IT-2000

UPLOAD/DOWNLOAD UTILITY
(LMWIN RS-232C/SCSI)

MANUAL

(Version 2.01 for Windows 95)

April 1998

Casio Computer Co., Ltd.

Table of Contents

< API >

< File Check Utility >

1. Introduction

This document is the User Manual for the Windows 95 Upload/Download utility software, version 2.01, for the IT-2000 system. It describes the operation, screen installation and configuration of the system. It also describes the application programming interface (API).
The object of the Upload/Download utility software is to perform the following functions:

1. Send and Receive files between the Host PC and the IT-2000.
2. Remote two-way and one-way command execution (buzzer activate, format)
3. Script file execution on Host PC

2. Terminology/Nomenclature

- **Download**
  Transfer of files/data from the Host PC to the IT-2000

- **Upload**
  Transfer of files/data from the IT-2000 to the Host PC

- **I/O Box**
  A device which acts as the interface between the Host PC and the IT-2000.
  This device can be a Master I/O Box or a Satellite I/O Box.

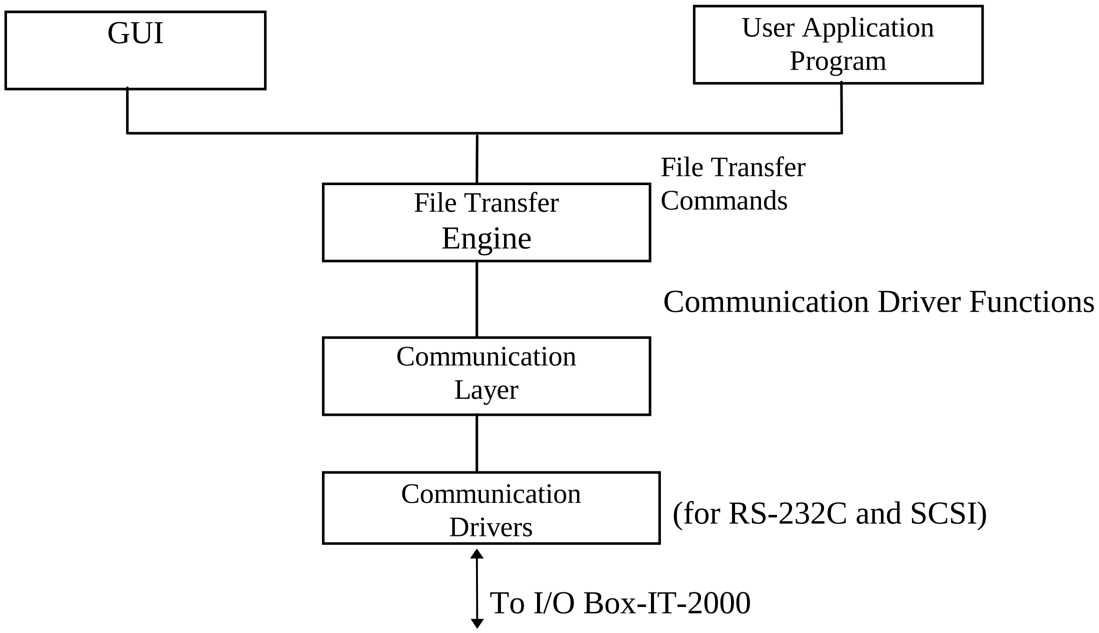
- **Master I/O Box**
  An I/O Box which connects to the Host PC through a SCSI link. A Master I/O Box has protocol translation capability.

- **Satellite I/O Box**
  An I/O Box which is connected to a Master I/O Box through RS-422 link or an I/O Box which can be connected to the Host PC through an RS-232C link. It basically acts as a protocol converter from RS-232C to IrDA.

- **Script File**
  An application specific batch file which is prepared on the Host PC, consisting of a number of commands to be executed in sequence. This file can be prepared on the Host PC using the Environment Setup Tool (part of the Upload/Download utility) or a Text Editor. This file can be designed to execute on the Host PC or the IT-2000. The script files shall have an extension of .**SCR**. The utility will not recognize any other extensions as script file.

- **Session**
  The activity between the Host PC and the IT-2000 from the start of a request by the IT-2000 to the termination of the activity is called a session.
  In the Windows 95 environment, multiple sessions can be active (one for each Master I/O Box) on the Host PC.

3. System Architecture



The Upload/Download System architecture is as shown above. It consists of the following components:

GUI

This is a Graphical User Interface of the Upload/Download utility. It consists of the following sub-systems:

- A script file editor which enables the user to create and edit script files in wizard form .
- A configuration file editor which enables the user to select the communication interface between the I/O Box and the Host PC and also to configure the settings (COM port, baud rate, etc.).

  These settings and selections are stored in the Configuration file **DEVICE.INI.**
- A Command Execution Screen which enables the user to start and stop the utility and to issue commands (send, receive, format, etc.).

User Application Program (optional)

This is an optional facility by which the user can make use of the File Transfer utility APIs to develop their own Applications or User Interfaces. This product provides a standard User Interface (GUI), but the user has the option to develop other interfaces or enhance the existing interface.

File Transfer Engine

This sub-system implements the File Transfer Protocol, and may be considered as the engine which carries out all file transfer and other commands.

Communication Layer

This layer provides a common interface between the various communication drivers (RS-232C, SCSI) and the File Transfer Utility layer. The objective of this is to provide a single application programming interface between the utility and the drivers.

Communication Drivers

This layer implements the communication drivers for RS-232C and SCSI. This is the lowest layer which interfaces directly between the hardware on the Host PC and the I/O Box.

4. Environment and Interfaces

The Upload/Download utility host program is available under the following operating environment.

IBM compatible PC AT

CPU	: Pentium
Clock	: 100 MHz (minimum)
Memory Capacity	: 16 Mbytes RAM or more

5. Functionality

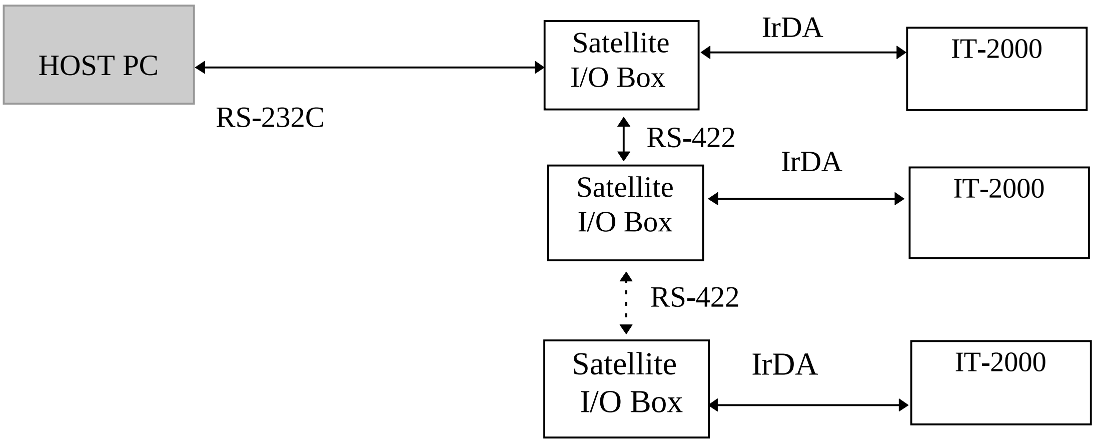
The Upload/Download utility provides the following functions between the Host PC and the IT-2000.

1. File Send
2. File Receive
3. File Send and Append
4. File Receive and Append
5. Buzzer activate
6. Format Disk
7. Set Time and Date
8. Delete
9. Move/Rename
10. Fetch Disk Information
11. Fetch File Information
12. Set File Information
13. End Session
14. Execute Child Process
15. Display String

