'Tab Code Mode' of the system environment settings.

[Remarks] Even if 'No conversion' is specified on the code conversion side, line feed code will be processed as described above.

2.5.7 Binary Transfer

In binary transfer, code conversion is not carried out. Data is treated as a simple byte sequence and transferred. The size of the send data and receive data becomes the same. When the Receive file is a fixed length file, the writing is completed until the end of the record and the data is written from the top of the next record.

[Note] When the host on the receiving side is i5OS, K, or Mainframe and the file of receiving side is fixed length, the sending data size, which is not a multiple of the record length of the Receive file, sets NULL to the remaining data of the Receive file.



2.5.8 Code Conversion Rules for Format Transfer

During format transfer, code conversion is carried out for each defined field type. The field type which can be defined is as follows. There are code conversion rules for each field type.

Windows, UNIX, or Linux does not recognize data records by linefeed codes. Therefore transferring to Mainframe or office computer does not delete linefeed codes from data. Meanwhile, transferring from Mainframe or office computer does not add linefeed codes to data.

Table 2.9 Format Field Type

Field Type		Size (Byte)	Field Explanation
X	Character	From 1 to 9999	Character data format that contains only ASCII code*1
M	Mixing of Kanji and Characters	From 1 to 9999	Character data format that ASCII code and double byte code coexist
N	Kanji	From 2 to 9998	Character data format that contains only doube byte code Only even numbered bytes can be handled*2
9	Unsigned External Decimal	From 1 to 18	Decimal data format of zoned numeric without sign
B	Binary	2,4,8	Numeric data format of signed integer
P	Signed Internal Decimal	From 1to 10	Packed decimal data format with sign Only up to 18 digits can be handled
S	Signed External Decimal	From 1 to 18	Decimal data format of zoned numeric with sign
F	Floating point	4,8	Floating point data format with sign Supports the floating decimal format of IEEE
I	Image	From 1 to 9999	Data format that indicates no conversion

*1 For conversion of X type, refer to "(2) Format Conversion Rule 2 (Character Type)."

*2 When ASCII code is included in the N type specified data, it is not converted normally. Also, at the time of conversion to Mainframe and office computers, shift code is not added for N type.

(1) Format Conversion Rule 1

The conversion rules for Kanji (N), Mixing of Kanji and Characters (M), Binary type (B), Unsigned external decimal (9), Floating point (F) and Image type (I) are as follows.

Table 2.10 Format Conversion Rule 1

Before Conversion			After Conversion		Remarks
Field Type		Byte Count	Field Type	Byte Count	
N	Kanji		Same as before conversion	Same as before conversion	*1
M	Mixing of Kanji and Characters		Same as before conversion	Same as before conversion	*1
B	Binary	2,4,8	Same as before conversion	Same as before conversion	Only byte order conversion
9	Unsigned External Decimal		Same as before conversion	Same as before conversion	*2
F	Floating Point	4,8	Same as before conversion	Same as before conversion	Only byte order conversion
I	Image		Same as before conversion	Same as before conversion	No conversion

*1 Kanji code is converted based on the Kanji code that corresponds to the 'Kanji Code Type' which is registered in the Host Information.

*2 When sending to UNIX, Linux, and Windows, it is converted to ASCII code. When sending to Mainframe and office computers, it is converted to EBCDIC code. When receiving from each host, it is converted to EBCDIC code.

(2) Format Conversion Rule 2 (Character Type)

As for the character type (X), the conversion rules vary according to the settings value of the X Type Conversion of the System Environment Settings file. Refer to the "3.4 System Environment Settings" for System Environment Settings.

Table 2.11 Format Conversion Rule 2

Before Conversion			After Conversion		Remarks
Field Type		Byte Count	Field Type	Byte Count	
X	Character		Same as before conversion	Same as before conversion	

a) When the X Type Conversion is Mode 0

Code conversion is carried out in the same manner as Mixing of Kanji and Characters (M). The Kanji code is converted based on the Kanji Code Type registered in the Host Information.

b) When the X Type Conversion is Mode 1

Treated as single byte code.

When sending and receiving among UNIX, Linux, and Windows, conversion is not executed because they all use the same ASCII code.

Conversion is not carried out when sending to and receiving from Mainframe and office computers. The EBCDIC code set is converted based on the code that corresponds to the EBCDIC Code Set registered in the Send Management Information and the Receive Management Information.

[Note]

- **In the Format Conversion of X Type, setting the X Type Conversion to 'Mode1' does not convert a half width katakana of EUC successfully, because it is a double-byte character.**
- **In the Format Conversion of X Type, setting the X Type Conversion to 'Mode1' does not convert UTF-8 successfully, because the codes include multiple-byte characters.**

(3) Format Conversion Rule 3 (Signed Decimal Attribute Conversion)

Depending on the settings value of the Pack Zone Conversion in the system environment settings, the data type and byte count is changed for the signed internal decimal (P) and the signed external decimal (S).

a) When Pack Zone Conversion mode is N/S/B/F/9

Even if the code conversion side and the code conversion host type are different, the conversion rules are the same.

Table 2.12 Format Conversion Rule 3-a)

Before Conversion			After Conversion							
			Pack Zone Conversion							
			Mode N Mode S		Mode B		Mode F		Mode 9	
Field Type		Byte Count	Field Type	Byte Count	Field Type	Byte Count	Field Type	Byte Count	Field Type	Byte Count
P	Signed Internal Decimal*3	1 to 2	Same as before conversion *4	Same as before conversion	Binary	2	Floating Point	4	Unsigned External Decimal	*5
		3 to 5				4		4		
		6 to 10				8		8		
S	Signed External Decimal	1 to 4	Same as before conversion *4	Same as before conversion	Binary	2	Floating Point	4	Unsigned External Decimal	Same as before conversion
		5 to 9				4		4		
		10 to 18				8		8		

*3 The significant digit (not including sign part) is not more than 18 digits.

*4 Only the sign part is converted in the case of mode S.

*5 The significant digit of signed internal decimal (not including sign part) is from one to 18 digits. If you convert signed internal decimal into unsigned external decimal with mode 9, the byte count after conversion is the same as the digit number before conversion. The tables of the digit number before conversion and the byte count after conversion are shown below:

Table 2.13 Conversion Rules of Signed Internal Decimal with Mode 9

Before Conversion		After Conversion
Signed Internal Decimal		Unsigned External Decimal
Number of Digits	Byte Count	Byte Count
1	1	1
2	2	2
3		3
4	3	4
5		5
6	4	6
7		7
8	5	8
9		9

Before Conversion		After Conversion
Signed Internal Decimal		Unsigned External Decimal
Number of Digits	Byte Count	Byte Count
10	6	10
11		11
12	7	12
13		13
14	8	14
15		15
16	9	16
17		17
18	10	18

[Note] When all the conditions mentioned below are satisfied, an error occurs.

- **Receiving side is Ver.5 or lowrer**
- **Sending side conversion**
- **The Interface DBID is specified**
- **The Pack Zone Conversion (PSCHG) is set to 'Mode S'**
- **The Sign for ASCII (PSASCII) is set to other than '1'**

b) When Pack Zone Conversion mode is 0/1 for sending from Windows

When format transfer is carried out from Windows, at the time of send side code conversion, conversion does not take place and data type and number of bytes become the same as one before conversion.

Table 2.14 Format Conversion Rule 3-b)

Before Conversion			After Conversion	
Field Type		Byte Count	Field Type	Byte Count
P	Signed Internal Decimal	1 to 2	Same as before conversion	Same as before conversion
		3 to 5		
		6 to 10		
S	Signed External Decimal	1 to 4	Same as before conversion	Same as before conversion
		5 to 9		
		10 to 18		

c) When Pack Zone Conversion mode is 0 for receiving to Windows

The conversion rule varies depending on the side of code conversion, host type, and remote host type.

Table 2.15 Format Conversion Rule 3-c)

Before Conversion			After Conversion							
			Mainframe, K				i5OS		UNIX, Windows	
			Sending Side Conversion		Receiving Side Conversion					
Field Type		Byte Count	Field Type	Byte Count	Field Type	Byte Count	Field Type	Byte Count	Field Type	Byte Count
P	Signed Internal Decimal	1 to 2	Binary	2	Binary	2	Binary	2	Same as before conversion	Same as before conversion
		3 to 5		4		4		4		
		6 to 10		8	Floating Point	8	Floating Point	8		
S	Signed External Decimal	1 to 4	Binary	2	Binary	2	Binary	2	Same as before conversion	Same as before conversion
		5 to 9		4		4		4		
		10 to 18		8	Floating Point	8	Floating Point	8		

d) When Pack Zone Conversion mode is 1 for receiving to Windows

The conversion rule varies depending on remote host type.

Table 2.16 Format Conversion Rule 3-d)

Before Conversion			After Conversion			
			Mainframe, K, i5OS		UNIX, Windows	
Field Type		Byte Count	Field Type	Byte Count	Field Type	Byte Count
P	Signed Internal Decimal	1to 2	Binary	2	Same as before conversion	Same as before conversion
		3 to 5		4		
		6 to 10		8		
S	Signed External Decimal	1 to 4	Binary	2	Same as before conversion	Same as before conversion
		5 to 9		4		
		10 to 18		8		

(4) Format Conversion Rule 4 (Handling of Sign and Zone for Signed Internal Decimal)

a) Sign and zone which can be used in HULFT

Sign and zone parts treated as transferable data in HULFT are mentioned in the table below.

Table 2.17 Sign and Zone Which can be Used in HULFT

Host Type	Sign		Zone
	Positive	Negative	
Mainframe	C,F	D	F
UNIX/Linux	3,4	5,7	3
Windows	3,4	5,7	3
i5OS	C,F	D	F
K	C,F	D	F

[Note] **In HULFT for Windows, the following exception values for sign and zone are supported in addition to the values mentioned in Table 2.17. However, these should be used only to maintain compatibility with a lower version of HULFT.**

- **Sign Positive: A, C, E, F/Negative: B, D**
- **Zone F**

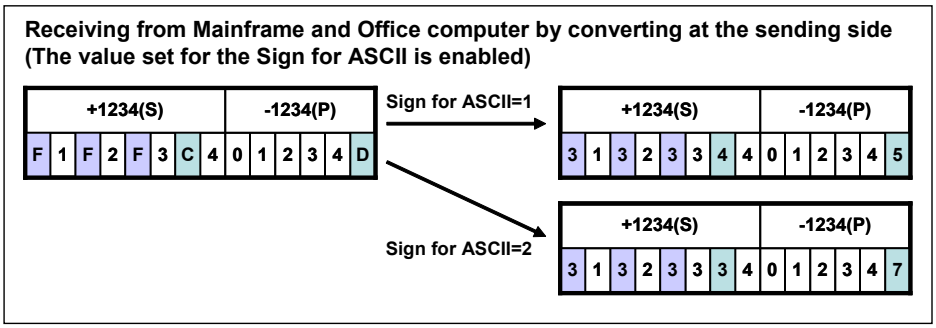
If any values other than described in Table 2.17 or above list is used in the sign, a conversion error occurs.

b) Selection of sign and zone

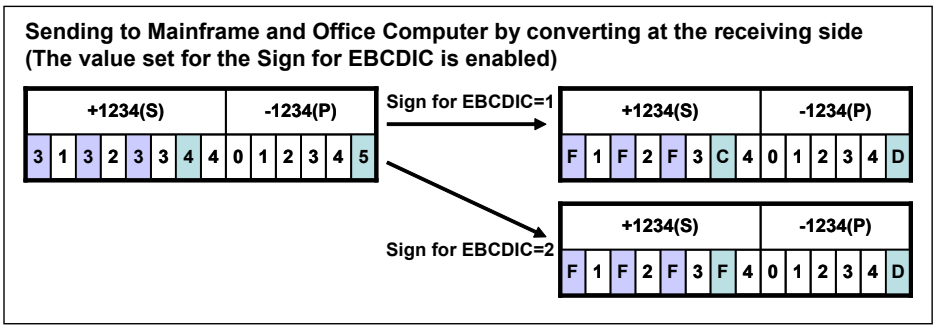
Sign and zone can be selected for format transfer or multi format transfer when converting attributes to signed internal decimal (P) type or signed external decimal (S) type.

This setting will be effective when 'S' is set as the value of the Pack Zone Conversion of the System Environment Settings file. Sign and zone at the time of conversion can be selected by the Sign for EBCDIC and the Sign for ASCII of the System Environment Settings file.

<When receiving from Mainframe and office computers>



<When sending to Mainframe and office computer>



(5) Conversion from UTF-8

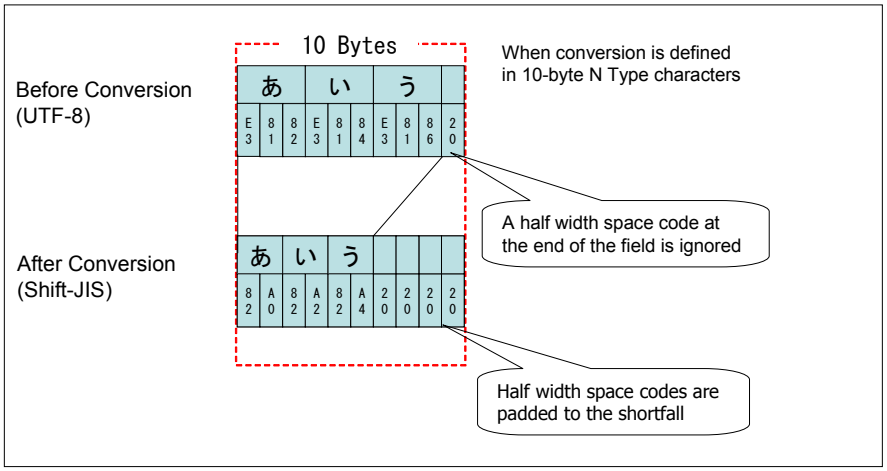
This section explains how HULFT operates when the application uses the Format Information of N type in the conversion from UTF-8.

a) When there is a half width space code at the end of a field

When there is a half width space code at the end of a field, HULFT ignores the space code and executes conversion. To adjust byte count of the fields, use half width space codes. Half width space codes are set to the shortfall after conversion.

Shown below is the example of conversion from UTF-8 to Shift-JIS:

<Example> Conversion from UTF-8 to Shift-JIS

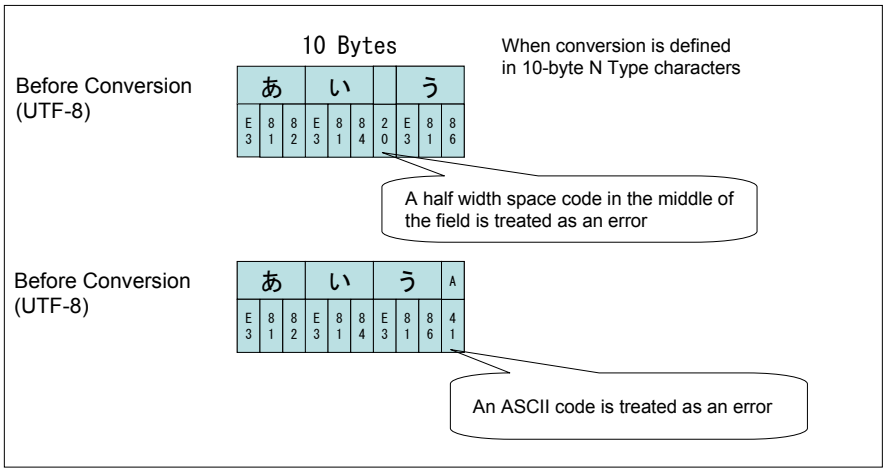


b) When data is turned to be wrong

Inclusion of ASCII codes in N type, or a half width space code in the middle of data turns the data into wrong one, and HULFT fails to transfer the data.

Shown below is the example of conversion from UTF-8 when data is wrong:

<Example> Conversion from UTF-8 (Wrong Data)



(6) Example of Code Conversion

An example of code conversion during Format Transfer is explained in this section.

Some data might be lost depending on the combination of the Field Type and Byte Count of the Format Information specified for format transfer and the Kanji Code Type registered in the Host Information. Specify these settings by considering how shift codes and half width Katakana of EUC and UTF-8 are handled after conversion.

Refer to "2.5.10 Rules of Shift Code Conversion" for the shift code conversion.

a) Missing of Data due to increase in data size

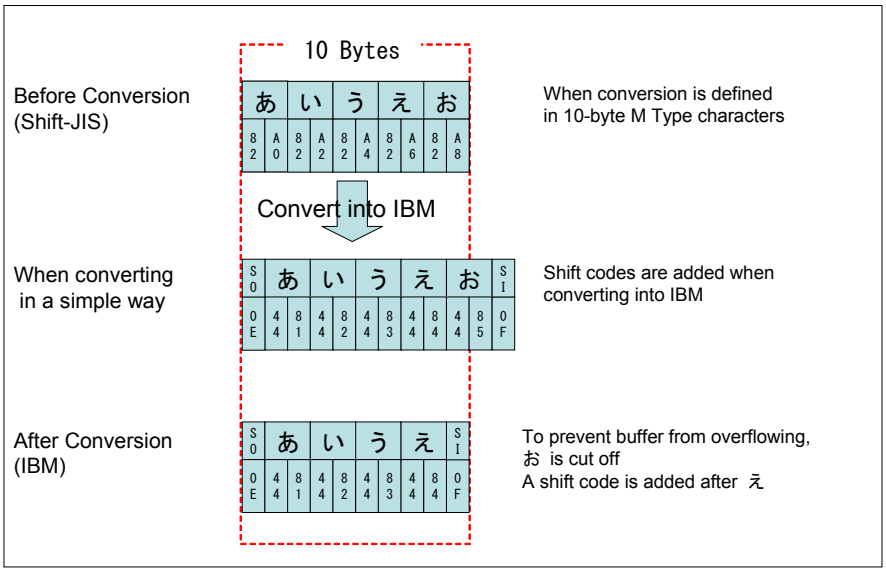
Code conversion may result in the extension of the data size, which makes the size longer than the field length in the Format Information. In such case, the data which has become longer than the field length is cut.

Shown below is the example of data missing in conversion.

Example1) Missing of data due to addition of shift codes

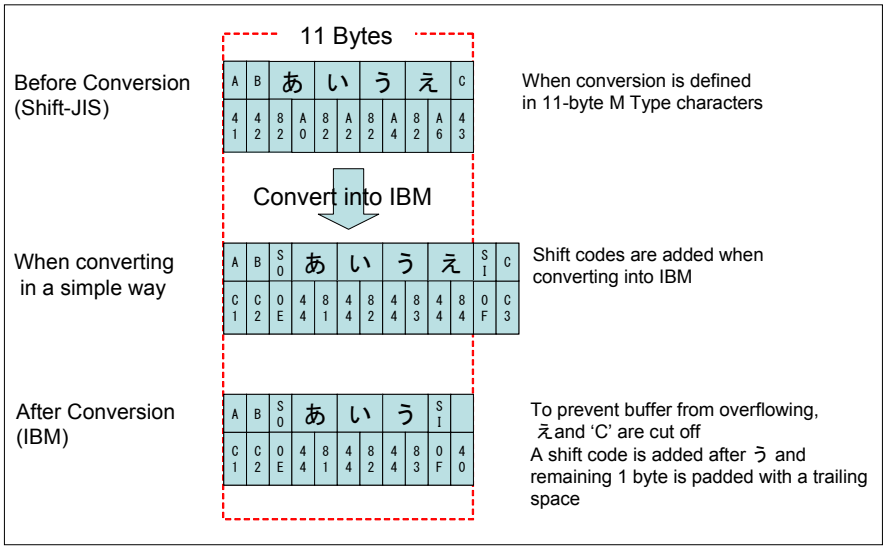
Execution of code conversion from the code types without shift codes (namely, Shift-JIS, EUC, and UTF-8) to those with shift codes (namely, JEF, IBM, KEIS, NEC) may cause missing of data, because shift codes are added by conversion.

<Example 1> Conversion from Shift-JIS to IBM



[Remarks] Above example shows the behavior when Shiftcode Mode in the Send Management information is set to 'Add.'

<Example 2> Conversion from Shift-JIS to IBM

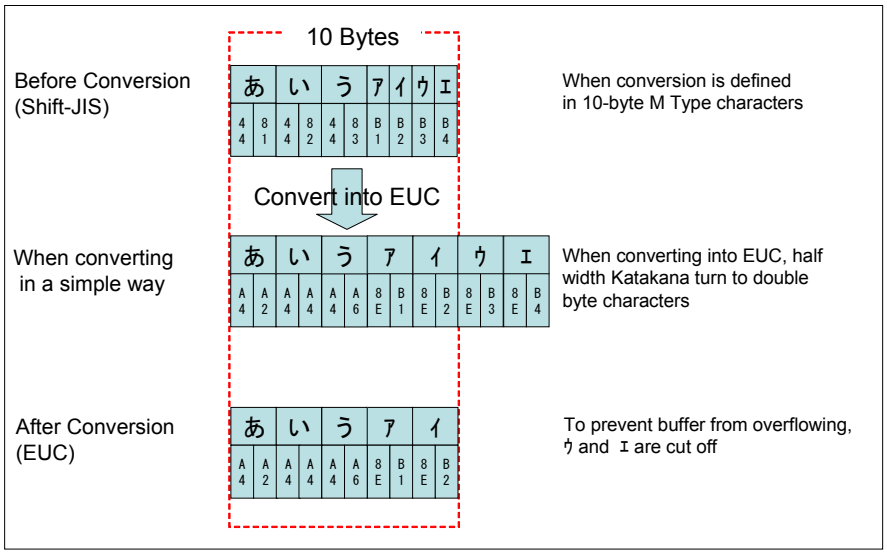


[Remarks] Above example shows the behavior when the Shiftcode Mode in the Send Management information is set to 'Add.'

Example2) Missing of data due to half width Katakana conversion

Execution of code conversion to EUC or UTF-8 may cause missing of data because half width Katakana in data are converted into multibyte characters.

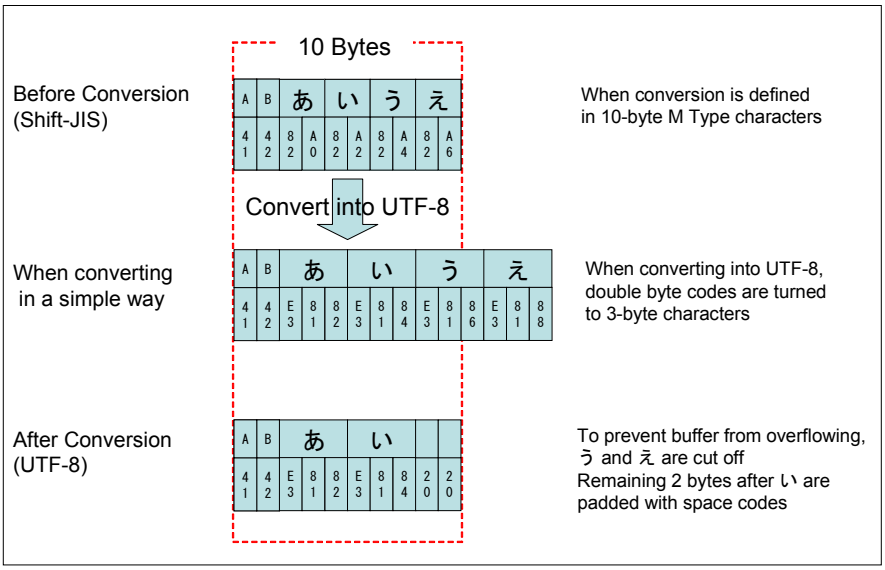
<Example> Conversion from Shift-JIS to EUC



Example3) Missing of data due to change in the length of Kanji code

Execution of code conversion to UTF-8 may cause missing of data, because double byte codes in the data are converted into multibyte codes.

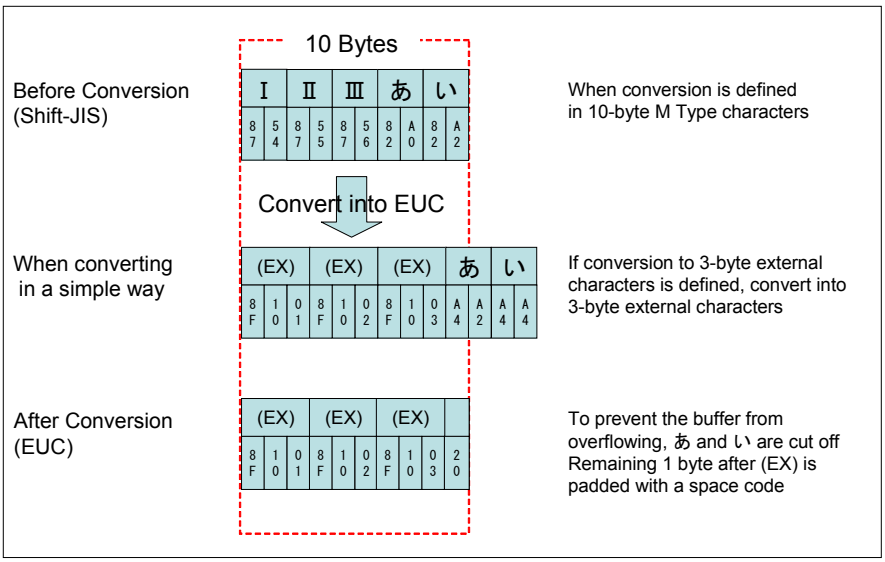
<Example> Conversion from Shift-JIS to UTF-8



Example4) Missing of data due to external character conversion

Execution of code conversion to EUC or UTF-8 converts external characters into the characters of 3 or 4 bytes. Setting Code Conversion Mode to HULFT7 mode (Mode1) can convert external characters into 3 bytes or more. Yet conversion under such settings may cause missing of data.

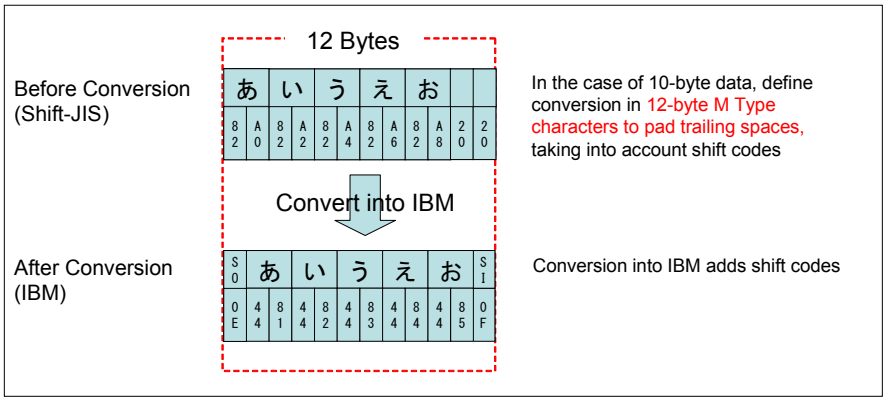
<Example> Conversion from Shift-JIS to EUC



• How to prevent missing of data

To prevent missing of data, calculate the increment in data size first, and set the data format and the Format Information based upon the calculation in advance. Set half width space codes that amount to the increment at the end of the data.

<Example> Conversion from Shift-JIS to IBM—how to prevent missing of data



[Remarks] Above example shows the behavior when the Shiftcode Mode in the Send Management information is set to 'Add.'

b) Padding of space codes due to decrease in data size

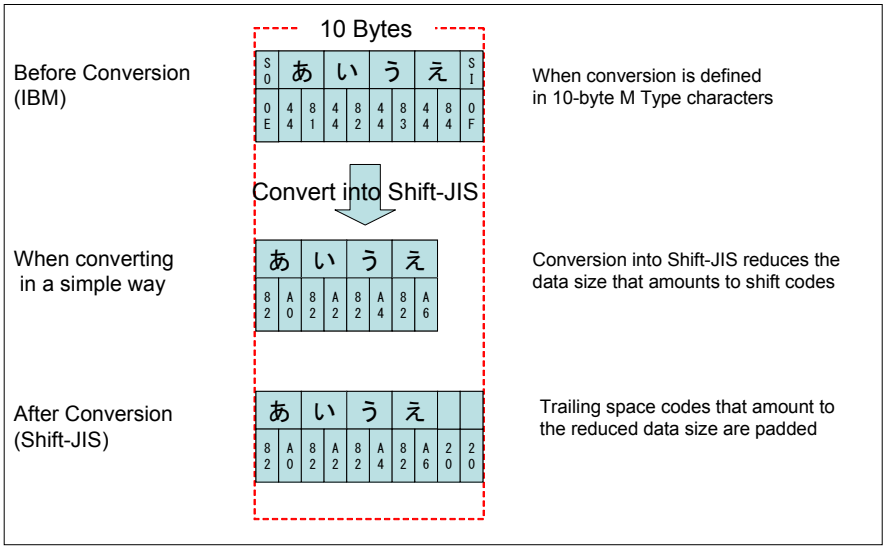
Code conversion may result in the reduction of the data size, which makes the data length shorter than the field length in the Format Information. HULFT pads the data with half width space codes after conversion, if the data becomes shorter than the field length.

[Remarks] The space codes to be padded are half width, regardless of the Field Type(M Type or N Type) specified in the Format Information.

Example1) Decrease in data due to cutting of shift code

Execution of code conversion from the codes with shift codes (namely, JEF, IBM, KEIS, NEC) to those without shift codes (namely, Shift-JIS, EUC, and UTF-8) pads the data with half width space codes because the data that amonts to shift codes are reduced by conversion.

<Example> Conversion from IBM to Shift-JIS

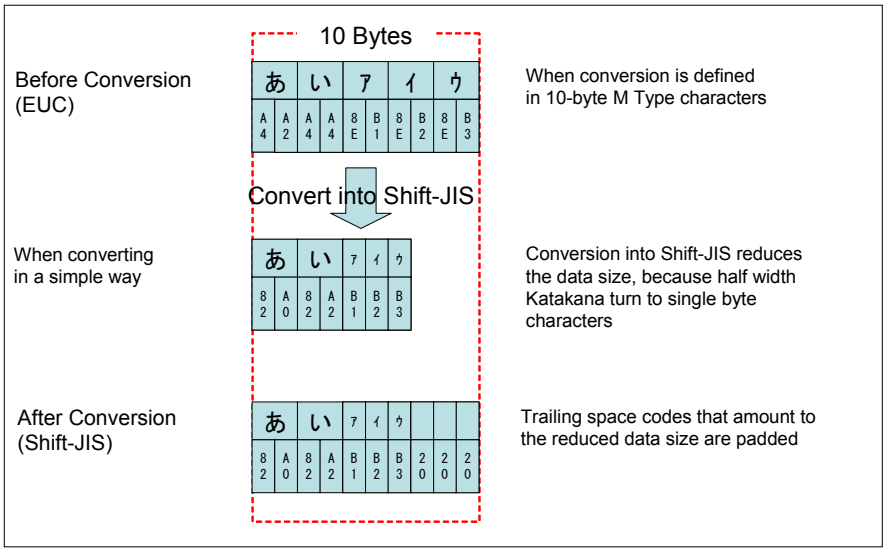


[Remarks] Above example shows the behavior when the Shiftcode Mode in the Send Management information is set to 'Cut.'

Example2) Decrease in data due to conversion of half width Katakana

Execution of code conversion from EUC or UTF-8 turns half width Katakana from multibyte codes to single byte codes. HULFT pads the data with half width space codes, because the data that amounts to multibyte in the half width Katakana is reduced.

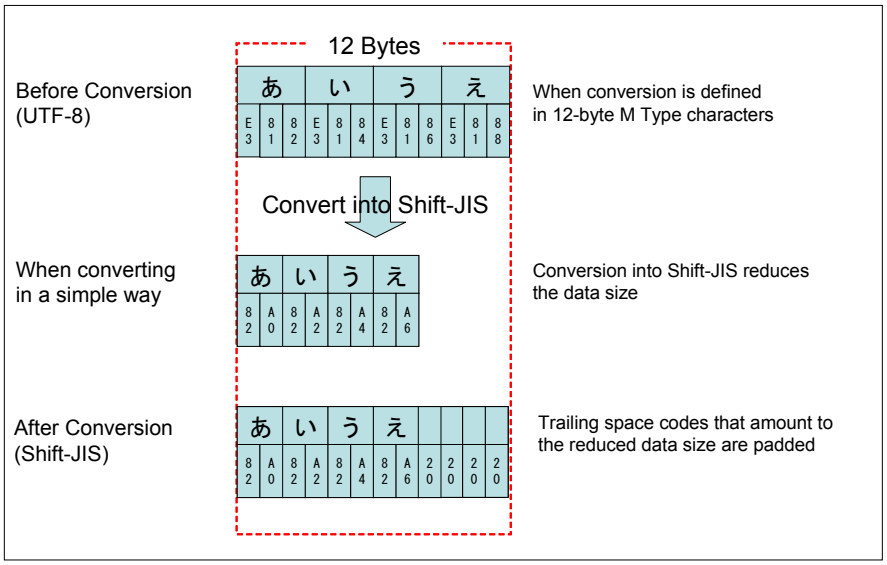
<Example> Conversion from EUC to Shift-JIS



Example3) Decrease in data size due to conversion of the code type of which Kanji code length varies

Execution of code conversion from UTF-8 reduces data size, because multibyte characters are converted into double byte characters. HULFT pads the data with half width space codes that amount to the reduced data.

<Example > Conversion from UTF-8 to Shift-JIS

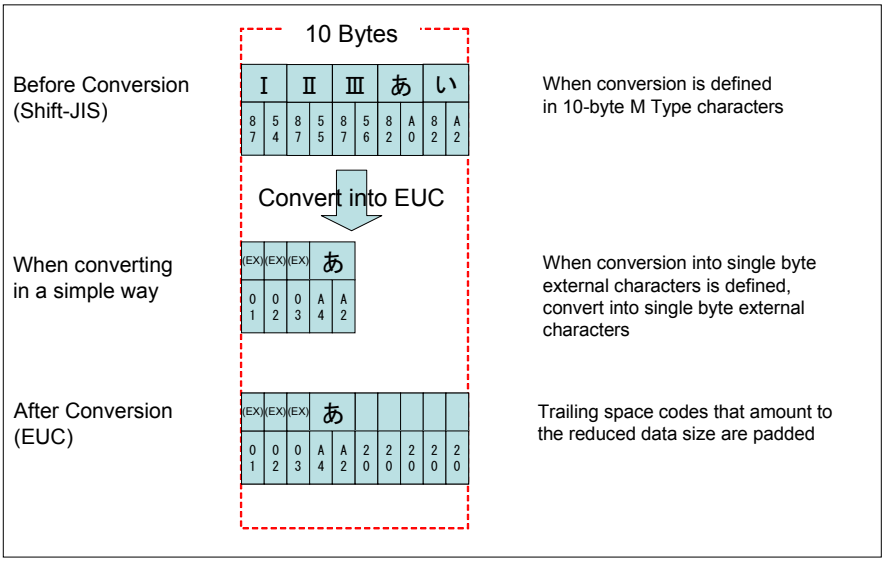


Example4) Decrease in data due to conversion of external characters

Setting the Code Conversion Mode to HULFT7 mode (Mode1) can convert external characters to single byte characters. Conversion under such circumstances may reduce data.

HULFT pads the data with half width space codes that amount to the reduced data.

<Example> Conversion from Shift-JIS to EUC



• How to prevent decrease in data

If the data is reduced, HULFT always pads the data with half width space codes. There is no way to avoid padding of space codes.

2.5.9 Rules of Space Code Conversion

There are two patterns in the rules of space code conversion. One is Mode 0, which carries out the same conversion as the lower version (Ver.1); and the other is Mode 1, which carries out conversion through treating 0x2020 of ASCII code as a half width character. HULFT allows you to select Mode 0 or Mode 1 by specifying '0' or '1,' respectively, for the Space Character Mode in the System Environment Settings.

Further, you can select the conversion of KEIS Em-size (double byte) space code by specifying the KEIS Em-size Space Mode (keisspmode) in the System Environment Settings, when you send a file in Mode 1 to the host that uses KEIS code. (Conversion on Sending Side)

The conversion rules of space code are explained below:

(1) Conversion Rules When Receiving from Mainframe/Office Computers with Receiving Side Conversion

a) Mode 0 (Lower Version Compatibility Mode)

In the Text Transfer, or in the Mixing of Kanji and Characters (M type) in the Format Transfer or in the Multi Format Transfer, Mode 0 converts 0x4040 in shift code into half width space codes (0x2020) of 2 bytes. If there is a full width space code, Mode 0 converts it into a full width space code.

Setting Kanji (N Type) in the Format Transfer or in the Multi Format Transfer converts 0x4040 into half width space codes (0x2020) of 2 bytes. If there is a full width space code, Mode 0 converts it into a full width space code.

b) Mode 1

In the Text Transfer, or in the Mixing of Kanji and Characters (M type) in the Format Transfer or in the Multi Format Transfer, Mode 1 converts 0x4040 in shift code into a full width space code. If there is a full width space code, Mode1 converts it into a full width space code.