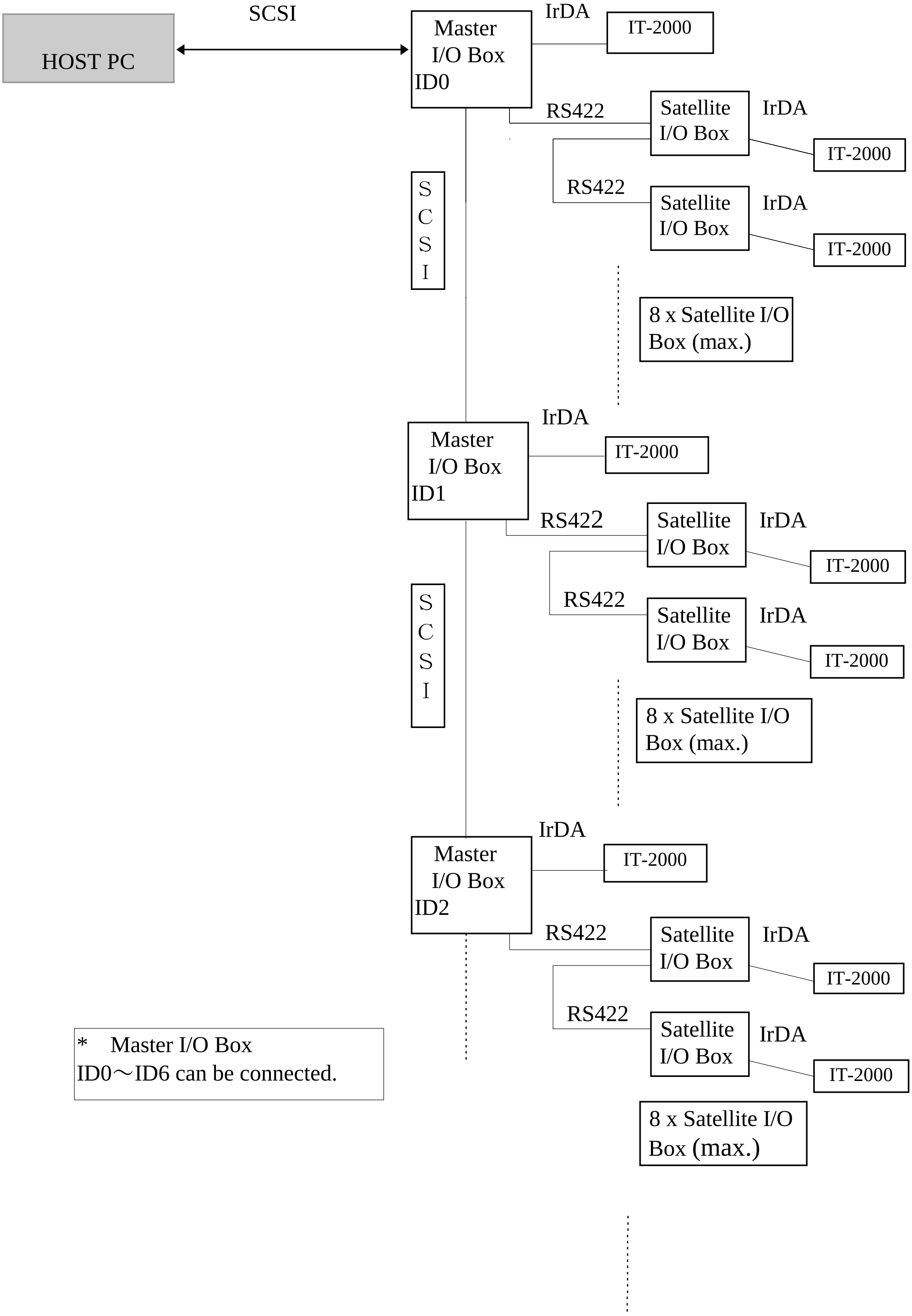
6. System Configuration

6.1 Connection of Satellite I/O Box through RS-232C

Satellite I/O Box can be connected to the Host PC through an RS-232C serial link. The satellite I/O Box may be daisy-chained to other satellite I/O Boxes (maximum of 8 units) through RS-422 serial links, as shown below, or may be stand alone.



6.2 Connection of Master I/O Box through SCSI



* Master I/O Box
ID0～ID6 can be connected.

7. Error Information

1. The Upload/Download System identifies, logs and displays the following types of errors:
   1) Communication errors
   2) File Transfer Utility or Protocol Errors
   3) User Command Syntax Errors

Error Messages are displayed in Error Message Dialog boxes. A maximum of 8 dialog boxes can be displayed on the screen simultaneously. If there are more error messages, the last Error Message box will display a message " More Errors Exist -- See Error Log for details".

All the errors are logged in the Error Log file (ERROR.LOG) in the directory specified in the **DEVICE.INI** file (configuration file).The Error Log file has the following information about the errors:

< Content in Error Log file >

Session No.	Date	Time	Error code	Description
0	6-9-1997	17-44-46	0 1	Undefined function code
0	6-9-1997	18- 1-24	0 5	Communication error
0	8-9-1997	11-53-29	4 5	Command timeout error

7.1 Error Codes

The following is a list of error codes and descriptions for errors returned by the File Transfer Protocol layer. These errors are returned to the application by the FTP API.

Error Code		**Error Message**	**Description**
00x00	0x00	NormalEnding	Normal end
0xDC	0x00	ADriverFormatNotice	Notification of format on Drive A
|	|	|	|
0xF5	0x00	ZDriveFormatNotice	Notification of format on Drive Z
0xF6	0x00	PowerOffEndingNotice	Notification of ending power shut down
0xF7	0x00	ResetEndingNotice	Notification of ending reset
0xF8	0x00	BreakKeyInterruptEndingNotice	Notification of ending interruption by user
0x01	0x00	UndefineFunctionCode	Protocol error (function code)
0x01	0x01	UndefineSubFunctioncode	Protocol error (sub-function code)
0x01	0x02	NotExecuteCommand	Command not executed
0x01	0x03	CheckSumError	Checksum error
0x01	0x04	CommandSequenceError	Command sequence error
0x01	0x05	SequenceNumberError	Sequence number error
0x01	0x06	OtherProtocolError	Illegal protocol
0x01	0x07	ParameterError	Parameter error
0x01	0x08	TimeoutError	Timeout error
0x01	0x10	DataLengthError	Protocol error (DATA LEN)
0x01	0x12	ProtocolVersionDifferenceError	Protocol version not mis-matched
0x01	0x13	MemoryAllocationError	Memory allocation not possible
0x01	0x15	FileSizeError	Protocol error (file size)
0x01	0x15	DateError	Protocol error (date)
0x01	0x16	TimeError	Protocol error (time)
0x01	0x17	FileAttributeError	Protocol error (attribute)

0x01	0x18	OverwriteOptionError	Protocol error (forced overwrite)
0x01	0x19	EOFFlagError	Protocol error (EOF)
0x02	0x02	FileNotFound	File not found
0x02	0x03	PathNotFound	Path not found
0x02	0x0B	InvalidFormat	Not formatted
0x02	0x0F	InvalidDiskDrive	Not right disk
0x02	0x10	DeleteRequestisCurrentDirectory	Delete request to current directory
0x02	0x11	NotSameDisk	Not the same disk
0x02	0x12	FileNothing	File not existed
0x03	0x13	WriteProtectError	Write protect error
0x03	0x14	UnknownUnit	Undefined unit
0x03	0x15	DriveNotReady	Drive not ready
0x03	0x17	DataError	Data error
0x03	0x19	SeekError	Seek error
0x03	0x1A	UnknownDiskFormat	Disk not formatted
0x03	0x1B	SectorNotFound	Sector not found
0x03	0x1D	WriteError	Write error
0x03	0x1E	ReadError	Read error
0x03	0x1F	UnknownError	Specified model of IT-2000 not correct
0x03	0x20	FileShareError	File share error
0x03	0x21	FileLockError	File lock error
0x03	0x22	InvalidDiskChanged	Illegal disk change
0x03	0x23	FCBFull	FCB full
0x03	0x53	FatalError	Fatal error
0x04	0x00	ReadOnlyFileAccessError	Access error to read-in only file
0x05	0x00	CommunicationError	Communication error
0x05	0x01	EndIndicationfromHT	End request from IT-2000
0x05	0x02	PCSendsEndIndication	End request from PC
0x05	0x03	SpawnError	Spawn error
0x05	0x04	CommandTimoutError	Command timeout error
0x05	0x05	ErrorOpeningErroLogFile	Fail to open log file
0x05	0x06	OptionError	Option error
0x05	0x07	StartupError	Start up error
0x05	0x08	OpenError	Open error
0x05	0x09	ListenError	Listening error
0x05	0x0A	AcceptError	Accept error
0x05	0x0B	NotEnoughMemoryToExecuteProcess	Not enough memory to execute process
0x05	0x0C	ArgumentListTooLargeForTheProcess	Too large process
0x05	0x0D	InvalidModeForTheProcess	Illegal end process
0x05	0x0E	ProcessTerminatedNormally	Normal end process
0x05	0x0F	NoiseError	Communication noise error

8. Installation

The following files are supplied as part of the LMWIN and are required to install the Upload/Download utility under the Windows 95:

File Name	Remarks
LMWIN32.EXE	Upload/download utility execute type (GUI mode)
driver32.dll	RS-232C driver
hfc32.dll	Command analysis driver
lman32.dll	Connection observation driver
scsidrv.dll	SCSI connection driver
tcpip.dll	Reserved file for TCP/IP connection
LMWIN.INI	Execution default setting file
DEVICE.INI	Configuration file
Other	File for Upload/Download utility/File for help

All the above files are provided in one package. Files and libraries necessary for all the installations are generated by the **Setup.exe** program.

8.1 Syntax of Installer Command

The Upload/Download utility for Windows 95 is installed by running the **Setup.exe** being installed in the setup directory in FD.

```
>setup
```

9. Command Interface

The following commands can be issued from the User Interface of the Upload/Download utility software. All the commands are script-executable.

1. send
2. recv
3. append send
4. append recv
5. delete
6. move
7. buzzer
8. set time date
9. display string
10. end session
11. exec child process
12. getdisk info
13. getfile info
14. set file info
15. format

9.1 Single Shot Mode/Batch Mode

Every Command and Script File can be executed in two modes : Single Shot Mode and Batch Mode. The Command Execution Wizard Screen gives the user the options to execute each command once (Single Shot Mode) or as a batch (many times - Batch Mode) by proper selection of the Mode option.
If the Batch Mode is selected, the command is executed repeatedly until the user selects the ABORT TRANSFER sub-menu on the EXECUTE menu.
The default mode is the Batch Mode.

Multiple Sessions

The Windows 95 version of the Upload/Download utility software supports multiple sessions. This means that the utility can communicate simultaneously with a number of I/O Boxes (if the SCSI interface is selected). The progress bars of all the active sessions can be displayed on the screen, and any or all the sessions can be aborted.

9.2 Command Detail

The detail of the utility commands are described as follow.