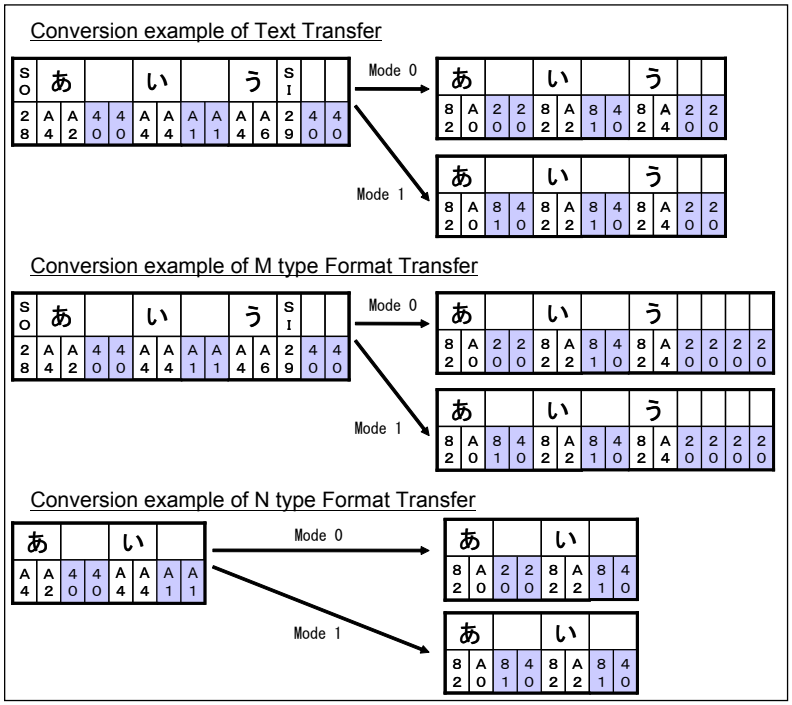
Setting Kanji (N Type) in the Format Transfer or in the Multi Format Transfer converts 0x4040 into full width space codes. If there is a full width space code, Mode 1 converts it into a full width space code.

Table 2.18 Space Code Conversion Table 1

Type	Before Conversion		After Conversion					
			Mode 0			Mode 1		
	Kanji Code Type	Code (hexadecimal number)	Shift-JIS	EUC	UTF-8	Shift-JIS	EUC	UTF-8
M type (in shift code)	JEF	0x4040	0X2020	0X2020	-	0X8140	0XA1A1	0XE38080
		0xA1A1	0X8140	0XA1A1	-	0X8140	0XA1A1	0XE38080
	IBM	0x4040	0X2020	0X2020	-	0X8140	0XA1A1	0XE38080
	KEIS	0x4040	0X2020	0X2020	-	0X8140	0XA1A1	0XE38080
		0xA1A1	0X8140	0XA1A1	-	0X8140	0XA1A1	0XE38080
	NEC	0x4F4F	0X2020	0X2020	-	0X8140	0XA1A1	0XE38080
N type	JEF	0x4040	0X2020	0X2020	-	0X8140	0XA1A1	0XE38080
		0xA1A1	0X8140	0XA1A1	-	0X8140	0XA1A1	0XE38080
	IBM	0x4040	0X2020	0X2020	-	0X8140	0XA1A1	0XE38080
	KEIS	0x4040	0X2020	0X2020	-	0X8140	0XA1A1	0XE38080
		0xA1A1	0X8140	0XA1A1	-	0X8140	0XA1A1	0XE38080
	NEC	0x4F4F	0X2020	0X2020	-	0X8140	0XA1A1	0XE38080

*If the code is UTF-8, conversion is always performed in Mode1.

<Example> Receiving from JEF to Shift-JIS



[Remarks] Above example shows the behavior when the Shiftcode Mode in the Send Management Information is set to 'Cut.'

(2) Conversion Rules When Sending to Mainframe/Office Computers with Sending Side Conversion

a) Mode 0 (Lower Version Compatibility Mode)

In the Text Transfer, or in the Mixing of Kanji and Characters (M type) in the Format Transfer or in the Multi Format Transfer, Mode 0 converts half width space codes (0x2020) of 2 bytes that follow a double byte character into 0x4040. Mode 0 does not insert a shift code between a double byte character and 0x4040.

Setting N Type in the Format Transfer or in the Multi Format Transfer converts half width space codes (0x2020) of 2 bytes into 0x4040. If there is a full width space code, Mode 0 converts it into a full width space code.

b) Mode 1

In the Text Transfer, or in the Mixing of Kanji and Characters (M type) in the Format Transfer or in the Multi Format Transfer, Mode 1 converts 0x2020 that follows a double byte character into 0x4040. Further, Mode 1 inserts a shift code between a double byte character and 0x4040.

Setting N Type in the Format Transfer or in the Multi Format Transfer converts half width space codes (0x2020) of 2 bytes into a full width space code. If there is a full width space code, Mode 1 converts it into a full width space code.

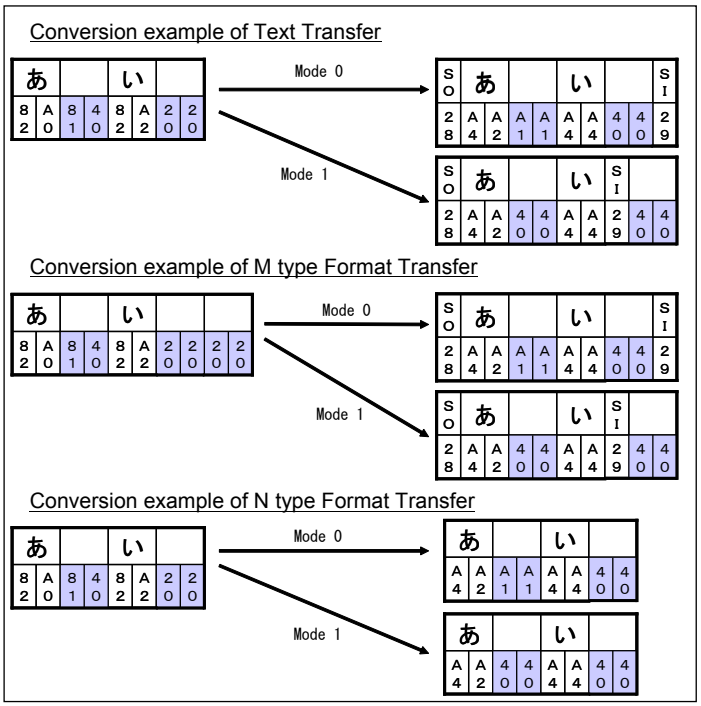
When you intend to convert codes into KEIS in Mode 1, you can select conversion into full width space codes by specifying the KEIS Em-size Space Mode(keisspmode) in the System Environment Settings. For details of the KEIS Em-size Space Mode, refer to " 3.4 System Environment Settings."

Table 2.19 Space Code Conversion Table 2

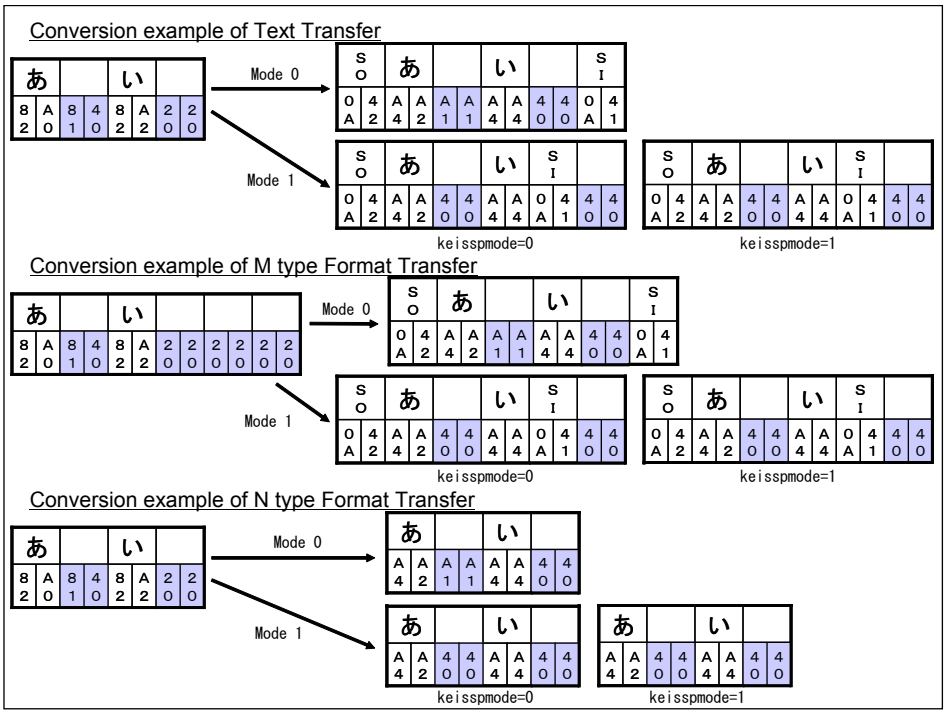
Type	Before Conversion		After Conversion							
			Mode 0				Mode 1			
	Kanji Code Type	Code (hexadecimal number)	JEF	IBM	KEIS	NEC	JEF	IBM	KEIS	NEC
M type (After double byte character)	Shift-JIS	0x2020	0X4040	0X4040	0X4040	0X4F4F	0X4040	0X4040	0X4040	0X4040
		0x8140	0XA1A1	0X4040	0XA1A1	0X4F4F	0X4040	0X4040	0X4040 0XA1A1	0X4F4F
	EUC	0x2020	0X4040	0X4040	0X4040	0X4F4F	0X4040	0X4040	0X4040	0X4040
		0xA1A1	0XA1A1	0X4040	0XA1A1	0X4F4F	0X4040	0X4040	0X4040 0XA1A1	0X4F4F
	UTF-8	0x2020	-	-	-	-	0X4040	0X4040	0X4040	0X4040
		0XE38080	-	-	-	-	0X4040	0X4040	0XA1A1	0X4F4F
N type	Shift-JIS	0x2020	0X4040	0X4040	0X4040	0X4F4F	0X4040	0X4040	0X4040	0X4F4F
		0x8140	0XA1A1	0X4040	0XA1A1	0X4F4F	0X4040	0X4040	0X4040 0XA1A1	0X4F4F
	EUC	0x2020	0X4040	0X4040	0X4040	0X4F4F	0X4040	0X4040	0X4040	0X4F4F
		0xA1A1	0XA1A1	0X4040	0XA1A1	0X4F4F	0X4040	0X4040	0X4040 0XA1A1	0X4F4F
	UTF-8	0x2020	-	-	-	-	0X4040	0X4040	0X4040	0X4F4F
		0XE38080	-	-	-	-	0X4040	0X4040	0XA1A1	0X4F4F

*If the code is UTF-8, conversion is always performed in Mode1.

<Example> Sending from Shift-JIS to JEF



<Example> Sending from Shift-JIS to KEIS



(3) Conversion Rules between ASCII Type Hosts (EUC, Shift-JIS, and UTF-8)

The conversion of the same sort is performed in terms of conversion between ASCII Type Hosts (EUC, Shift-JIS, and UTF-8), regardless of code conversion side, sending or receiving.

If the code is UTF-8, conversion is performed in Mode1.

a) Mode 0

Mode 0 leaves half width space codes (0x2020) of 2 bytes as they are.

If there is a full width space code, Mode 0 converts it into a full width space code.

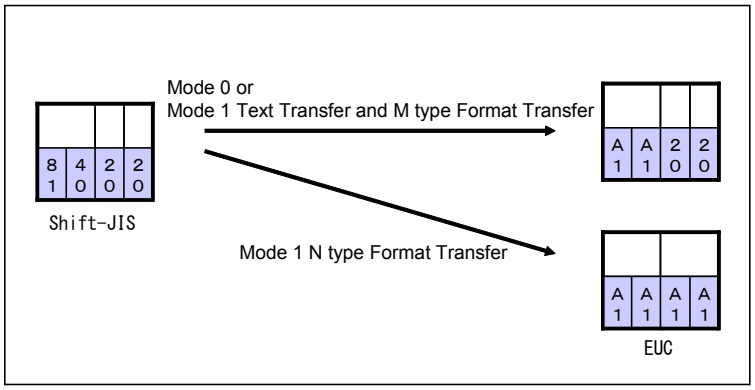
b) Mode 1

Setting N Type in the Format Transfer or in the Multi Format Transfer converts half width space codes (0x2020) of 2 bytes into a full width space code.

If there is a full width space code, Mode 1 converts it into a full width space code.

In the Text Transfer, or in the M type in the Format Transfer or in the Multi Format Transfer, Mode 1 performs the same sort of conversion as Mode 0.

<Example> Sending from Shift-JIS to EUC



2.5.10 Rules of Shift Code Conversion

In Mainframe and office computers, shift codes are used to identify double byte codes.

The length of record increases when the shift code is added at the time of transfer from Windows to Mainframe and office computers. Similarly, the record length decreases when the shift code is deleted in transfer from Mainframe and office computer to Windows. Handling of shift codes can be specified to prevent such increase and decrease of record length. The Shiftcode Mode should be set on the sending side irrespective of the side that performs code conversion.

(1) Conversion Rules for Sending to Mainframe and Office Computers

When data is transmitted from Windows to Mainframe or office computers, the option whether or not to add the shift code can be set in the 'Shiftcode Mode' in the Send Management Information on the sending side Windows. Refer to *Operation Manual* for the Send Management Information.

a) When shift code is added

In the Text transfer, or in the Mixing of Kanji and Characters (M type) in the Format Transfer or in the Multi Format Transfer, setting that adds a shift code adds a shift code to a double byte character.

The setting is disabled in the Kanji (N) type in the Format Transfer or in the Multi Format Transfer.

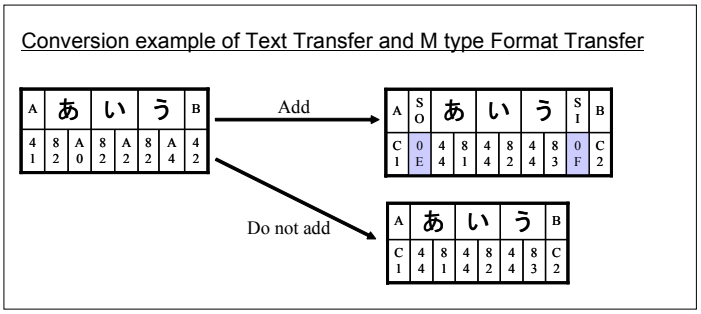
[Note] As for the Mixing of Kanji and Characters (M type) in the Format Transfer or in the Multi Format Transfer, specify data format and the Format Information by considering the extra shift codes to be added so that no data is missed. For details, refer to "2.5.8 Code Conversion Rules for Format Transfer."

b) When shift code is not added

In the Text transfer, or in the Mixing of Kanji and Characters (M type) in the Format Transfer or in the Multi Format Transfer, the setting that does not add a a shift code does not add a shift code to a double byte character.

The setting is disabled in the Kanji (N) type in the Format Transfer or in the Multi Format Transfer.

<Example> Conversion from Shift-JIS to IBM



(2) Conversion Rules for Receiving from Mainframe and Office Computers

When receiving from the host on Mainframe or office computers, how to handle shift codes is determined by the settings of the Shiftcode Mode in the Send Management Information of Mainframe or office computers on sending side as well as the setting in the System Environment Settings (The Shiftcode Mode for Send Files in terms of Mainframe, and the Fixed Length Transfer in terms of i5OS).

Refer to the *Administration Manual* and *Operation Manual* of the remote host on Mainframe and the office computer for the setting details.

a) When shift code is deleted (text transfer only)

In the Text Transfer, shift codes are deleted from the data when the following settings are configured:

- Send Management Information Shiftcode Mode is 'C (Cut)'
- System Environment Settings

  In Mainframe : Shiftcode Mode for Send Files is 'N'

  In i5OS : Fixed Length Transfer is 'N'

  ('N' : Space is not added)

b) When shift code is deleted and space is added at the end of the data

In the Text Transfer, shift codes are deleted from the data, and spaces of the same size as the deleted shift codes are added at the end when the following settings are configured:

- Send Management Information Shiftcode Mode is 'C (Cut)'
- System Environment Settings

  In Mainframe : Shiftcode Mode for Send File is 'Y'

  In i5OS : Fixed Length Transfer is 'Y'

  ('Y' : Space is added)

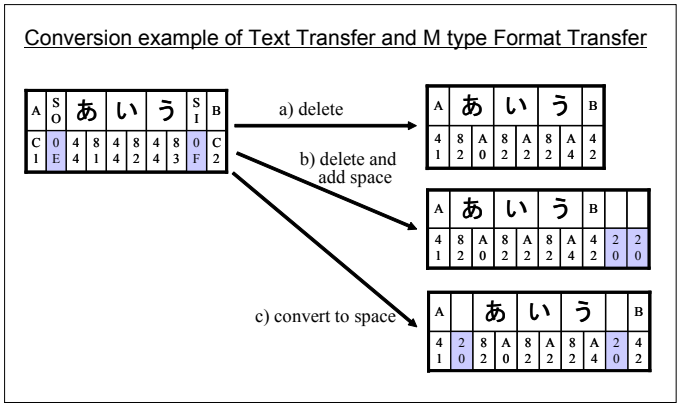
In the case of the Mixing of Kanji and Characters (M type) in the Format Transfer or in the Multi Format Transfer, setting the Shiftcode Mode in the Send Management Information to 'C (Cut)' adds the spaces that amount the size of the deleted shift codes at the end of the data.

c) When the shift code is converted into space

In the Text Transfer, or in the Mixing of Kanji and Characters (M type) in the Format Transfer or in the Multi Format Transfer, setting the Shiftcode Mode in the Send Management Information to 'S (Space),' and if there is any, converts a shift code into a space.

[Remarks] In the case of a), the record length becomes shorter by the size of the deleted shift codes, while the length is maintained in the case of b) and c).

<Example> Conversion from IBM to Shift-JIS



2.6 Operation Log

When executing a command or accessing to a file, HULFT can keep the record (operation log) that indicates what having been done by whom. When suspicious log records of sending or receiving are found, or an improper value is set to management information, you can identify unlawful computer access or the operational mistakes and the like by examining the corresponding operation logs.

[Remarks] To use operation log output function, it is necessary to set the Operation Log Output Option (oplselect) in the System Environment Settings to a value other than '0.'

2.6.1 Outline of Operation Log

Operation logs are output when a request is issued to HULFT, a command of HULFT is executed, or a system file is accessed by HULFT.

The operation logs output by the Send and Receive processes are shown below:

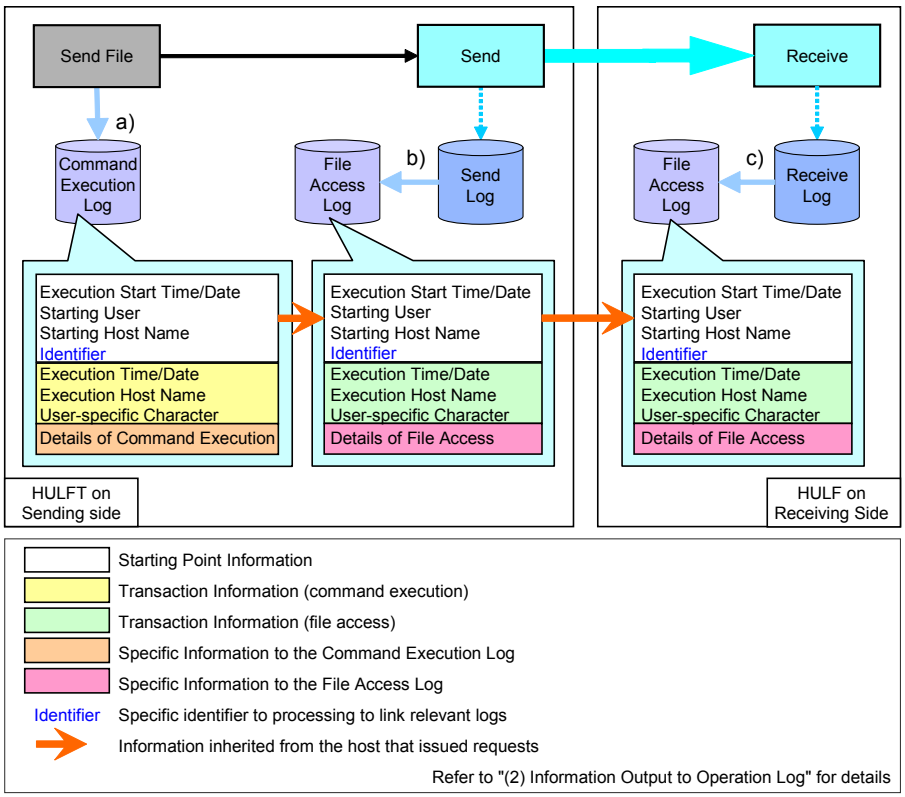


Figure 2.37 Operation Logs in Send and Receive Processes

a) When the Send File is issued on the Management screen or in the command line, the date of issuing the request, the User ID, the Host Name, and the like are output to the Command Execution Log of HULFT on the sending side.

b) When the Send process is executed based on the Send File and a Send Log is added, the date when the log was added, the information that identifies the added log, and the like are output to the File Access Log of HULFT on the sending side.

c) When the Receive process is executed and a Receive Log is added, the date when the log was added, the information that identifies the added log, and the like are output to the File Access Log of HULFT on the receiving side.

(1) File Access Log and Command Execution Log

The records of file access and the records of request issuance or command execution are output to separate logs. (The File Access Log and the Command Execution Log, respectively)

Table 2.20 Variety of Operation Log

Variety of Operation Log	Output Destination	Explanation
File Access Log	HULPATH\ opl\huloplfile.csv	The Log is output when information is added, updated, or deleted in system files, such as System Environment Settings, system management information, and various logs and the like
Command Execution Log	HULPATH\ opl\huloplcmd.csv	The Log is output when a request is issued or a command is executed

Refer to "Appendix 1 I/O Format of Operation Log" for target system files, target types of file access, and target requests or commands of operation logs.

[Remarks]

- The File Access Log is output when the contents in the system file are modified. Mere reference of the file without modification does not output the Log.
- The Command Execution Log is output by the host on the side where a request or command was executed. The Request Acknowledge Log as well as the File Access Log that carries the record of the addition in the Request Acknowledge Log are output by the host on the side where a request was acknowledged.
- Where multiple records are deleted from system management information or logs by specifying the target range, the Command Execution Log outputs only one entry, while the File Access Log outputs one entry per deleted record.
- The operation logs do not target internal behavior of the system because the log aims at keeping the record of the operation by users.

(2) Information Output to Operation Log

Information output to the operation log is divided roughly into following categories: information on request issuance or command execution that is an origin of a series of transaction (starting point information), information on the transaction performed according to request issuance or command execution (transaction information), information specific to the File Access Log, and information specific to the Command Execution Log.

The following information is output to the operation log in CSV format:

<Starting Point Information>

Starting point information is the information at the time when request issuance or command execution is started.

The information is notified to the connection destination, and the same information is used by all the File Access Logs and the Command Execution Logs within one transaction.

[Remarks] The fields other than user ID in starting point information are always notified to the connection destination when the operation log output function is used.

Execution Start Time/Execution Start Date

Time and date that request issuance or command execution started

It is a date set on OS of starting host.

User ID (OS) and User ID (Management Screen)

User ID of the user who performed operation

They are OS user ID of starting host, and the user ID of the Management screen that issued the request, respectively.

Their definition varies depending on the types of starting host:

Table 2.21 Difference in Definition of User ID (OS) or User ID (Management Screen) by Host Types

Host Type	User ID (OS)	User ID (Management Screen)
Mainframe	User ID of OS	User ID of the user who used the Management screen
UNIX/Linux	User Name of OS	Same as 'User ID(OS)' (When the Management Screen Security is disabled, indicated as "")
Windows	User Name of OS	User Name registered on the Management Screen Security (When the Password is disabled, indicated as "")
i5OS	User ID of OS	Same as 'User ID(OS)' (When the Management Screen Security is disabled, indicated as "")
HULFT Manager	User Name of OS	User ID (Management Screen) of HULFT at connection destination
HULFT-HUB Server	User Name of OS	Always indicated as ""because there is no Management screen
HULFT-HUB Manager	User Name of OS	The User Name that logged on HULFT-HUB Server

Starting Host Name

The name of local host that performed the operation

Refer to "4.2 Local Host Name" for the Local Host Name.

Identifier

Identifier of transaction

In each transaction, different identifier is generated. There are two types of identifier: 'Latest Identifier' and 'Starting Identifier.'

Refer to "2.6.2 Identifier" for the Latest Identifier and the Starting Identifier.

Identifiers can link the record output to the operation log files with the record in the Send/Receive/Request Acknowledge Logs, because they are output to the Send/Receive/Request Acknowledge Logs.

<Transaction Information>

Transaction information is the information at the time when a file is accessed, or request issuance or command execution is processed.

Execution Time/Execution Date

Time and date when file access, or request issuance or command execution was processed

It is a time and date set on OS of execution host.

Execution Host Name

Local Host Name of the host to which file access, or request issuance or command execution was processed

Refer to "4.2 Local Host Name" for the Local Host Name.

Operation Log User-specified Character

Character string to identify HULFT that outputs the operation log

The Operation Log User-specified Character is a character string to identify which HULFT outputs which operation log when the operation logs output by two or more HULFTs are accumulated.

Table 2.22 Values Set to Operation Log User-specified Character

Host Type	Value Set to Operation Log User-specified Character
Mainframe	Operation Log User-specified Character in the System Environment Settings
UNIX/Linux	HULFT Identifier in the System Environment Settings
Windows	Operation Log User-specified Character in the System Environment Settings
i5OS	Operation Log User-specified Character in the System Environment Settings
HULFT Manager	User-specified Character in the System Environment Settings
HULFT-HUB Server	Service Name in the System Environment Settings
HULFT-HUB Manager	User-specified Character in the System Environment Settings

<Specific Information to File Access Log>

Specific information to the File Access Log is the information on the types and objects of file access. It is output only to the File Access Log.

Refer to "Appendix 1 I/O Format of Operation Log" for the list of the output values.

File Access Type

Types of file access

The File Access Type means varieties of operation, such as addition, modification, deletion and the like.

System File Type

The system file that is an object of file access

It is an identifier that indicates the object system file of file access.

File Key Information

Fields in the file that are objects of file access

The File Key Information is an additional information to identify the record that was accessed, when two or more records are stored in one file, such as system management information, or Send/Receive/Request Acknowledge Logs and the like.

<Specific Information to Command Execution Log>

Specific information to the Command Execution Log is the information on the processed requests or commands. The Information is output only to the Command Execution Log.

Refer to "Appendix 1 I/O Format of Operation Log" for the list of the output values.

Command Execution Key

Varieties of requests or commands

It is an identifier that indicates the processed requests or commands.

Command Issued by

Issuer of requests or commands

It is an identifier that indicates the source of requests or commands, such as whether the command was executed in the command line, and whether the request was issued on HULFT Management screen.

Command Parameter

The Command Parameter is a parameter specified when the parameter command is executed.

2.6.2 Identifier

The Identifier is a peculiar identifier generated each time a request is issued or a command is executed.

The Identifier can link all the operation logs and the Send/Receive/Request Acknowledge Logs that are diversified into two or more hosts, because the same value is output to all the operation logs and the Send/Receive/Request Acknowledge Logs that are generated from request issuance or command execution at one time.

Example 1) Investigating the user that issued a request by referring the Send Log and the Receive Log

Confirming the identifier of the Send Log and the Receive Log, and searching it in the File Access Log by specifying the identifier of the log allows you to identify the user or starting host that performs the operation.

Specifying the same identifier to search it in the Command Execution Log of starting host provides you with the details of request issuance or command execution.

Example 2) Investigating the user that modified the system management information

Specifying the system management information and access type (=update) and searching them in the File Access Log allows you to identify the user that modified the management information.

Specifying the identifier to search it in the Command Execution Log provides you with the details of request issuance or command execution.

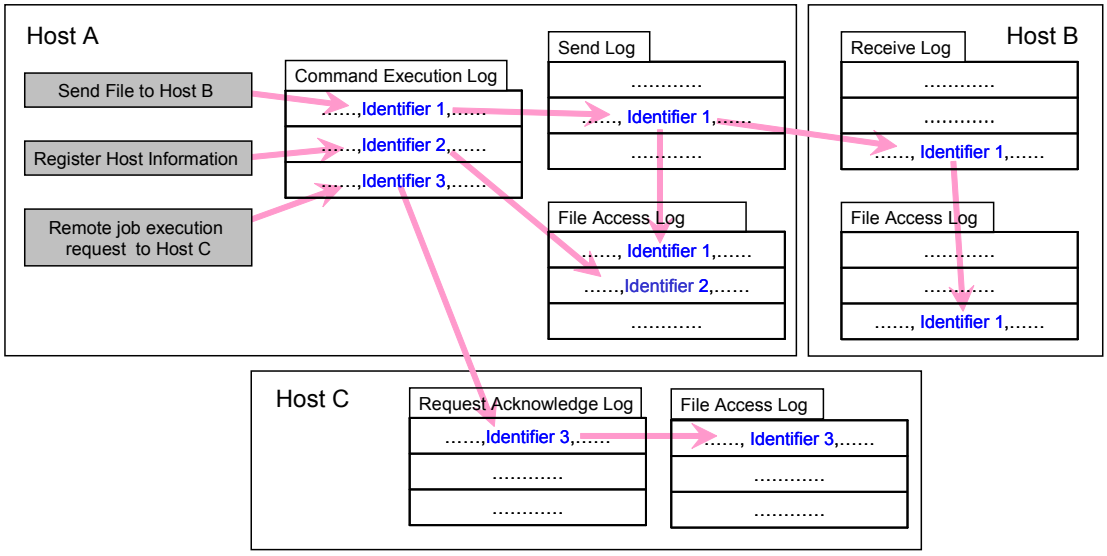


Figure 2.38 Linking Logs by Identifier

Using Identifier in Combination with HULFT-HUB

The Identifier is classified into the Latest Identifier and the Starting Identifier.

Usually, the values of two identifiers are identical. The value is not identical when HULFT received data that were accumulated in HULFT-HUB Server.

In such case, different identifiers are generated, because HULFT regards accumulation and delivery are different transactions. Therefore, HULFT offers two types of identifier, because such behavior by HULFT does not allow users to link the operation logs and the Send/Receive/Request Acknowledge Logs on the sending side with those on the receiving side.

The Starting Identifier maintains the value generated upon accumulating, while the Latest Identifier generates new value each time delivery is performed.

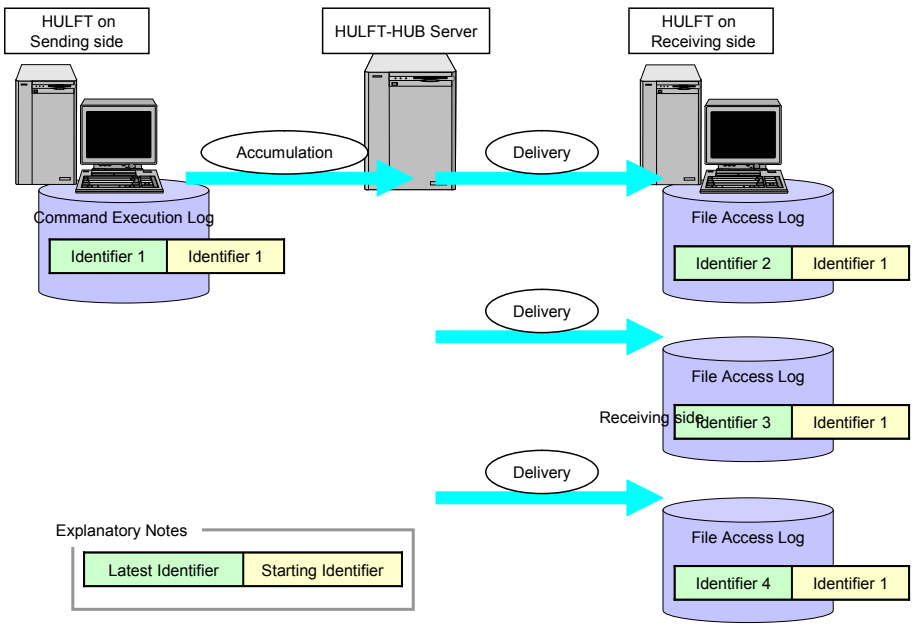


Figure 2.39 Latest Identifier and Starting Identifier

[Note] **When HULFT-HUB Server delivers data, the User ID and the Starting Host Name is overwritten in addition to the Latest Identifier.**

Therefore, the information that links the sending side with the receiving side is only the Starting Identifier, when the data are accumulated and delivered by HULFT-HUB Server.

Output the File Access Log and the Command Execution Log by using Operation Log List Display commands, and confirm the Starting Identifier.

2.6.3 Notify User Details

In line with the operating environment, you can select whether to notify the user ID to the connection destination host.

You can set this option by specifying the Notify User Details (USRNOTIFY) in the Host Information. You can apply the operation such as notifying the User ID to the host in the same office, while withholding the User ID from the host in a different office, because it is possible to set this field host by host.

For the host that is not notified the User ID the character string "N/A" is output instead of the User ID. Even when the User ID is not notified to the connection destination host, the User ID is output to the operation log of local host.

[Remarks] The default value of the Notify User Details (USRNOTIFY) is '(N) No.'

When the operation log output function is used, it is necessary to set this field to 'Yes (Y)' on each host to whom you intend to notify the User ID.

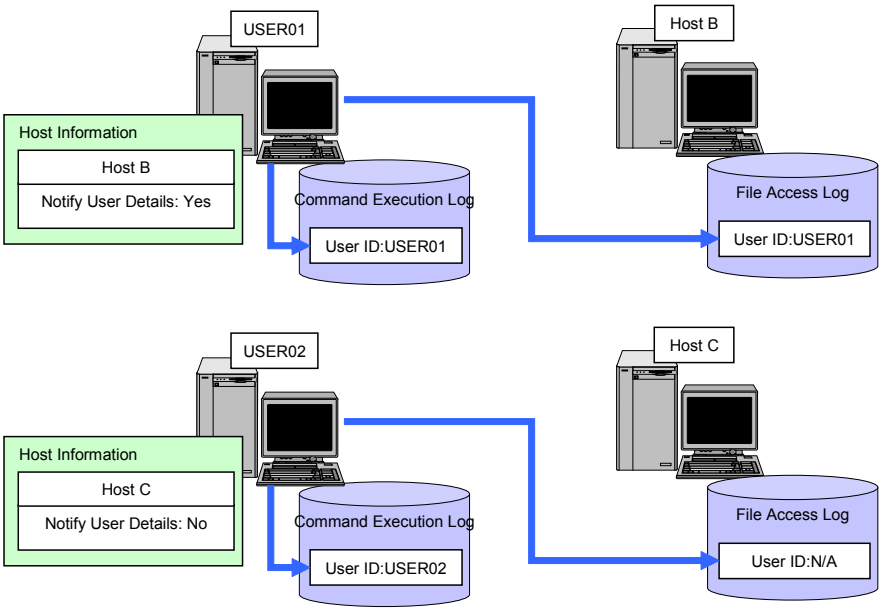


Figure 2.40 Notify User Details

2.6.4 Automatic Switching of Operation Log

This function automatically saves the output contents as a backup file when the File Access Log or the Command Execution Log reaches a certain size. As a result, you can suppress the disk utilization of the operation log so as not to exceed a certain amount, or save the operation log on a log server at regular intervals.

The automatic switching of the operation log can be set in the Operation Log Auto Switch Size and the Operation Log Switch Generation Count in the System Environment Settings.

In outputting the operation log, the original file is renamed to be backed up and an output file is newly created, if the file size exceeds the setting in the Operation Log Auto Switch Size. The backup file is created in the same folder as the output destination of the operation log. The backup file is saved until a generation count reaches the value specified in the Operation Log Switch Generation Count, while a backup file is overwritten in order from the Generation File Number 0001, if the value exceeds the setting.

The backup file names are indicated as follows:

File Access Log: Huloplfile_bk_nnnn.csv
Command Execution Log: Huloplcmd_bk_nnnn.csv

*nnnn is Generation File Number (four digits, ranging from 0001 to 9999).

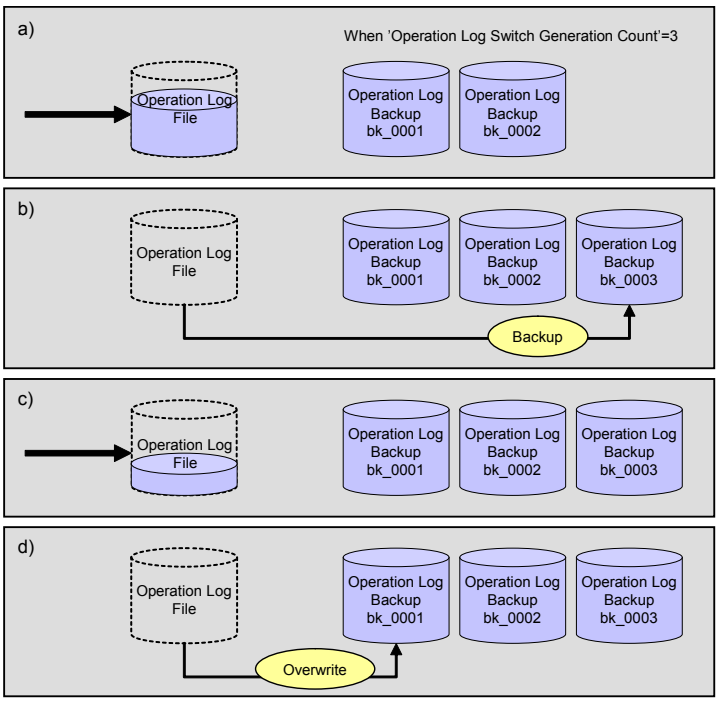


Figure 2.41 Automatic Switching of Operation Log

a) Size check

Check the file size before outputting the operation log.