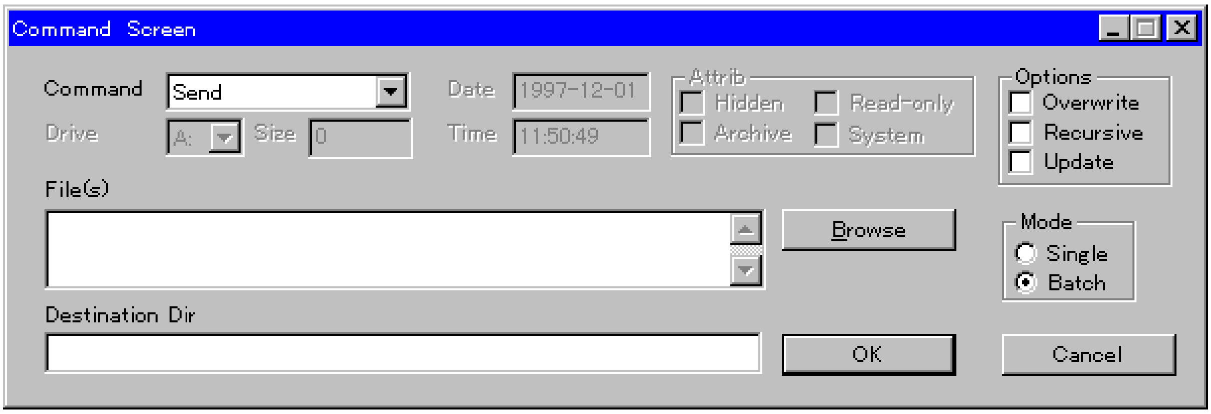
9.2.1 Send

The files reside on the Host PC will be transferred to the IT-2000.
Here, destination directory means the directory into which the send files shall be copied on the side of the station which receives the files.

Way to describe in the script file :

```
/S[O|U|R]  <file name> <destination directory>
```

<Communication Command Screen>



Command: Send
Drive: A: Size: 0
Date: 1997-12-01
Time: 11:50:49
Attrib: □ Hidden □ Read-only □ Archive □ System
Options: □ Overwrite □ Recursive □ Update
Mode: ○ Single ● Batch
File(s):
Destination Dir:

File
Specifies name of file in full path name that is to be sent.

Destination directory
Specifies directory of IT-2000 that is to be sent.

Option

O (Forced overwrite) :
Specifies the forced overwrite in read-only file.
When this option is specified, the process of write is executed in a read-only file.

U (Date verification) :
When this option is specified, later files in the receive directory will not be over written..

R (Recursive) :

- When this option is specified, all files under the directory which is specified by send path file name will be target for transmission file.
  If sub-directory is existed under the specified directory, its sub-directory name is also appended, and the send process is done.

- Even if this option is specified, send path file name must be specified by the full path name.

- When this option is not specified, only files specified by send path file name will be target for transmission file.

U (Date verification) :

- When this option is specified, a latest file in the storage directory will not be over written.

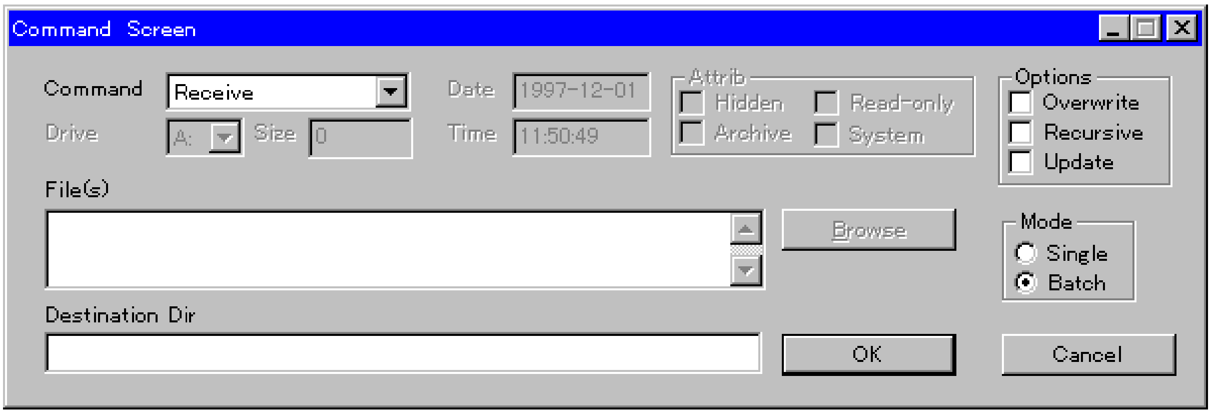
9.2.2 Receive

This function is to receive a file, by specifying request path name, which is resided in the IT-2000. If the specified directory by the receive directory is not resided in the IT-2000, the directory is automatically generated in the PC.

Way to describe in the script file :

```
/R[O|U|R]  <file name> <destination directory>
```

<Communication Command Screen>



Command: Receive
Drive: A:
Size: 0
Date: 1997-12-01
Time: 11:50:49

Attrib
□ Hidden □ Read-only
□ Archive □ System

Options
□ Overwrite
□ Recursive
□ Update

Mode
○ Single
● Batch

File(s)

Destination Dir

File

Specifies the name of file on the IT-2000 in full path name that is to be received by the Host PC.

Destination directory

Specifies the directory in the Host PC that is to recceive files from the IT-2000.

Option

O (Forced Overwrite) :

- Specify forced over-write even to read-only files.
  When this option is specified, a write is executed even into read-only file.

R (With Directory) :

- All files under the directory specified by requested path file name will be target for transmission file.

- If sub-directory exists under the specified directory, its sub-directory name and files under the sub-directory are transmitted.

- Even if this option is specified, requested path file shall be specified by the full path name.

U (Date Change) :

- When this option is specified, later files in the receive directory will not be over written.

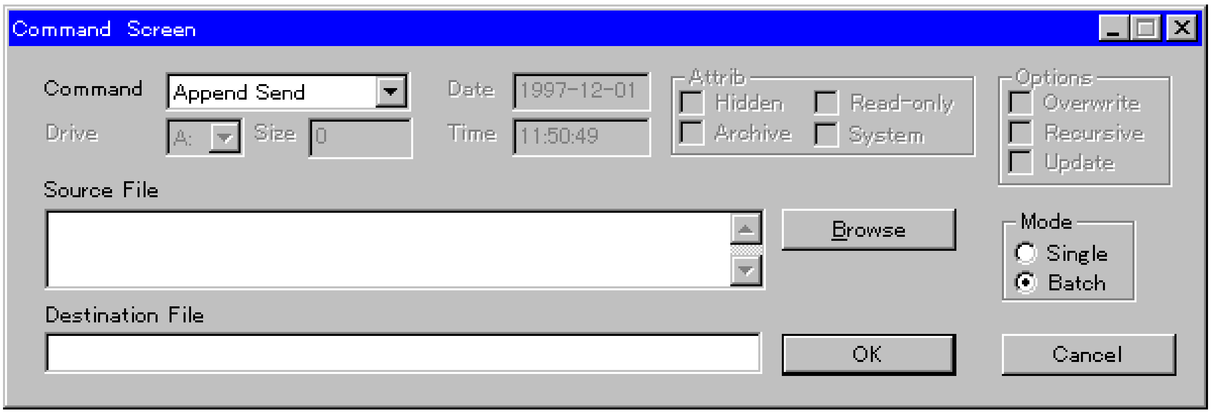
9.2.3 Append File Send

This function is to append the content of a file in the Host PC which is specified by the append path file name to a file on the IT-2000 which is specified by the target path file name.
A wild card cannot be specified.
File is appended under the binary system (following the EOF code, it is appended.).

Way to describe in the script file :

```
/AR   <source file name> <destination file name>
```

<Communication Command Screen>



Command: Append Send
Drive: A: Size: 0
Date: 1997-12-01
Time: 11:50:49
Attrib: ☐ Hidden ☐ Read-only ☐ Archive ☐ System
Options: ☐ Overwrite ☐ Recursive ☐ Update
Mode: ○ Single ● Batch
Source File:
Destination File:

Source File

Specifies a file in PC by full path name.

Destination File

Specifies a file of IT-2000 which connects with a file in PC.

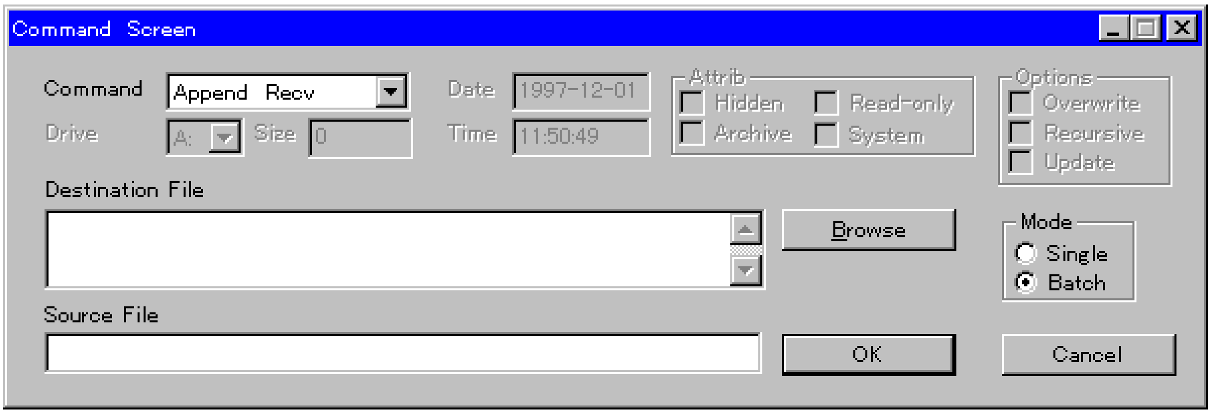
9.2.4 Append File Receive

This function is to append the content of a file in the IT-2000 which is specified by the append path file name to a file in the Host PC which is specified by the target path file name.
A wild card cannot be specified. The source file is appended as binary. (Even if there is an EOF code at the end of destination file, source file is appended.).