Output the operation log as it is and end the processing, when the file size does not exceed the setting in the Operation Log Auto Switch Size.

b) Switching

Rename and back up the operation log file, when the file size exceeds the setting in the Operation Log Auto Switch Size.

c) Outputting Operation Log

Create fresh operation log file, and output the operation log.

d) Exceeding Generation Count

When the number of backup files exceeds the setting in the Operation Log Switch Generation Count, a backup file is overwritten in order from the Generation File Number 0001.

2.6.5 Manual Switching of Operation Log

Processing similar to "2.6.4 Automatic Switching of Operation Log" can be executed at any point by using a command. The details of processing and the file name of the backup file to be created are the same as the case of automatic switching.

Manual switching of the operation log allows you to acquire the operation logs generated till then to store them in a log server and analyze the logs, without waiting for the automatic switching or stopping HULFT.

Refer to the explanation of the Operation Log Manual Switch command (utloplchg) in *Operation manual* for details.

2.6.6 Search and Output of Operation Log

By specifying conditions, the relevant operation log can be output in a list or to a file. All the operation logs become the objects of searching and outputting, including the backups created by manual or automatic switching.

Refer to the explanation of utlopllist in the section of the Operation Log Control Commands in *Operation manual* for details.

2.6.7 Points to be Noted on Operation Log

(1) Execution Date

- Displaying the operation log in a list by specifying date may not work properly, when the date set on OS of each host does not correspond each other.

(2) User ID

- Executing a command does not output anything in the field of the User ID (Management Screen).
- Failure in obtaining a User ID outputs "FAILED" as User ID. Do not use the term "FAILED" for the User ID.
- When a User ID is not notified by the starting point host, "N/A" is output as the User ID. Do not use "N/A" for the User ID.
- Do not use double byte code characters other than JIS Level 1 or JIS Level 2.

(3) Identifier

- Occurrence of an error before obtaining an Identifier from a starting point host does not output the operation log to an execution host.
- The accounts in this manual does not guarantee that HULFT does not generate the same Identifier into the future. The same Identifier might be generated for a different transaction depending on the circumstances.

(4) Output Strings

- Using a double quotation in a parameter such as -msg of utlsend, utlrecv and the like prevents the operation log from handling the string as CSV properly.

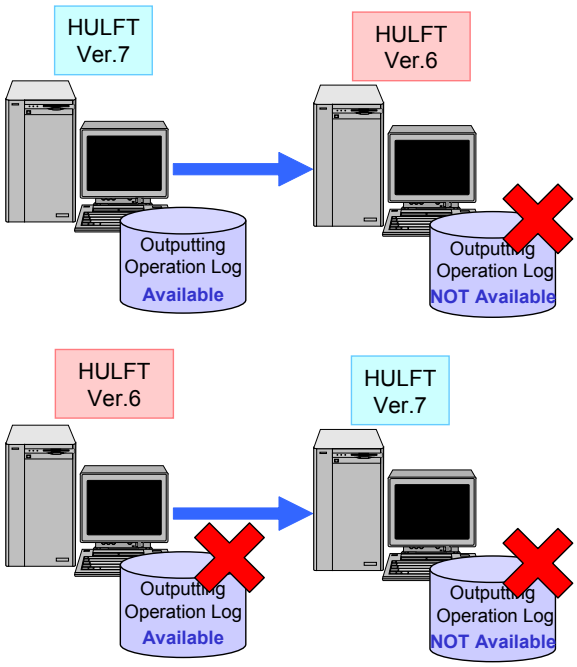
  (ex.) "cannot be used for -msg message in utlsend.

(5) Switching of Operation Log

- When the number of backups exceeds the setting in the Operation Log Switch Generation Count due to switching of the operation log, a backup file is overwritten beginning with the chronologically oldest file. Even defaulting the Operation Log Switch Generation Count reverts the Generation File Number to 0001, when the number reaches to 9999. Therefore, should the old file exist, it is overwritten. Save the backup in log server and the like at regular intervals, if necessary.

(6) Restriction on Outputting Operation Log

The operation via HULFT of which version is lower than Ver.7 does not output the operation logs.



(7) Output Processing of Operation Log

When the operation log files (the Command Execution Log and the File Access Log) have been exclusively opened to other process, writing process into the operation log basically keeps waiting for the release of the exclusive access control.

Therefore, a response might not be returned when the operation that outputs the operation log (namely, starting the Management screen, upgrading the management information, request issuance, command execution and the like) is executed, while you have opened the logs on the application such as Microsoft Excel and the like.

In such case, exit the process which opens the operation log file for a moment. The release of exclusive access control allows HULFT to write into the operation log and the processing continues.

2.7 Security

HULFT provides security functions such as the encryption of data in the transfer file, the verification of the consistency of the transfer data, and the checking authorization when being connected from other hosts and the like. This section explains the operation concerning security using the function of HULFT.

2.7.1 Encryption

HULFT can encrypt file contents to transfer them. There are two types of the Encryption Scheme as shown below:

- HULFT Encryption Scheme
- Encryption Scheme by exit routine

Two or more encryption schemes cannot coexist in HULFT.

However, HULFT transfers a file with HULFT Encryption Scheme when the specification of the Encryption Scheme is different between the Send System and the Receive system. (Namely, in the case where the file transfer between HULFT that specifies HULFT Encryption Scheme and the other specifies Encryption Scheme by the exit routine)

Specify an encryption type in the Encryption Scheme in the System Environment Settings file. Refer to "3.4 System Environment Settings" for the System Environment Settings file.

(1) HULFT Encryption Scheme

HULFT transfers a file by using HULFT Encryption Scheme, when a user sets the Encryption Scheme in the System Environment Settings file to enable HULFT Encryption Scheme and specifies the same Encryption Key both in the Send Management Information and the Receive Management Information.

When the settings of the Encryption Key on the sending side and the receiving side are not the same, or when only HULFT on the sending side specifies the Encryption Key, data is not decoded correctly.

Besides, when the setting of the Encryption Key on the sending side is disabled, data is not encoded. In such case, the Encryption Key of the receiving side is ignored.

(2) Encryption Scheme by Exit Routine

Because the interface is open to public as the encryption exit routine, users can incorporate any commercially available encryption logics or user-defined encryption logics into HULFT.

HULFT can transfer a file with user-defined the encryption scheme through the incorporation of an exit routine by specifying the Encryption Scheme in the System Environment Settings file so as to use other encryption scheme.

Shown below is the flow of the processing when using an encryption exit routine:

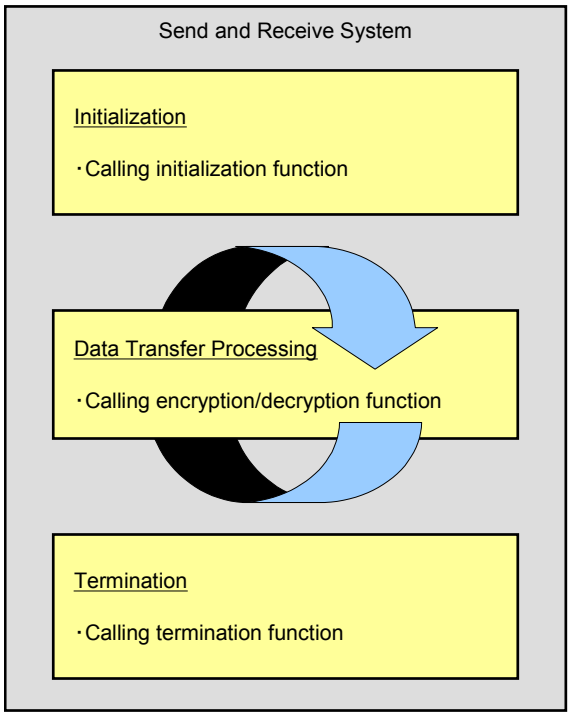


Figure 2.42 Encryption Exit Routine

Refer to the "3.10 Encryption Exit Routine" for the method of incorporating an encryption exit routine.

2.7.2 Verifying Consistency of Transfer Data

HULFT can verify whether the data received by the host on the receiving side and the data transferred from the host on the sending side consist or not in the Send and Receive systems. This function is called data verification.

The data verification function handles the data as an error, when the inconsistency in the data transferred from the sending side turns out through the data verification. In such case, HULFT notifies an error message to the host on the sending side.

The range of data that data verification function guarantees the consistency is the data from the beginning to the end of transferring. The data after transferring is not included within the range of the guarantee.

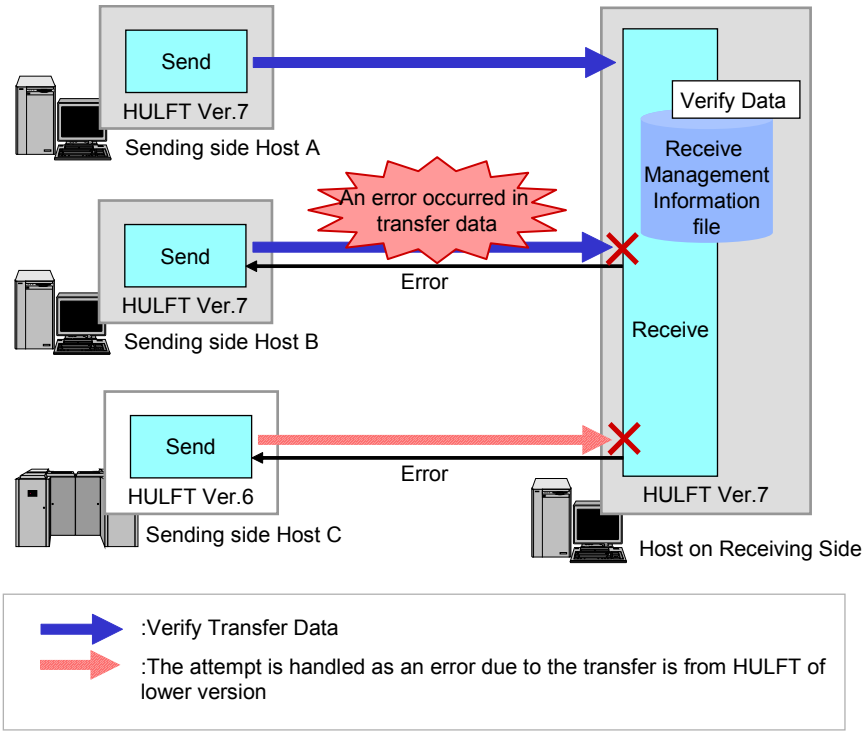


Figure 2.43 Data Verification Function

To use data verification function, specify '1' (Yes) for the Verify Data in the Receive Management Information that corresponds to the target file on the host on the receiving side. In operation to send the identical file to two or more hosts (multicasting), set the Verify Data in the Receive Management Information of all HULFTs of the hosts on the receiving side.

Refer to *Operation manual* for the Receive Management Information.

[Note] **To use data verification function, HULFTs both on the sending side and the receiving side are required to have installed HULFT Ver.7 or higher. If HULFT on the sending side is lower than Ver.7, the specification of the Verify Data in HULFT on the receiving side is handled as a transfer error.**

2.7.3 Request Acknowledge Host Check Function When Acknowledging Service Request

When a service request is acknowledged by the Request Acknowledge system, HULFT checks whether the request was issued by the host registered in the Host Information or not, which allows a user to refuse the connection from unintended hosts. This function is called requestor host check function.

The requestor host check function can be applied to the services shown below:

- Send Request (SEND)
- Resend Request (RESEND)
- Post-receive Job Result Inquiry Request (HULJOB)
- Job Execution Result Notification (HULSNDRC)
- Remote Job Execution (HULRJOB)

To use requestor host check function, enable the Request Acknowledge Host Check in the System Environment Settings of HULFT on the request acknowledge side. Refer to "3.4 System Environment Settings" for the details of the System Environment Settings.

This function is based on the assumption that the information on the host that you permit connection (Host Name and the like) is registered in the Host Information of HULFT. Refer to *Operation manual* for the details of the Host Information.

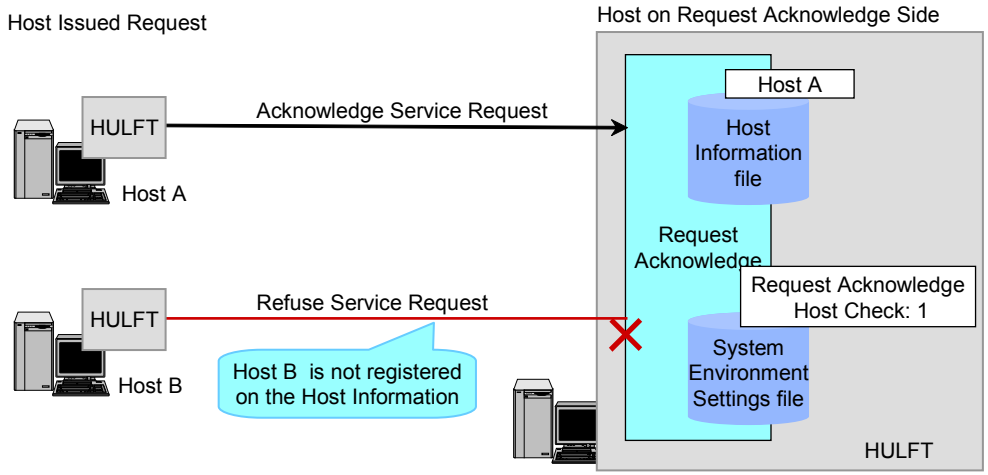


Figure 2.44 Requestor Host Check Function

In system operation, HULFT refuses to acknowledge the service request from the requestor hosts that are not registered in the Host Information, and handles the requests as errors.

In the example shown in the figure, HULFT refuses the service request from 'Host B' that is not registered in the Host Information.

[Remarks] The requestor host check function is even applied to the service request issued in the environment that the requestor host has installed HULFT lower than Ver.7.

2.7.4 Request Acknowledge Setting by Service Request

In acknowledgement of service requests by the Request Acknowledge system, you can set whether to grant acknowledgement by service types issued by the host registered in the Host Information of HULFT.

This function is called request acknowledge setting function. The request acknowledge setting function can realize the operation that HULFT acknowledges only the specified types of service request, even if the request is issued by the host that is permitted to connect, which allows a user to refuse the request from other services.

The request acknowledge setting function is applied to the services shown below:

- Send Request/Resend Request (SEND/RESEND)
- Post-receive Job Result Inquiry Request (HULJOB)
- Job Execution Result Notification (HULSNDRC)
- Remote Job Execution (HULRJOB)

The request acknowledge setting function is based on the assumption that the information on the host that you permit connection (Host Name and the like) is registered in the Host Information of HULFT. When you intend to refuse the request from the specified service, specify 'N' for the service name you intend to refuse in the Host Information.

Refer to *Operation manual* for the details of the Host Information.

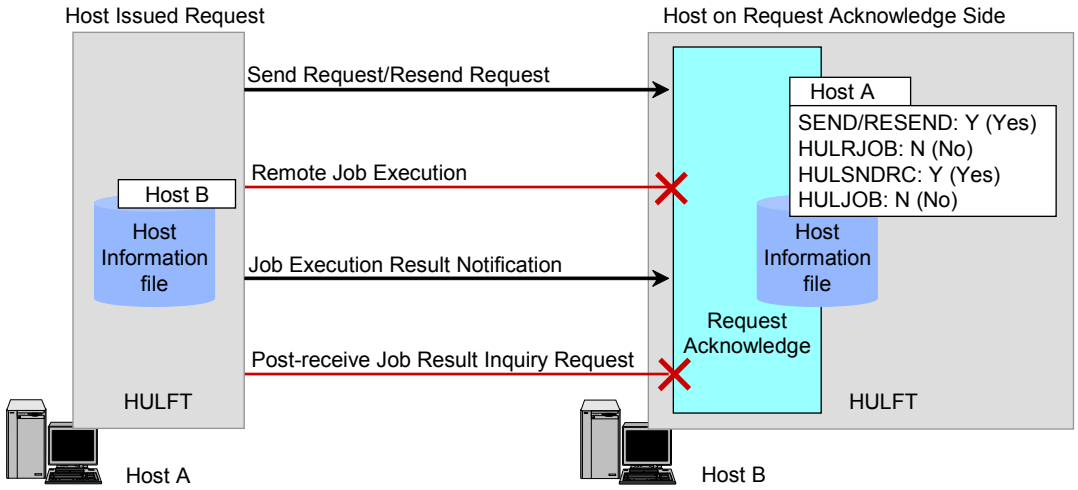


Figure 2.45 Request Acknowledge Setting Function

In system operation, HULFT refuses the acknowledgement of the service requests that are set 'N' in the Host Information and handles the request as an error.

In the example shown in the figure, HULFT refuses the connection from the services of the Remote Job Execution (HULRJOB) and the Post-receive Job Result Inquiry Request (HULJOB).

[Remarks] Request acknowledge setting function is even applied to the service request issued in the environment that host issued the request has installed HULFT lower than Ver.7.

2.7.5 Confirmation of Sending Host Based on Transfer Group ID

When receiving a Receive file in the Send and Receive systems, HULFT can check whether the host on the sending side is registered in the specified Transfer Group ID in the Receive Management Information or not. As a result, HULFT can block the sending of the intended file from the unintended hosts. This function is called transfer group check.

Refer to "4.3 Receiving from Authorized Host (Using Transfer Group ID)" for the necessary settings to use the Transfer Group ID and the Transfer Group Check function.

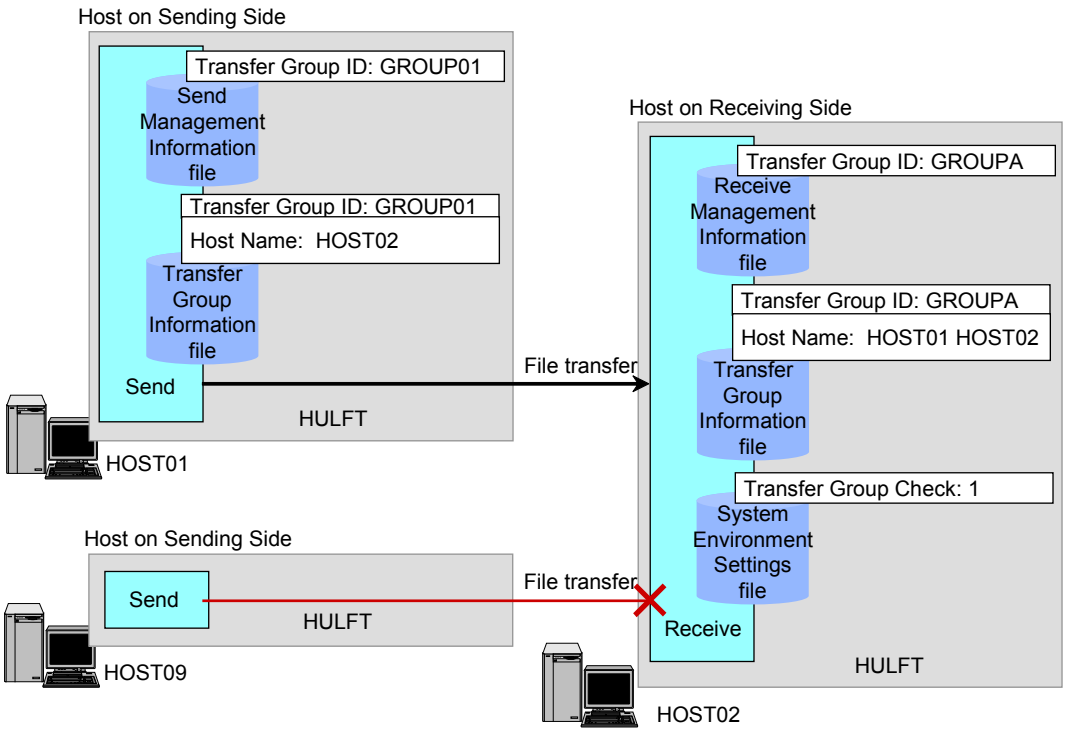


Figure 2.46 Transfer Group Check Function

In the example shown in above figure, HULFT refuses the file transfer from the host 'HOST09' that is not registered in the Transfer Group ID 'GROUPA' of the Transfer Group Information specified in the Receive Management Information.

Chapter 3

Environment Settings for Windows

This chapter explains HULFT environment settings of Windows side and the setting method of the system environment.

3.1 Operating Environment

This section explains HULFT environment.

3.1.1 Program Structure

The program structure of HULFT is shown in Table 3.1 to Table 3.3. The files in this structure are stored at the time of installation.

Table 3.1 HULFT Program Structure 1

Name	Install Output Destination	File Name
Management Screen		
Management screen DLL	binnt	hmifiles.dll
Management screen		HulWin.exe
Splash Window		hulftsp.bmp
Message File (for Management screen)		hulmsg.dat
Management screen DLL		Adbridge_DLL.dll
iconv DLL		iconv.dll *1
libxml2 DLL		libxml2.dll
log4net DLL		log4net.dll
Management screen DLL		OperationLog.dll
Management screen DLL		PluginExtension.dll
Management screen DLL		Saison.DF.dll
zlib1 DLL		zlib1.dll
Common Screen Resource File	binnt\sys	hullftUIRes_JP.xml
Common Message Resource File		hulPluginGUImsg_JP.xml
System File	binnt\sys\xml	hubadmin2.dat
System File		huladmin7.dat
System File		seq_huladmin7.dat
System File		structure.dat
Management screen DLL	binnt\en-US	HulWin.resources.dll
Manual File	binnt\WebHelp	
Error Definition File	HULPATH\errfile	NT.dat
Error Definition File		MAL.dat
Help		
Help File	binnt	help.exe
DLL (Common to 32-bit and 64-bit)		
HULFT API DLL	binnt	hulapi.dll
HULFT Common DLL		hulftrt.dll
DLL Offered by Microsoft	%windir%\WinSxS	msvcm80.dll
DLL Offered by Microsoft		msvcp80.dll
DLL Offered by Microsoft		msvcr80.dll
DLL Offered by Microsoft		mfc80.dll
DLL Offered by Microsoft		mfc80u.dll
DLL Offered by Microsoft		mfcm80.dll
DLL Offered by Microsoft		mfcm80u.dll
DLL(32-bit)		
HULFT Installation DLL	binnt	hulinst.dll
HULFT XML DLL		hulexgxml.dll
DLL(64-bit)		
HULFT API DLL	binnt	hulapi64.dll
HULFT Common DLL		hulftrt64.dll
HULFT Installation DLL		hulinst64.dll
HULFT XML DLL		hulexgxml64.dll
DLL Offered by Microsoft	%windir%\WinSxS	msvcm80.dll
DLL Offered by Microsoft		msvcp80.dll
DLL Offered by Microsoft		msvcr80.dll
DLL Offered by Microsoft		mfc80.dll
DLL Offered by Microsoft		mfc80u.dll
DLL Offered by Microsoft		mfcm80.dll
DLL Offered by Microsoft		mfcm80u.dll
Readme		
HULFT Windows Readme	HULPATH	hulft7e.readme.win.txt

*Available only in 32-bit edition.

Table 3.2 HULFT Program Structure 2

Name	Install Output Destination	File Name
Common Commands		
Send File Command/Resend File Command	binnt	utlsend.exe
Send Request Command/Resend Request Command Receive Ready Notification Command		utlrecv.exe
Management Information Batch Registration Command		utliupdt.exe
Management Information Deletetion Command		utlirm.exe
Management Information Record Deletion Command		utlcomp.exe
Management Information Parameter File Generation Command		utligen.exe
Format Information Display Command		utlilist.exe
Management Information Related Display Command		utlidlist.exe
Send and Receive Log List Display Command		utllist.exe
Request Acknowledge Log List Display Command		utlobslist.exe
Send and Receive Log Deletion Command		utllog.exe
Request Acknowledge Log Deletion Command		utlobsrm.exe
Multiple File Join Command		utljoin.exe
Joined File Break Command		utlbreak.exe
Joined File Display Command		utldspfil.exe
File Record Edit Command		utllf.exe
Multiple Receive File Edit Multiple Receive File Data Delete Command Multiple Receive File Data Extract Command Multiple Receive Information File List Display Command		utlstore.exe
External Character Table Expand Command		utlgtfextdV.exe
Unicode Conversion Command		utlchgunicode.exe
Management Information Conversion Command		hulconv.exe
Transfer Summary Command		utllogcnt.exe
Main (Service Main Module)		hulsrvc.exe
Send Main Module		hulsdd.exe
Request Acknowledge Main Module		hulobs.exe
Receive Main Module		hulrcv.exe
Send Cancellation Command		utlscan.exe
Send Status Display Command		utlsdisp.exe
Resend Queue Status List Deletion Command		utlrsdrm.exe
Unsent Status Queue Change Command		utlschange.exe
Job Execution Result Notification Command		hulsndrc.exe
Remote Job Execution Command		utlrjob.exe
Receive Cancellation Command		utlrcan.exe
Receive Status Display Command		utlrdisp.exe
Alert Notification Utility Alert Notification Command Alert Resend Command		utlalert.exe
Service Registration/Deletion Command		utlservice.exe
HULFT Startup/Termination Command		utlsvcctl.exe
Debug Mode Change Command		utldebug.exe
Remote Job Execution Command		hulrjob.exe
Mail Interface Module		utlsendmail.exe
Operation Log List Display File Access Log List Display Command Command Execution Log List Display Command		utlopllist.exe
Operation Log Manual Switch Command		utloplchg.exe

Table 3.3 HULFT Program Structure 3

Name	Install Output Destination	File Name
Environment Settings		
HULFT System Environment Settings File	binnt	hulft.ini
System Environment Settings File	HULPATH	hulenv.cnf
Message File (for Main Module)		hulmsg.fil
Request Acknowledge Information File		service.dat
HULFT System File		hulft.sys
External Character Table (EUC to SJIS)	HULPATH\gtf	gtetos.xtd
External Character Table (IBM to SJIS)		gtitos.xtd
External Character Table (JEF to SJIS)		gtjtos.xtd
External Character Table (KEIS to SJIS)		gtktos.xtd
External Character Table (SJIS to EUC)		gtstoe.xtd
External Character Table (SJIS to IBM)		gtstoi.xtd
External Character Table (SJIS to JEF)		gtstoj.xtd
External Character Table (SJIS to KEIS)		gtstok.xtd
External Character Table (SJIS to NEC)		gtston.xtd
External character table (NEC to SJIS)		gtntos.xtd
External Character Table (SJIS to UTF-8)		gtsto8.xtd
External Character Table (UTF-8 to SJIS)		gt8tos.xtd
External Character Table Template File (EUC to SJIS)		ktetos.txt
External Character Table Template File (IBM to SJIS)		ktitos.txt
External Character Table Template File (JEF to SJIS)		ktjtos.txt
External Character Table Template File (KEIS to SJIS)		ktktos.txt
External Character Table Template File (SJIS to EUC)		ktstoe.txt
External Character Table Template File (SJIS to IBM)		ktstoi.txt
External Character Table Template File (SJIS to JEF)		ktstoj.txt
External Character Table Template File (SJIS to KEIS)		ktstok.txt
External Character Table Template File (SJIS to NEC)		ktston.txt
External Character Table Template File (NEC to SJIS)		ktntos.txt
External Character Table Template File (SJIS to UTF-8)		ktsto8.txt
External Character Table Template File (UTF-8 to SJIS)		kt8tos.txt
Make File for External Character Conversion		gaiji.mak
Library for External Character Converter		utlgtfextd.lib
External Character Table Template Sample File (SJIS to UTF-8)		sjistoutf8-34kanji.txt
External Character Table Template Sample File (UTF-8 to SJIS)		utf8tosjis-34kanji.txt
Conversion Template ASCII (ASCII to EBCDIC)	HULPATH\ucf	AS2EBAC.UCF
Conversion Template ASPEN (ASCII to EBCDIC)		AS2EBAP.UCF
Conversion Template Lowercase (ASCII to EBCDIC)		AS2EBEL.UCF
Conversion Template Standard (ASCII to EBCDIC)		AS2EBIBM.UCF
Conversion Template Kana Characters (ASCII to EBCDIC)		AS2EBKN.UCF
Conversion Template Standard Extend (ASCII to EBCDIC)		AS2EBEXT.UCF
Conversion Template NEC Kana (ASCII to EBCDIC)		AS2EBNEC.UCF
Conversion Template ASCII (EBCDIC to ASCII)		EB2ASAC.UCF
Conversion Template ASPEN (EBCDIC to ASCII)		EB2ASAP.UCF
Conversion Template Lowercase (EBCDIC to ASCII)		EB2ASEL.UCF
Conversion Template Standard (EBCDIC to ASCII)		EB2ASIBM.UCF
Conversion Template Kana Characters (EBCDIC to ASCII)		EB2ASKN.UCF
Conversion Template Standard Extend (EBCDIC to ASCII)		EB2ASEXT.UCF
Conversion Template NEC Kana (EBCDIC to ASCII)		EB2ASNEC.UCF

* Conversion template indicates the EBCDIC code set conversion template

3.1.2 File Structure

HULFT files consist of the system files and the log files. You can operate HULFT even if you are not aware of the file names.

(1) System Files

The system files are shown in Table 3.4 to Table 3.6.

[Remarks] *: Environment Variables and tag names used by Path.

- HULPATH is a directory where the environment settings files used by HULFT are stored. For details, refer to "3.2.1 Directory of HULFT Environment Settings Files."

Table 3.4 Types of System Files1

Name	Path*	Main File Name	Sub File Name
Management Information			
System Environment Settings File	HULPATH	hulenv.cnf	hulenv.bkup.cnf
Request Acknowledge Definition File	HULPATH	service.dat	service.bkup.dat
Account File	HULPATH	psaccount.dat	psaccount.bkup.dat
Mail Environment Settings File	HULPATH\mail	Sendmail.ini	Sendmail.bkup.ini
Send Management Information File	HULPATH	hulsnddb.dat	hulsnddb.bkup.dat
	HULPATH	hulsnddb.idx	hulsnddb.bkup.idx
Receive Management Information File	HULPATH	hulrcvdb.dat	hulrcvdb.bkup.dat
	HULPATH	hulrcvdb.idx	hulrcvdb.bkup.idx
Host Information File	HULPATH	hulhstdb.dat	hulhstdb.bkup.dat
	HULPATH	hulhstdb.idx	hulhstdb.bkup.idx
Transfer Group Information File	HULPATH	hulrhtdb.dat	hulrhtdb.bkup.dat
	HULPATH	hulrhtdb.idx	hulrhtdb.bkup.idx
Job Information File	HULPATH	hulexedb.dat	hulexedb.bkup.dat
	HULPATH	hulexedb.idx	hulexedb.bkup.idx
Format Information File	HULPATH	hulfmtdb.dat	hulfmtdb.bkup.dat
	HULPATH	hulfmtdb.idx	hulfmtdb.bkup.idx
	HULPATH	FormatID.fmt	FormatID.bkup.fmt
Multi Format Information File	HULPATH	hulmfmtdb.dat	hulmfmtdb.bkup.dat
	HULPATH	hulmfmtdb.idx	hulmfmtdb.bkup.idx
CSV Environment Settings File	HULPATH	hulcsv.inf	hulcsv.bkup.inf
Password Management File	HULPATH	huladmin.dat	huladmin.bkup.dat
Mail Interface Information File	HULPATH\mail	Sendmail.lst	Sendmail.bkup.lst
	HULPATH\mail	MailID.mal	MailID.bkup.mal
Addressbook File	HULPATH\mail	Sendmail.adr	Sendmail.bkup.adr
XML Environment Settings File	HULPATH\xml	<host name>.<format segment field>.<format ID field>.inf	<host name>.<format segment field>.<format ID field>.bkup.inf
Generation File			
Generation File Information File	HULPATH	<file ID>.info	<file ID>.bkup.info

Table 3.5 Types of System File 2

Name	Path*	Main File Name	Sub File Name
External Character File			
SJIS to IBM	HULPATH\gtf	gtstoi.xtd	gtstoi.bkup.xtd
SJIS to KEIS	HULPATH\gtf	gtstok.xtd	gtstok.bkup.xtd
SJIS to JEF	HULPATH\gtf	gtstoj.xtd	gtstoj.bkup.xtd
SJIS to EUC	HULPATH\gtf	gtstoe.xtd	gtstoe.bkup.xtd
SJIS to NEC	HULPATH\gtf	gtston.xtd	gtston.bkup.xtd
SJIS to UTF-8	HULPATH\gtf	gtsto8.xtd	gtsto8.bkup.xtd
KEIS to SJIS	HULPATH\gtf	gtktos.xtd	gtktos.bkup.xtd
EUC to SJIS	HULPATH\gtf	gtetos.xtd	gtetos.bkup.xtd
IBM to SJIS	HULPATH\gtf	gtitos.xtd	gtitos.bkup.xtd
JEF to SJIS	HULPATH\gtf	gtjtos.xtd	gtjtos.bkup.xtd
NEC to SJIS	HULPATH\gtf	gtntos.xtd	gtntos.bkup.xtd
UTF-8 to SJIS	HULPATH\gtf	gt8tos.xtd	gt8tos.bkup.xtd
EBCDIC Code Set Conversion Template			
Kana Characters (EBCDIC to ASCII)	HULPATH\ucf	EB2ASKN.UCF	
IBM Standard (EBCDIC to ASCII)	HULPATH\ucf	EB2ASIBM.UCF	
Lowercase (EBCDIC to ASCII)	HULPATH\ucf	EB2ASEL.UCF	
ASPEN (EBCDIC to ASCII)	HULPATH\ucf	EB2ASAP.UCF	
ASCII (EBCDIC to ASCII)	HULPATH\ucf	EB2ASAC.UCF	
IBM Standard Extension (EBCDIC to ASCII)	HULPATH\ucf	EB2ASEXT.UCF	
NEC Kana (EBCDIC to ASCII)	HULPATH\ucf	EB2ASNEC.UCF	
Kana Characters (ASCII to EBCDIC)	HULPATH\ucf	AS2EBKN.UCF	
IBM Standard (ASCII to EBCDIC)	HULPATH\ucf	AS2EBIBM.UCF	
ASPEN (ASCII to EBCDIC)	HULPATH\ucf	AS2EBAP.UCF	
Lowercase (ASCII to EBCDIC)	HULPATH\ucf	AS2EBEL.UCF	
ASCII (ASCII to EBCDIC)	HULPATH\ucf	AS2EBAC.UCF	
IBM Standard Extension (ASCII to EBCDIC)	HULPATH\ucf	AS2EBEXT.UCF	
NEC Kana (ASCII to EBCDIC)	HULPATH\ucf	AS2EBNEC.UCF	
EBCDIC user table 1	HULPATH	user1.ucf	user1.bkup.ucf
EBCDIC user table 2	HULPATH	user2.ucf	user2.bkup.ucf
EBCDIC user table 3	HULPATH	user3.ucf	user3.bkup.ucf

Table 3.6 Types of System File 3

Name		Path*	Main File Name	Sub File Name
Temporary				
	Send Log Deletion Temporary File	HULPATH	hulsndlg.dat.tmp.tid.pid	
	Receive Log Deletion Temporary File	HULPATH	hulrcvlg.dat.tmp.tid.pid	
	Send Job Log Deletion Temporary File	HULPATH	hulexlgs.dat.tmp.tid.pid	
	Receive Job Log Deletion Temporary File	HULPATH	hulexlgr.dat.tmp.tid.pid	
	Request Acknowledge Log Deletion Temporary File	HULPATH	hulobslg.dat.tmp.tid.pid	
		HULPATH	hulschlg.dat.tmp.tid.pid	
	Multiple Receive Temporary File	Temporary path	rcvtmp.<receive start time>tid.tmp.pid	
	Generation Receive Temporary File	Receive file folder	rcvtmp.<receive start time>tid.tmp.pid	
	Backup File for Restoration	Receive file folder	<receive file name>.tmp .tid.pid	
	Format Information Registration Temporary File	HULPATH	fmttmp.<file ID>. tid.pid.<any value>	
	Temporary File for hulenv.cnf Update	HULPATH	sysenvtmp.tid.pid	
	Temporary File for utlcomp	HULPATH	tmp_file.dat	
		HULPATH	tmp_file.idx	
	utlalert Temporary File	Temporary path	alerttmp.pid	
	File during Send Process Operation	HULPATH	hulsdd.act	
	File during Receive Process Operation	HULPATH	hulrcv.act	
	File during Request Acknowledge Process Operation	HULPATH	hulobs.act	
	utlstore Receive Information Registration Temporary File	HULPATH	<receive file ID>.str.PID	
	utlstore Receive File Temporary File	Receive file folder	<receive file name>.PID	
	XML Conversion Information Temporary File	HULPATH\xml	<host name>.<format segment field>.<format ID field>.inf.tmp	
Others				
	Send Control File	HULPATH	sddreqcp.dat	
	Resend Queue File	HULPATH	sddreqls.dat	
	Processing Status Storage File	HULPATH	hulconsole1.log	
	Console Log	binnt	hulcon.log	
	HULFT System Environment Settings File	binnt	hulft.ini	
	Management Screen Environment File	binnt	hulwin.ini	
	HULFT System File	HULPATH	hulft.sys	
	HULFT Message File	HULPATH	hulmsg.fil	
	Error Definition File	HULPATH\errfile	NT.dat	
	Error Definition File	HULPATH\errfile	MAL.dat	
	utlalert Resend Message File	Not fixed	Not fixed	

(2) Log File

Log files are shown in Table 3.7.

Table 3.7 Type of Log Fles

Name		Path*	Main File Name	Sub File Name
Logs				
	Send Log	HULPATH	hulsndlg.dat	hulsndlg.bkup.dat
	Send Log (switch)	HULPATH	hulsndlg.sw.dat	hulsndlg.sw.bkup.dat
	Post-send Job Execution Log	HULPATH	hulexlgs.dat	hulexlgs.bkup.dat
	Post-send Job Execution Log (switch)	HULPATH	hulexlgs.sw.dat	hulexlgs.sw.bkup.dat
	Receive Log	HULPATH	hulrcvlg.dat	hulrcvlg.bkup.dat
	Receive Log (switch)	HULPATH	hulrcvlg.sw.dat	hulrcvlg.sw.bkup.dat
	Post-receive Job Execution Log	HULPATH	hulexlgr.dat	hulexlgr.bkup.dat
	Post-receive Job Execution Log (switch)	HULPATH	hulexlgr.sw.dat	hulexlgr.sw.bkup.dat
	Request Acknowledge Log	HULPATH	hulobslg.dat	hulobslg.bkup.dat
	Request Acknowledge Log (switch)	HULPATH	hulobslg.sw.dat	hulobslg.sw.bkup.dat
	Job Notification Log (hulsndrc.dat)	Not fixed	Not fixed (default: hulsndrc.dat)	<main file name>.bkup (default: hulsndrc.dat.bkup)
	hulrjob Execution Log (joblog.dat) for Server	Not fixed	Not fixed (default: joblog.dat)	<main file name>.bkup (default: joblog.dat.bkup)
	utlrjob Execution Log (joblog.dat) for Client	Not fixed	Not fixed	<main file name>.bkup
	utlalert Execution Log	HULPATH	Not fixed	<main file name>.bkup
Receive Information				
	Multiple Receive Information File	HULPATH	<file ID>.str	<file ID>.bkup.str

(3) Operation Log File

Operation log files are shown in Table 3.8.

Table 3.8 Type of Operation Log Files

Name		Path*	Main File Name	Sub File Name
Operation Logs				
	File Access Log File	HULPATH\opl	huloplfile.csv	huloplfile.bkup.csv
	Command Execution Log File	HULPATH\opl	huloplcmd.csv	huloplcmd.bkup.csv
	File Access Log Switch Generation Management File	HULPATH\opl	huloplfile.info	huloplfile.bkup.info
	Command Execution Log Switch Generation Management File	HULPATH\opl	huloplcmd.info	huloplcmd.bkup.info

3.2 Environment Settings

This section explains the methods of setting HULFT environment.

The explanation provided in this section assumes that HULFT is already installed.

3.2.1 Directory of HULFT Environment Settings Files

Set the directory where the environmental settings files used by HULFT exist in the 'hulft.ini.' The section name is 'PATH', and the entry name is 'HULPATH'. At the time of installation, the installation program automatically set these. If you wish to change the settings, modify the 'hulft.ini' file, by using an editor such as Notepad.

<Description example>

[PATH]

```
HULPATH=C:\HULFT Family\hulft7\etc
```

[Remarks] Explanation of HULPATH

- HULPATH is the location where the management information necessary for HULFT Send and Receive operations is stored.

3.2.2 Access Permission Settings

For the following files and directories, set the access permission of HULFT operation account.

* HULFT operation account means the account registered in account settings of the Extension Settings tab of System Environment Settings dialog box (this can be displayed by the System Environment Settings command from the [System Configuration] menu on HULFT Management screen). When it is not registered, the default value is SYSTEM account.

- Service Execution
  Execution permissions
- Work File Generation Directory
  Read and write permissions
- Send and Receive Files Storage Directory and Send and Receive Files
  Read and write permissions
- All the Job Execution Application (if any job is executed)
  Execution permissions
- Directories under HULPATH (set in hulft.ini)
  Read and write permissions
- Directories under the Installation folder
  Read, write and execution permissions

3.3 Send and Receive Files

This section explains about files which can be sent and received with HULFT on Windows.

(1) Record Size

For text transfer, the maximum record length (up to carriage return) is 32,767 bytes excluding the carriage return.

Even if the file does not exceed the maximum size mentioned above, it cannot be handled if the file size exceeds 32,767 bytes during Kanji code conversion.

For format transfer and multi format transfer, the maximum length of a single record should be 19,997 bytes.

(2) File Size

There are no limitations for the file size. However, there are cases where Send and Receive operations cannot be carried out due to the restrictions on file size by the operating system.

(3)Format Field Size and the Number of Fields

The lengths of one field at Format Transfer and Multi-Format Transfer are as follows:

The number of fields that can be registered on single record is up to 1000 fields.

Table 3.9 Format Field Size

Field type	Size (bytes)	Remarks
X	1 to 9999	
M	1 to 9999	
N	2 to 9998	Even values only can be handled
9	1 to 18	
B	2, 4, 8	
P	1 to 10	Up to 18 digits can be handled
S	1 to 18	
F	4, 8	
I	1 to 9999	

(4) Sign of Signed External Decimal

The followings are handled as the sign of signed external decimal.

3, 4, 5, 7, A, B, C, D, E, F

(5) Network Resource

File that uses the network resource cannot be sent or received.

(6) Shortcut

Shortcut files cannot be sent or received.

(7) File Name

Specify the file name by the complete path.

3.4 System Environment Settings

The System Environment Settings file is a file to set the environment of HULFT system. Change the default contents set during installation to suit the system environment you use.

There are two methods in setting as shown below:

- Editing the file directly by using an editor program
- Setting the fields interactively by using the Management screen

In this manual, the setup method by using an editor is described. Refer to *Operation Manual* for the method of setting by the Management screen

3.4.1 Method of Settings

Set the system environment of HULFT according to the following procedure:

(1) Exit from HULFT service and the Management screen.

(2) Start a text editor, and open the System Environment Settings file.

(3) Edit the System Environment Settings file and save it.

Refer to "3.4.3 System Environment Settings List" for the details of fields in the System Environment Settings file.

(4)Restart HULFT service.

HULFT loads the System Environment Settings file upon starting.

To reflect the modification in the System Environment Settings file, exit from HULFT service to modify the file, and then restart the service.

If the Management screen has started, it is necessary to restart the screen as well.

3.4.2 System Environment Settings File

(1) Types of System Environment Setting

The following types of System Environment Settings for each functions are available.

- Basic Settings
- Extension Settings
- Mail Interface
- Other

(2) File Name

The system environment is set in a file according to the function. The names of these files cannot be changed.

The storage directories should correspond to HULPATH set in 'hulft.ini.'

- System Environment Settings file: HULPATH\hulenv.cnf

  The functions of the Basic Settings, the Extension Settings and the Other are set.

- Request Acknowledge Definition file: HULPATH\service.dat

  Request Acknowledge Log Output settings in the Basic Settings are set. For details, refer to "3.7.1 Request Acknowledge Definition File."

- Account Setup file: HULPATH\psaccount.dat

  Account settings in the Extension Settings are set.

- Mail Environment Settings file: HULPATH\Mail\sendmail.ini

  The Mail Interface function is set.

(3) Description Format

a) System Environment Settings file (hulenv.cnf)

It is considered as a comment line when a single byte number sign (#) exists at the beginning of the line. One or more blank lines are accepted. The line where a field is set should be described in the following format.

```
tag_name = set_value
```

- The tag name represent the name of the field to be set and uses a unique name for each field. The tag names are fixed and cannot be changed.
- Use a single byte equal sign (=) to separate the tag name and the set value. Spaces before and after the equal sign are acceptable.
- A carriage return code (0x0d0a) needs to be added after the set value.

b) Mail Environment Settings file (sendmail.ini)

It is considered as a comment line when a single byte semicolon (;) exists at the beginning of the line. One or more blank lines are accepted. The line where a field is set should be described in the following format.

```
tag_name = set_value
```

- The tag name represent the name of the field to be set and uses a unique name for each field. The tag names are fixed and cannot be changed.
- Use a single byte equal sign (=) to separate the tag name and the set value. Spaces before and after the equal sign are acceptable.

c) Request Acknowledge Definition file (service.dat)

For details, refer to "3.7.1 Request Acknowledge Definition File."

d) Account settings file (psaccount.dat)

This file can be set only on the Management screen or via HULFT Manager. For details, refer to *Operation Manual* or Online Help of HULFT Manager.

3.4.3 System Environment Settings List

Table 3.10 List of System Environment Settings 1

Field Name	Tag Name	Default Value	Setting Value	On-screen Setting	Remarks
Basic Settings					
Work File Generation Path	tmpdir	.	Characters	✓	within 50 bytes
Receive Port No.	rcvport	30000	from 1 to 65535	✓	
Request Acknowledge Port No.	obsport	31000	from 1 to 65535	✓	
Command Acknowledge Port No.	sddport	65535	from 1 to 65535	✓	
Service Port No.	srvport	65533	from 1 to 65535	✓	
Send Process Multiplex Level	sndpsnum	10	from 1 to 999	✓	
Receive Process Multiplex Level	rcvpsnum	999	from 1 to 999	✓	
JIS Year	jistype	83	78,83	✓	
Space Character Mode	spcode	1	0,1	✓	
Tab Code Mode	tabchange	0	0,1	✓	
Automatic Process Startup	autostart	111		✓	0,1
Receive File Max. Size	rcvmaxfilesize	*1	from 1 to *1	✓	Bytes
Request Acknowledge Log Output	-	1	0,1	✓	
Extension Settings					
Retry Count on Locked Send File	lockcnt	0	from 0 to 99999	✓	
Retry Interval on Locked Send File	locktime	0	from 0 to 99999	✓	
Retry Count on Locked Receive File	rretrycnt	0	from 0 to 99999	✓	
Retry Interval on Locked Receive File	rretrytime	0	from 0 to 99999	✓	
Connect Retry Count	retrycnt	10	from 0 to 99999	✓	
Connect Retry Interval	retrytime	1	from 0 to 99999	✓	
Socket Read Timeout	socktime	3600	from 10 to 259200	✓	
Socket Buffer Size	socksize	12288	0, from 4096 to 65520	✓	
Console Log Size	logsize	999999	from 1024 to 102400	✓	999999:unlimited
Console Log Backup	backuplog	1	0,1	✓	
Job Timeout	jobwtimeout	999999	from 10 to 259200	✓	999999:unlimited
Clear Unsent File	delreqcpfile	0	0,1	✓	
Clear Resend Queue File	delresendfile	0	0,1	✓	
Output to Event Logs	eventlog	000	0,1	✓	
Account Name	-			✓	*3
Password	-			✓	*3
Domain Name	-			✓	*3
Extension2 Settings					
Status Display Selection	statusdisplay	1	0,1	✓	
Pack Zone Conversion	pschg	0	0,1,B,F,9,S,N	✓	
Sign for ASCII	psascii	1	1,2	✓	
Sign for EBCDIC	psebcdic	1	1,2	✓	
Auto Resend Retry Count	sockerr_autoretry	0	from 0 to 9	✓	
Log Switch Count	logdelcount	0	from 0 to 999999	✓	
Send Transfer Error Recovery	enderrmode	0	0,1	✓	*2
External Character Table Mode	gaijifile	0	0,1,2	✓	
Process when External character has not been Registered	gaijierr	0	0,1	✓	
Receive Multiplex Level Over Retry	rcvover_rty	0	0,1	✓	
KEIS Em-size Space Mode	keisspmode	0	0,1	✓	
Criteria to Delete Resend Queue	resenddel	1	1,2	✓	
HULFT-SAN V5 Product	san_ver5	H	F,H	✓	
Process Auto Reboot Max. Count	procrestartcnt	0	from 0 to 99	✓	
Process Auto Reboot Retry Interval	procrestarttime	60	from 0 to 99	✓	
Request Acknowledge Host Check	obshstchk	0	0,1	✓	
Framed Message Transfer Type	proctranstype	A	N,S,R,A	✓	
Code Conversion Mode	codechangemode	0	0,1	✓	

*1 9,223,372,036,854,775,808 (The 63rd power of two)

*2 '1' is set for new installation

*3 Only on-screen setting is available.

Table 3.11 List of System Environment Settings 2

Field Name	Tag Name	Default Value	Setting Value	On-screen Setting	Remarks
Others					
Password Check	password	0	0,1		
Management Screen Security	admcheck	0	0,1		
Retry Count on Send File	sretrycnt	10	from 0 to 99999		
Retry Interval on Send File	sretrytime	1	from 0 to 99999		
Dynamic Parameter Specification	dynparam	0	0,1		
Message Dynamic Parameter Specification	msgdynparam	0	0,1,2,3		
Send Unit Selection	sndpsend	1	0,1		
Service Connect Timeout	srvtime	3600	from 1 to 65535		
IP Version	ipversion	4	4,4/6,6/4,6		
Transfer Group Check	tgrpchk	0	0,1,2		
Transfer Status Transfer Count	disptranscnt	100	from 1 to 100		
Encryption Scheme	ciphertype	0	0,1		
X Type Conversion	xmode	0	0,1		
HULFT Manager Timeout	admsocktime	3600	from 10 to 259200		
Receive File Lock Standby	rcvfilelockwait	1	0,1		
Queue Of Pending Connect	listen	5	from 1 to 512		
Local Host Name	myhostname	*4			Within 68 bytes
Receive Ready Notification	rcvcmd	*4			
Operation Log Output Option	oplselect	0	0,1,2,3		
Operation Log Auto Switch Size	oplchangesize	0	from 0 to 9999		0: unlimited
Operation Log Switch Generation Count	oplgenerationcount	9999	from 2 to 9999		
Operation Log User-specified Character	oplcharacter		Alphanumeric Characters		Within 8 bytes
Operation Log User ID Output Format	opluseridtype	0	0,1		1 byte

*4 No tag is applied upon installation

Table 3.12 List of System Environment Settings 3

Field Name	Tag Name	Default Value	Setting Value	On-screen Setting	Remarks
Mail Interface					
Mail Account Name	User			✓	
Mail Server Host Name	MailHost			✓	
SMTP Port No.	SmtpPort		from 1 to 65535	✓	
Full Name	FullName			✓	
Domain Name	DomainName			✓	

3.4.4 Explanation of Each Field

This section explains about each field of the System Environment Settings. Change the default contents set during installation to suit the system environment you use. Tag name is given within brackets.

Each field is explained in relation to System Environment Settings screen of the Management screen. Refer to *Operation Manual* for the System Environment Settings screen. The default value will be used when omitted.

<Basic Settings: System Environment Settings file>

[Remarks] < XXX:YYY > XXX indicates a category of System Environment Settings. Regarding a field that end users can configure on the Management screen, you can set a value with a tab that bears identical name to the category. HULFT saves such values in YYY.

Work File Generation Path (tmpdir)

The path storing HULFT temporary files is specified.

There are cases where HULFT temporarily processes send and receive data and creates a work file. Therefore, check whether there is sufficient free disk space in the specified drive.

Receive Port No. (rcvport)

Specifies the port number used by the Receive process when accepting data sent from another host. Make sure this port number does not overlap with the port number reserved by the operating system or the port number used by other application programs.

Request Acknowledge Port No. (obsport)

Specifies the port number used by the Request Acknowledge process when accepting a request from another host. Make sure this port number does not overlap with the port number reserved by the operating system or the port number use by other application programs.

Command Acknowledge Port No. (sddport)

Specifies the port number to be used when accepting Send File and Resend File.

Make sure this port number does not overlap with the port number reserved by the operating system or the port number use by other application programs.

Service Port No. (srvport)

Specifies the port number used by HULFT program internally when accepting a request from another host.

Make sure this port number does not overlap with the port number reserved by the operating system or the port number use by other application programs.

Send Process Multiplex Level (sndpsnum)

Specify the maximum number of Send threads started simultaneously by the Send process. In the Send process, when number of the simultaneous send hosts increases, load on the system or communication lines also increases. Therefore, this option should be set in line with the condition of the system currently in use.

Receive Process Multiplex Level (rcvpsnum)

Specify the maximum number of the Receive threads that start at the same time. The number of threads that can be received cannot exceed this value. During the Receive process, when the number of simultaneous receive hosts are increased, load on the system or communication lines also increases. Therefore, this option should be set in line with the condition of the system currently in use.

JIS Year (jistype)

Specify JIS year of the Kanji code of the system in use.

78: 78 JIS
83: 83 JIS

Space Character Mode (spcode)

This field specifies the conversion rule of space code.

0: Mode 0
1: Mode 1

For conversion rule of the space code, refer to the "2.5.9 Rules of Space Code Conversion."

Tab Code Mode (tabchange)

When text transfer to a EBCDIC host is carried out at the sending side, whether to handle the tab code (0x09) as a space code (0x20) or as a tab code (0x09) is specified.

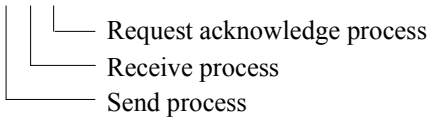
0: Handle as a space code
1: Handle as a tab code

Automatic Process Startup (autostart)

Each process is activated upon starting up HULFT Service. Whether to start a process automatically or not can be set process by process, by specifying 0 or 1 in a 3-digit field. The specification of '1' for the certain digit starts the specified process automatically, while setting '0' to a certain field does not start the process. From the left, the three-digit number represents 'Send process,' 'Receive process,' and 'Request Acknowledge process,' respectively.

0: Disable automatic startup
1: Enable automatic startup

<Example> autostart = 1 0 1



The specification as above starts the 'Send process' and the 'Request Acknowledge process' upon the startup of HULFT service, yet the application does not start the 'Receive process.'

Receive File Max. Size (rcvmaxfilesize)

The maximum size of the file which can be received in the Receive process is specified. Depending on the supported OS, there are some host types that do not support transferring of file that exceeds 2 to the power of 31 bytes. When file transfer is carried out with such hosts, specify the maximum size supported by that host.

<Basic Settings: Request Acknowledge Definition file>

Request Acknowledge Log Output

Specify whether the Request Acknowledge Log will be output or not for each service to be acknowledged.

The following service names can be specified.

SEND:	Send Request
RESEND:	Resend Request
HULADMIN:	Connection request from HULFT Manager
HULSNDRC:	Job status notification request
HULJOB:	Job Monitor Request
HULRJOB:	Remote Job Execution

<Extension Settings: System Environment Settings file>

Retry Count on Locked Send File (lockcnt)

Specify the number of time reopening is tried when the Send file is used by another processes or application programs.

When '0' is specified, no retry will be carried out.

Retry Interval on Locked Send File (locktime)

Specify the interval in seconds until the next reopning attempt when opening of the Send file fails.

Retry Count on Locked Receive File (rretrycnt)

Specify the number of times reopening is tried when the Receive file is used by another processes or application programs.

When '0' is specified, no retry will be carried out.

[Note] Only when the Receive file is used by a process other than HULFT, the settings of Retry Count on Locked Receive File and Retry Interval on Locked Receive File become valid. When the Receive file is used by any HULFT process, the setting of Receive File Lock Standby takes precedence over the settings of Retry Count on Locked Receive File and Retry Interval on Locked Receive File. Refer to the explanations of Retry Interval on Locked Receive File and Receive File Lock Standby.

Retry Interval on Locked Receive File (rretrytime)

Specify the interval in seconds until the next reopning attempt when opening of the Receive file fails.

[Note] Only when the Receive file is used by a process other than HULFT, the settings of Retry Count on Locked Receive File and Retry Interval on Locked Receive File become valid. When the Receive file is used by any HULFT process, the setting of Receive File Lock Standby takes precedence over the settings of Retry Count on Locked Receive File and Retry Interval on Locked Receive File. Refer to the explanations of Retry Count on Locked Receive File and Receive File Lock Standby.

Connect Retry Count (retrycnt)

Reconnect retry count is specified in case the socket connection fails.

When '0' is specified, reconnection is not carried out.

Connect Retry Interval (retrytime)

This field specifies the retry interval for reconnection where the socket connection fails by seconds.

Socket Read Timeout (socktime)

This field specifies a timeout interval in seconds where there is no response from remote host during socket communication. If the processing reached timeout, HULFT regards the state as a transmission error, and then it will cut off the communication. Because the setting value indicates the time used for error detection in network, select suitable value for the network in use.

Socket Buffer Size (socksize)

Specify the input buffer size for socket communication. When you specify 0, HULFT applies the value configured in its environment (OS) to operate.

Console Log Size (logsize)

Specify the maximum file size of the Console Log file 'hulcon.log' in KB.

When the size of the Console Log file 'hulcon.log' exceeds the specified value, the file contents will be cleared.

Console Log Backup (backuplog)

Specify whether or not to back up the Console Log file 'hulcon.log' before the file is cleared when the size of the Console Log file reaches the value specified in logsize.

The backed up Console Log is saved by the file name 'hulcon.YYYYMMDDHHMMSS' (the extension refers to the date and time of backup).

0: Do not back up the log file
1: Back up the log file

Job Timeout (jobwtimeout)

The timeout period to wait for the job termination is specified in seconds.

HULFT enables the setting in the case of Pre-send Job or where you configure Notification to be sent on the completion of Successful Job. Also, when you specify Synchronization for Remote Job Execution, HULFT enables the set value of this field if it satisfies either of the following conditions:

- Where the set value for the Job Timeout is less than that of timeout specified for Remote Job Execution
- Where you do not specify the value of timeout for Remote Job Execution

Regarding other job execution, HULFT waits for the termination of a job without limitation. For the points where HULFT enables the Job Timeout, refer to the "4.4 Timeout Settings."

Clear Unsent File (delreqcpfile)

Specify whether or not to delete the Send Control file (sddreqcp.dat) when the Send process starts up.

0: Do not delete
1: Delete

Clear Resend Queue File (delresendfile)

Specify whether or not to delete the Resend Queue file (sddreqcp.dat) when the Send process starts up.

0: Do not delete
1: Delete

Output To Event Logs (eventlog)

Whether to output a message to an event log or not can be set by message types, by specifying 0 or 1 in a 3-digit field. The specification of '1' for the certain digit outputs the specified type of message to an event log, while setting '0' to a certain field does not output the message. From the left, the three-digit number represents 'Information,' 'Warning,' and 'Minor and critical Error,' respectively.

0: Do not output to event log
1: Output to event log

<Example> eventlog = 1 0 1

- 1 → Information
- 0 → Warning
- 1 → Minor and critical error

The specification as above outputs the 'Information' and the 'Minor and critical Error,' yet the application does not output the 'Warning.'

[Note] Refer to *Error Codes and Messages* for message types.

<Extension Settings: Account file>

Account Name

Specify an account name to operate Send, Receive, and Request Acknowledge process; and Pre-send Job.

Specify in not more than 20 characters. Regardless of full width or half width, count one letter as one character. Half width spaces before or after a character are deleted.

The account name should be specified by the account name registered in Windows. When this field is defaulted, Send, Receive and Request Acknowledge processes are operated under the system account. On the other hand, Pre-send Job is operated under the account of the user who issues Send File.

The user entered the account settings has to satisfy the following conditions by using the user manager.

1. The account belongs to the 'Administrator' group of the local machine.
2. The account is given the following user permission
   - to function as a part of the operating system
   - to increase memory quota of process (to increase quota in the case of Windows 2000)
   - to replace tokens of process level
   - to restore files and directories
   - to back up files and directories

[Note] It is necessary to log on Windows under the user name entered for account setting.

[Remarks] Refer to the relevant manual of the OS for setting user permissions.

Password

The password for above account name is specified. Specify the field within 14 half width characters.

Specify the password registered in Windows.

Domain Name

In the case of domain at where the above account is located, you should specify the name of the domain. Specify the field in half width characters within 64 bytes. The spaces before or after a characters are deleted.

Specify the domain name registered in Windows.

<Extension2 Settings: System Environment Settings file>

Status Display Selection (statusdisplay)

Specify the type of transfer status to be displayed in the Transfer Status List screen of the Management screen.

0: Display only the send status
1: Display both the send and the receive status

Pack Zone Conversion (pschg)

Specify the conversion rules of signed internal decimal (P) and signed external decimal (S) for format transfer or multi format transfer.

0: Mode 0
1: Mode 1
B: Mode B
F: Mode F
9: Mode 9
S: Mode S
N: Mode N

For details on the pack zone conversion rules, refer to "2.5.8 Code Conversion Rules for Format Transfer."

Sign for ASCII (psascii)

The sign part is specified. This is used when signed external decimal (S) and signed internal decimal (P) are converted into ASCII data during format transfer or multi format transfer.

This setting is valid only when the 'Pack Zone Conversion' is mode S.

Table 3.13 Signs for ACSII

Setting value	Zone	Sign	
		Positive	Negative
1	3	4	5
2	3	3	7

Sign for EBCDIC (psebcdic)

The sign part is specified. This is used when signed external decimal (S) and signed internal decimal (P) are converted into EBCDIC data during format transfer or multi format transfer.

This setting is valid only when the 'Pack Zone Conversion' (pschg) is mode S.

Table 3.14 Sign for EBCDIC

Setting value	Zone	Sign	
		Positive	Negative
1	F	C	D
2	F	F	D

Auto Resend Retry Count (sockerr_autoretry)

Automatic resend retry count is specified in case a communication error occurs during the transfer process. When '0' is set, automatic resend is not carried out.

Log Switch Count (logdelcount)

When the Send Log, Receive Log and Request Acknowledge Log are automatically switched, the timing of switch is specified by the number of log count. When any of log file exceeds the specified value, the log is switched. When '0' is specified, the log is not switched.

Send Transfer Error Recovery (enderrmode)

This field specifies when to clear the Send file or how to handle the Send file where the delete processing is terminated unsuccessfully.

The clearing or deletion of Send file is executed when the transfer processing has been terminated successfully.

0 : Handle it as an unsuccessful termination and execute the Unsuccessful Job
1 : Handle it as a successful end and execute the Successful Job

External Character Table Mode (gaijifile)

The process of external character table conversion is specified.

0 : Convert all the external character codes to '□' without using the external character table
1 : Convert external character codes to relevant codes by referring to the external character table
2 : Convert to relevant codes by referring to the external character table first

Refer to section "3.9 External Character Table" for details of the external character table.

Process when External character has not been Registered (gaijierr)

This field specifies the handling of the codes which are not listed in JIS Level 1 and JIS Level 2, nor registered in the External Character Table.

0: Convert to the default code
1: Handle it as an error

Receive Multiplex Level Over Retry (rcvover_rty)

This field specifies whether to retry the connection where an error occurs due to excess of the Receive Multiplex Level during socket connection to a host on the receiving side.

0: Do not retry and return an error
1: Retry the connection

For reconnect retry count and interval, 'Connect Retry Count' and 'Connect Retry Interval' are used respectively.

KEIS Em-size Space Mode (keisspmode)

This field specifies the conversion rules of em-size (full width) space code. This tag is valid only when the Space Character Mode (spcode) is set to 1.

0: Convert the em-size space code into 0x4040
1: Convert the em-size space code into 0xA1A1

Criteria to Delete Resend Queue (resenddel)

This field specifies criteria to delete records from Resend Queue where Send File is issued for the File ID that exists in the Resend Queue.

1: Delete the records of which File ID and Host Name agree to those of the file to send.
2: Delete the records of which File ID, Host Name, and the File Name agree to those of the file to send.

HULFT-SAN V5 Product (san_ver5)

This field is not available in English edition.

Process Auto Reboot Max. Count (procrestartcnt)

Specify the maximum number of automatic reboot attempts when a process terminates unsuccessfully.

When '0' is specified, the process will not be restarted automatically.

Process Auto Reboot Retry Interval (procrestarttime)

Specify the interval in seconds until the next restart attempt when automatic process reboot is executed upon unsuccessful termination of the process.

Request Acknowledge Host Check (obshstchk)

In the acknowledgement of a service request, this field specifies the operation by HULFT where the host that issued the request is not registered in the Host Information.

The target services of this field are as follows; Send Request(SEND), Resend Request (RESEND), Post-receive Job Result Inquiry Request(HULJOB), Job Execution Result Notification (HULSNDRC), and Remote Job Execution (HULRJOB)

0: Continue the Request Acknowledge processing.
1: Refuse the connection from the host that issued the request, and handle the case as an error.

Framed Message Transfer Type (proctranstype)

Specify a data transfer method.

N: Error Detection Priority Mode

HULFT operates in the Error Detection Priority Mode at the time both sending and receiving. The data transfer method, the timing for the error detection, and the timing of cancellation acknowledgement are the same as those of HULFT lower than Ver.7.

S: Send Speed Priority Mode

As for sending, HULFT operates in the Transfer Speed Priority Mode, while the application operates in the Error Detection Priority Mode for receiving.

In the Send Speed Priority Mode, utilization of resources at the time of sending varies more widely than the conventional transfer method of HULFT lower than Ver.7 which may cause delay in the timing of error detection and the cancellation acknowledgement.

When HULFT on the receiving side is not in the Transfer Speed Priority Mode, the Error Detection Priority Mode is applied.

R: Receive Speed Priority Mode

As for receiving, HULFT operates in the Transfer Speed Priority Mode, while the application operates in the Error Detection Priority Mode for sending.

In the Receive Speed Priority Mode, utilization of resources at the time of receiving varies more widely than the conventional transfer method of HULFT lower than Ver.7 which may cause delay in the timing of error detection and the cancellation acknowledgement.

When HULFT on the sending side is not in the Transfer Speed Priority Mode, the Error Detection Priority Mode is applied.

A: Send and Receive Speed Priority Mode

HULFT operates in the Transfer Speed Priority Mode at the time of both sending and receiving.

In the Send and Receive Speed Priority Mode, utilization of resources at the time of sending and receiving varies more widely than the conventional transfer method of HULFT lower than Ver.7 which may cause delay in the timing of error detection and the cancellation acknowledgement.

When HULFT on the other side is not in the Transfer Speed Priority Mode, the Error Detection Priority Mode is applied.

Code Conversion Mode (codechangemode)

This field specifies how to treat external characters strings.

0: Mode 0

In code conversion, this mode treats the codes of conversion destination as fixed length and executes conversion to external character

(except for UTF-8)

1: Mode 1

In code conversion, this mode treats the codes of conversion destination as variable length and executes conversion to external character

Some of the fields mentioned below cannot be set on the Management screen. In such cases, edit the file directly.

<Others: System Environment Settings file>

Password Check (password)

Specify whether or not to carry out user and password check when being connected from the Management screen or HULFT Manager.

0: Disable password check
1: Enable password check

Management Screen Security (admcheck)

Specify whether or not to use the Management Screen Security function. Note that the Management Screen Security function becomes invalid unless 'Password Check' is set as '1' (enable password check).

0: Disable the Management Screen Security function
1: Enable the Management Screen Security function

Refer to "3.6 Management Screen Security Setting" for details on the security function of the Management screen.

Retry Count on Send File (sretrycnt)

Specify the number of request re-issue attempts when connection to the Send process by using the Send File command (utlsend.exe) fails.

When '0' is specified, re-issue will not be attempted.

Retry Interval on Send File (sretrytime)

Specify the interval until the next attempt of request re-issue in seconds when connection to the Send process by using the Send File command (utlsend.exe) fails.

Dynamic Parameter Specification (dynparam)

Specify whether or not file names, transfer group names and host names can be dynamically changed in the Send File command (utlsend.exe).

0: Disable dynamic specification
1: Enable dynamic specification

[Note] When the resulting byte count of dynamic specification exceeds the limitation of the byte count for that field, an error occurs.

Message Dynamic Parameter Specification (msgdynparam)

The mode used while receiving the message is specified.

Specify whether or not to replace the message and environment variable ($MSGn) entered in the management information with the message specified at the sending side.

0: Never replace any messages in management information.
1: Replace messages in the Send Management Information and the Receive Management Information; do not replace messages in the Job Information and the Mail Interface Information.
2: Do not replace messages in the Send Management Information and the Receive Management Information; replace messages in the Job Information and the Mail Interface Information.
3: Replace all the messages in management information

Send Unit Selection (sndpsend)

Specify the timing when HULFT completes a Send process. The timing to update the number of sending processings that are executed concurrently changes depending on this value.

0: At the time of subsequent job completion
1: Immediately before subsequent job start-up

Service Connect Timeout (srvtime)

Specify the timeout period from the service command until connection of the Request Acknowledge process in seconds when a service request is acknowledged from another host. Increase this value if a large amount of service requests are accepted and timeout errors occur frequently.

IP Version (ipversion)

Specify the version of the IP protocol for socket communication.

4: Use IPv4 when connecting to the remote host, use IPv4 only when being connected
4/6: Use IPv4 first when connecting to the remote host, try with IPv6 if connection fails; use both IPv4 and IPv6 when being connected
6/4: Use IPv6 first when connecting to the remote host, try with IPv4 if connection fails; use both IPv4 and IPv6 when being connected
6: Use IPv6 when connecting to the remote host; use only IPv6 when being connected

[Note] Register the IPv6 address of the local host in DNS or the 'hosts' file when specifying '4/6, '6/4' or '6'.

Transfer Group Check (tgrpchk)

Specify transfer group check process when receiving.

The host name of the sending side is checked against the host name registered in the Transfer Group ID of the Receive Management Information are checked, and when it is not registered, it can be considered as an error.

0: Do not check
1: Check only when Transfer Group IDs are set in the Receive Management Information; do not check when no ID is set
2: Check only when Transfer Group IDs are set in the Receive Management Information; return an error when no ID is set

Transfer Status Transfer Count (disptranscnt)

Specify the number of transfer data count acquired at one time when the transfer status is displayed.

Encryption Scheme (ciphertype)

Specify the method of encryption that HULFT uses.

0: Use HULFT Encryption Scheme
1: Use other encryption scheme by incorporating encryption exit routine

X Type Conversion (xmode)

Specify the conversion rules of character type (X) for format transfer or multi format transfer.

0: Mode 0
1: Mode 1

For details on conversion rules of X type, refer to "2.5.8 Code Conversion Rules for Format Transfer."

HULFT Manager Timeout (admsocktime)

Specify the timeout period to break the communication in seconds while the Request Acknowledge process is executed but there is no reply from HULFT Manager.

Receive File Lock Standby (rcvfilelockwait)

Specify the exclusive control process of the Receive file.

0: Handle it as an error when the Receive file is locked
1: Wait until the file is unlocked

[Note] If the processing by HULFT locks the Receive file, HULFT conforms to the setting in Receive File Lock Standby. In that case, HULFT disables the settings of Retry Count on Locked Receive File and Retry Interval on Locked Receive File. If processing performed by other than HULFT locks the Receive file, HULFT disables the settings in Receive File Lock Standby yet it enables the settings in Retry Count on Locked Receive File and Retry Interval on Locked Receive File.

Queue Of Pending Connect (listen)

Specify the maximum number of connections that a Send, a Receive, or a Request Acknowledge process can accept at the same time. An error occurs at the connection side when the number of connections at one time exceeds this value. Note that this is not the maximum number of communications allowed at one time.

The value that can be specified depends on the OS where HULFT is operating. For more details, refer to the relevant manual of OS.

Local Host Name (myhostname)

Normally, the host name set in the OS is used as the local host name. This field is used when setting a different name for the remote host for a reason of administration.

When the destination host type is K, a string up to 8 bytes is valid for this value.

[Note] This tag does not exist in the initial condition of HULFT installation.

Receive Ready Notification (rcvcmd)

Use of Receive Ready Notification functionality is specified. Resend Request command (utlrecv.exe -a) is specified. For details regarding Receive Ready Notification, refer to "3.5.5 Method of Automatic Startup upon Power On."

[Note] This tag does not exist at the time of initial HULFT installation.

Operation Log Output Option (oplselect)

Specify whether to output the Operation Log.

0: Do not output the Operation Log
1: Output only the File Access Log
2: Output only the Command Execution Log
3: Output both File Access Log and Command Execution Log

Operation Log Auto Switch Size (oplchangesize)

Specify the maximum size of the Operation Log file in the unit of MB.

If the size of the Operation Log file exceeds this value, the file output till then is renamed to be backed up and the Operation Log file is switched to a newly-created one automatically. The backup file is created in the same output folder as the Operation Log file.

If you specify '0' for this field, HULFT does not switch the Operation Log automatically.

Refer to "2.6.4 Automatic Switching of Operation Log" for details.