Way to describe in the script file :

```
/AR <source file name> <destination file name>
```

<Communication Command Screen>



Command Screen

Command: Append Recv

Drive: A: Size: 0

Date: 1997-12-01 Time: 11:50:49

Attrib: □ Hidden □ Read-only □ Archive □ System

Options: □ Overwrite □ Recursive □ Update

Mode: ○ Single ● Batch

Destination File:

Source File:

Source File

Specifies a file in the IT-2000 by full path name.

Destination File

Specifies a file of the Host PC which connects with a file in the IT-2000.

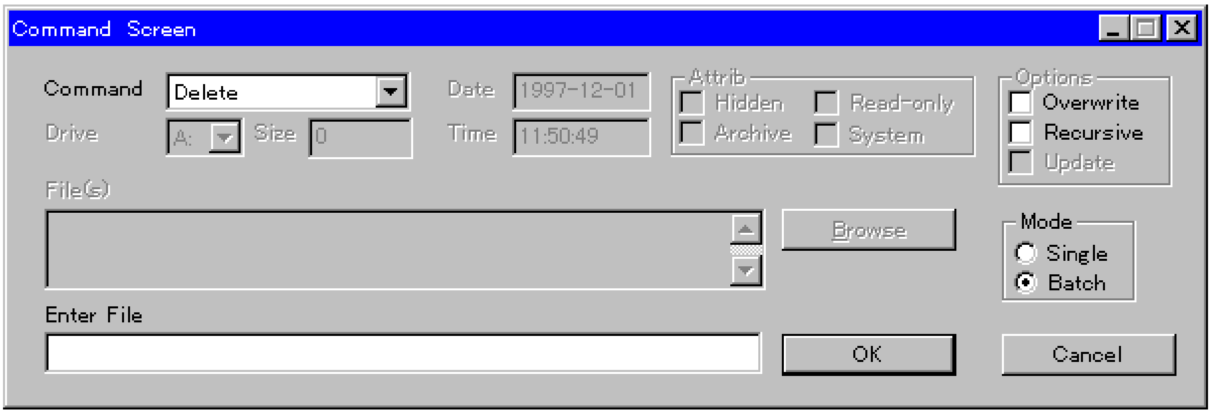
9.2.5 File / Directory Delete

A file being resided in the IT-2000 can be deleted by specifying it to
< delete file name >.

Way to describe in the script file :

```
/D  <delete file name>
```

<Communication Command Screen>



File
Specifies a file to be deleted in the IT-2000.

Option
Not available.

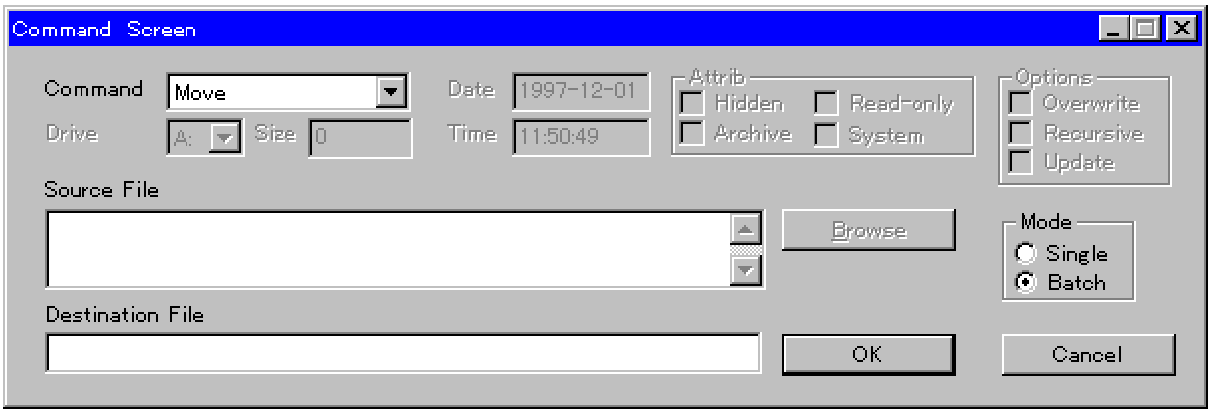
9.2.6 Move / Rename

This command is to move the specified file (with the full path name) to another directory in the same drive of the same IT-2000 with the same file name or different file name (specify in full path too). The function can be used to move or to modify file on the same drive.

Way to describe in the script file :

```
/N  <source file name> <destination file name>
```

<Communication Command Screen>



Command Screen

Command: Move
Drive: A: Size: 0
Date: 1997-12-01
Time: 11:50:49

Attrib
□ Hidden □ Read-only
□ Archive □ System

Options
□ Overwrite
□ Recursive
□ Update

Mode
○ Single
● Batch

Source File

Destination File

Source File

Specifies a file in the IT-2000 by full path name.

Destination File

Specifies a file in the IT-2000 by full path name which the file in the IT-2000 is moved to.

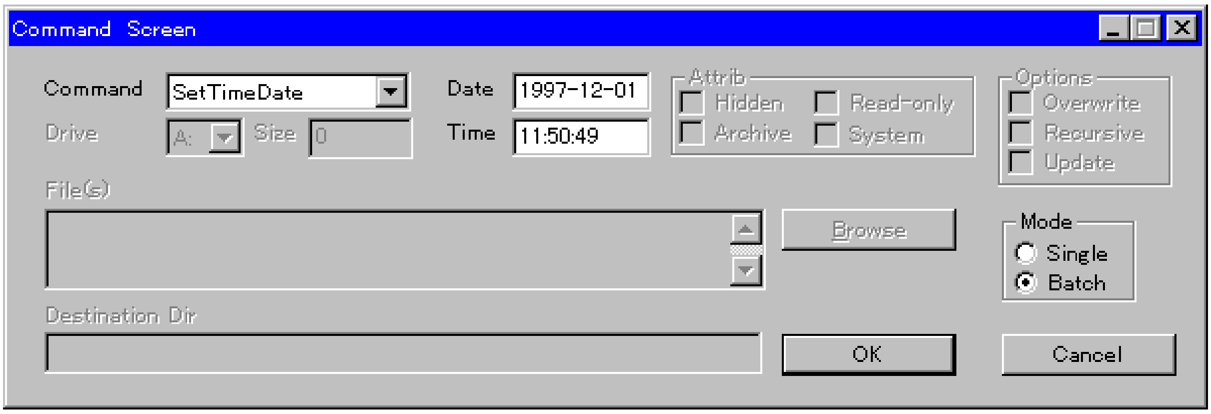
9.2.7 Time Date Set

Sets the current time and date on the IT-2000. If time and date are not specified by the script file, the time and date in the Host PC are set on the IT-2000.

Way to describe in the script file :

```
/T  <YYYY-MM-DD HH:MM:SS>
```

<Communication Command Screen>



Command SetTimeDate

Date 1997-12-01

Time 11:50:49

Mode ○ Single ● Batch

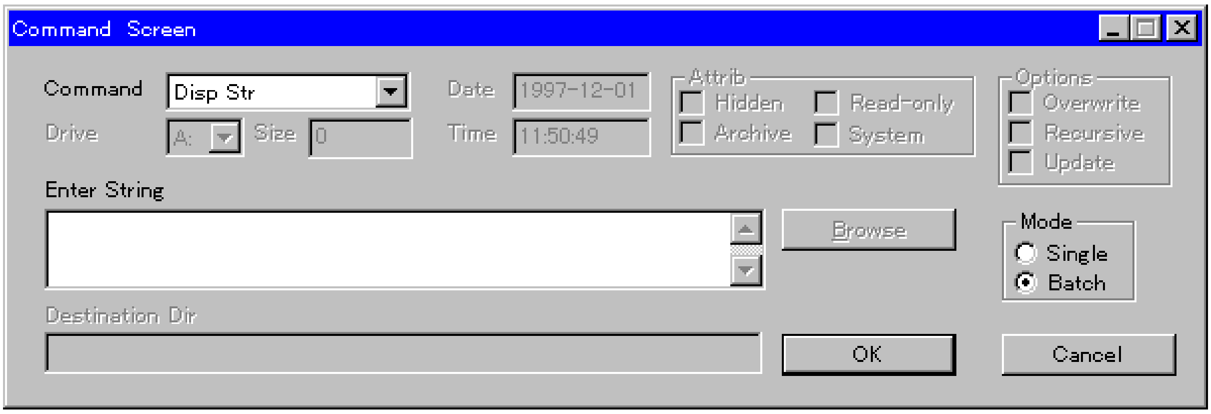
9.2.8 Character String Display

Displays character strings on the screen display of the IT-2000. Character string(s) is enclosed by (“).

Way to describe in the script file :

```
/P  <character string to be displayed>
```

<Communication Command Screen>



Command Screen

Command: Disp Str

Drive: A:

Size: 0

Date: 1997-12-01

Time: 11:50:49

Attrib: □ Hidden □ Read-only □ Archive □ System

Options: □ Overwrite □ Recursive □ Update

Enter String

Mode: ○ Single ● Batch

Destination Dir

Character String :
A character string to be displayed on the IT-2000 is written.