Operation Log Switch Generation Count (oplgenerationcount)

Specify how many generations of backup files of the Operation Log file should be kept when the Operation Log file is switched.

When the number of backup files exceeds this value through automatic or manual switching, the Operation Log file is overwritten sequentially from the oldest file.

Specifying the value other than '0' in the Operation Log Auto Switch Size enables this field.

Defaulting this field sets '9999' for the value.

Refer to "2.6.4 Automatic Switching of Operation Log" for details.

Operation Log User-specified Character (oplcharacter)

Specify the character string to identify HULFT that outputs the Operation Log or the Trace Log, when two or more HULFTs operate in the same environment.

Specify the field in alphanumeric characters within 8 bytes. When 9 bytes or more are specified, the characters after the ninth byte are ignored.

Defaulting this field does not display anything.

Operation Log User ID Output Format (opluseridtype)

Specify the output format of the Operation Log User ID (OS).

0: Down level log-on name
1: User principal name (UPN)

Specifying '0' outputs the User ID in the form of 'Net BIOS Domain Name\User Name.'

Example) TEST_DOMAIN\User

Specifying '1' outputs the User ID in the form of 'User Name @DNS Domain Name.'

Example) User@test_domain.hulft.com

However, when the user is not a domain user account, the User ID is output in the form of '0,' because HULFT cannot acquire UPN even if this format is specified.

<Mail Interface: Mail Environment Settings file>

Mail Account Name (User)

Specify the account name of the SMTP server.

This is specified within 20 alphanumeric characters. This is mandatory when using the mail interface function.

Usually, enter the same string present on the left of the 'At' symbol (@) of the e-mail address.

The character string, in which the values of 'mail account name', '@', and 'Domain Name' are enlinked, is notified to the receiving side of the mail.

Mail Server Host Name (MailHost)

This field specifies the SMTP server name used for message transmission in alphanumeric character within 20 bytes. If you have any questions, ask your internet service provider or the administrator of your local area network.

[Note] When a Mail Server with SMTP authentication is specified, proper operation of the mail interface function is not guaranteed.

SMTP Port No. (SmtpPort)

Specify the port number that the SMTP server uses.

A value from '1' to '65535' can be specified. In general, '25' is used.

Full Name (FullName)

Specify the full name of the sender for mail interface.

This is optional even when using the mail interface function.

Domain Name (DomainName)

Specify the domain name within 64 alphanumeric characters.

Usually, enter the same string present on the right of the 'At' symbol (@) of the e-mail address.

The character string, in which the values of 'mail account name', '@', and 'Domain Name' are enlinked, is notified to the receiving side of the mail.

3.5 HULFT System Startup and Stop

In order to use HULFT system, it is necessary to start the following processes.

- Send process
- Receive process
- Request Acknowledge process

3.5.1 Service

When HULFT is installed, HULFT service is registered with the OS. If the service is not registered, the transfer function of HULFT cannot be used.

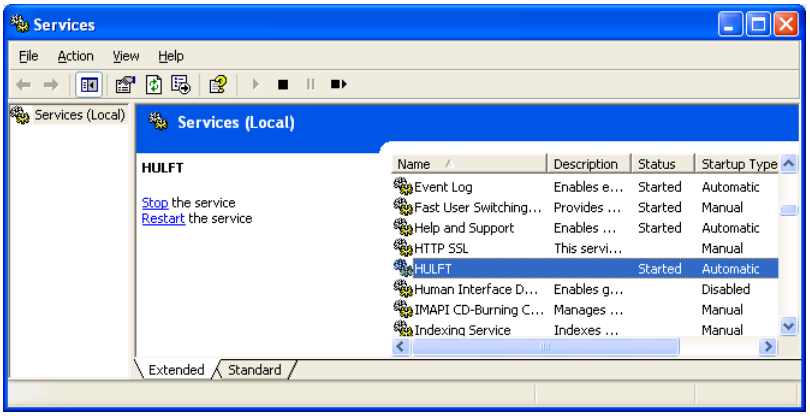
(1) Method to Confirm Service Registration

In the case of Microsoft WindowsXP Professional, whether the service is registered or not can be checked by the following procedure.

[Remarks] As the method to display the Service screen varies depending on the OS, refer to the relevant manual of the OS for the procedure.

(1) Click [Start] button, select [Control Panel] > [Management Tool], then double click [Service].

(2) The list of the services is displayed. Check whether the service name of HULFT in use is registered in the list.



If the service name of HULFT in use does not exist in the service list, register the service name.

[Note] It is not necessary to register or delete every time HULFT is started and stopped.

(2) Method of Service Registration / Deletion

Register or delete HULFT service from the OS services by using the following command:

• Service Registration/Deletion command

```
utlservice -m {add [servicename]|delete}
```

Parameter explanation

```
-m {add[servicename]|delete}
```

Selection of Registration/Deletion (Mandatory)

- add: Registration of service.
- *servicename*: Registration of specific 'HULFT(*servicename*)'
  By default, Register as 'HULFT'
- delete: Deletion of service.

[Note]

- **When HULFT is installed, the installer registers the service. It is not necessary to register/delete every time HULFT is started/ stopped.**
- **In order to delete the service, it is necessary for HULFT service to be stopped.**
- **When you uninstall HULFT, do so after the service has been deleted. In case uninstallation is carried out before deletion of the service, re-install and then again uninstall the software after deleting the service.**
- **When you execute utlservice command, execute it on the command prompt activated by administrative privilege.**

3.5.2 HULFT Startup Method

Refer to *Operation Manual* for details on HULFT startup method.

3.5.3 HULFT Exit Method

Refer to *Operation Manual* for details on HULFT stop method.

3.5.4 User Permission

As an essential requirement, a user who operates HULFT should belong to Administrator group of the local machine.

Ensure to carry out installation under the user who has permission to register the service.

3.5.5 Method of Automatic Startup upon Power on

(1) Automatic HULFT Service Startup

HULFT service can be automatically started at the time of OS reboot. The settings of automatic startup differ according to the OS in use. The method of setting automatic startup in Windows XP Professional is described below. The following tasks should be carried out by the system administrator.

a) Select [Start] > [Control Panel] > [Management Tool] > [Service]

b) Double click the service name from the list of the registered services, or select the service name in the list and click the [Properties] button.

c) Select [Automatic] from the [Startup Type] dropdown list and click [OK].

Registering by following this procedure ensures that HULFT service starts up when the OS is started.

(2) Issue of Receive Ready Notification (Automatic Startup of Resend Request Command)

If the Send process is executed when the Receive process of receiving side is not operational, it would result in an error and the process would then turn into Resend Queue status. By executing Receive Ready Notification command upon starting the Receive process on the receiving side, HULFT can issue Resend File to the Send files placed on the Resend Queue on the sending side. This mechanism allows users to operate HULFT without fearing the omission of sending.

[Method to set Receive Ready Notification]

Add the tag mentioned below to the System Environment Settings file.

```
rcvcmd = utlrecv.exe -a
```

Resend Request is issued to all hosts that are registered in Host Information, whenever the Receive process starts. Here the request would fail if error occurs even in one host or the Resend Queue file does not exist in the Request destination host. (Resend is done to the host only when the request was correctly accepted).

Refer to *Operation Manual* for details on the Receive Ready Notification command.

[Note] When a large number of host names are registered, it may take some time for the command to end.

3.6 Management Screen Security Settings

3.6.1 Password Check Function

Any illegal access can be prevented by making users to enter the user name and the password when the Management screen is started.

(1) File Name

The name of the Password Management file is 'huladmin.dat'. This name cannot be changed. The Password Management file is stored in the path where the system file used by HULFT exists (HULPATH).

Note that the 'huladmin.dat' file should not be deleted since all the registration information will be lost by deleting this file.

(2) Setup Method

a) Enable the password function

Specify in the System Environment Settings file to carry out password check. Describe as follows.

```
password = 1
```

Refer to "3.4.3 System Environment Settings List" for details.

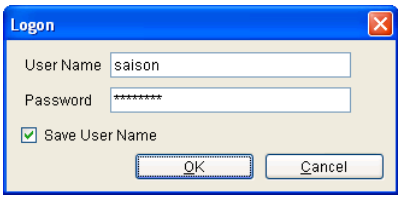
b) Register user names and passwords

It is necessary to set the user names and the passwords of the users who are allowed to connect. Set these from the User Information Registration screen. Refer to "3.6.2 Management Screen Security Settings."

[Note] When only password check is carried out, all permissions will be given irrespective of the registration contents.

(3) Startup of Management Screen

When the Management screen is started, the dialog box shown below is displayed in the beginning.



Enter the user name and password registered in the step (2) in this dialog box. When authentication is completed successfully, the message 'Logged-on normally' is displayed. When an illegal user name or password is input, the Management screen is not displayed and input is requested again.

3.6.2 Settings for Management Screen Security

User permissions for following seven types of operation on HULFT Management screen can be given to each user:

- Log Read Permission
- Log Delete Permission
- Issue Request Permission
- Permission to View System Management Information
- Permission to View and Update System Management Information
- Permission to View and Delete System Management Information
- Permission to Access Directory via HULFT Manager

(1) Setup Method

In order to add security while using the Management screen, do the following:

a) Enabling the password function

Refer to the "3.6.1 Password Check Function" for the specification methods.

b) Enabling the Management Screen Security function

It is specified in the System Environment Settings file to carry out the Management Screen Security function.

Specification method is as follows:

```
admcheck = 1
```

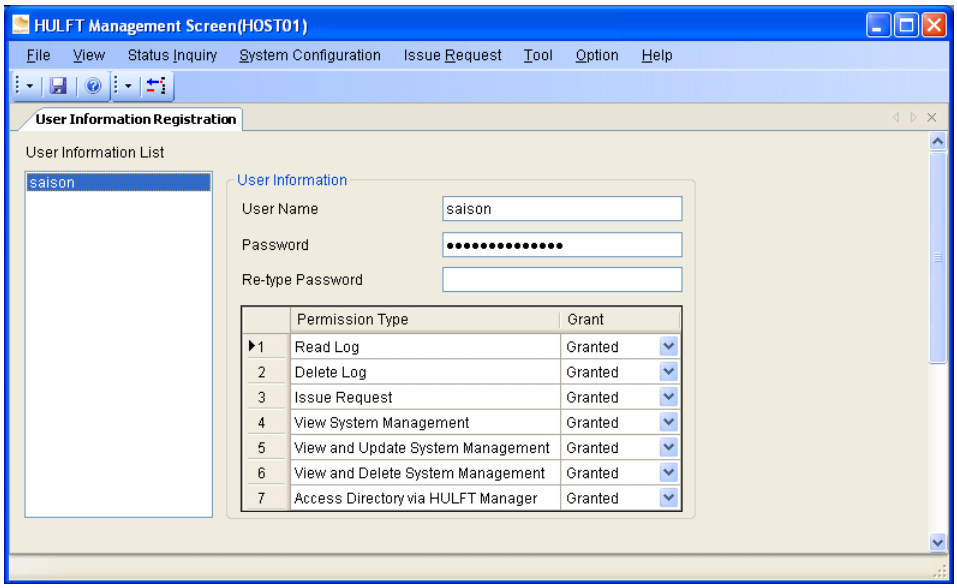
For details, refer to the "3.4.3 System Environment Settings List."

c) Registering user name, password and permission information

It s necessary to set the user name, the password and the permission of the user who is allowed to connect. Carry out the setup method on the User Information Registration screen.

(2) Explanation of Each Field

The explanation of each field is given below.



User Name

Specify the user name whose permission is set within 20 bytes of alphanumeric characters.

Password

Specify the password of the above user within 14 bytes of alphanumeric characters.

Log Read Permission (on-screen name: Read Log)

Specify the log read permission on the Management screen.

Granted : Log Read Permission is granted
Not Granted : Log Read Permission is not granted

The Log Read Permission of the screens listed below is specified by above setting.

Status Inquiry menu:

- Send Log Inquiry screen
- Receive Log Inquiry screen
- Request Status Confirmation screen
- Transfer Status List screen
- Resend Queue Status List screen

Log Delete Permission (on-screen name: Delete Log)

Specify the log delete permission on the Management screen.

Granted : Log Delete Permission is granted
Not Granted : Log Delete Permission is not granted

The Log Delete Permission of the screens listed below is specified by above setting.

Status Inquiry menu:

- Send Log Inquiry screen
- Receive Log Inquiry screen
- Request Status Confirmation screen

Issue Request Permission (on-screen name: Issue Request)

Specify the request issue permission on the Management screen.

Granted : Issue permission of Send File, Send Request, and the like is granted
Not Granted : Issue permission of Send File, Send Request, and the like is not granted

The Issue Request Permission of the screens listed below is specified by above setting.

Send File:

- Send Log List
- Send Management Information List

Send Request:

- Receive Log List
- Receive Management Information List

Cancel Request:

- Transfer Status List (Sending or Receiving)

Resend File / Resend Queue Deletion Request:

- Resend Queue Status List screen

Permission to View System Management Information
(on-screen name: View System Management)

This field specifies the permission to view management information on the Management screen.

Granted : Permission to view system management information is granted
Not Granted : Permission to view system management information is not granted

[Remarks] When you set 'Permission to View System Management Information' to 'Not Granted,' you should specify 'Not Granted' for the Permission to View and Update System Management Information and the Permission to View and Delete System Management Information as well.

Permission to View and Update System Management Information
(on-screen name: View and Update System Management)

This field specifies the permission to view and update management information on the Management screen.

Granted : Permission to View and Update system management information is granted
Not Granted : Permission to View and Update system management information is not granted

Permission to View and Delete System Management Information
(on-screen name: View and Delete System Management)

This field specifies the permission to view and delete management information on the Management screen.

Granted : Permission to View and Delete system management information is granted
Not Granted : Permission to View and Delete system management information is not granted

Above permissions, namely Permission to View System Management Information, Permission to View and Update System Management Information, and Permission to View and Delete System Management Information, specify the permission to view, update and delete system management information on the following screens:

System Configuration menu:

- Send Management Information
- Receive Management Information
- Job Information
- Host Information
- Transfer Group Information
- Format Information
- Multi Format Information

- Mail Interface Information
- System Environment Settings

Tool menu:

- Process Controller
- CSV Conversion Information Registration
- User Information Registration
- External Character Table Registration
- EBCDIC User Table Registration

Permission to Access Directory via HULFT Manager

(on-screen name: Access Directory via HULFT Manager)

This field specifies the permission to view directories via HULFT Manager. It is a permission to view directories when you select files via the screen of HULFT Manager.

Granted :	Permission to view directories is granted
Not Granted :	Permission to view directories is not granted

3.7 Request Acknowledge Definition Settings

Request Acknowledge Definitions can be set for each service.

The following seven types of services are set. The service name is provided within parenthesis.

- Send Request (SEND)
- Resend Request (RESEND)
- Job Execution Result Notification (HULSNDRC)
- Remote Job Execution (HULRJOB)
- HULFT Manager (HULADMIN)
- Post-receive Job Result Inquiry (HULJOB)

3.7.1 Request Acknowledge Definition File

Change the default contents set during installation in line with the system environment in use.

(1) File Name

The file name of the Request Acknowledge Definition file is 'service.dat'. This name cannot be changed. The Request Acknowledge Definition file is stored in the path where the system file used by HULFT exists (HULPATH).

(2) Description Format

It is considered as a comment line when a single byte number sign (#) exists at the beginning of the line. One or more blank lines are accepted.

A space code (0x20) or a tab code (0x09) is used as the delimiter between fields.

[Note]

- **If there are any incorrect input in the Request Acknowledge Definition file (service.dat), the request will not be accepted correctly. When modifying the Request Acknowledge Definition file, make sure to create a backup.**
- **When you would like to use space in startup command, enclose the whole message within double quotations.**

(3) Record layout

Because the Request Acknowledge Definition file (service.dat) does not include binary data, it is possible to create it by using an editor such as Notepad. Record should be described for each service.

(4) Explanation of Each Field

The explanation of each field is provided below.

Log output

The output mode of the Request Acknowledge Log output related to the corresponding service name is specified.

0 : Do not output the Request Acknowledge Log for the service
1 : Output the Request Acknowledge Log for the service

Specifying '0' also executes the Request Acknowledge process normally.

Service name

The defined service name is described.

Program name

The service program that is started up for the service name is described.

(5) Setting Example

[Description example]

```
# Copyright (c) 1995-2009, SISCO
#
# logflg log output flag 0 : Will not be output 1 : Will be output
# Service name
# Command Start-up command
#
# logflg Service Command
1 SEND hulsendex SEND
1 RESEND hulsendex RESEND
1 HULSNDRC hulsndrc -l c:\temp\hulsndrc.log
1 HULADMIN
1 HULJOB huljob
1 HULRJOB hulrjob -l
```

3.7.2 Send Request Service Settings

It is not necessary to change the Request Acknowledge Definition file.

3.7.3 Resend Request Service Settings

It is not necessary to change the Request Acknowledge Definition file.

3.7.4 Job Execution Result Notification Service Settings

If HULFT receives Job Execution Result Notification, it changes the Request Acknowledge Definition file in accordance with the environment that the application works.

Description method of the Request Acknowledge Definition file

```
HULSNDRC hulsndrc -l [filename]
```

Parameter explanation

`-l filename`

Specify the file name to which the message is output.

The message from the remote host will be output to the file.

The file name is specified by the full path within the 256 bytes. When no file name is specified (when specified only '-l'), 'hulsndrc.dat' is set as the default value.

- Message output contents

hulsndrc : HOSTNAME = Host name JOBNAME = Job name DATE = Date TIME = Time RC = Status code MESSAGE = Message DN = Domain name

<Message explanation>

HOSTNAME : Host Name that issues the Job Execution Result Notification (8bytes)
JOBNAME : Job name (60 bytes)
DATE : Job end date (YYYY/MM/DD)
TIME : Job end time (hh:mm:ss)
RC : Status code (4 bytes)
MESSAGE : Message (128 bytes)
DN : Domain name (68 bytes)

[Note] For the host name, the first 8 bytes of the domain name are displayed. The message record length is 344 bytes.

3.7.5 Remote Job Execution Service Settings

If HULFT receives Remote Job Execution, it changes the Request Acknowledge Definition file in accordance with the environment that the application works.

Description method of the Request Acknowledge Definition file

```
HULRJOB -l [filename]
```

Parameter explanation

```
-l   filename
```

Specify the file name to which the job execution result is output.

The file name is specified by the full path within 256 bytes.

When no file name is specified (when specified only '-l'), 'joblog.dat' is set as the default value.

Job result output contents

JOBID = Job ID HOST = Host name

DATE = Date STIME = Start time ETIME = End time RC=0000

JOB = Job

<Output example>

JOBID=JOB01 HOST=xs3475

DATE=2003/ 9/21 STIME=21:50:28 ETIME=21:50:33 RC=0000 JOB=C:\WINNT\Notepad.exe

DATE=2003/ 9/21 STIME=21:50:33 ETIME=21:50:35 RC=0000 JOB=C:\WINNT\Notepad.exe

DATE=2003/ 9/21 STIME=21:50:35 ETIME=21:50:37 RC=0000 JOB=C:\WINNT\Notepad.exe

[Remarks] When multiple jobs are registered in Job ID, the execution result of the jobs will be output with items from 'DATE' to 'JOB' displayed for each of the jobs.

<Message explanation>

JOBID :	Job ID (8 bytes)
HOST :	Request issue host name
DATE :	Job start date (YYYY/MM/DD)
TIME :	Job start time (hh:mm:ss)
TIME :	Job end time (hh:mm:ss)
RC :	Job status code (4 bytes)
JOBID :	Job Execution

3.7.6 HULFT Manager Service Settings

It is not necessary to change the Request Acknowledge Definition file.

3.7.7 Post-receive Job Result Inquiry Request Service Settings

It is not necessary to change the Request Acknowledge Definition file.

3.8 EBCDIC User Table Settings

When a user table is used at the time of single byte code conversion with EBCDIC hosts, it is necessary to register the EBCDIC user table.

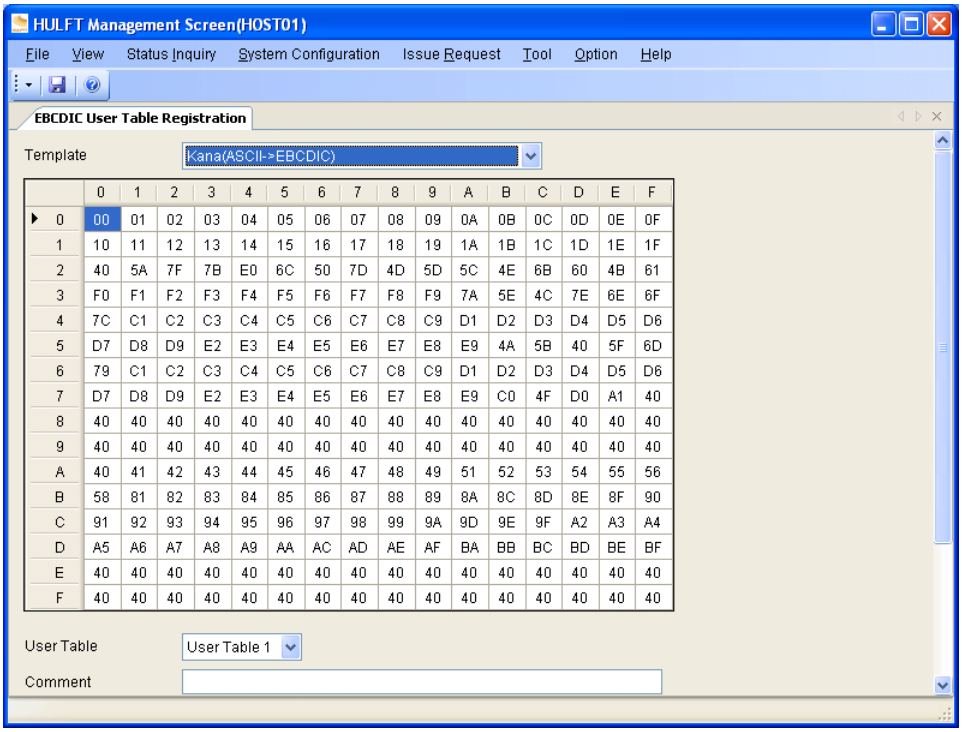
A maximum of three user code tables can be registered from the available templates.

Refer to "2.5 Code Conversion" for details on code conversion.

3.8.1 Registration of EBCDIC User Table

Registration of the EBCDIC user table is carried out from the EBCDIC User Table Registration screen.

The method of registering the EBCDIC user table is described below.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
1	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
2	40	5A	7F	7B	E0	6C	50	7D	4D	5D	5C	4E	6B	60	4B	61
3	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	7A	5E	4C	7E	6E	6F
4	7C	C1	C2	C3	C4	C5	C6	C7	C8	C9	D1	D2	D3	D4	D5	D6
5	D7	D8	D9	E2	E3	E4	E5	E6	E7	E8	E9	4A	5B	40	5F	6D
6	79	C1	C2	C3	C4	C5	C6	C7	C8	C9	D1	D2	D3	D4	D5	D6
7	D7	D8	D9	E2	E3	E4	E5	E6	E7	E8	E9	C0	4F	D0	A1	40
8	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
9	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
A	40	41	42	43	44	45	46	47	48	49	51	52	53	54	55	56
B	58	81	82	83	84	85	86	87	88	89	8A	8C	8D	8E	8F	90
C	91	92	93	94	95	96	97	98	99	9A	9D	9E	9F	A2	A3	A4
D	A5	A6	A7	A8	A9	AA	AC	AD	AE	AF	BA	BB	BC	BD	BE	BF
E	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
F	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40

(1) Registration Method

a) When registering for the first time

By selecting a conversion pattern from the template on the EBCDIC User Table Registration screen, the template code will be expanded. In User Table 1 to User Table 3 of template, '00' is stored for all the records by default. After entering the code which is to be converted, specify the output destination in the User Table field (User Table 1 to 3) and click [Save].

b) When updating the registered information

When updating the already registered information, the registered code will be expanded on the EBCDIC User Table Registration screen by selecting a user table (User Table 1 to 3) from the Template field. After entering a code you wish to modify, specify the output destination in the User Table field (User Table 1 to 3) and click [Save].

(2) Explanation of Each Field

- Template
  Specify the template for conversion code.
  There are 14 types of templates.
- User table
  Select the user table to be registered from 'User Table 1' to 'User Table 3'.
- Code input area
  Conversion character is specified. You can input from '00' to 'FF'. When not specified, '00' is set. The input area is from top left '0x00', '0x01', '0x02' to top right, and the bottom right corresponds to '0xFF'.
- Comment
  Specify the comment for this code setting within 60 bytes.

3.9 External Character Table

The External Character Table Expand command or the External Character Table Registration screen is used when Creating the external character table.

To use the external character table, it is necessary to set External Character Table Mode of the System Environment Settings file to either '1' or '2.'

3.9.1 Registration Method Using The Command

Create an external character table file, and create the expand file by using the 'External Character Table Expand' command.

Table 3.15 List of External Character Tables and Files

Input code	Output code	External Character Table	External Character Table Expand File	Default Value When Undefined
EUC	Shift-JIS	gtetos.txt	gtetos.xtd	0x81A0
Shift-JIS	EUC	gtstoe.txt	gtstoe.xtd	0xA2A2
Shift-JIS	IBM	gtstoi.txt	gtstoi.xtd	0x44E9
Shift-JIS	JEF	gtstoj.txt	gtstoj.xtd	0xA2A2
Shift-JIS	KEIS	gtstok.txt	gtstok.xtd	0xA2A2
Shift-JIS	NEC	gtston.txt	gtston.xtd	0x7F7F
Shift-JIS	UTF-8	gtsto8.txt	gtsto8.xtd	0xE296A1
IBM	Shift-JIS	gtitos.txt	gtitos.xtd	0x81A0
JEF	Shift-JIS	gtjtos.txt	gtjtos.xtd	0x81A0
KEIS	Shift-JIS	gtktos.txt	gtktos.xtd	0x81A0
NEC	Shift-JIS	gtntos.txt	gtntos.xtd	0x81A0
UTF-8	Shift-JIS	gt8tos.txt	gt8tos.xtd	0x81A0

• External Character Table Expand command

```
utlgtfextdV   [-i es| se| si| sj| sk| sn| s8| is| js| ks| ns| 8s| a]
              [-o es| se| si| sj| sk| sn| s8| is| js| ks| ns| 8s| a]
```

Parameter explanation

`-i es| se| si| sj| sk| sn| s8| is| js| ks| ns| 8s| a`

Create the external character table expand file (Optional)

es :	Expands an external character table from EUC to Shift-JIS
se :	Expands an external character table from Shift-JIS to EUC
si :	Expands an external character table from Shift-JIS to IBM
sj :	Expands an external character table from Shift-JIS to JEF
sk :	Expands an external character table from Shift-JIS to KEIS
sn :	Expands an external character table from Shift-JIS to NEC
s8 :	Expands an external character table from Shift-JIS to UTF-8
is :	Expands an external character table from IBM to Shift-JIS
js :	Expands an external character table from JEF to Shift-JIS
ks :	Expands an external character table from KEIS to Shift-JIS
ns :	Expands an external character table from NEC to Shift-JIS
8s :	Expands an external character table from UTF-8 to Shift-JIS
a :	Expands all external character tables

`-o es| se| si| sj| sk| sn| s8| is| js| ks| ns| 8s| a`

Display contents of external character table expand file to standard output (Optional)

es :	display contents of external character table of Shift-JIS from EUC
se :	display contents of external character table of EUC from Shift-JIS
si :	display contents of external character table of IBM from Shift-JIS
sj :	display contents of external character table of JEF from Shift-JIS
sk :	display contents of external character table of KEIS from Shift-JIS
sn :	display contents of external character table of NEC from Shift-JIS
s8 :	display contents of external character table of UTF-8 from Shift-JIS
is :	display contents of external character table of Shift-JIS from IBM
js :	display contents of external character table of Shift-JIS from JEF
ks:	display contents of external character table of Shift-JIS from KEIS
ns :	display contents of external character table of Shift-JIS from NEC
8s :	display contents of external character table of Shift-JIS from UTF-8
a :	displays contents of all external character tables

3.9.2 External Character Table Inclusion Example

The method of including ㈱ (Shift-JIS : 0x878A, IBM : 0x446D) in the external character table when converting from Shift-JIS code to IBM code is shown below as a creation example.

(1) Editing the external character table file from Shift-JIS to IBM by using Notepad

[Note]

- **HULFT for Windows doesn't offer the External Character Table file. Create the file according to the example mentioned below. Moreover, refer to List of External Character Table Files of Table 3.14 for the file name of each expand code.**
- **Store the created file in 'HULPATH\gtf'.**

Content of external character table (gtstoi.txt)

```
#
# Shift-JIS -> IBM
#

# default output code
defaultcode=0x0000

#incode #outcode
0x878A 0x446D
```

The line beginning with a number sign (#) is a comment.

(2) Creating the external character table expand file

Start the command prompt. Move to the binnt folder in HULFT installation folder, and execute the following command:

```
> utlgtfextdV -i si
```

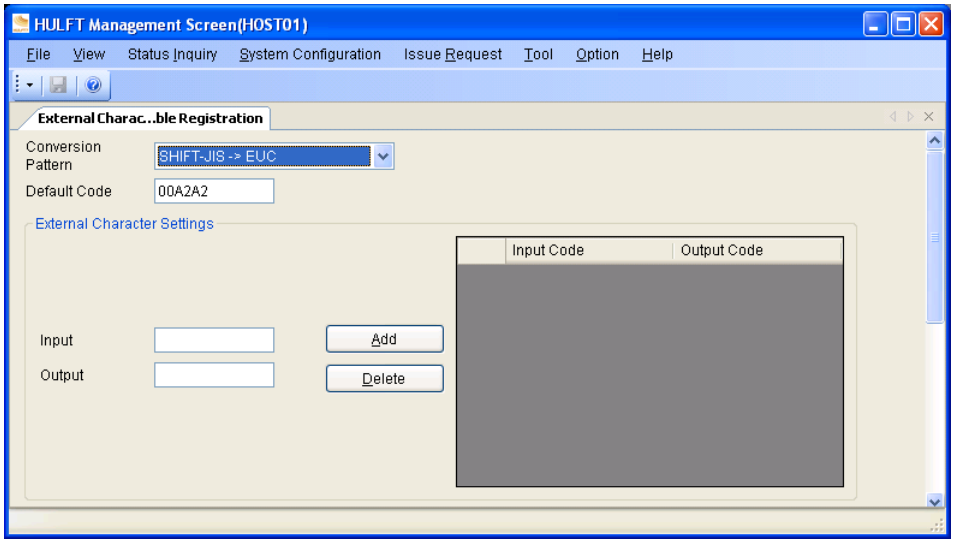
(3) Verifying the contents of the expand file with the standard output

Start the command prompt. Move to the binnt folder in HULFT installation folder, and execute the following command:

```
> utlgtfextdV -o si
```

3.9.3 External Character Registration Method Using Screens

Create an external character table on the External Character Table Registration screen.



(1) Registration Method

The conversion patterns where an external character is added or deleted are explained below:

a) Adding an external character

When updating the information that has already been registered, enter values for the Input Code and the Output Code first, click the [Add] to set and then click the [Save].

b) Deleting an external character

When deleting the information which has already been registered, select the code which is to be deleted from conversion code list and click [Delete].

(2) Explanation of Each Field

- **Conversion pattern**

  Specifies the conversion pattern where the external character is used.

- **Default code**

  The conversion characters other than the first level, second level, and the external character are input. By default, it is converted into JIS code 0x2222 (□).

- **Input code**

  The code of the conversion source that is to be registered as an external character is specified by the hexadecimal number.

- **Output code**

  The code after the conversion that is to be registered as an external character is specified by the hexadecimal number.

[Remarks] When the specified code is EUC and if the code is entered in double byte, it is displayed on the list in 3 bytes with the string '00' added at the beginning.

3.9.4 NEC Special Character Table

Although external character tables are not provided with HULFT (refer to"3.9.2 External Character Table Inclusion Example"), some of the NEC special characters are used often. Therefore, these are provided as a template table. It is imported and used in the existing external character table file.

The template of NEC special character table provided is stored in 'HULPATH\gtf'.

[Remarks] The following characters are provided as the NEC special characters.

①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫	⑬	⑭	⑮	⑯
⑰	⑱	⑲	⑳	Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ	Ⅷ	Ⅸ	Ⅹ	㍉	㌔
㌢	㍍	㌘	㌧	㌃	㌶	㍑	㌦	㌫	㍗	㌍	㌢	㍊	㌻	㎜	㎝
㎞	㎎	㎏	㏄	㎡	㍻	№.	㏍	㍾	㊤	㊥	㊦	㊧	㊨	㈱	㈲
㈹	㍽	㍼	㍻	∮	∟	∠									

Table 3.16 File List of NEC Special Character Table Template

Input code	Output code	Template File Name	Default value
EUC	Shift-JIS	ktetos.txt	0x81A0
Shift-JIS	EUC	ktstoe.txt	0xA2A2
Shift-JIS	IBM	ktstoi.txt	0x44E9
Shift-JIS	JEF	ktstoj.txt	0xA2A2
Shift-JIS	KEIS	ktstok.txt	0xA2A2
Shift-JIS	NEC	ktston.txt	0x7F7F
Shift-JIS	UTF-8	ktsto8.txt	0xE296A1
IBM	Shift-JIS	ktitos.txt	0x81A0
JEF	Shift-JIS	ktjtos.txt	0x81A0
KEIS	Shift-JIS	ktktos.txt	0x81A0
NEC	Shift-JIS	ktntos.txt	0x81A0
UTF-8	Shift-JIS	kt8tos.txt	0x81A0

The inclusion method of conversion of external characters of NEC special characters when converting from IBM code to Shift-JIS code is shown below as a creation example.

(1) Creating the external character table file (gtitos.txt).

a) When the external character table for IBM to Shift-JIS already exists, create the external character table file (gtitos.txt) of existing external characters.

Start the command prompt. Move to the binnt folder in the installation folder, and execute the following command to create an External Character file:

```
> utlgtfextdV -o is > gtitos.txt
```

Move the created external character table file to 'HULPATH\gtf'.

[Note] When the following processing is carried out without creating an existing external character table file, the existing external character table is annulled.

b) When any external character table does not exist, create the external character table file (gtitos.txt) in 'HULPATH\gtf' as follows.

```
#
# IBM -> Shift-JIS
#

# default output code
defaultcode=0x81A0

#incode #outcode
```

• Line starting with '#' is a comment.

(2) Copying the contents of NEC special character table from IBM to Shift-JIS.

The contents of NEC special character table (ktitos.txt):

```
#
# IBM -> Shift-JIS
#
#incode  #outcode
0xE270   0x8740
0xE271   0x8741
0xE272   0x8742
  •        •
  •        •
```

a) The text in bold is copied.

b) Line starting with '#' is a comment.

(3) Pasting the contents that were copied from (3) and (2) in the external character table file for IBM to Shift-JIS (refer to Table 3.15).

The contents of the external character table (gtitos.txt):

```
#
# IBM -> Shift-JIS
#
# default output code
defaultcode=0x81A0
#incode    #outcode
0x676B     0xFBFC
0x5294     0xFA61
0x576B     0x8BA7
0xE270     0x8740
0xE271     0x8741
0xE272     0x8742
```

a) The text in bold is copied.

b) Line starting with '#' is a comment.

(4) Creating the external character table expand file.

Start the command prompt. Move to the binnt folder in the installation folder, and execute the following command:

```
> utlgtfextdV -i is
```

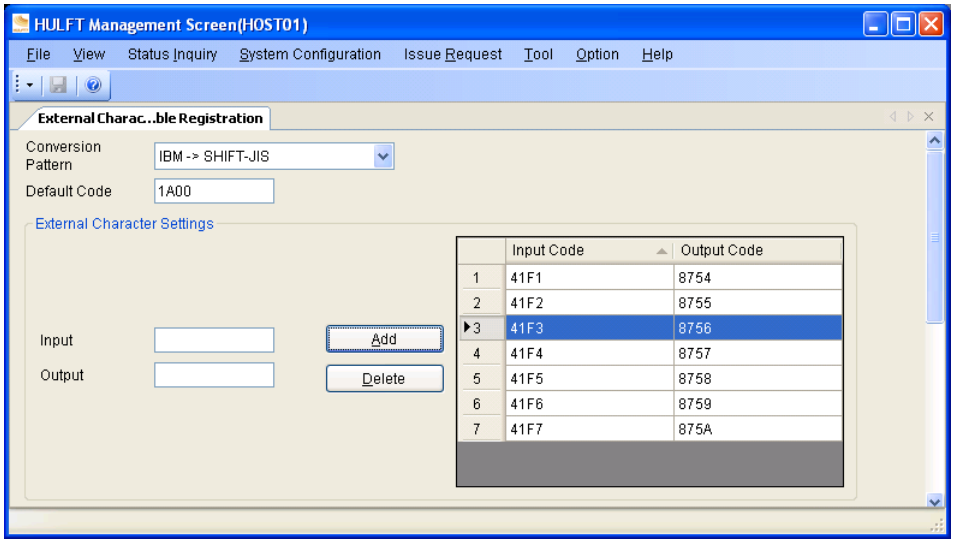
(5) Verifying the contents of the expand file.

a) Verification by the command

Start the command prompt. Move to the binnt folder in the installation folder, and execute the following command:

```
> utlgtfextdV -o is
```

b) Verification by using the External Character Table Registration screen



3.9.5 Incorporation of Windows Vista JIS2004 Character

In code conversion, users of Windows Vista and Windows Server 2008 should incorporate the additional characters in JIS2004 into the external character table, because HULFT treats the JIS2004 characters as external characters.

Among the characters that have been newly added to JIS level 3 and JIS level 4, HULFT offers template sample files of 2642 characters prepared by Shift-JIS/UTF-8. Copy the required part onto the external character table file to use the characters.

The External Character Table Template Sample file is stored in 'HULPATH\gtf.'

Table 3.17 JIS2004 Additional Character Table Template Sample File List

Code Conversion from	Code Conversion to	Template Sample File Name
Shift-JIS	UTF-8	sjistoutf8-34kanji.txt
UTF-8	Shift-JIS	utf8tosjis-34kanji.txt

The following section explains how to incorporate the additional characters in JIS2004 into the external character table, when converting from Shift-JIS to UTF-8.

(1) Copy the contents of the External Character Table Template Sample file (sjistoutf8-34kanji.txt)

Contents of the External Character Table Template Sample file (sjistoutf8-34kanji.txt)

```
#
#
SJIS->UTF-8
#
  •         •
  •         •
0xfa5c 0xe7ba8a
0xfa5d 0xe8a49c
0xfa5e 0xe98d88
  •         •
  •         •
```

a) Indicated in bold is the pasted part.

b) The line that starts with "#" is a comment.

(2) Paste the copied font table file on the External Character Table file. (gtsto8.txt)

Contents of the External Character Table (gtsto8.txt)

```
#
#
Shift JIS->UTF8
#
#
default output code
defaultcode=0xe296a1
#incode #outcode
0x8740 0xe291a0
0x8741 0xe291a1
0x8742 0xe291a2
0xfa5c 0xe7ba8a
0xfa5d 0xe8a49c
0xfa5e 0xe98d88
  •         •
  •         •
```

a) Indicated in bold is the pasted part.

b) The line that starts with "#" is a comment.

(3) Create external character table expand file

Start command prompt. Move to the binnt folder in HULFT installation folder, and execute the following command:

```
> utlgtfextdV -i s8
```

(4) Confirm the contents of the external character table expand file in standard output.

Start command prompt. Move to the binnt folder in HULFT installation folder, and execute the following command:

```
> utlgtfextdV -o s8
```

3.10 Encryption Exit Routine

It is necessary to create a DLL based on the interface specifications to include a commercially available encryption logic or user-defined encryption logic.

3.10.1 Method of Encryption Exit Routine Inclusion

(1) Creation of Functions

Create functions based on the interface specifications.

Refer to "3.10.2 Interface Specifications (Functions)" for function specifications.

(2) Creation of DLL

Create a DLL based on the interface specifications.

Refer to "3.10.3 Interface Specifications (DLL)" for DLL specifications.

(3) Inclusion in HULFT

Set the Encryption Scheme of System Environment Settings file as Another Encryption Scheme and restart HULFT.

3.10.2 Interface Specifications (Functions)

(1) Initialization Function

The Send process or the Receive process calls initialization function as the initialization process of the transfer file when the transfer is started.

When the return value of the initialization function indicates an error, the system regards it as an error in the encryption exit routine and the process will be terminated unsuccessfully without executing the file transfer.

Syntax

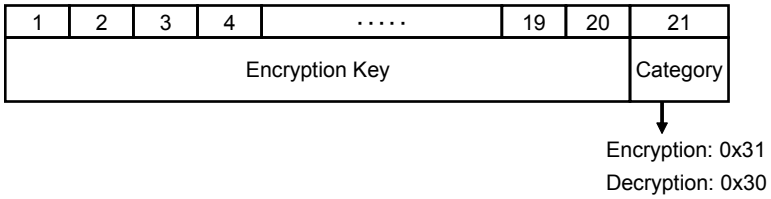
```
int_stdcall hul_usrcipher_init( void *usrdat, int usrdatlen )
```

Parameter explanation

`void *usrdat(IN)`

User specified data.

The encryption key and the category are set. When the encryption key is less than 20 bytes, the remaining part is filled with NULL. In the category, 0x31 is specified for the encryption (sending) side and 0x30 is specified for the decryption (receiving) side.



`int usrdatlen(IN)`

User specified data length

'21' is set as a fixed value.

Return value

Successful: 0
Unsuccessful: -9999 to -1, 1 to 9999

The return value will be output as the Detail code to HULFT log (However, negative value will be output by absolute value). If a value other than '-9999' to '9999' is specified, it is handled as '9999'.

(2) Encryption/Decryption Function

The Send process or the Receive process calls the Encryption or Decryption function at the time of file transfer. Because the data is passed to the Encryption or Decryption function as an input buffer per record, store that data in output buffer after encryption or decryption of the data.

When the return value of the encryption/decryption function indicates an error, the system regards it as an error in the encryption exit routine and the process will be terminated unsuccessfully without executing the file transfer.

Syntax

```
int __stdcall hul_usrcipher( void *usrdat, int inlen, char *inbuff,
int *outlen, char *outbuff )
```

Parameter explanation

`void *usrdat(IN)`

User specified data address.

Set the address of user specification data specified in initialization function.

`int inlen(IN)`

Input buffer length.

Set the size of input data buffer.

`char *inbuff(IN)`

Input buffer.

Set the buffer of input data.

`int *outlen(IN/OUT)`

Output buffer length.

Set the size of output data buffer. After completion of processing, setup the size that was output to output buffer.

[Note] Input buffer length and the output buffer length at the time of output must be the same.

`char *outbuff(OUT)`

Output buffer.

Set the buffer of output data.

Return value

Successful: 0
Unsuccessful: -9999 to -1, 1 to 9999

The return value will be output as the Detail code to HULFT log (However, negative value will be output by absolute value). If a value other than '-9999' to '9999' is specified, it is handled as '9999.'

(3) Termination Function

The Send process or the Receive process calls the termination function at the time of file transfer completion.

When the return value of the termination function indicates an error, the system regards it as an error in the encryption exit routine and the process will be terminated unsuccessfully.

Syntax

```
int __stdcall hul_usrcipher_fin( void *usrdat )
```

Parameter explanation

```
void *usrdat (IN/OUT)
```

User specification data address.

Set the address of user specification data that is specified in initialization function.

Return value

Successful: 0
Unsuccessful: -1 to -9999, 1 to 9999

The return value will be output as the Detail code in HULFT log (However, negative value will be output by absolute value). If a value other than '-9999' to '9999' is specified, it is handled as '9999.'

3.10.3 Interface Specifications (DLL)

When the Encryption Scheme is set to use other encryption scheme in the System Environment Settings file and HULFT Cipher Option is not installed, the Send process or the Receive process loads the encryption exit routine DLL as a part of initialization processing at the time of startup.

On the contrary, the Send process or the Receive process unloads the encryption exit routine DLL as a part of termination processing at the time of termination.

DLL name

hulusrcipher.dll

DLL storage folder

'HULFT installation folder'\binnt

3.10.4 Points to be Noted

(1) When Including Encryption Exit Routine

- Encryption Exit Routine cannot be used for any purpose other than encryption.
- If the Encryption Key is not set in the sending side, Encryption Exit Routine cannot be called.
- When compression is specified, if the Encryption Keys of the sending side and receiving side are different, it may result in a compression/decompression error.

(2) When Creating Encryption Exit Routine

- Create the encryption exit routine in C language. The target compiler is 'Microsoft Visual C++.net.' Use '-stdcall' in accordance with the call conventions.
- Set the return value of each function and return it to the calling side. Make sure that the function does not exit within itself.
- After including the Encryption Exit Routine, if there is any problem generated in the Received file contents, first verify the data contents in the created function.
- When the variable is declared in the function, do not use static variables and global variables.

3.11 Exclusive Control between HULFT and User Job

If you need to control files exclusively between HULFT services when developing business application, the method can be used for this purpose.

As the method to lock Send files and Receive files when Send file deletion is specified, mutex object (refer to 'Win32SDK' or 'Visual C++' for details) is used in HULFT.

To control Send and Receive files exclusively in the business application, use the mutex object in the same way with the use of the CreateMutex() function and the ReleaseMutex() function.

[Note]

- **Any processes on a file should be executed after acquiring ownership of the mutex object.**
- **When the Notification is set as Successful Job Completion and you would like to access the Receive file via Post-receive Job, HULFT has already locked the file with mutex object.**

  Therefore, do not lock the file with mutex during Post-receive Job.

<Mutex naming conventions>

A mutex name is created in the following naming conventions.

Mutex name= 'Global\' + 'HUL' + 'Send or Receive file name'

'Global \'

Specify 'Global\' at the beginning of the mutex name.

'HUL'

A fixed string 'HUL' is used.

'Send and Receive file name'

A Send File Name in the Send Management Information or a Receive File Name in the Receive Management Information is used by the full path.

However, '\' must be replaced with '?'. Also, the name is case-sensitive.

<Example of mutex name>

When the Send File Name is 'c:\temp\sendfile.txt', the mutex name is as follows:

```
Global\HULc:?temp?sendfile.txt
```

3.12 Backup of System Files

From the point of view of safe operation of the system, take a periodic backup of the system files in case of emergencies such as hard disk crush.

(1) Backup Method

HULFT backup can be made by the following procedure. It is recommended that the following tasks should be performed by the user who had installed HULFT or the system administrator. In this procedure, it is assumed that 'HULPATH' at the time of HULFT operation is 'HULFT_folder\etc.' If a different HULPATH is in use, replace the path with that of your environment.

a) Move to the HULFT installation folder.

b) Check the environment of HULFT

c) Move to 'etc,' which is HULPATH environment.

d) Check the contents of the HULPATH environment.

Confirm that it is the folder which is described in the 'hulft.ini' file in the binnt folder.

e) Create a backup.

Copy all the files under the environment of HULPATH into arbitrary folder.

Here, all files that exist in this folder are the target of the backup. Therefore, log file etc. have also become objects of backup. When log file is to be excluded from target, please delete it by log deletion command beforehand.

f) Verify whether a backup has been created.

g) The backup is now complete.

Retain the backup for recovery.

(2) Method of Restoration from Backup

You cam restore the backup by the following procedure. It is recommended to perform the following operations by the user who installed HULFT or by the system administrator. In this procedure, it is assumed that 'HULPATH' at the time of HULFT operation is 'Hulft_folder\etc.' If a different HULPATH is in use, replace it with that of your environment.

a) Move to the directory where the backup of HULFT management information is stored.

b) Verify the existence of HULFT management information backup file.

c) Move to the HULPATH environment 'etc'.

d) Check the contents of the HULPATH environment.

Confirm that it is the folder which is described in the 'hulft.ini' file in the binnt folder.

e) Copy and then restore the backup file.

After copying, all management information will return to information at the time of backup creation. Note that it is not possible to return after restoring.

f) Check whether the environment is reflected by using the Management screen and the like.

g) Once you confirm the environment is returned to the state when the backup is created, the restoration is complete. The backup file can be deleted if it is not required.

[Note]

- **Be sure to exit from HULFT before you carry out backup or restoration work.**
- **You cannot apply the backup file to the environment that is different from the one you have created the backup file.**

Chapter 4

Points to be Noted for HULFT Operation

This chapter provides you with some points to be noted when you operate HULFT.

4.1 Handling of Host Names

The remote hosts targeted for transfer are managed by the 'Host Information' in HULFT. The Host Information manages the Host Type, Kanji Code Type, etc. with the host name using as the key, and does not use the IP Address for this purpose (Excluding K).

It is necessary to register a remote host name in the Host Information of HULFT when carrying out file transfer. In addition, it is necessary to register the host name and IP address in the HOSTS file of each machine or DNS server.

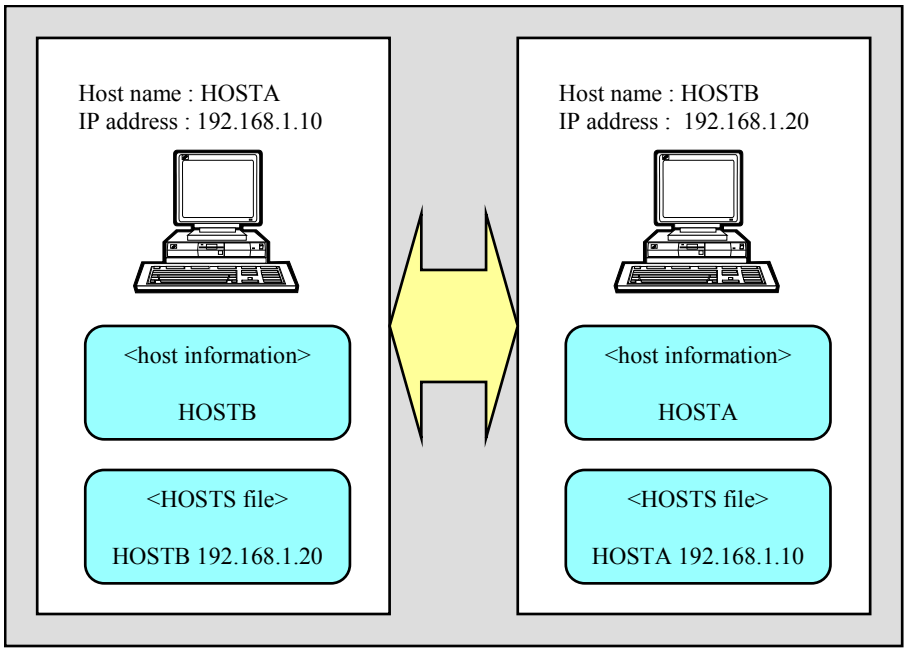


Figure 4.1 Handling of host names

4.2 Local Host Name

It is necessary to determine the local host name before HULFT installation. If your host name recognized by your machine (The local host name) and your host name recognized by the remote host (The remote host name) are different, transfer may not be successful. In order to avoid this, your host name (The local host name) should be determined first, and then register it in the remote machine.

The method of setting the local host name for each model is explained below.

4.2.1 Method of Setting Mainframe

There are 'local host name settings' and 'Selection of upper case or lower case for notification' in the settings related to the local host name.

In 'Select upper case or lower case for notification', you can select whether to notify using upper case or lower case when the local host name is notified to the remote host.

These settings are described in the System Environment Settings file as follows :

(1) System Environment Settings file

HULFT parameter library (HULPRM)

(2) Description method

- Local host name settings
  HOST-NAME=local host name (When within eight characters)
  DOMAIN=domain name (Use this when nine or more characters)
- Selection of upper case or lower case for notification
  HSTCHA=select upper case or lower case for local host name
  (L: lowercase characters, U: uppercase characters)

(3) Default values

- **Local host name**
  None
- **Selection of upper case or lower case for notification**
  Lowercase characters

(4) Note

When both parameters (HOST-NAME, DOMAIN) are defined in the Local host name settings, the 'DOMAIN' parameter is given priority.

The name set in DOMAIN (HOST-NAME) is notified to the remote host in the format specified in HSTCHA.

When the remote host is UNIX, Linux, Windows or Nonstop, the name should be registered in the Host Information of each host using the format specified in HSTCHA, since these host types are case-sensitive.

When the remote host is Mainframe, i5OS or K, register in upper case in the Host Information of each host irrespective of the specification in HSTCHA.

(5) Description Example

```
HSTCHA=L
DOMAIN=HOSTA
```

In the above description, the local host name will be notified as follows :

For UNIX, Linux, Windows, and Nonstop : hosta
For Mainframe, i5OS, K : HOSTA

4.2.2 Method of Setting i5OS

The local host name is set as the 'local host name' by using the Management screen. Uppercase characters are used when the local host name is notified to the remote host.

(1) Method of Setting

'Local Host Name' of the System Environment Settings screen.

(2) Default values

None.

(3) Note

The host name set in Local Host Name is notified to the remote host.

Register it using uppercase characters at the remote host side.

4.2.3 Method of Setting UNIX/Linux

The local host name is set as the 'Local Host Name' by using the Management screen or the System Environment Settings file.

(1) System Environment Settings file

HULPATH/hulenv.conf

• Tag name : myhostname

(2) Method of setting

• Settings by the Management screen

'Local Host Name' of 'System Environment Settings' screen

• Description in the System Environment Settings

myhostname = local host name

(3) Default values

Host name obtained by the hostname command.

(4) Note

The host name set in Local Host Name (myhostname) is notified to the remote host. If the local host name (myhostname) is not set, the default value (host name obtained by hostname command) is notified.

Use uppercase and lowercase characters correctly when registering at the remote host side. When the remote host is Mainframe, i5OS or K, register in upper case in the Host Information of each host.

4.2.4 Method of Setting Windows

The local host name is described in Local Host Name in the System Environment Settings file.

(1) System Environment Settings file

HULPATH\hulenv.cnf

• Tag name: myhostname

(2) Description method

myhostname=local host name

(3) Default values

Host name obtained by the hostname command.

(4) Note

The host name set in Local Host Name (myhostname) is notified to the remote host. If the local host name (myhostname) is not set, the default value (Host name obtained by hostname command) is notified.

Use uppercase and lowercase characters correctly when registering at the remote host side. When the remote host is Mainframe, i5OS or K, register in upper case in the Host Information of each host.

4.3 Receiving from Authorized Host (Using Transfer Group ID)

You can set the Transfer Group ID in the Receive Management Information for receiving operation. If a host is not registered in the Receive Management Information, you can 'refuse' to receive the files from non-registered hosts.

4.3.1 Transfer Group ID

An ID is composed of a maximum of eight alphanumeric characters. This ID is used to describe the remote host name to be targeted for transfer.

The number of host names which can be defined is one to 48 for i5OS, and one to 1000 for other types, respectively.

4.3.2 Specification Location and Usage of Transfer Group ID

The Transfer Group ID is defined in the Send Management Information or the Receive Management Information.

The Transfer Group ID in the Send Management Information is not an optional field. When Send File is issued, this ID is used to decide the host to which the file is transferred.

Meanwhile, the Transfer Group ID in the Receive Management Information is an optional field. Transfer Group ID in the Receive Management Information is not required when the host on the receiving side only waits for the files to be sent without starting any programs.

When the Transfer Group ID is specified in the Receive Management Information, you can omit the parameter of the Host Name at the time of Send Request or you can issue Send Request on the screen.

4.3.3 Check of Send Destination Host Using the Transfer Group ID

You can check whether the host can receive files from a host other than those registered in the specified Transfer Group ID and handles any process as an error if a file is sent from a non-registered host.

There are three options of settings :

Mode 0: Do not check.
Mode 1: Check when any Transfer Group ID is set in the Receive Management Information; do not check if any Transfer Group ID is not set.
Mode 2: Check when any Transfer Group ID is set in the Receive Management Information; handle the process as an error if any Transfer Group ID is not set.

(1) Method of Setting Mainframe

The settings of whether or not to check the send destination host using Transfer Group ID are defined in the System Environment Settings file.

System Environment Settings file

HULFT parameter library (HULPRM)

Description method

TGRPCHK=Transfer Group Check (0, 1, 2)

Default values

0: Does not check

(2) Method of Setting i5OS

The settings of whether or not to check the send destination host using Transfer Group ID are set by using the Management screen.

Method of setting

Transfer Group Check of the System Environment Settings screen

Default values

0: Does not check

(3) Method of Setting UNIX/Linux

The settings of whether or not to check the send destination host using transfer group ID are set by using the Management screen or the System Environment Settings file.

System Environment Settings file

HULPATH/hulenv.conf

Method of setting

- Settings by the Management screen.

  On the System Environment Settings screen, select the submenu of the Send And Receive Related Settings, and specify the field of the Transfer Group Check screen.

- Description in the System Environment Settings file

  tgrpchk=transfer group check (0,1,2)

Default values

0: Does not check

(4) Method of Setting Windows

The settings of whether or not to check the send destination host using Transfer Group ID are defined in the System Environment Settings file.

System Environment Settings file

HULPATH\hulenv.cnf

Description method

tgrpchk=Transfer Group Check (0,1,2)

Default values

0: Does not check

4.4 Timeout Settings

HULFT sets the value for timeout to control processing time when the application executes file transfer or jobs before and/or after file transfer. Moreover, when network failure happened, HULFT sets communication timeout to detect any faults in communication. This section explains varying patterns of timeout and the value used as timeout for each pattern.

4.4.1 Pre-send Job Timeout

In the Pre-send Job, the program to execute the job specified by Pre-send Job is different according to the methods of issuing requests. Therefore, in some cases the value to detect timeout may be different. There are two main patterns as shown below:

- When you execute Send File by Send File command or on the Management screen
- When you execute Send Request or when you execute Send File via HULFT Manager

(1) When you execute Send File by Send File command or on the Management screen

When you execute Send File by Send File command or on the Management screen, the host on the sending side issues a request. The explanation on such case is provided below:

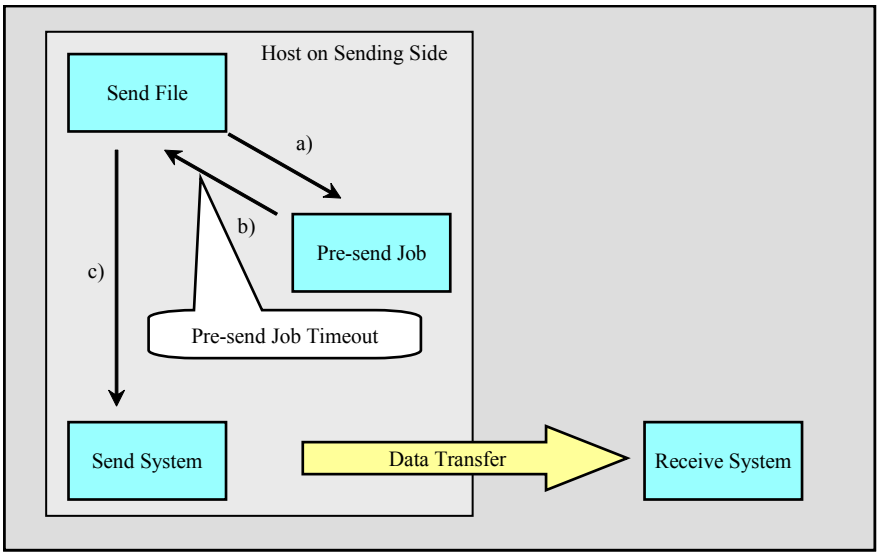


Figure 4.2 Pre-send Job Timeout (ex. Sending started from Send File command)

First, when the Send File command is issued, HULFT starts the User Job specified as a Pre-send Job. a) Send File command does not do anything and remains in waiting mode, until the Pre-send Job is completed. HULFT then receives the result of Pre-send Job. b) After the application confirms successful job termination, HULFT issued a request to carry out transfer to the Send system c).

However, if the time set to wait for the result of the Pre-send Job is not long enough b), the Send File command is terminated due to timeout. In the case of Mainframe and UNIX, HULFT continuously executes the job. However, for i5OS, Linux, and Windows, HULFT forcibly terminates the job after the expiration of timeout period. HULFT sets the timeout period of the Job shown as b) as Job Timeout.

<Specification Method>

Mainframe

Send File command

Startup JCL of the Send File program (XRSNDGO), XRSYSIN Definition Card

• JOBTIME=Timeout at the time of Job Execution (minutes)

Send File from Management screen

Command procedure of Startup command (HULFT) of HULFT and XRSYSIN definition card

• JOBTIME = Timeout at Job Execution (minutes)

i5OS

Send File command (UTLSEND), Send File on the Management screen

System Environment Settings

• JOB Timeout (seconds) (tag name: JOBWTIMEOUT)

UNIX/Linux

Send File command (utlsend), Send File on the Management screen,

System Environment Settings

• Job Termination Timeout (seconds) (tag name: jobwtimeout)

Windows

Send File command (utlsend.exe), Send File on the Management screen,

System Environment Settings

• Job Timeout (seconds) (tag name: jobwtimeout)

(2) When you execute Send Request or When you execute Send File via HULFT Manager

When you execute Send Request or when you execute Send File via HULFT Manager, the request is issued by another host and Request Acknowledge system accepts the request. The explanation on such case is provided below.

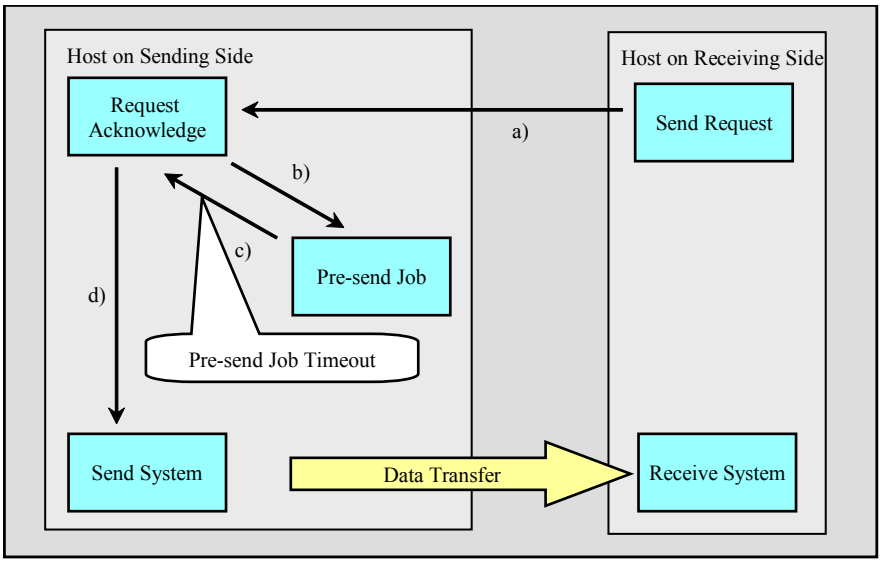


Figure 4.3 Pre-send Job Timeout (Sending started from Send Request command)

When the Send Request command is issued on the receiving side a), firstly Request Acknowledge system accepts the request. Then HULFT starts the User Job specified as a Pre-send Job b). Request Acknowledge system does not do anything and remains in waiting mode, until the Pre-send Job is completed. HULFT then receives the result of Pre-send Job. c) After the application confirms successful job termination, HULFT issued a request to carry out transfer to the Send system. d)

However, if the time set to wait for the result of the Pre-send Job c) is not long enough, the Send Request command is terminated due to timeout. In the case of Mainframe and UNIX, HULFT continuously executes the Job. However, For i5OS, Linux, and Windows, HULFT forcibly terminates the job after the expiration of timeout period. HULFT sets the timeout period of the Job of c) as Job Timeout.

<Specification Method>

Mainframe

Send Request command (XRRCVREQ), Send Request on the Management screen, Send File and Send Request via HULFT Manager

Startup JCL of the Request Acknowledge program (XRACCPT), XRSYSIN Definition Card

• JOBTIME=Timeout at the time of Job Execution (minutes)

i5OS

Send Request command (UTLRECV), Send Request on the Management screen, Send File and Send Request via HULFT Manager

System Environment Settings

• JOB Timeout (seconds) (tag name: JOBTIMEOUT)

UNIX/Linux

Send Request command (utlrecv), Send Request on the Management screen, Send File and Send Request via HULFT Manager

System Environment Settings

• Job Termination Timeout (seconds) (tag name: jobwtimeout)

Windows

Send Request command (utlrecv.exe), Send Request on the Management screen, Send File and Send Request from HULFT Manager

System Environment Settings

• Job Timeout (seconds) (tag name: jobwtimeout)

4.4.2 Receive Completion Notification and Successful Job Timeout

When the Notification in the Receive Management Information is specified as Successful Job Completion, the transfer process will not complete until the User Job specified in the Post-receive Successful Job ends.

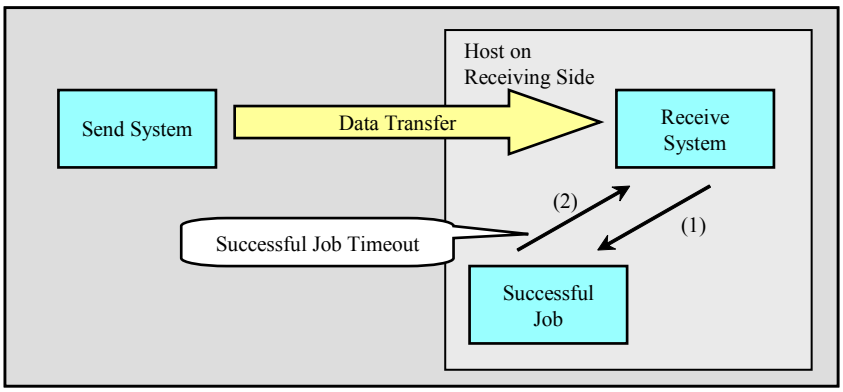


Figure 4.4 Successful Job Timeout

After the file transfer is ended successfully, HULFT activated the User Job specified as Successful Job (1). Until this job is completed, the receiving remains in the waiting status without doing anything else. When HULFT receives the result of the Successful Job (2), the application returns the result which includes the job completion result to the sending side.

However, if the time set to wait for the result of the Successful Job (2) is not long enough, the file transfer is terminated due to timeout even if the Post-receive Successful Job terminated successfully. HULFT specifies the wait time for job in (2) as Job Timeout.

<Specification Method>

Mainframe

Receive

Startup JCL of the Receive Resident program (XRRCV), EXEC statement parameter

• JOBWTIME=Post-receive Job Execution Timeout (min.)

i5OS

Receive

System Environment Settings

• Job Timeout (seconds) (tag name: JOBWTIMEOUT)

UNIX/Linux

Receive

System Environment Settings

• Job Termination Timeout (seconds) (tag name :jobwtimeout)

Windows

Receive

System Environment Settings

• Job Timeout (seconds) (tag name: jobwtimeout)

4.4.3 Timeout When Synchronous Transfer is Specified

Regarding Send File and Send Request, the two modes can be selected, namely HULFT waits for the transfer result (synchronous transfer) and HULFT doesnot wait for the transfer result (asynchronous transfer). When the transfer is executed in Synchronous Transfer mode, it is necessary to set the length of time to wait for the transfer results.

As for the Synchronous Transfer, methods for transfer between processes and/or the programs that wait for processing during Synchronous Transfer differ depending on how you issue the request. Therefore, in some cases the value to detect the Timeout may be different. There are two main patterns as shown below.

- In the case of Send File
- In the case of Send Request

(1) In the case of Send File

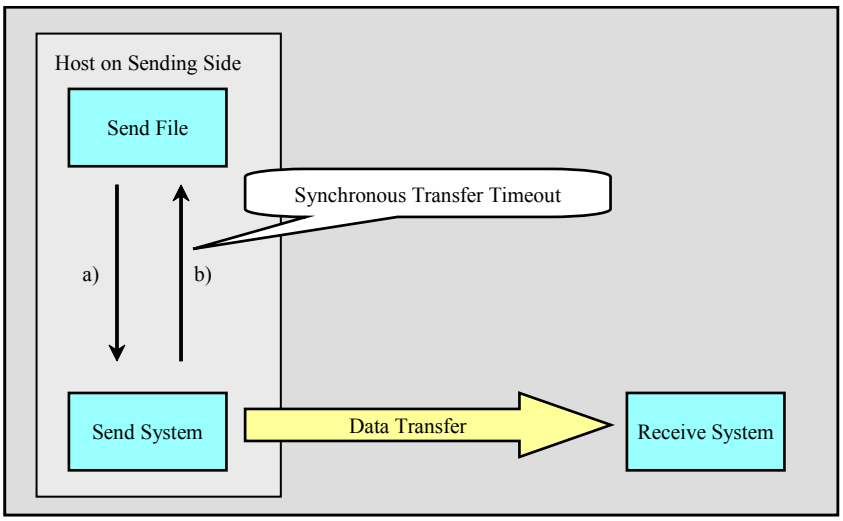


Figure 4.5 Synchronous Transfer Timeout (Send File)

When the Send File command is issued, HULFT requests sending to the Send system a). In the case of Synchronous Transfer, the Send File command remains in the waiting status until the Send system completes the file transfer. The system receives the transfer result after the file transfer, then ends the process b).

However, if the time set to wait for the result of the file transfer b) is not long enough, the Send File command is terminated due to timeout even if the file transfer is ended successfully. HULFT specifies the wait time as Synchronous Transfer Timeout.

[Note]

- **If the Send Management Information fulfills the condition stated below, it is necessary to take into account the execution time of Pre-send Job.**
  - → **Pre-send Job ID is specified.**
- **If the Receive Management Information fulfills all the conditions stated below, it is necessary to take into account the execution time of Post-receive Job.**
  - → **Successful Job ID is specified.**
  - → **'Successful Job Completion' is specified for the Notification.**

<Specification Method>

Mainframe

Send File program (XRSNDGO)

Because Mainframe cannot specify the Synchronous Transfer Timeout, HULFT will not time out due to its setting.

i5OS

Send File command (UTLSEND)

Parameter of the command

• 'W=' wait time (seconds)

If the above parameter is not specified, the timeout value becomes indefinite.

UNIX/Linux

Send File command (utlsend)

Parameter of the command

• -w wait time (seconds)

If the above parameter is not specified, the Utility Timeout in the System Environment Settings is used.

Windows

Send File command (utlsend.exe)

Parameter of the command

• -w wait time (seconds)

If the above parameter is not specified, the Socket Read Timeout in the System Environment Settings is used.

(2) In the case of Send Request

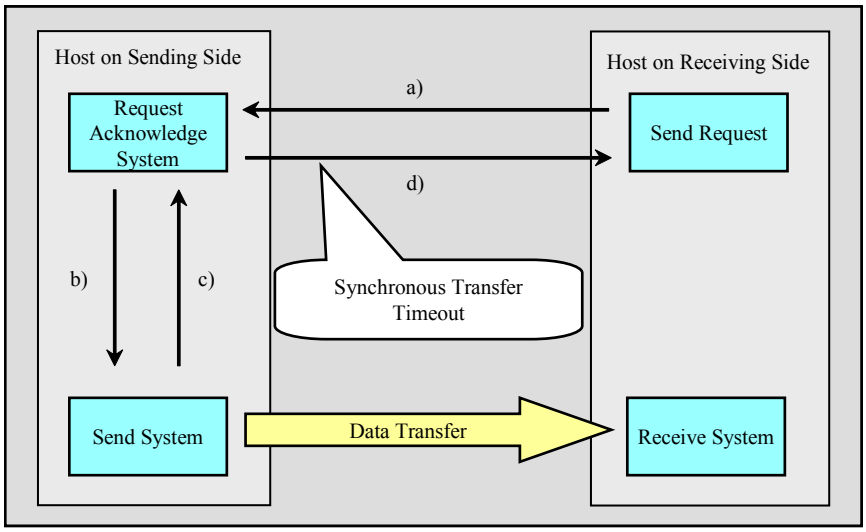


Figure 4.6 Synchronous transfer timeout (Send Request)

When the Send Request command is issued, the request is made to the Request Acknowledge system of the sending side a). The Send Request command remains in waiting status until the result is notified by the Request Acknowledge system.

The Request Acknowledge system requests sending to the Send system b), and waits until the Send system completes the file transfer. After the completion of file transfer, the system returns the transfer result notified by the Send system c) to the Send Request command d).

However, if the time set to wait for the result of the file transfer b) is not long enough, the Send Request command is terminated due to timeout even if the file transfer is ended successfully. HULFT specifies the wait time as Synchronous Transfer Timeout.

[Note]

- **When Request Acknowledge side (sender) is Mainframe, the set value of the Synchronous Transfer Timeout is valid only when all of the following requirements are fulfilled.**
  - → **The versions of HULFT on both request issuing side and Request Acknowledge side are Ver. 6.3 or later.**
  - → **Wait Synchronous Request Acknowledge Mode (REQWAITMODE) in the System Environment Settings on the Request Acknowledge side is set as '1.'**

  In the case other than above , HULFT for Mainframe on the Request Acknowledge side does not time out despite the specification of the Synchronous Transfer Timeout on the issuing side of the Send Request. HULFT on the issuing side of the Send Request times out when HULFT on the Request Acknowledge side does not respond even after the expiration of setting time for the timeout.
- **When the control is not released from Request Acknowledge side despite the expiration of Non Communication Timeout due to error in network and so forth, such case is handled as a communication timeout.***
- **When HULFT on the Request Acknowledge side (sender) is i5OS, you should enable the Send Packet in the System Environment Settings.**
- **When HULFT on Request Acknowledge side (sender) is K, HULFT cannot perform Synchronous Transfer. Instead, HULFT performs Asynchronous Transfer.**

• If the Receive Management Information fulfills all the conditions stated below, the waiting time for the execution of the Successful Job on the receiving side is also included in the Synchronous Transfer Timeout.

→ Successful Job ID is specified.

→ 'Successful Job Completion' is specified for the Notification.

Determine the setting of the Synchronous Transfer Timeout by taking into account above respect.

• Only for XSP, if the Send Management Information fulfills the condition stated below, the waiting time for the execution of the Successful Job is also included in the Synchronous Transfer Timeout.

→ Pre-send Job ID is specified.