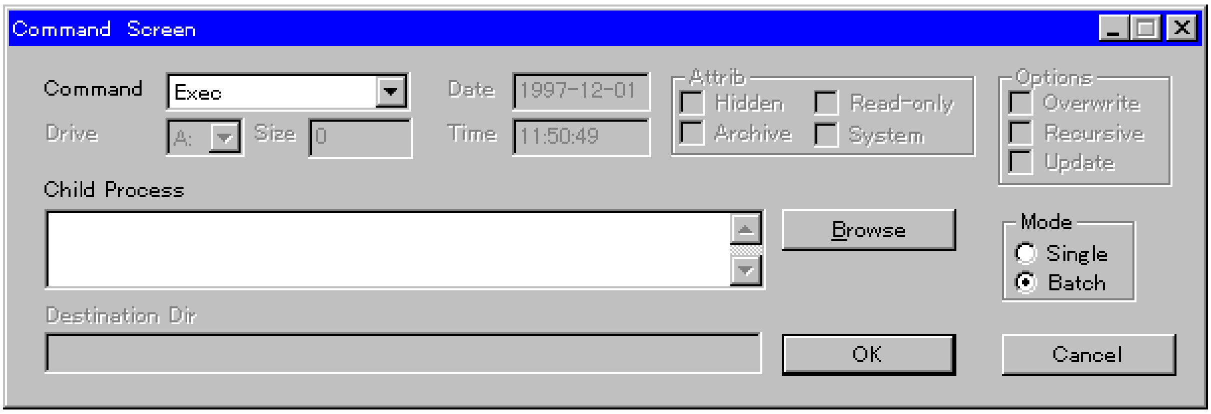
9.2.9 Child Process Execution

Executes the execute file on the Host PC specified by <execution file name>.

Way to describe in the script file :

```
/C  <child process>
```

<Communication Command Screen>



Command Screen

Command: Exec

Drive: A: Size: 0

Date: 1997-12-01

Time: 11:50:49

Attrib: □ Hidden □ Read-only □ Archive □ System

Options: □ Overwrite □ Recursive □ Update

Child Process:

Destination Dir:

Mode: ○ Single ● Batch

Child-process :

Write an executable file on the Host PC to be executed.

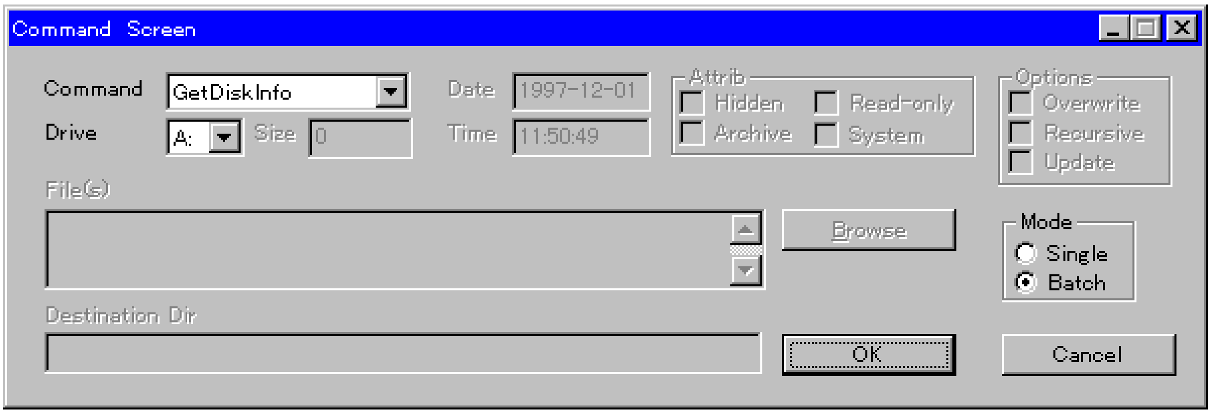
9.2.10 Drive Information Fetching

The drive information on the HT will be appended to “ diskinfo.clp “.

Way to describe in the script file :

```
/I  <drive name>
```

<Communication Command Screen>



Command Screen

Command: GetDiskInfo

Drive: A:

Size: 0

Date: 1997-12-01

Time: 11:50:49

Attrib: □ Hidden □ Read-only □ Archive □ System

Options: □ Overwrite □ Recursive □ Update

Mode: ○ Single ● Batch

File(s)

Destination Dir

Drive :

Specify the drive name on the IT-2000 you wish to fetch the information from.

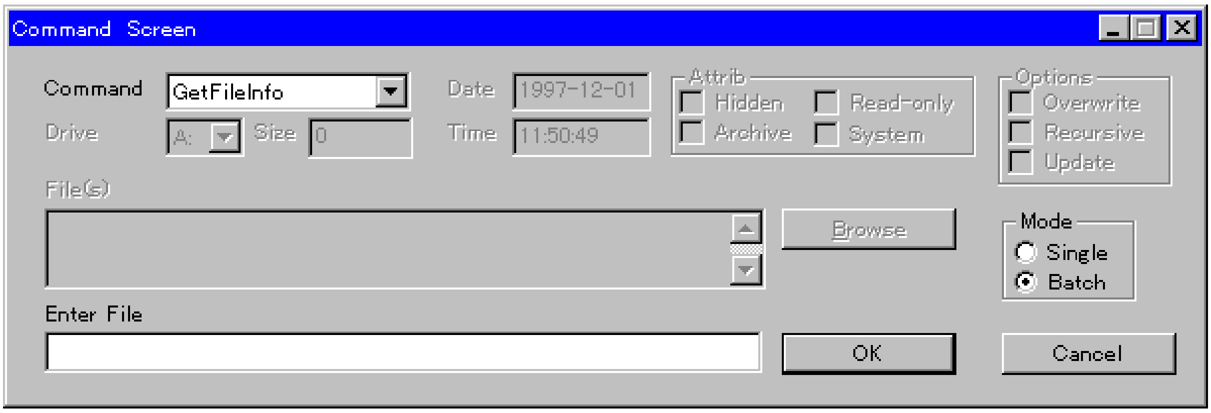
9.2.11 File Information Fetching

The file information on the IT-2000 will be appended to “ fileinfo.clp “.

Way to describe in the script file :

```
/I  <file name>
```

<Communication Command Screen>



Command Screen

Command: GetFileInfo

Drive: A:

Size: 0

Date: 1997-12-01

Time: 11:50:49

Attrib: □ Hidden □ Read-only □ Archive □ System

Options: □ Overwrite □ Recursive □ Update

Mode: ○ Single ● Batch

File :

Specify the file on the IT-2000 by full path name from which you wish to fetch the information.

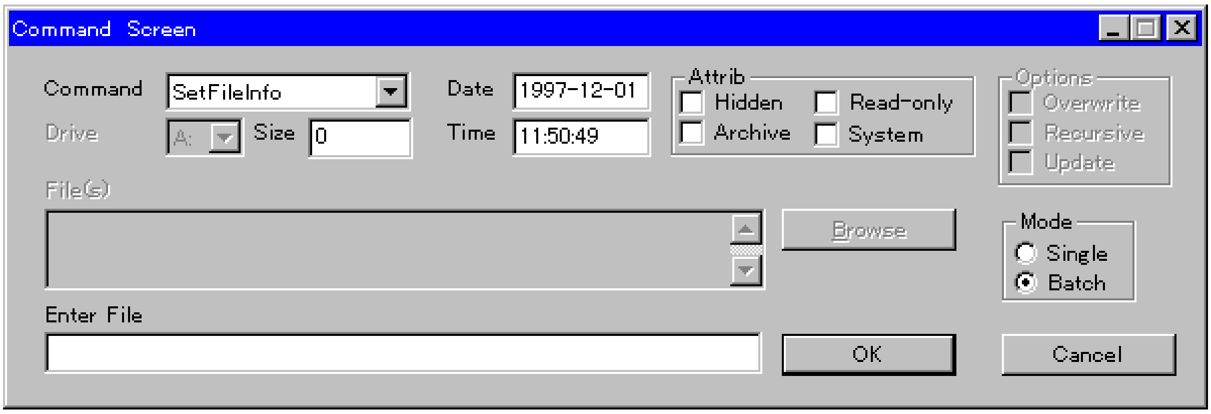
9.2.12 File Information Setting

Sets for the information of file <updated file name> on the IT-2000.
<size> means the size of file (0 specifies no update of file.).

Way to describe in the script file :

```
/X[R|H|S|A]  <updated file name> <size> <YY-MM-DD> <hh:mm:ss>
```

<Communication Command Screen>



Command Screen

Command: SetFileInfo
Drive: A:
Size: 0
Date: 1997-12-01
Time: 11:50:49
Attrib: □ Hidden □ Read-only □ Archive □ System
Options: □ Overwrite □ Recursive □ Update
Mode: ○ Single ● Batch

Size :

Specify the size of a file (0 specifies no update).

Attibute :

Change the attribute to one of the followings.