Determine the setting of the Synchronous Transfer Timeout by taking into account above respect.

*** For Non Communication Timeout, refer to "4.4.5 Non Communication Timeout."**

<Specification Method>

Mainframe

Send Request program (XRRCVREQ)

Startup JCL, XRCARD Definition Card

• TIME=Synchronous Transfer Timeout (seconds)

If above parameter is not specified, Request Acknowledge will not time out due to its setting. However, Send Request will be treated as Non Communication Timeout when there is no response from Request Acknowledge even if the Non Communication Timeout period expires. For details, refer to "4.4.5 Non Communication Timeout."

i5OS

Send Request command (UTLRECV)

Parameter of the command

• 'W=' Synchronous Transfer Timeout (seconds)

If above parameter is not specified, the Socket Read Timeout in the System Environment Settings is used.

UNIX/Linux

Send Request command (utlrecv)

Parameter of the command

• -w Synchronous Transfer Timeout (seconds)

If above parameter is not specified, Utility Timeout (seconds) of the System Environment Settings is used.

Windows

Send Request command (utlrecv.exe)

Parameter of the command

• -w Synchronous Transfer Timeout (seconds)

If above parameter is not specified, the Socket Read Timeout in the System Environment Settings is used.

4.4.4 Remote Job Timeout When Synchronous Transfer is Specified

In Remote Job Execution, you can select the mode in which HULFT waits for the job completion executed by the remote host (synchronization) or the mode in which the application doesnot wait for the job completion. (asynchronization) When you execute in the mode of waiting for job completion (synchronization), you should specify the length of time to wait for the job results.

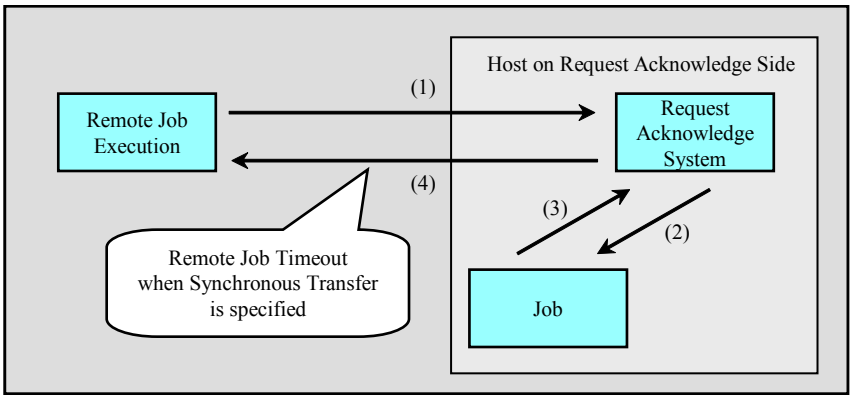


Figure 4.7 Remote Job Timeout

The Remote Job Execution command requests job execution to the Request Acknowledge system of the remote host (1) and enters into the wait status until the job execution is completed.

The Request Acknowledge system executes the User Job (2) and waits for the completion of the job (3).

After the completion of the job execution, the system notifies the job completion result to the Remote Job Execution command (4).

[Note]

- **If the Request Acknowledges side is Mainframe (or if the remote job of Mainframe is executed), the Remote Job Timeout you specified is rounded up to be handled in minute. Ex. If you specify 30 seconds, the value of Remote Job Timeout will be 1 minute (=60 seconds).**
- **If the Request Acknowledges side is K, the Remote Job Timeout will not be applied thus Hulft on the Request Acknowledge side will not time out, even if you specify the value by Remote Job Execution command. In such case, if the processing takes longer than Non Communication Timeout period, the Remote Job Execution command will be subject to Non Communication Timeout.***
- **When the control is not released from Request Acknowledge side despite the expiration of Non Communication Timeout due to error in network and so forth, such case is handled as a communication timeout.***

*** For Non Communication Timeout, refer to "4.4.5 Non Communication Timeout."**

<Specification Method>

Mainframe

Remote Job Execution program (XRRJOBEX)

Startup JCL and XRCARD Definition Card

• TIMEOUT = Wait Time (second)

If above parameter is not specified, 300 (seconds) is applied.

i5OS

Remote Job Execution command (UTLRJOB)

Parameter of the command

• 'W=' wait time (seconds)

If above parameter is not specified, the Job Timeout in the System Environment Settings is used.

UNIX/Linux

Remote Job Execution command (utlrjob)

Parameter of the command

• -w wait time (seconds)

If above parameter is not specified, Job Termination Timeout (seconds) of the System Environment Settings is used.

Windows

Remote Job Execution command (utlrjob.exe)

Parameter of the command

• -w wait time (seconds)

If above parameter is not specified, the Job Timeout in the System Environment Settings is used.

4.4.5 Non Communication Timeout

When the session is established and the file is being transferred, some packets may be lost due to network failure or other reasons.

As long as TCP/IP does not detect this, HULFT also does not detect this and enters into the waiting state. The Non Communication Timeout is set as a restriction on this wait time.

HULFT can detect network failure promptly by making the setting of Non Communication Timeout short.

As for the Non Communication Timeout, the value to detect the timeout might be different according to the difference in processing methods. There are three main patterns as shown below.

- Send and Receive processing
- Request Acknowledge processing
- Send Request When Synchronization is Specified and/or Remote Job Execution.

(1) Send and Receive Processing

During Send and Receive processing, HULFT applies Non Communication Timeout specified by respective local hosts to detect network failure.

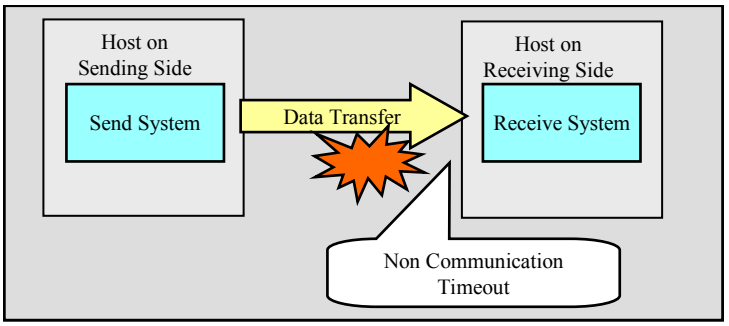


Figure 4.8 Non Communication Timeout
(Send and Receive process)

HULFT sends and receives small amount of packet at fixed intervals to prevent the Communication Timeout during successful processing, when there is time-consuming processing before and after transfer. HULFT adopts the Non Communication Timeout as a wait time for response from packet at the time of network failure to detect failure.

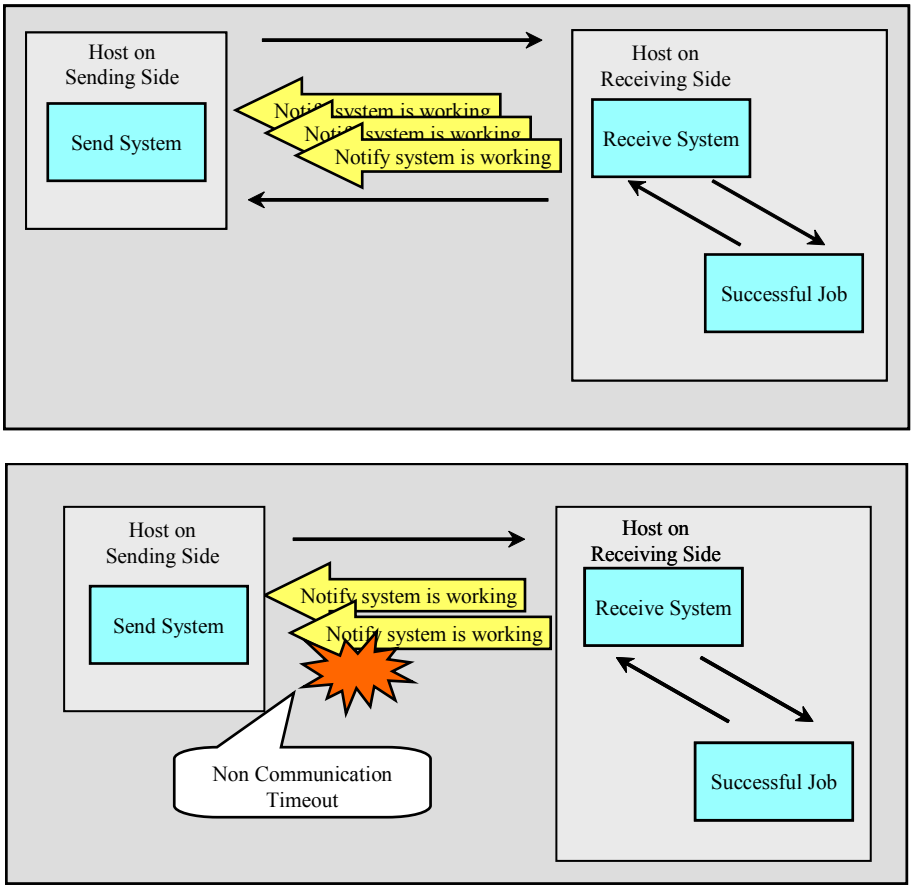


Figure 4.9 Non Communication Timeout
ex. Executing Successful Job when specifying 'Successful Job Completion' for Notification

<Specification method>

Mainframe

■ zOS

Send

Startup JCL of the Send Resident program (XRSND), XRSYSIN Definition Card

• TIMEOUT=Socket Read Timeout (seconds)

Receive

Startup JCL of the Receive Resident program (XRRCV), EXEC sentence parameter

• TIMEOUT=Socket Read Timeout (seconds)

■ VOS3

Send

Startup JCL of the Send Resident program (XRSND), XRSYSIN Definition Card

• TIMEOUT=Socket Read Timeout (seconds)

Receive

Startup JCL of the Receive Resident program (XRRCV), XRSYSIN Definition Card

• TIMEOUT=Socket Read Timeout (seconds)

■ MSP and XSP

HULFT will not time out due to its setting.

Set the Non Communication Timeout (minutes) by the TCMTIME parameter of the ISP operating environment of TISP. Specify a time shorter than the setting for TCMTIME(a value which is 1/10 of TCMTIME is recommended) as a value for Response Confirmation Time (Seconds).

Startup JCL of Send File resident program(XRSND), XRSYSIN Definition Card

• Response Confirmation Time (Tag name: KEEPALIVE)

Startup JCL of Receive File resident program (XRRCV), XRSYSIN Definition Card

• Response Confirmation Time (Tag name: KEEPALIVE)

i5OS

System Environment Settings

• Socket Read Timeout (tag name : SOCKTIME)

UNIX/Linux

System Environment Settings

• Socket Timeout (tag name : socktime)

Windows

System Environment Settings

• Socket Read Timeout (tag name : socktime)

(2) Request Acknowledge Processing

Request Acknowledge system (server program) sets the Non Communication Timeout as wait time for the response from the issued request commands or HULFT Manager.

Meanwhile, request issue systems (client programs) specify the Non Communication Timeout as wait time for the response from Request Acknowledge system.

At the time of network failure, HULFT detects timeout according to the setting of each system.

In HULFT Manager, if the operation (browsing another screens and so forth) is not carried out for the period longer than specified Non Communication Timeout, Non Communication Timeout occurs.

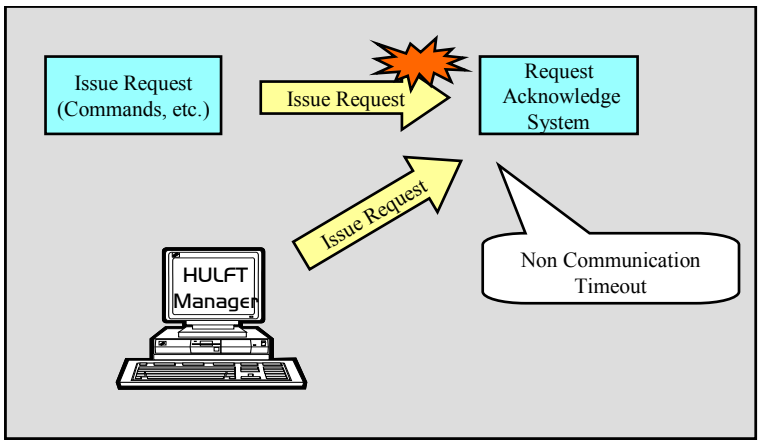


Figure 4.10 Non Communication Timeout
(Request Acknowledge process)

[Remarks] The request issue (client program) is requests for connection shown below:

- Send Request
- Resend Request
- Job Execution Result Notification
- Post-receive Job Result Inquiry
- Remote Job Execution
- Connection Request from HULFT Manager
  (Only when HULFT ManagerConnection Option is introduced)
- Data Conversion Connection Request from HULFT Manager
  (Only when HULFT DataConversion Pro, or HULFT DataConversion Ver.6 for UNIX/Linux is installed)

For the value of the Non Communication Timeout in request issue programs that HULFT Manager uses, refer to the relevant manual.

<Specification method>

Mainframe

■ zOS

Request Acknowledge

Startup JCL of Request Acknowledge resident program (XRACCPT) and EXEC statement parameter

• TIMEOUT=Socket Read Timeout (seconds)

request issue programs

Startup JCL of request issue programs, XRSYSIN Definition Card

• TIMEOUT=Socket Read Timeout (seconds)

■ VOS3

Request Acknowledge

Startup JCL of the Request Acknowledge Resident program (XRACCPT), XRSYSIN Definition Card

• TIMEOUT=Socket Read Timeout (seconds)

request issue programs

Startup JCL of request issue programs, XRSYSIN Definition Card

• TIMEOUT2=Socket Read Timeout (seconds)

■ MSP and XSP

HULFT will not time out due to its setting.

Set the Non Communication Timeout by the TCMTIME parameter of the ISP operating environment of TISP.

i5OS

Request Acknowledge (except for HULFT Manager)

System Environment Settings

• Socket Read Timeout (tag name: SOCKTIME)

Request Acknowledge (HULFT Manager)

System Environment Settings

• HULFT Manager Timeout (Tag name: ADMSOCKTIME)

UNIX/Linux

Request Acknowledge

System Environment Settings

•Request Acknowledge Timeout (Tag name: obssocktime)

request issue programs

System Environment Settings

•Socket Timeout (Tag name: socktime)

Windows

Request Acknowledge (When connecting to Send Request, Resend Request, Job Execution Result Notification, and Remote Job Execution Request)

System Environment Settings

• Socket Read Timeout (tag name: socktime)

Request Acknowledge (When connecting to HULFT Manager, and Post-receive Job Result Inquiry)

System Environment Settings

• HULFT Manager Timeout (tag name: admsocktime)

request issue programs

System Environment Settings

• Socket Read Timeout (tag name: socktime)

(3) Send Request When Synchronization is Specified, or when Remote Job is Executed

HULFT sends and receives small amount of packet at fixed intervals to prevent the communication timeout during transfer processing or job execution, when you specify Send Request and Synchronization mode in Remote Job Timeout for HULFT Ver.6.3 or later. At the time of network failure, HULFT adopts the Non Communication Timeout as the wait time for the packet response to detect failure.

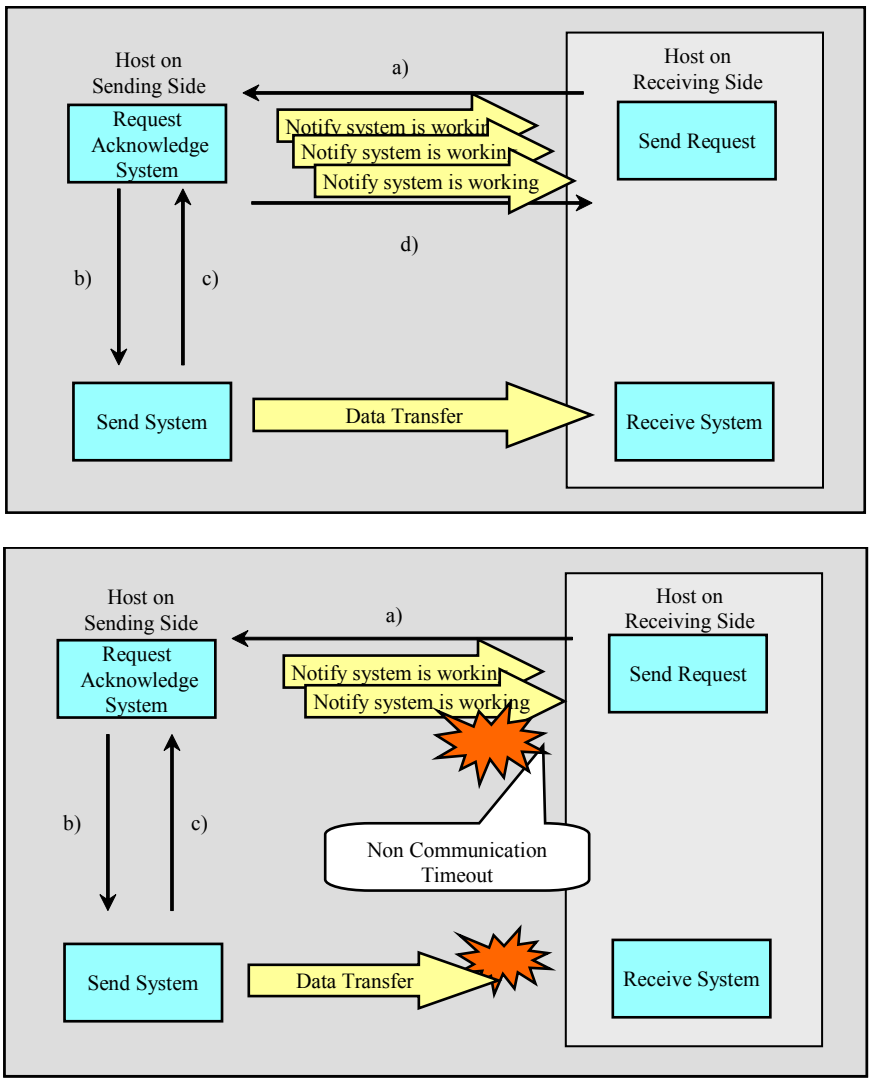


Figure 4.11 Non Communication Timeout
(Ex. Send Request on Synchronous Transfer)

[Note] **In the following cases, HULFT doesnot carry out packet sending and receiving to prevent the Non Communication Timeout.**

- **When either of HULFT on the Request Issue side or on the Request Acknowledge side is lower than Ver.6.3**
- **In Send Request, when there is no specification for wait time in the parameter of the command.**

Therefore, when you make the setting of Non Communication Timeout short, some models adopt the Communication Timeout (Non Communication Timeout compatible with lower version) which takes into account the time for Receive

processing result (Synchronous Transfer Timeout, Remote Job Timeout When Synchronous Transfer).

Refer to "4.4.3 Timeout When Synchronous Transfer Is Specified" for the Synchronous Transfer Timeout, and "4.4 4. Remote Job Timeout When Synchronous Transfer Is Specified" for Remote Job Timeout, respectively.

<Specification method>

Mainframe

■ zOS

Send Request

Startup JCL of Send Request program (XRRCVREQ), XRSYSIN Definition Card

• TIMEOUT=Socket Read Timeout (seconds)

Remote Job Execution

Startup JCL of Remote Job Execution program (XRRJOBEX), XRSYSIN Definition Card

• TIMEOUT=Socket Read Timeout (seconds)

■ VOS3

Send Request

Startup JCL of Send Request program (XRRCVREQ), XRSYSIN Definition Card

• TIMEOUT2=Socket Read Timeout (seconds)

Remote Job Execution

Startup JCL of Remote Job Execution program (XRRJOBEX), XRSYSIN Definition Card

• TIMEOUT2=Socket Read Timeout (seconds)

■ MSP and XSP

HULFT will not time out due to its setting.

Set the Non Communication Timeout (minutes) by the TCMTIME parameter of the ISP operating environment of TISP. Specify a time which is shorter than the setting for TCMTIME(a value which is 1/10 of TCMTIME is recommended) as a value for Response Confirmation Time (seconds).

Startup JCL of Send Request resident program (XRRCVREQ), XRSYSIN Definition Card

• Response Confirmation Time (Tag name: KEEPALIVE)

Startup JCL of Remote Job Execution Program (XRRJOBEX) and XRSYSIN definition card

• Response Confirmation Time (Tag name: KEEPALIVE)

i5OS

Send Request Command (UTLRECV)

System Environment Settings

• Socket Read Timeout (tag name: SOCKTIME)

* Non Communication Timeout compatible with lower version

The value of Socket Read Timeout or the value of Synchronous Transfer Timeout + 60 seconds, whichever is greater

[Note] In the following cases, HULFT applies Non Communication Timeout that is compatible with lower version.

- **There is no specification of wait time (-w) by parameter of the command**
- **Send Packet in the System Environment Settings is set as 'Disable Packet Transmission.'**

Remote Job Execution command (UTLRJOB)

System Environment Settings

• Socket Read Timeout (tag name : SOCKTIME)

* Non Communication Timeout compatible with lower version

The value of Socket Read Timeout or the value of Synchronous Transfer Timeout + 60 seconds, whichever is greater

UNIX/Linux

Send Request command (utlrecv)

System Environment Settings

• Socket Timeout (Tag name: socktime)

* Non Communication Timeout compatible with lower version

The value of Socket Timeout or the value of Synchronous Transfer Timeout + 60 seconds, whichever is greater

[Note] When the Synchronous Transfer Timeout is not specified as a command parameter, the Non Communication Timeout that is compatible with lower version is applied.

Remote Job Execution command (utlrjob)

System Environment Settings

• Socket Timeout (Tag name: socktime)

* Non Communication Timeout compatible with lower version

The value of Socket Timeout or the value of Synchronous Transfer Timeout + 60 seconds, whichever is greater

Windows

Send Request command (utlrecv)

System Environment Settings

• Socket Read Timeout (tag name: socktime)

* Non Communication Timeout compatible with lower version

The value of the Socket Read Timeout or the value of the Synchronous Transfer Timeout + 60 seconds, whichever is greater

[Note] When the Synchronous Transfer Timeout is not specified as a command parameter, the Non Communication Timeout that is compatible with lower version is applied.

Remote Job Execution command (utlrjob)

System Environment Settings

• Socket Read Timeout (tag name: socktime)

* Non Communication Timeout compatible with lower version

The value of the Socket Read Timeout or the value of the Synchronous Transfer Timeout + 60 seconds, whichever is greater

4.5 Operation of the Log File

HULFT outputs a log record for each host everytime the application performs transfer.

Because the record of the log is not deleted automatically, it is necessary to delete log records on a regular basis.

The types of Send and Receive Logs of each host and the deletion method is explained below.

4.5.1 Operation for Mainframe

(1) HULFT Management screen (status inquiry)

Output destination file name

Send and Receive Log Information file (VSAM)

This file is created at the time of installation. Each of log information (Send, Receive, Post-send Job, Post-receive Job, Resend and Request Acknowledge) is managed in one file.

Method of deletion

a) Log record deletion (Manual operation)

It is possible to delete from the deletion programs (XRLOGDEL, XRRQLDEL) or HULFT Management screen.

Refer to the *Operation Manual* for the details on the deletion method.

b) Log record deletion (Automatic execution)

Log records can be deleted automatically by setting the following parameters in the System Environment Settings.

LOGDELTHRESHOLD=Log Deletion Threshold

LOGKEEPCOUNT=Log Keep Count

Example:

The following setting keeps the latest 200 records and deletes the remaining 800 records automatically when the number of log records exceeds 1000.

LOGDELTHRESHOLD=1000

LOGKEEPCOUNT=200

c) Re-creation of Send and Receive Log files

Even if the above-mentioned methods a) and b) are executed, increase in the unused space of DASD may cause problems in operation efficiency.

We recommend that you re-create the log files after you delete log records.

Refer to the *Operation Manual* for the details of the re-creation method.

(2) System console

Output destination file name

System console

This is output in a string starting with 'XR.' Error codes (information) and so forth, returned from Operation System are also displayed. You can investigate the trouble based on this information.

Method of deletion

Follow the system console management method.

4.5.2 Operation for i5OS

(1) HULFT Management screen (status inquiry)

Output destination file name

Send Log file (SNDLOG)

This is where the Send Log is stored.

Receive Log file (RCVLOG)

This is where the Receive Log is stored.

Post-send Job Execution Log File (EXECLOGS)

This is where the execution log of Post-send Job is stored.

Post-receive Job Execution Log File (EXECLOGR)

This is where the execution log of Post-receive Job is stored.

Request Acknowledge Log file (REQLOG)

This is where the Request Acknowledge Log is stored.

Resend Queue file (SDDREQ)

This is where the status of resend is stored.

Method of deletion

a) Log record deletion (manual operation)

It is possible to delete log records by utilities (UTLLOG, UTLOBSRM, UTLRESENDRM) to delete log or on HULFT Management screen.

Refer to the *Operation Manual* for the details on the deletion method.

b) Log record deletion (automatic execution)

Except for Resend Queue file, log records are deleted automatically by setting the following parameter in the System Environment Settings:

LOGDELCOUNT=Log Deletion Threshold

Example:

The following setting deletes one log record, which is the oldest one, when the number of log records exceeds 1000.

LOGDELCOUNT=1000

(2) Log to be output at resident job startup (if specified)

Output destination file name

File arbitrarily specified by user

Method of deletion

Delete the relevant files physically, or delete by using a user application.

[Note] When you delete files physically, firstly you must stop HULFT.

4.5.3 Operation for UNIX/Linux

(1) HULFT Management screen (status inquiry)

Output destination file name

Send Log file (hulsndlog.dat)

This is where the Send Log is stored.

Receive Log file (hulrcvlog.dat)

This is where the Receive Log is stored.

Post-send Job Execution Log file (hulexeclogs.dat)

This is where the execution log of Post-send Job is stored.

Post-receive Job Execution Log file (hulexeclogr.dat)

This is where the execution log of Post-receive Job is stored.

Request Log file (hulaccreqlog.dat)

This is where the Request Log is stored.

Send Control file (sddsendlist.dat)

This is where the status of resend is stored.

Method of deletion

a) Log record deletion (manual operation)

It is possible to delete log records by utilities (utllog, utlobsrm, and utlresendrm) to delete log records or on HULFT Management screen.

Refer to the *Operation Manual* for the details on the deletion method.

b) Log record deletion (automatic execution)

Log records can be deleted automatically by setting the following parameters in the System Environment Settings.

logdelcount=Log Switch Count

Example:

The following setting renames the current log file and creates fresh log file when the number of log records in a log file exceeds 1000. Further, when the number of log records in current log file exceeds another 1000, HULFT deletes the renamed log file and renames the current log file. (When the total number of log records reaches 2000, HULFT deletes the records to keep the number of log records 1000.)

logdelcount=1000

[Note] As for the Send Control file, HULFT doesnot carry out automatic deletion of log records explained in b).

(2) Trace Log

Output destination file name

File arbitrarily specified by user

Message are output to the specified file by setting the following parameter in System Environment Settings.

tlogmode=Trace Output Mode

tlogfile=Trace Output File Name

tlogsize=Trace Output File Size

Method of deletion

There are settings for the maximum size of log file and the backup method on the System Environment Settings screen. In accordance with these settings, HULFT carries out deletion automatically.

(3) System log (SYSLOG)

Output destination file name

System log (SYSLOG)

Method of Deletion

In the case of Files arbitrarily specified by user, delete the relevant files physically, or delete the files by using a user application.

For system log (SYSLOG), follow the system console management method.

[Note] When you delete files physically, firstly you must stop HULFT.

4.5.4 Operation for Windows

(1) HULFT Management screen (status inquiry)

Output destination file name

Send Log file (hulsndlg.dat)

This is where the Send Log is stored.

Receive Log file (hulrcvlg.dat)

This is where the Receive Log is stored.

Post-send Job Execution Log file (hulexlgs.dat)

This is where the execution log of Post-send Job is stored.

Post-receive Job Execution Log file (hulexlgr.dat)

This is where the execution log of Post-receive Job is stored.

Request Acknowledge Log file (hulobslg.dat)

This is where the Request Acknowledge Log is stored.

Resend Status file (sddreqls.dat)

This is where the status of re-sending is stored.

Method of deletion

a) Log record deletion (Manual operation)

It is possible to delete log records by utilities (utllog.exe, utlobsrm.exe, and utlrsdrm.exe) to delete log or on HULFT Management screen.

Refer to the *Operation Manual* for the details on the deletion method.

b) Log record deletion (Automatic execution)

Log records can be deleted automatically by setting the following parameters in the System Environment Settings file.

logdelcount=Log Switch Count

Example:

The following setting renames the current log file and creates fresh log file when the number of log records in a log file exceeds 1000. Further, when the number of log records in current log file exceeds another 1000, HULFT deletes the renamed log file and renames the current log file. (When the total number of log records reaches 2000, HULFT deletes the records to keep the number of log records 1000.)

logdelcount=1000

(2) HULFT console

Output destination file name

Console log (hulcon.log)

Method of deletion

There are settings for the Console Log Size and the Console Log Backup on the System Environment Settings screen. In accordance with these settings, HULFT carries out deletion automatically.

4.6 Send and Receive File Lock

The locking methods when HULFT opens a Send file and/or a Receive file differ depending upon the settings of management information, or upon the difference between the locking used in HULFT and the locking applied by user jobs.

4.6.1 Method of Locking

This section explains the locking methods used in HULFT.

For details, refer to "Chapter 3 Exclusive Control between HULFT and User Job."

Mainframe

HULFT allocates the file in either shared mode (DISP=SHR) or in exclusive mode (DISP=OLD) before the file is opened.

i5OS

The locking method differs depending on the setting values of the Lock Type.

When '0' is set, the file is locked by the lock() function before the file is opened.

When '1' is set, the file is locked by the ALCOBJ command before the file is opened.

UNIX/Linux

HULFT locks the target file by the lockf () function after the file is opened. If the file is locked by the lockf() function and closed without unlocking it, the file is automatically unlocked.

Meanwhile, OS that is capable of handling large size file uses lockf64() function.

As to the Receive file, in addition to the locking method with lockf() function, HULFT locks a real file name with its unique lock method before the file is opened. (Available from Ver.6.3.1 onwards).

This is a locking method of Receive file in HULFT Receive system and it controls the locking in Receive File Lock Control file ($HULPATH/.#hulrcv_file.lock).

[Remarks] Regarding the exclusive control between HULFT and user jobs in Receive file, you should establish interface using the locking method that employs lockf() function. You cannot use the unique locking method of HULFT against the real file name.

Windows

HULFT locks the file with the mutex object before the file is opened.

Further, the application specifies shared mode by using CreateFile() when opening the file.

4.6.2 Setting of File Open and Lock

In HULFT, there are several settings related to file opening and locking when you handle Send and Receive files. Setting values on the sending side is used for the Send file, while setting values on the receiving side is used for the Receive file. For the details on setting values, refer to "Chapter 3 System Environment Settings."

Mainframe

Send

When you use the Send file registered in Send Management Information:

System Environment Settings file

- Send File Retry Count (tag name: SNDRETRY)
- Send File Retry Interval (tag name: SNDRTIME)

When you specify the Send file by DD(FD) statement in Send program JCL:

HULFT keeps waiting until the Send file is unlocked when Send File command is issued.

Receive

Startup JCL of the receive resident programs (XRRCV), EXEC statement parameter

- Mode When the Receive File is Being Used (parameter name : RWAIT)
- Receive File Retry Count (parameter name: RCVRETRY)
- Receive File Retry Interval (parameter name: RCVRTIME)

i5OS

Send

System Environment Settings file

- Lock Type (tag name: LOCKTYPE)
- Retry Count on Locked Send File (tag name: LOCKCNT)
- Retry Interval on Locked Send File (tag name: LOCKTIME)

Receive

System Environment Settings file

- Lock Type (tag name: LOCKTYPE)
- Retry Count on Locked Receive File (tag name: RCVLOCKCNT)
- Retry Interval on Locked Receive File (tag name: RCVLOCKTIME)
- Retry Count on Locked System File (tag name: SYSLOCKCNT)
- Retry Interval on Locked System File (tag name: SYSLOCKTIME)

UNIX/Linux

Receive

System Environment Settings file

- Receive Open Mode (tag name: rcvfilewait)

Windows

Send

System Environment Settings file

- Retry Count on Locked Send File (tag name: lockcnt)
- Retry Interval on Locked Send File (tag name: locktime)

Receive

System Environment Settings file

- Receive File Lock Standby (tag name : rcvfilelockwait)
- Retry Count on Locked Receive File (tag name : rretrycnt)
- Retry Interval on Locked Receive File (tag name : rretrytime)

4.6.3 Send File Mode

Regarding Send file, the locking methods are different depending on the setting values of Send File Mode in the Send Management Information.

(1) When the Send File Mode is set to 'Keep'

Mainframe

HULFT allocates a Send file in shared mode.

When another HULFT or non-HULFT job has already obtained the Send file in exclusive mode, HULFT retries for accessing the file based on the settings of the Send File Retry Count and the Send File Retry Count Interval. When HULFT or non-HULFT job releases the control of the file, OS carries out the allocation of the file.

i5OS

HULFT opens the file without locking the Send file.

The locking status of the opened file becomes *SHRRD in the case of physical data file and *SHRNUP in the case of archive file, respectively.

UNIX/Linux

HULFT opens the file without locking the Send file.

The application does not check whether or not the file is locked by a non-HULFT process.

Windows

HULFT opens the file in shared-for-read mode without locking the Send file with mutex object.

If a non-HULFT process has opened the Send file without sharing, HULFT retries for opening based on the setting of the Retry Count on Locked Send File and the Retry Interval on Locked Send File.

(2) When the Send File Mode is set to 'Clear' or 'Lock'

Mainframe

HULFT allocates the Send file in exclusive mode.

When another HULFT or a non-HULFT job has already obtained the Send file in exclusive mode, HULFT retries for accessing the file based on the settings of the Send File Retry Count and the Send File Retry Count Interval. When HULFT or non-HULFT job releases the control of the file, OS carries out the allocation of the file.

i5OS

In the case of the physical data files, HULFT locks the Send file with lock status as *EXCL before it opens the file. If another HULFT or a non-HULFT job has already locked the Send file, the file is locked based on the setting of Retry Interval on Locked Send File. In the case of archive files, the Send file is opened without being locked. In such cases, the locking status of the opened file becomes *SHRNUP.

UNIX/Linux

HULFT locks the Send file after the file is opened.

If another HULFT or a non-HULFT process has already locked the Send file, the process stops until the file is unlocked.

Windows

HULFT opens the Send file not in sharing state, without locking it with mutex object.

If a non-HULFT process has already opened the Send file without sharing, HULFT retries for opening the file based on the setting of the Retry Count on Locked Send File and the Retry Interval on Locked Send File.

(3) When the Send File Mode is set to 'Delete'

Mainframe

Same as 4.6.3 (2) When the Send File Mode is set to 'Clear' or 'Lock'

i5OS

Same as 4.6.3 (2) When the Send File Mode is set to 'Clear' or 'Lock'

UNIX/Linux

Same as 4.6.3 (2) When the Send File Mode is set to 'Clear' or 'Lock'

Windows

After locking the Send file with mutex object, HULFT opens the file without sharing.

If another HULFT or a non-HULFT process has already locked the Send file with mutex object or they have opened the file without sharing, HULFT retries for opening the file based on the setting of the Retry Count on Locked Send File and the Retry Interval on Locked Send File.

4.6.4 Receive File Mode

In the case of Receive file, the locking method differs depending on the setting values of Receive Mode and Notification of the Receive Management Information.

This section also explains the locking methods in the case of creating a temporary file, such as in generation management.

(1) When the Receive Mode is set to 'Single Receive'

Mainframe

HULFT allocates the Receive file in exclusive mode.

When the Receive file has already been obtained in another Receive job of HULFT, the Receive process either waits until the the control of the file is released or ends in error, depending on the setting of the Mode When the Receive File is Being Used.

When a non-HULFT job has already obtained the Receive file in shared mode or in exclusive mode, HULFT retries for accessing the file, based on the setting of the Receive File Retry Count and the Receive File Retry Intervals. OS carries out the allocation of the file when another HULFT or a non-HULFT job release the control of the file.

i5OS

In the case of physical data files, HULFT locks the Receive file assuming the locking status as *EXCLRD before the application opens the file. If another HULFT or non-HULFT job has already locked the Receive file, HULFT locks the file based on the settings of the Retry Count on Locked Receive File and the Retry Interval on Locked Receive File.

In the case of archive files, HULFT opens the Receive file without locking. In such cases, the locking status of the opened file becomes *SHRNUP.

If another HULFT or non-HULFT job has already locked the Receive file, HULFT opens the file based on the settings of the Retry Count on Locked Receive File and the Retry Interval on Locked Receive File.

UNIX/Linux

HULFT locks the real file names with its unique locking method before it opens the Receive file. In addition, after opening the Receive file, HULFT locks the file with the lockf() function.

As soon as HULFT terminates transfer, the application closes the Receive file. (Namely, HULFT releases the lock with lockf() function)

HULFT releases its unique lock against the real file name without waiting for establishing interface with CSV or XML, nor the termination of the Successful Job or the Unsuccessful Job.

If another HULFT or non-HULFT process has already locked the Receive file, Receive process either waits until it is unlocked or ends in error, depending on the setting of the Receive Open Mode.

Windows

After locking the Receive file with mutex object, HULFT opens the file without sharing.

If another HULFT or a non-HULFT process has already locked the Receive file with mutex object, the Receive process either waits until it is unlocked or ends in error, depending on the setting of the Receive File Lock Standby.

If a non-HULFT process has already opened the Receive file, HULFT opens the file based on the settings of the Retry Count on Locked Receive File and the Retry Interval Count on Locked Receive File.

(2) When Receive Mode is set to 'Multiple Receive'

Mainframe

HULFT allocates receive work file in exclusive mode.

When any of receive jobs of HULFT have already obtained the receive work file, the Receive process either waits until the control of the file is released or ends in error, depending on the setting of the Mode When the Receive File is Being Used.

If a non-HULFT job has already obtained the receive work file in shared mode or in exclusive mode, HULFT retries for accessing the file, based on the settings of the Receive File Retry Count and Receive File Retry Interval. OS carries out the allocation of the file when the control of the work file is released.

i5OS

In the case of physical data files, HULFT opens the receive work file without locking it and writes the received data. After the completion of data receiving, HULFT locks the Receive file with the lock status as *EXCLRD before the file is opened. After HULFT writes appended data in the work file, the application deletes the receive work file.

When another HULFT or a non-HULFT job has already locked the Receive file, HULFT locks the file based on the setting of the Retry Count on Locked Receive File and the Retry Interval on Locked Receive File.

Regarding archive files, HULFT cannot perform receiving in Multiple Receive mode.

UNIX/Linux

HULFT locks the receive work file with lockf() function after the file is opened. Then the application writes the received data into the file.

After the completion of data receiving, HULFT locks the real file name by using its unique lock before the it opens the Receive file. In addition, the application locks the file with lockf() function after the file is opened. After locking the file, HULFT writes appended data obtained from the receive work file, then it deletes the receive work file after the completion of writing.

As soon as HULFT terminates transfer, the application closes the Receive file. (Namely, HULFT releases the lock with lockf() function). HULFT releases its unique lock after the completion of various processing, namely outputting logs, Successful/Unsuccessful Jobs, or establishing interface with CSV or XML.

If another HULFT or a non-HULFT process has already locked the Receive file, the application waits until the file is unlocked, regardless of the setting in the Receive Open Mode.

Windows

HULFT opens the receive work file in not sharing state, without locking it with mutex object and then writes the received data. After the completion of data receiving, HULFT locks the Receive file with mutex object, opens the file without sharing, and then writes aappended data obtained from the work file. After the writing is completed, HULFT deletes the Receive work file.

If another HULFT or a non-HULFT process has already opened the receive work file without sharing, the Receive process ends in error. In addition, if another HULFT or a non-HULFT process has already locked the Receive file with mutex object, the Receive process either ends in error or waits until the file is unlocked, depending on the setting of the Receive File Lock Standby.

If a non-HULFT process has already opened the Receive file, the file is opened based upon the setting of the Retry Count on Locked Receive File and the Retry Interval Count on Locked Receive File.

(3) When the Generation File is set to 'Enabled'

Mainframe

Same as that of 4.6.4 (1) When the Receive Mode is 'Single Receive'

However, the file names allocated to exclusive mode should be the one registered in the Receive Management Information, not in the file that is numbered Generation File Number.

i5OS

In the case of physical data files, HULFT locks the receive work file with the locking status as *EXCLRD before the file is opened and writes the received data into the file.

After the completion of data receiving, the application locks the Generation File Information file with the lock status as *EXCLRD before the file is opened to obtain the Generation File Number. Then HULFT renames the receive work file into the name which is numbered Generation File Number, and unlocks Generation File Information file

If another HULFT or a non-HULFT job has already locked the Receive work file or Receive file, HULFT locks the file based on the setting of the Retry Count on Locked Receive File and the Retry Interval on Locked Receive File.

Also, if another HULFT job or a non-HULFT job has already locked the Generation File Information file, HULFT locks the file based on the setting of the Retry Count on Locked System File and the Retry Interval on Locked System File.

In the case of archive files, HULFT opens the file without locking the Receive work file. In such cases, the locking status of the opened file becomes *SHRNUP. After the completion of data receiving, the application locks the Generation File Information file with the locking status as *EXCLRD before the file is opened to obtain the Generation File Number. Then it rename the receive work file into the name which is numbered Generation File Number to unlock the Generation File Information file.

If another HULFT or a non-HULFT job has already locked the receive work file or Receive file, HULFT locks the file based on the setting of the Retry Count on Locked Receive File and the Retry Interval on Locked Receive File.

If the another HULFT or a non-HULFT job has already locked the Generation File Information file, HULFT locks the file based on the setting of the Retry Count on Locked System File and the Retry Interval on Locked System File.

UNIX/Linux

After opening the Receive work file, HULFT locks the work file with lockf() function and writes the received data into the file. After the completion of data receiving, HULFT opens the Generation File Information file at first and then locks the Generation File Information file by using lockf() function to obtain Generation File Number. Then HULFT renames the receive work file into the name which is numbered Generation File Number to unlock the Generation File Information file.

If you enable the Generation File, HULFT does not use its unique lock against real file names.

If another HULFT or a non-HULFT process has already locked the Generation File Information file, HULFT waits until it is unlocked.

Windows

HULFT opens the receive work file in not sharing state, without locking it with mutex object and writes the received data into the file. After the completion of data receiving, HULFT obtains the Generation File Numbers. Then the application rename the receive work file into the name which is numbered Generation File Number.

If another HULFT or a non-HULFT process has already opened the receive work file, the Receive process ends in error.

If another HULFT or a non-HULFT process has already opened the Receive file which is numbered Generation File Number, the Receive process ends in error.

(4) When Notification is set to 'Successful Job Completion'

Mainframe

Same as that of 4.6.1 (1) When the Receive Mode is 'Single Receive.'

i5OS

In the case of a physical data file, HULFT unlocks the files after the completion of data receiving, where the Notification is set to the 'Receive Completion.' However, when the field is set to the 'Successful Job Completion,' the file remains in locked status during the Successful Job. HULFT unlocks the file after the completion of the Successful Job.

In the case of archive files, HULFT unlocks the files after the completion of data receiving, regardless of the setting of the Notification.

UNIX/Linux

If the Notification is set to 'Receive Completion,' HULFT unlocks its unique lock against real file name without waiting for the completion of the Successful Job. If the field is set to 'Successful Job Completion,' HULFT waits for the completion of the Successful Job to unlock the file.

Windows

If the Notification is set to 'Receive Completion,' HULFT releases the lock with mutex object after the completion of data receiving. If the field is set to the 'Successful Job Completion,' HULFT closes the Receive file after the completion of data receiving and releases the lock with mutex object after the completion of the Successful Job.

~ Memo ~

Appendix 1

I/O Format of Operation Log

App.1.1 Operation Log File Format

Table App.1.1 File Access Log

<table>
<tr><th>Field Name</th><th>Maximum Length</th><th>Classification</th><th>Details</th></tr>
<tr><td>Execution Date</td><td>10</td><td>Fixed Length</td><td>Date of file access</td></tr>
<tr><td>Execution Time</td><td>12</td><td>Fixed Length</td><td>Time of file access</td></tr>
<tr><td>Execution Start Date</td><td>10</td><td>Fixed Length</td><td>Start date of request issuance or command execution</td></tr>
<tr><td>Execution Start Time</td><td>12</td><td>Fixed Length</td><td>Start time of request issuance or command execution</td></tr>
<tr><td>User ID (OS)</td><td>561</td><td>Variable Length</td><td>OS User ID of the user who performed the operation *1</td></tr>
<tr><td>User ID
(Management Screen)</td><td>32</td><td>Variable Length</td><td>Management Screen Security User ID of the user who performed the operation *1</td></tr>
<tr><td>Starting Host Name</td><td>68</td><td>Variable Length</td><td>Name of the host that started the operation</td></tr>
<tr><td>Latest Identifier</td><td>34</td><td>Variable Length</td><td>Identifier that links the record output to the Operation Log file with the record in the Send/Receive/Request Acknowledge Log
(Identifier at the time of delivery, when receiving data accumulated in HULFT-HUB Server)</td></tr>
<tr><td>Starting Identifier</td><td>34</td><td>Variable Length</td><td>Identifier that links the record output to Operation Log file with the record in Send/Receive/Request Acknowledge Log
(Identifier at the time of accumulation, when receiving data accumulated in HULFT-HUB Server)</td></tr>
<tr><td>Execution Host Name</td><td>68</td><td>Variable Length</td><td>Name of the host that executed the transaction</td></tr>
<tr><td>Operation Log
User-specified Character</td><td>HULFT: 8
HUB: 100</td><td>Variable Length</td><td>Characters to identify the environment that outputs the Operation Log</td></tr>
<tr><td>File Access Type</td><td>10</td><td>Variable Length</td><td>Types of file access<table><tr><th>File Access Types</th><th>Description</th></tr><tr><td>NEW</td><td>New(Add)</td></tr><tr><td>UPDATE</td><td>Modify</td></tr><tr><td>DELETE</td><td>Delete</td></tr><tr><td>COMPRESS</td><td>Compress(Except for Mainframe)</td></tr><tr><td>COMMIT</td><td>Commit(Only for HULFT-HUB)</td></tr><tr><td>ROLLBACK</td><td>Rollback(Only for HULFT-HUB)</td></tr></table></td></tr>
<tr><td>System File Type</td><td>60</td><td>Variable Length</td><td>Object of File Access *2</td></tr>
<tr><td>File Key Information</td><td>MF: 500
UX: Unlimited
NT: Unlimited
AS: 110
HUB: Unlimited</td><td>Variable Length</td><td>Details of File Access *2</td></tr>
</table>

*1 Capable of multi byte code. Regarding i5OS, the ID is fixed length (105 bytes).

*2 Refer to "App.1.2 System File List."

Each field is enclosed within enclosed character (") and set off with delimiter (,) to be output.

Example:

```
"2008/07/01","12:34:56.789","2008/07/01","12:34:56.789","user01","user01",
"HOST01","AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA","AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA",
"HOST01","","NEW","HULSND","FILE01"
```

Table App.1.2 Command Execution Log

<table>
<tr><th>Field Name</th><th>Maximum Length</th><th>Classification</th><th>Details</th></tr>
<tr><td>Execution Date</td><td>10</td><td>Fixed Length</td><td>Date of command execution</td></tr>
<tr><td>Execution Time</td><td>12</td><td>Fixed Length</td><td>Time of command execution</td></tr>
<tr><td>Execution Start Date</td><td>10</td><td>Fixed Length</td><td>Start date of request issuance or command execution</td></tr>
<tr><td>Execution Start Time</td><td>12</td><td>Fixed Length</td><td>Start time of request issuance or command execution</td></tr>
<tr><td>User ID(OS)</td><td>561</td><td>Variable Length</td><td>OS User ID of the user who performed the operation *1</td></tr>
<tr><td>User ID
(Management Screen)</td><td>32</td><td>Variable Length</td><td>Management Screen Security User ID of the user who performed the operation</td></tr>
<tr><td>Starting Host Name</td><td>68</td><td>Variable Length</td><td>Name of the host that started the operation</td></tr>
<tr><td>Latest Identifier</td><td>34</td><td>Variable Length</td><td>Identifier that links the record output to the Operation Log file with the record in the Send/Receive/Request
Acknowledge Log
(Identifier at the time of delivery, when receiving data accumulated in HULFT-HUB Server)</td></tr>
<tr><td>Starting Identifier</td><td>34</td><td>Variable Length</td><td>Identifier that links the record output to the Operation Log file with the record in the Send/Receive/Request Acknowledge Log
(Identifier at the time of accumulation, when receiving data accumulated in HULFT-HUB Server)</td></tr>
<tr><td>Execution Host Name</td><td>68</td><td>Variable Length</td><td>Name of the host that executed the transaction</td></tr>
<tr><td>Operation Log
User-specified Character</td><td>HULFT: 8
HUB: 100</td><td>Variable Length</td><td>Characters to identify the environment that outputs the Operation Log</td></tr>
<tr><td>Command Execution Key</td><td>50</td><td>Variable Length</td><td>Name of issued request or executed command *2</td></tr>
<tr><td>Command Issued by</td><td>30</td><td>Variable Length</td><td>Location where the command was executed
<table>
<tr><th>Command Issued by</th><th>Description</th></tr>
<tr><td>HULFT_MANAGER</td><td>HULFT Manager Management Screen</td></tr>
<tr><td>HULFT_MANAGER_COMMAND</td><td>HULFT Manager Command</td></tr>
<tr><td>HULFT_MANAGEMENT_SCREEN</td><td>HULFT Management Screen</td></tr>
<tr><td>HULFT_COMMAND</td><td>HULFT Command</td></tr>
<tr><td>HULFT_CLS_COMMAND</td><td>HULFT Cluster Command</td></tr>
<tr><td>HULFT_HUB_MANAGER</td><td>HULFT-HUB Manager Management Screen</td></tr>
<tr><td>HULFT_HUB_SERVER_COMMAND</td><td>HULFT-HUB Server Command</td></tr>
<tr><td>HULFT_HUB_SERVER_CLS_COMMAND</td><td>HULFT-HUB Server Cluster Command</td></tr>
</table></td></tr>
<tr><td>Command Parameter</td><td>MF: 500
UX: Unlimited
NT: Unlimited
AS: 500
HUB: Unlimited</td><td>Variable Length</td><td>Parameters specified at the time of Command Execution</td></tr>
</table>

*1 Capable of multi byte code. Regarding i5OS, the ID is fixed length (105 bytes).

*2 Refer to "App.1.3 Command Execution Keys and Search Keys."

Each field is enclosed within enclosed character (") and set off with delimiter (,) to be output.

Example:

```
"2008/07/01","12:34:56.789","2008/07/01","12:34:56.789","user01","user01",
"HOST01","AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA","AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA",
"HOST01","","SEND_STATUS_DISPLAY","HULFT_COMMAND",""
```

App.1.2 System File List

Table App.1.3 System File List (HULFT)

File Name	System File Type	Key Information in File	Platform			
			MF	i5OS	UNIX	WIN
Management Information						
System Environment Settings File	HULENV	-		✓		✓
Account File	PSACCOUNT	Account Name				✓
Mail Environment Settings File	SENDMAILINI	Mail Account Name, Mail Server Host Name, Domain Name				✓
Send Management Information File	HULSND	File ID	✓	✓	✓	✓
Receive Management Information File	HULRCV	File ID	✓	✓	✓	✓
Host Information File	HULHST	Host Name	✓	✓	✓	✓
Transfer Group Information File	HULRHT	Transfer Group ID	✓	✓	✓	✓
Job Information File	HULEXE	Job ID	✓	✓	✓	✓
Format Information DB File	HULFMT	Format ID		✓	✓	✓
Format Information File			✓	✓	✓	✓
Multi Format Information File	HULMFMT	Multi Format ID	✓	✓	✓	✓
CSV Environment Settings File	HULCSV	File ID				✓
Password Management File	HULADMIN	User ID				✓
Mail Interface Information File	SENDMAILLST	Mail Interface ID				✓
Address Book File	SENDMAILADR	Address				✓
XML Environment Settings File	XML	File Name is set as Key → Host Name, Multi/Format Segment (f/m)+ Wild card (w/f), Format ID			✓	✓
External Character File						
SJIS → IBM	GAIJISI	-		✓	✓	✓
SJIS → KEIS	GAIJISK	-			✓	✓
SJIS → JEF	GAIJISJ	-			✓	✓
SJIS → EUC	GAIJISE	-			✓	✓
SJIS → NEC	GAIJISN	-			✓	✓
SJIS → UTF-8	GAIJIS8	-			✓	✓
IBM → SJIS	GAIJIIS	-		✓	✓	✓
IBM → KEIS	GAIJIIK	-		✓		
IBM → JEF	GAIJIIJ	-		✓		
IBM → EUC	GAIJIIE	-		✓	✓	
IBM → NEC	GAIJIIN	-		✓		
IBM → UTF-8	GAIJII8	-		✓	✓	
KEIS → SJIS	GAIJIKS	-			✓	✓
KEIS → IBM	GAIJIKI	-		✓		
KEIS → EUC	GAIJIKE	-			✓	
KEIS → UTF-8	GAIJIK8	-			✓	
JEF → SJIS	GAIJIJS	-			✓	✓
JEF → IBM	GAIJIJI	-		✓		
JEF → EUC	GAIJIJE	-			✓	
JEF → UTF-8	GAIJIJ8	-			✓	
NEC → SJIS	GAIJINS	-			✓	✓
NEC → IBM	GAIJINI	-		✓		
NEC → EUC	GAIJINE	-			✓	
NEC → UTF-8	GAIJIN8	-			✓	
EUC → SJIS	GAIJIES	-			✓	✓
EUC → IBM	GAIJIEI	-		✓	✓	
EUC → JEF	GAIJIEJ	-			✓	
EUC → KEIS	GAIJIEK	-			✓	
EUC → NEC	GAIJIEN	-			✓	
EUC → UTF-8	GAIJIE8	-			✓	
UTF-8 → SJIS	GAIJI8S	-			✓	✓
UTF-8 → IBM	GAIJI8I	-		✓	✓	
UTF-8 → EUC	GAIJI8E	-			✓	
UTF-8 → JEF	GAIJI8J	-			✓	
UTF-8 → KEIS	GAIJI8K	-			✓	
UTF-8 → NEC	GAIJI8K	-			✓	

File	System File Type	Key Information in File	Platform			
			MF	i5OS	UNIX	WIN
Others						
User-defined File	USRINFO	User ID	✓			
Send Control File (Resend Queue File)	RESENDREQ	File ID, Host Name, Date, Time, File Name	✓	✓	✓	✓
Log Files						
Send Log	HULSNDLOG	File ID, Host Name, Start Date, Start Time		✓	✓	✓
Receive Log	HULRCVLOG	File ID, Host Name, Start Date, Start Time		✓	✓	✓
Multiple Receive Information File	RCVSTORE	Host Name, Receive Date, Receive Time		✓	✓	✓
Request Acknowledge Log	HULOBSLOG	Service Name, Host Name, Acknowledge Date, Acknowledge Time		✓	✓	✓
Send and Receive Log File	HULSNDLOG HULRCVLOG	File ID, Host Name, Start Date, Start Time	✓			
	HULOBSLOG	Service Name, Host Name, Acknowledge Date, Acknowledge Time	✓			

Table App.1.4 System File List (HULFT-HUB Server)

File Name	System File Type	Access Type	Key Information in File
System Environment Settings File	hulhub.conf	Modify	-
Alternative Management Information	hub_Agent.db	New/Modify/Delete	Table Name, Current Host Name, Current Service Name, Current Port No., Alternative Host Name, Alternative Service Name, Alternative Port No.
		Commit/Rollback	-
User Management Information	hub_User.db	New/Modify/Delete	Table Name, User ID
		Commit/Rollback	-
Business Group Information	hub_BusinessGroup.db	New/Modify/Delete	Table Name, Business Group ID
		Commit/Rollback	-
Business Permission Information	hub_Authority.db	New/Modify/Delete	Table Name, Business Group ID, User ID
		Commit/Rollback	-
Client/Group Information	hub_ClientGroup.db	New/Modify/Delete	Table Name, Business Group ID, Host Name, Service Name
		Commit/Rollback	-
Transfer Information	hub_TransferData.db	New/Modify/Delete	[Master] Table Name, Send Host Name, Send Service Name, File ID [Slave] Table Name, Send Host Name, Send Service Name, File ID, Transfer Destination Information
		Commit/Rollback	-
Host Group Information	hub_HostGroupData.db	New/Modify/Delete	[Master] Table Name, Host Group Name [Slave] Table Name, Host Group Name, Host Name, Service Name
		Commit/Rollback	-
Transfer Destination Host Information	hub_TransferHost.db	New/Modify/Delete	Table Name, Host Name, Service Name
		Commit/Rollback	-
Transfer Conditions	hub_TransferCondition.db	New/Modify/Delete	Table Name, Send Host Name, Send Service Name, File ID Transfer Destination Host Name, Transfer Destination Service Name
		Commit/Rollback	-
HUB Mailbox	hub_MailBox.db	New/Modify/Delete	Table Name, HMB Classification, HMBID(Host Name), Service Name
		Commit/Rollback	-
Routing Information	hub_TransferRoute.db	New/Modify/Delete	Table Name, Network Address, Host Name, Service Name
		Commit/Rollback	-
Neighboring Server Information	hub_NeighborServer.db	New/Modify/Delete	Table Name, Host Name, Service Name
		Commit/Rollback	-
Controlled Client Information	hub_Client.db	New/Modify/Delete	Table Name, Host Name, Service Name
		Commit/Rollback	-
Transfer Log Table	hub_TransferHistory.db	NEW	Table Name, File ID, Send Host Name, Send Service Name, Receive Host Name, Receive Service Name, Transfer Start Date, Transfer Start Time
		Delete	[Screen] Table Name, Internal Key Information1, Internal Key Information2 [Command] Table Name, Parameter Information

App.1.3 Command Execution Keys and Search Keys

This section explains the host types that can use command execution keys and search keys.

z: available on zOS
i: available on i5OS
U: available on UNIX/Linux
W: available on Windows
H: available on HULFT-HUB Server*

*HULFT7 English edition does not support HULFT-HUB Server.

Table App.1.5 Command Execution Keys and Search Keys

Issued Requests and Commands	Command Execution Key	Operation Log Search Key	Command Issued by				
			HULFT			HUB Server	
			HULFT_MANAGEMENT_SCREEN Management Screen	HULFT_COMMAND Command	HULFT_CLS_COMMAND Cluster	HULFT_HUB_SERVER_COMMAND Command	HULFT_HUB_SERVER_CLS_COMMAND Cluster
Send File	SEND_FILE	SEND	z, i, U, W	z, i, U, W	-	-	-
Send Request	SEND_REQUEST	RECV	z, i, U, W	z, i, U, W	-	-	-
Resend File	RESEND_FILE	RESEND	z, i, U, W	z, i, U, W	-	-	-
Resend Request	RESEND_REQUEST	RERECV	z, i, W	z, i, U, W	-	-	-
Send Cancellation	SEND_CANCEL	SCAN	z, i, U, W	z, i, U, W	-	-	-
Receive Cancellation	RECEIVE_CANCEL	RCAN	z, i, U, W	z, i, U, W	-	-	-
Request Acknowledge Cancel	REQUEST_CANCEL	REQCAN	-	z	-	-	-
Send Status Display	SEND_STATUS_DISPLAY	SDISP	-	z, i, U, W	-	-	-
Send Task Display	SEND_TASK_STATUS_DISPLAY	SSTS	-	Z	-	-	-
Receive Status Display	RECEIVE_STATUS_DISPLAY	RDISP	-	z, i, U, W	-	-	-
Receive Task Display	RECEIVE_TASK_STATUS_DISPLAY	RSTS	-	z	-	-	-
Request Acknowledge Processing Status Display	REQUEST_ACKNOWLEDGE_PROCESSING_STATUS_DISPLAY	REQDISP	-	z	-	-	-
Request Acknowledge Task Status Display	REQUEST_ACKNOWLEDGE_TASK_STATUS_DISPLAY	REQSTS	-	z	-	-	-
Request Acknowledge Start Date	REQUEST_START_DATE	REQTIME	-	z	-	-	-
Unsent Status Queue Modification	UNSEND_STATUS_QUEUE_MODIFICATION	SCHANGE	-	z, i, U, W	-	-	-
Resend Queue list Display	RESEND_QUEUE_LIST_DISPLAY	RESNDLIST	-	z	-	-	-
Resend Queue Deletion	RESEND_QUEUE_DELETION	RESENDRM	z, i, U, W	z, i, U, W	-	-	-
Job Execution Result Notification	JOB_EXECUTION_RESULT_NOTIFICATION	SNDRC	-	z, i, U, W	-	-	-
Remote Job Execution	REMOTE_JOB_EXECUTION	RJOB	-	z, i, U, W	-	-	-
Management Information Duplication	MANAGEMENT_INFORMATION_DOUBLING	ADMFILEADJOIN	-	W	-	-	-
Management Information Registration	MANAGEMENT_INFORMATION_REGISTRATION	IUPDT	z, i, U, W	z, i, U, W	-	-	-
Management Information Deletion	MANAGEMENT_INFORMATION_DELETION	IRM	z, i, U, W	z, i, U, W	-	-	-

Issued Requests and Commands	Command Execution Key	Operation Log Search Key	Command Issued by				
			HULFT			HUB Server	
			HULFT_MANAGEMENT_SCREEN Management Screen	HULFT_COMMAND Command	HULFT_CLS_COMMAND Cluster	HULFT_HUB_SERVER_COMMAND Command	HULFT_HUB_SERVER_CLS_COMMAND Cluster
Management Information Record Deletion	MANAGEMENT_INFORMATION_RECORD_DELETION	COMP	-	i, U, W	-	-	-
Management Information File Generation	MANAGEMENT_INFORMATION_FILE_GENERATION	IGEN	-	z, i, U, W	-	-	-
Management Information Display	SEND_INFORMATION_DISPLAY	SLIST	-	z	-	-	-
	RECEIVE_INFORMATION_DISPLAY	RLIST					
	HOST_INFORMATION_DISPLAY	HLIST					
	MULTI_FORMAT_INFORMATION_DISPLAY	MLIST					
Format Information Display	FORMAT_INFORMATION_DISPLAY	ILIST	-	z, i, U, W	-	-	-
Management Information Related Display	MANAGEMENT_INFORMATION_RELATIONSHIP_DISPLAY	IDLIST	-	z, i, U, W	-	-	-
Send and Receive Log Deletion	SEND_AND_RECEIVE_LOG_DELETION	LOGRM	z, i, U, W	z, i, U, W	-	-	-
Request Acknowledge Log Deletion	REQUEST_LOG_DELETION	OBSRM	z, i, U, W	z, i, U, W	-	-	-
Send and Receive Log Display	SEND_AND_RECEIVE_LOG_DISPLAY	LOGLIST	-	i, U, W	-	-	-
Send Log Display	SEND_LOG_DISPLAY	SLOGLIST	-	z	-	-	-
Receive Log Display	RECEIVE_LOG_DISPLAY	RLOGLIST	-	z	-	-	-
Request Acknowledge Log Display	REQUEST_LOG_DISPLAY	OBSLIST	-	z, i, U, W	-	-	-
Transfer Summary	TRANSFER_SUMMARY	LOGSUM	-	W	-	-	-
Multiple File Join	FILE_JOINING	JOIN	-	z, i, U, W	-	-	-
Joined File Break	JOINED_FILE_BREAK	BREAK	-	z, i, U, W	-	-	-
Joined File Display	JOINED_FILE_CONTENT_DISPLAY	DSPFIL	-	z, i, U, W	-	-	-
Multiple Receive File Edit	MULTIPLE_RECEIVE_FILE_EDIT	STORE	-	z, U, W	-	-	-
Alert Notification Utility	ALERT_UTILITY	ALERT	-	z, i, U, W	-	-	-
File Record Edit	FILE_RECORD_EDIT	LF	-	U, W	-	-	-
External Character Table Expand	EXTERNAL_CHARACTER_TABLE_EXPAND	GTFEXTD	W	z, i, U, W	-	-	-
EBCDIC User Table Expand	EBCDIC_USER_TABLE_EXPAND	UCEXTD	W	i, U	-	-	-
Format Information Registration	FORMAT_INFORMATION_REGISTRATION	DTOF	-	i	-	-	-
Unicode Conversion	UNICODE_CONVERSION	CHGUNICODE	-	W	-	-	-
Product Confirmation	PRODUCT_CONFIRMATION	VERINFO	-	U		U	-
Send and Receive Log File Reorganization	REORGANIZATION_OF_SEND_AND_RECEIVE_LOG_FILE	REORG	-	z	-	-	-
Send and Receive Log File Open	LOG_FILE_OPEN	LOGOPEN	-	z	-	-	-
Management Screen Security	MANAGEMENT_SCREEN_SECURITY	USRADD	W	z, W	-	-	-
Scheduler Cancellation	SCHEDULER_CANCELLATION	SCHCAN	W	W	-	-	-
Schedule Log Deletion	SCHEDULE_LOG_DELETION	SCHRMLOG	W	W	-	-	-
Schedule Log List Display	SCHEDULE_LOG_LIST_DISPLAY	SCHLOGLIST	-	W	-	-	-
Debug Mode	DEBUG_MODE	DEBUG	-	i, U, W	-	-	-
Service Register/Delete	HULFT_SERVICE_SETUP	SVCSET	-	W	-	-	-
HULFT Send Startup	HULFT_SEND_STARTUP	SNDD	-	z, i, U, W	U	-	-

Issued Requests and Commands	Command Execution Key	Operation Log Search Key	Command Issued by				
			HULFT			HUB Server	
			HULFT_MANAGEMENT_SCREEN Management Screen	HULFT_COMMAND Command	HULFT_CLS_COMMAND Cluster	HULFT_HUB_SERVER_COMMAND Command	HULFT_HUB_SERVER_CLS_COMMAND Cluster
HULFT Send Termination	HULFT_SEND_END	KILLSND	-	z, i, U, W	-	-	-
HULFT Receive Startup	HULFT_RECEIVE_STARTUP	RCVD	-	z, i, U, W	U	-	-
HULFT Receive Termination	HULFT_RECEIVE_END	KILLRCV	-	z, i, U, W	-	-	-
HULFT Request Acknowledge Startup	HULFT_REQUEST_ACKNOWLEDGE_STARTUP	OBSD	-	z, i, U, W	U	-	-
HULFT Request Acknowledge Termination	HULFT_REQUEST_ACKNOWLEDGE_END	KILLOBS	-	z, i, U, W	-	-	-
HULFT Service Startup	HULFT_SVC_STARTUP	SVCD	-	W	-	-	-
HULFT Service Termination	HULFT_SVC_END	KILLSVCD	-	W	-	-	-
HULFT Scheduler Startup	HULFT_SCH_STARTUP	SCHD	-	W	-	-	-
HULFT Scheduler Termination	HULFT_SCH_END	KILLSCHD	-	W	-	-	-
HULFT Management Screen Startup	HULFT_MANAGEMENT_SCREEN_START	ADMSTR	z, i, U, W	-	-	-	-
HULFT Management Screen Log-in	HULFT_MANAGEMENT_SCREEN_LOGIN	ADMLOGIN	i, U, W	-	-	-	-
HULFT Management Screen Termination	HULFT_MANAGEMENT_SCREEN_FINISH	ADMSTP	z, i, U, W	-	-	-	-
Receive Job Execution Information	HULJOB_REQUEST	HULJOB	z, i, U, W	-	-	-	-
System Environment Settings File Update	HULFT_SYSTEM_ENVIRONMENT_SETTING_UPDATE	HULFTENVUPDT	i, U	-	-	-	-
Operation Log Manual Switch	OPERATION_LOG_SWITCH	OPLCHG	-	z, i, U, W	-	U	-
Operation Log List Display	OPERATION_LOG_LIST_DISPLAY	OPLLIST	-	z, i, U, W	-	U	-
Send Daemon Startup Synchronization Command	CLUSTER_SND_DAEMON_START	CLSNDSTR	-	-	U	-	-
Receive Daemon Startup Synchronization Command	CLUSTER_RECEIVE_DAEMON_START	CLRCVSTR	-	-	U	-	-
Request Acknowledge Daemon Startup Synchronization Command	CLUSTER_REQUEST_ACKNOWLEDGE_DAEMON_START	CLOBSSTR	-	-	U	-	-
Send Daemon Heartbeat Command	CLUSTER_SND_DAEMON_STATUS	CLSNDSTS	-	-	U	-	-
Receive Daemon Heartbeat Command	CLUSTER_RECEIVE_DAEMON_STATUS	CLRCVSTS	-	-	U	-	-
Request Acknowledge Heartbeat Command	CLUSTER_REQUEST_ACKNOWLEDGE_DAEMON_STATUS	CLOBSSTS	-	-	U	-	-
Send Daemon Termination Synchronization Command	CLUSTER_SND_DAEMON_STOP	CLSNDSTP	-	-	U	-	-
Receive Daemon Termination Synchronization Command	CLUSTER_RECEIVE_DAEMON_STOP	CLRCVSTP	-	-	U	-	-
Request Acknowledge Daemon Termination Synchronization Command	CLUSTER_REQUEST_ACKNOWLEDGE_DAEMON_STOP	CLOBSSTP	-	-	U	-	-
Send Daemon System Environment Settings Query Command	CLUSTER_SND_DAEMON_QUERY	CLSNDQRY	-	-	U	-	-
Receive Daemon System Environment Settings Query Command	CLUSTER_RECEIVE_DAEMON_QUERY	CLRCVQRY	-	-	U	-	-
Request Acknowledge Daemon System Environment Settings Query Command	CLUSTER_REQUEST_ACKNOWLEDGE_DAEMON_QUERY	CLOBSQRY	-	-	U	-	-
Send Daemon System Environment Settings Reset Command	CLUSTER_SND_DAEMON_SET	CLSNDSET	-	-	U	-	-

Issued Requests and Commands	Command Execution Key	Operation Log Search Key	Command Issued by				
			HULFT			HUB Server	
			HULFT_MANAGEMENT_SCREEN Management Screen	HULFT_COMMAND Command	HULFT_CLS_COMMAND Cluster	HULFT_HUB_SERVER_COMMAND Command	HULFT_HUB_SERVER_CLS_COMMAND Cluster
Receive Daemon System Environment Settings Reset Command	CLUSTER_RECEIVE_DAEMON_SET	CLRCVSET	-	-	U	-	-
Request Acknowledge Daemon System Environment Settings Reset Command	CLUSTER_REQUEST_ACKNOWLEDGE_DAEMON_SET	CLOBSSET	-	-	U	-	-
Send File API Send File	SEND_FILE_FUNCTION	SENDFUNC	-	z, W	-	-	-
Send File Extension API Send File	SEND_FILE_EXTENSION_FUNCTION	SENDEXFUNC	-	W	-	-	-
Send Request API Send Request	SEND_REQUEST_FUNCTION	RECVFUNC	-	z, W	-	-	-
Send Request Extension API Send Request	SEND_REQUEST_EXTENSION_FUNCTION	RECVEXFUNC	-	W	-	-	-
Send File API Resend File	RESEND_FILE_FUNCTION	RESENDFUNC	-	W	-	-	-
Send File Extension API Resend File	RESEND_FILE_EXTENSION_FUNCTION	RESENDEXFUNC	-	W	-	-	-
Send Request API Resend Request	RESEND_REQUEST_FUNCTION	RERECVFUNC	-	W	-	-	-
Send Request Extension API Resend Request	RESEND_REQUEST_EXTENSION_FUNCTION	RERECVEXFUNC	-	W	-	-	-
Job Execution Result Notification API	JOB_EXECUTION_RESULT_NOTIFICATION_FUNCTION	SNDRCFUNC	-	W	-	-	-
Log Search API	LOG_SEARCH_FUNCTION	LOGLISTFUNC	-	W	-	-	-
Remote Job Execution API	REMOTE_JOB_EXECUTION_FUNCTION	RJOBFUNC	-	z	-	-	-
Cluster Startup Synchronization Command	CLUSTER_HUB_SERVER_START	CLHUBSTR	-	-	-	U	-
Cluster Heartbeat Command	CLUSTER_HUB_SERVER_STATUS	CLHUBSTS	-	-	-	U	-
Cluster Termination Synchronization Command	CLUSTER_HUB_SERVER_STOP	CLHUBSTP	-	-	-	U	-
Delivery Request	HUB_SEND_REQUEST	HUBSEND	-	-	-	U	-
Totaled Result CSV Output	TOTAL_RESULT_CSV_OUTPUT	HUBSUMCSV	-	-	-	U	-
Client Suspend/Release	CLIENT_RESERVATION_AND_RELEASE	HUBCHGSTS	-	-	-	U	-
Transfer Log Delete	TRANSFER_LOG_DELETE	HUBTRNSLOGRM	-	-	-	U	-
HUB-Server Startup	HUB_SERVER_STARTUP	HUBSTR	-	-	-	U	U
HUB-Server Termination	HUB_SERVER_END	HUBSTP	-	-	-	U	-
Address Book File Addition	MAIL_ADDRESS_ADDITION	MAILADADD	W	-	-	-	-
Address Book File Update	MAIL_ADDRESS_UPDATE	MAILADUPDT	W	-	-	-	-
Address Book File Deletion	MAIL_ADDRESS_DELETION	MAILADRM	W	-	-	-	-
CSV Conversion Information Register/Update	CSV_EXCHANGE_SETTING_UPDATE	CSVUPDT	W	-	-	-	-
CSV Conversion Information Register/Delete	CSV_EXCHANGE_SETTING_DELETION	CSVRM	W	-	-	-	-
Menu Option Setting/Update	OPTION_SETTING_UPDATE	SETUP	W	-	-	-	-

~ Memo ~

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

HULFT7e Windows

Administration Manual

First Edition: September 1, 2009

SAISON INFORMATION SYSTEMS CO., LTD.

SAISON
INFORMATION
SYSTEMS
CO., LTD.