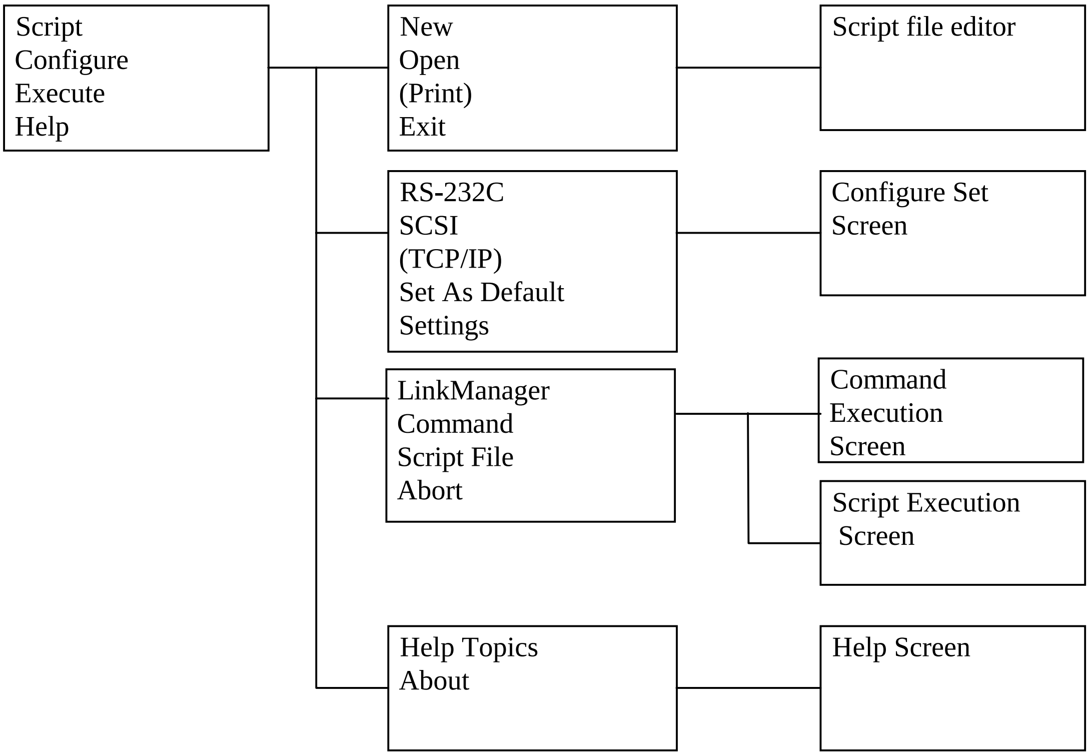
Write prohibited (R)	: Read only file
Hidden (H)	: Hidden file
System (S)	: System file
Archive (A)	: Archive file

Date / Time :

Specify the date and time of a file.

10. Screen Transition (GUI)

The following describes the screen transition for the Upload/Download utility (GUI).

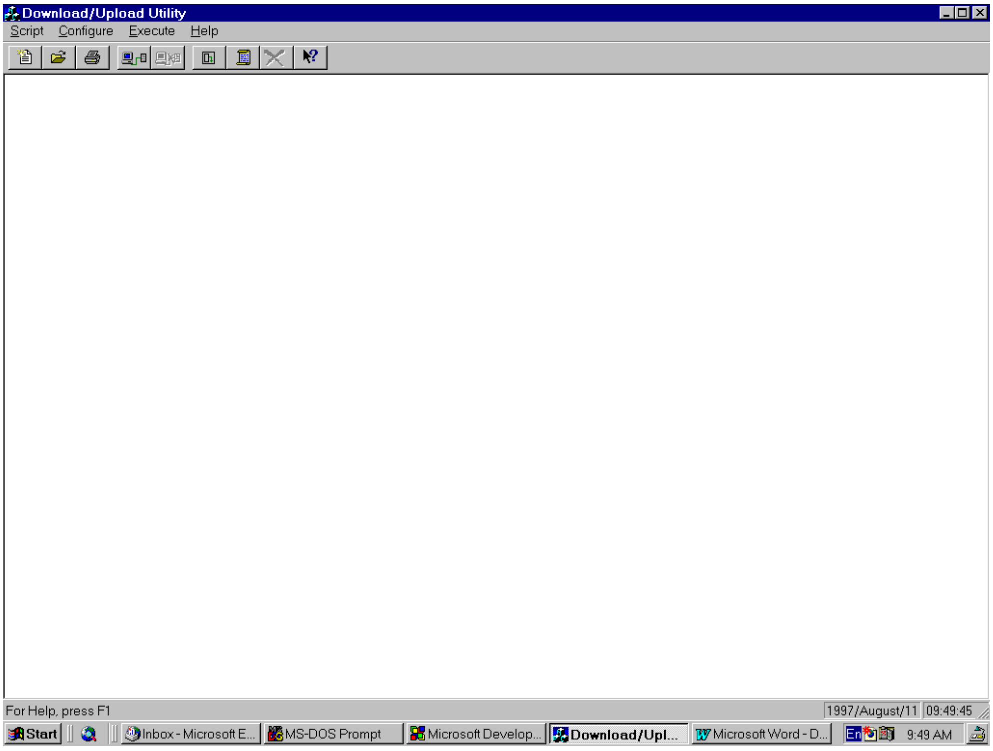


11. Screen Descriptions

11.1 Menu Screen

The following screen will be displayed, which has four menus on the Menu Bar. They are:

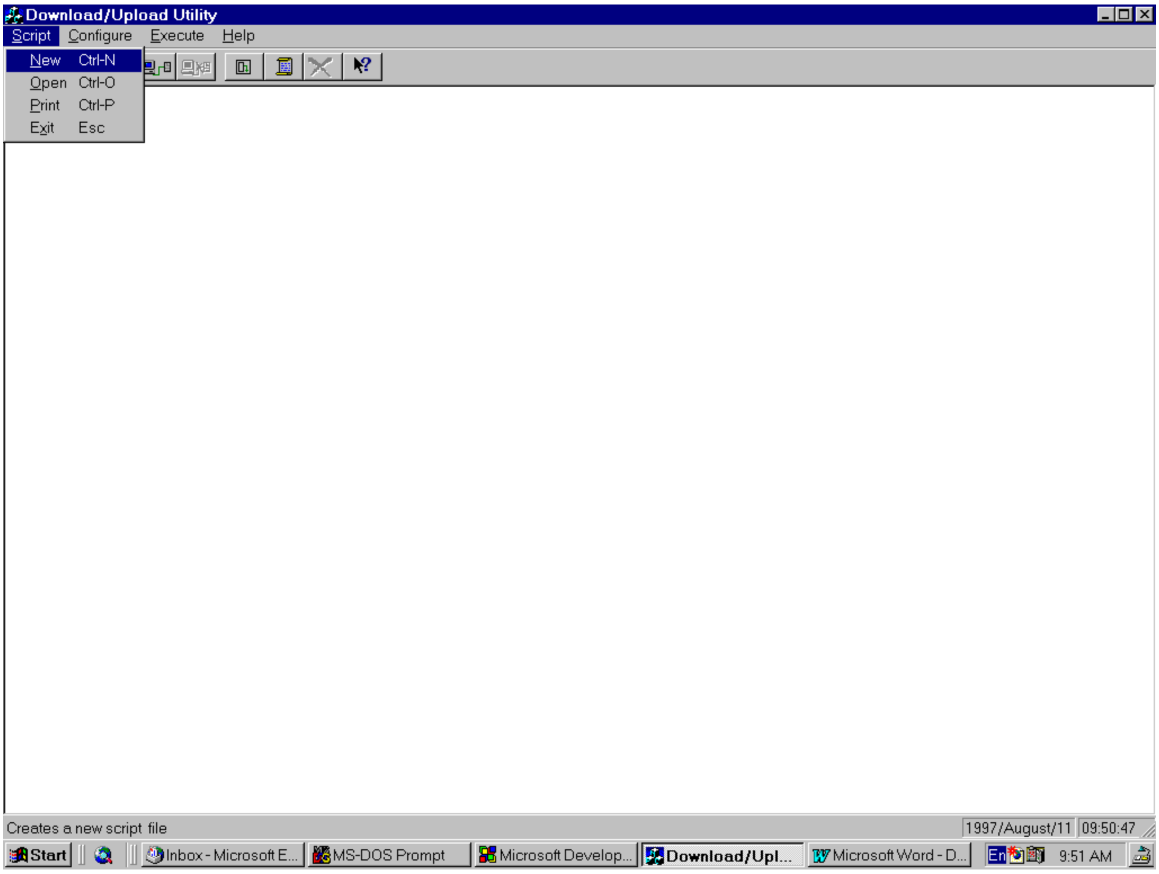
1. Script : This leads to the Script File Editor Wizard screens
2. Configure : This leads to the Configuration File Editor Wizard screen
3. Execute : This leads to the Command Execution Wizard screen
4. Help : This leads to the Help screens



11.2 Script Menu

This menu has the following pull down sub-menus:

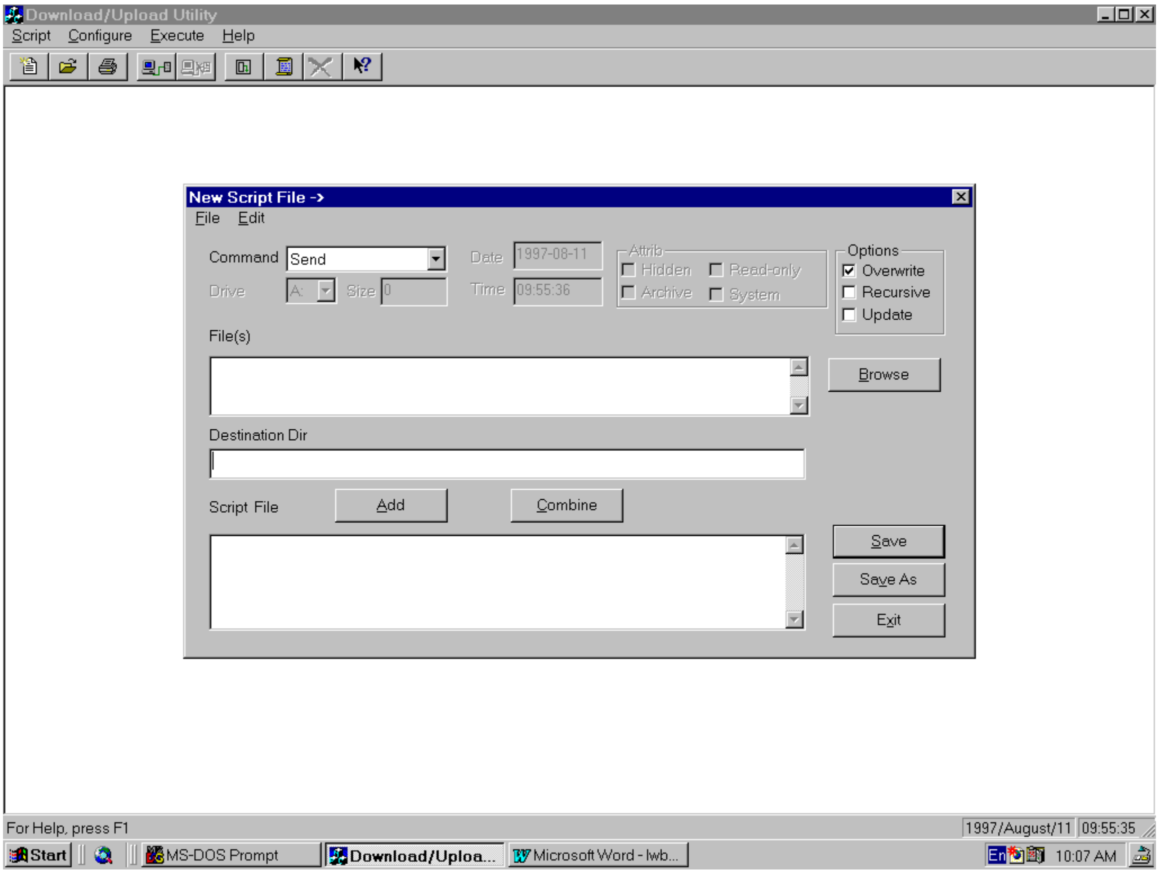
New : The New sub-menu is used for creating a new script file (having an extension **.SCR**)
Open : The Open sub-menu is used for opening and editing an existing script file.
Print : The Print sub-menu is used for printing a script file. This is not implemented as of now.
Exit : The Exit sub-menu is used for exiting the Upload/Download utility.



Creates a new script file

11.2.1 Script File Editor

On selecting the New sub-menu option, the following screen is displayed:



Command: Send
Drive: A:
Size: 0
Date: 1997-08-11
Time: 09:55:36

Attrib
□ Hidden □ Read-only
□ Archive □ System

Options
☑ Overwrite
□ Recursive
□ Update

For Help, press F1
1997/August/11 09:55:35

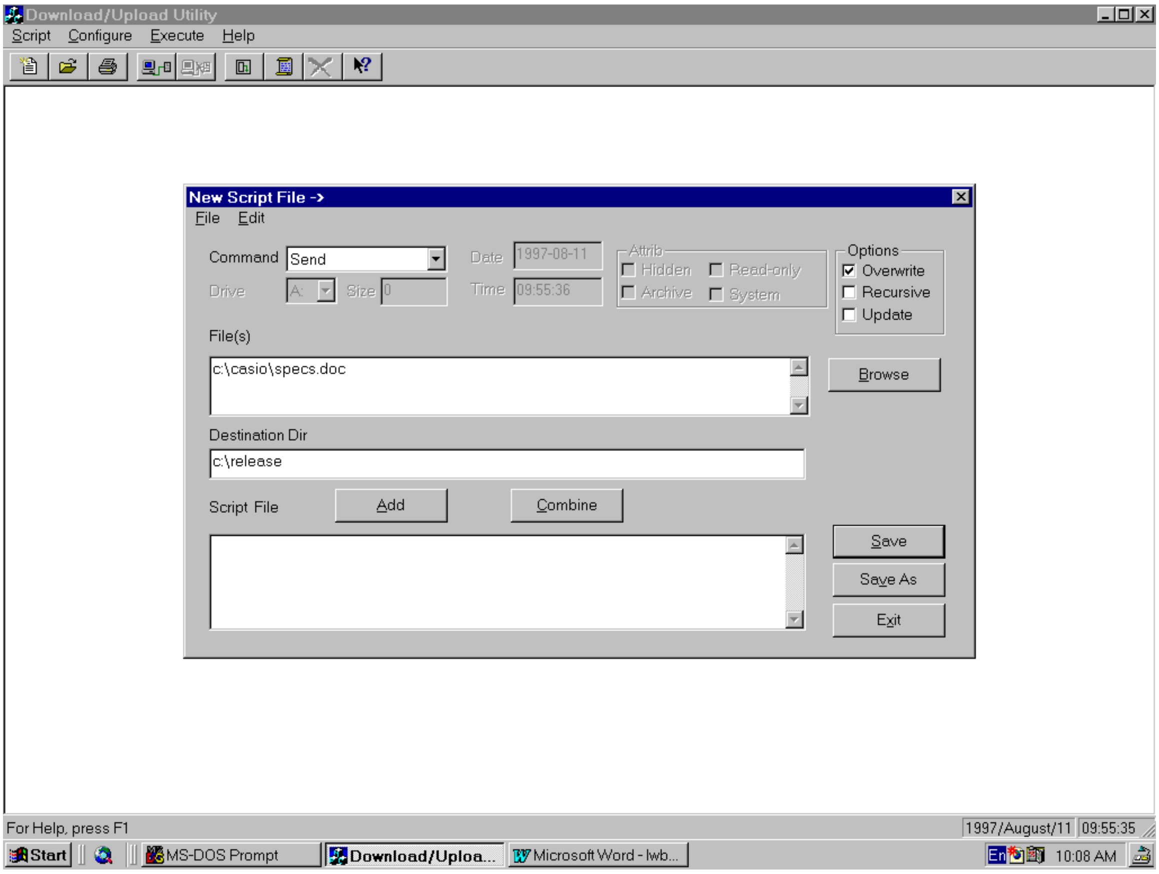
From the menu above, the user can select the required command from the Command box.
On selecting the command, the options corresponding to the command are enabled. For example, if the command chosen is “Send”, the following options are enabled:

File(s), Destination Dir, Options

After the options are selected, the user can do one of the followings:

1. Select “Exit” to exit the Script File Editor
2. Select “Add” to add the command to the script file
3. Select “Save” to save the script file
4. Select “Save As” to save the script file with a different name
5. Select “Combine” to add the currently selected command with the previous command on the same line of the script file.

The Browse button can be pressed to obtain a Directory Service Dialog Box from which the user can select the files to be sent.
The following figure shows the screen with the command and options selected, but before the Add is pressed.



Command: Send

Drive: A:

Size: 0

Date: 1997-08-11

Time: 09:55:36

Attrib: ☐ Hidden ☐ Read-only ☐ Archive ☐ System

Options: ☑ Overwrite ☐ Recursive ☐ Update

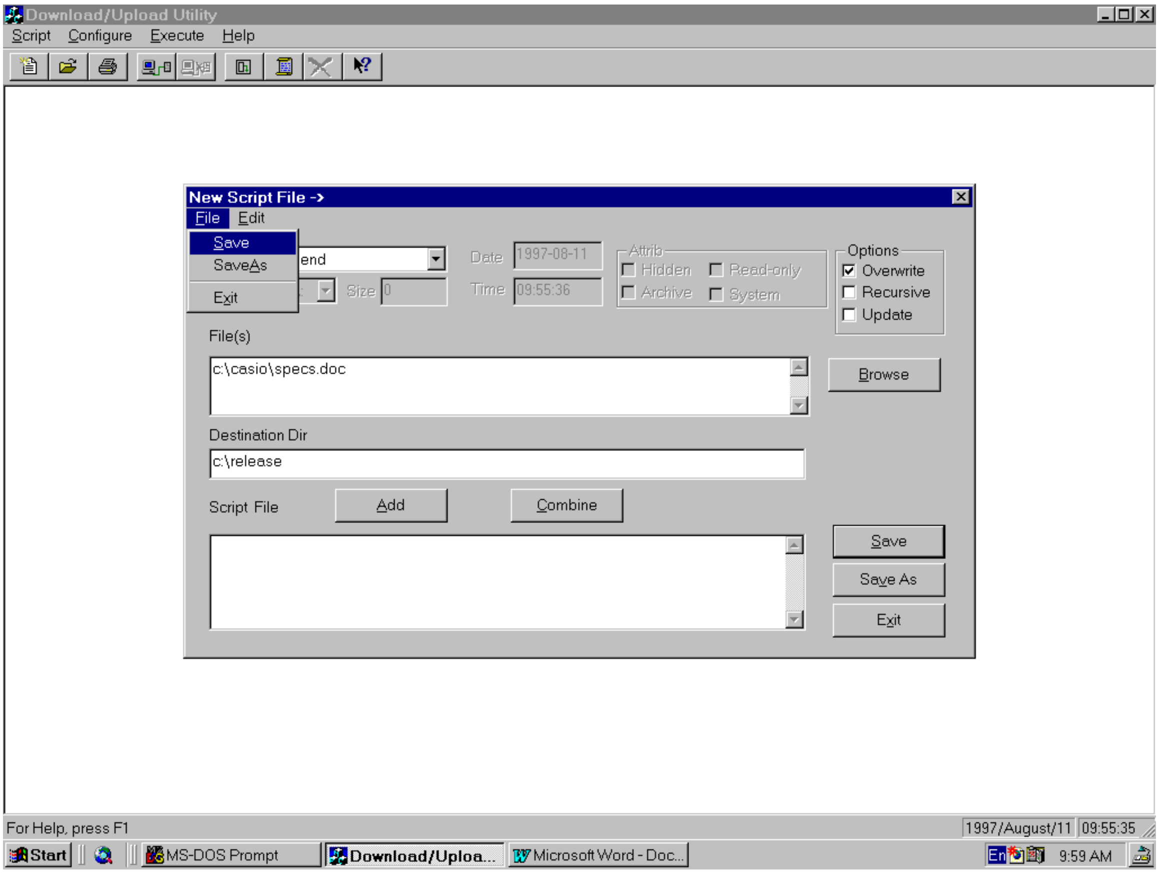
File(s): c:\casio\specs.doc

Destination Dir: c:\release

Script File:

For Help, press F1 | 1997/August/11 | 09:55:35

The screen on the next page shows the Script File Editor Wizard before the Add button has been selected. When the Add button is pressed, the new line of the script file will appear in the Script File Window at the bottom of the screen. The screen also shows the Save and Save As sub-menus on the File menu of the Script File Editor.



Command: ...end
Date 1997-08-11
Time 09:55:36
Size 0

Attrib
□ Hidden □ Read-only
□ Archive □ System

Options
☑ Overwrite
□ Recursive
□ Update

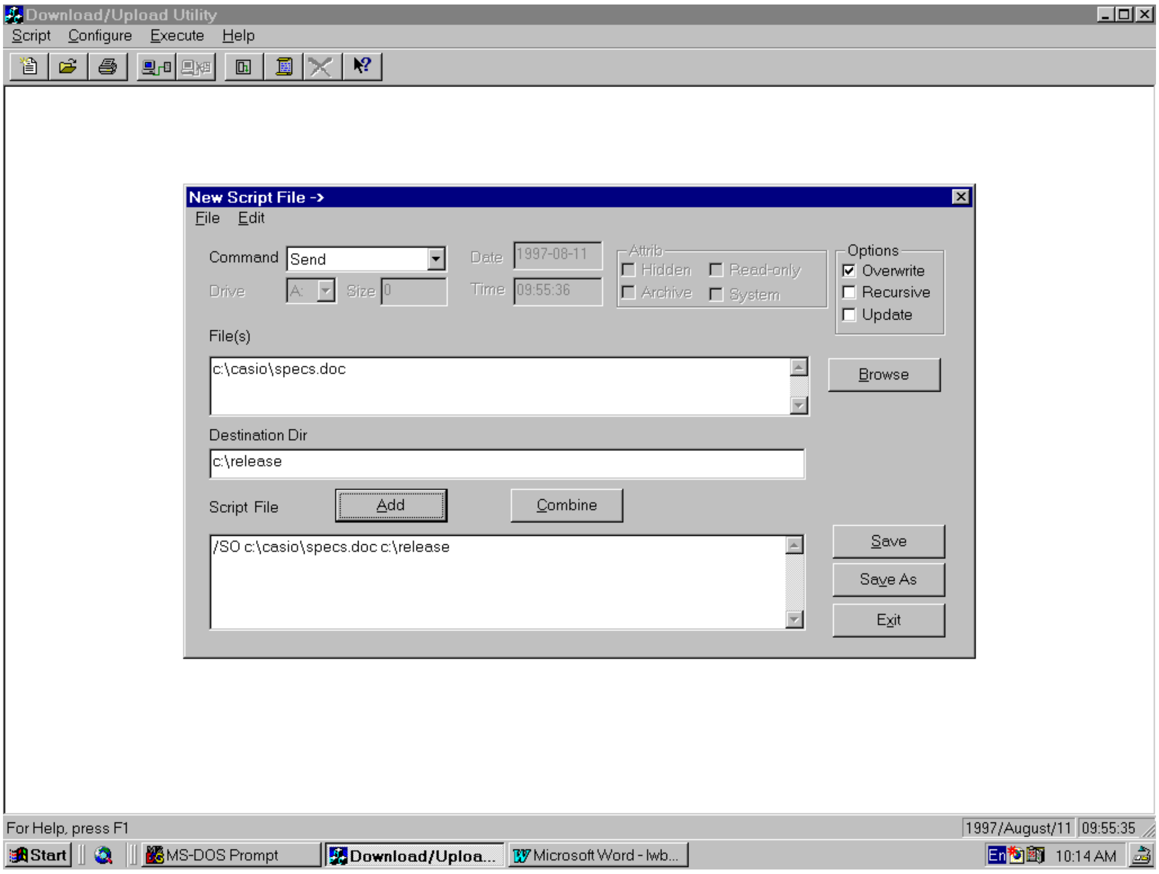
File(s)
c:\casio\specs.doc

Destination Dir
c:\release

Script File

For Help, press F1 1997/August/11 09:55:35

When the Add button is pressed, the screen appears as follows:



Command Send
Drive A:
Size 0
Date 1997-08-11
Time 09:55:36

Attrib
□ Hidden □ Read-only
□ Archive □ System

Options
☑ Overwrite
□ Recursive
□ Update

File(s)
c:\casio\specs.doc

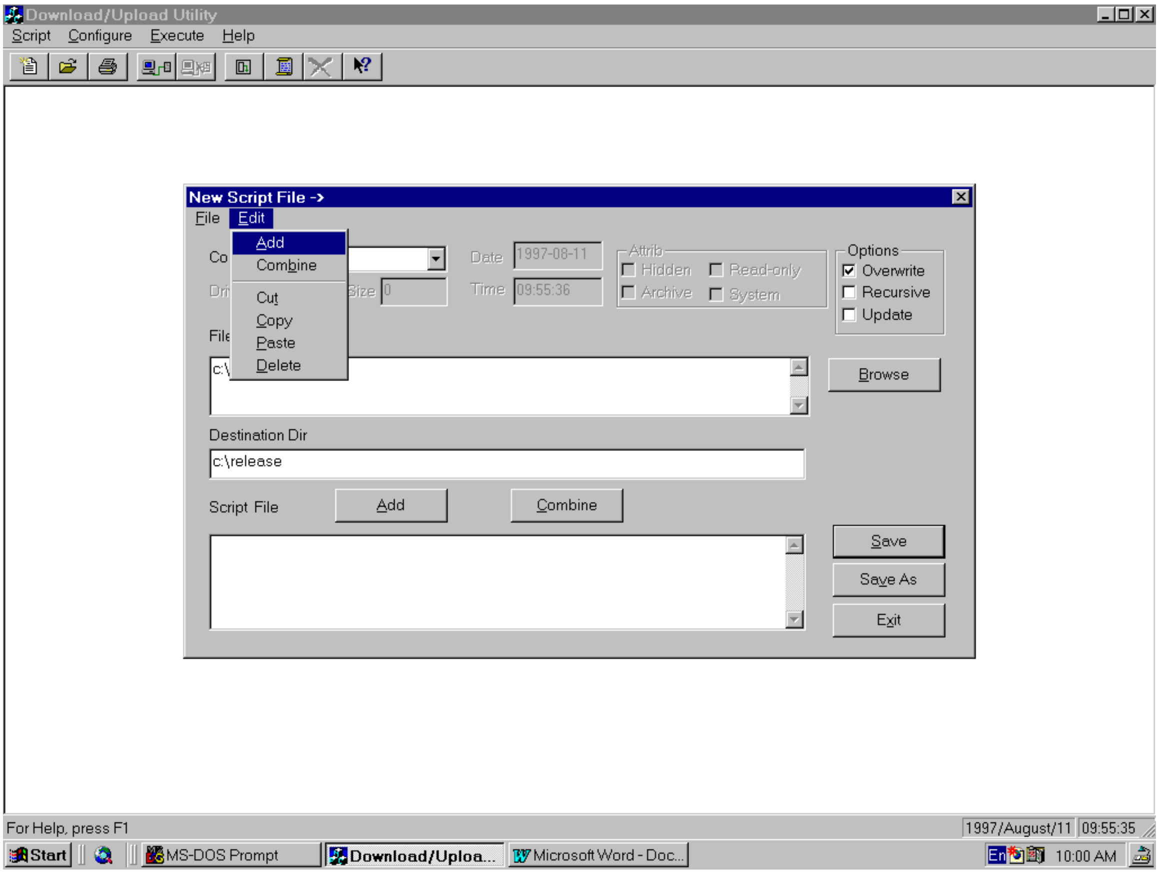
Destination Dir
c:\release

Script File

```
/SO c:\casio\specs.doc c:\release
```

For Help, press F1 1997/August/11 09:55:35

The Add, Combine and Cut, Copy, Paste, Delete tool-bar buttons are used to perform Add, Combine, Cut, Copy, Paste and Delete operations on the script file in the Script File Window. They are not valid on any other parts of the Script File Editor.



New Script File ->

Date 1997-08-11

Time 09:55:36

Size 0

Attrib
☐ Hidden ☐ Read-only
☐ Archive ☐ System

Options
☑ Overwrite
☐ Recursive
☐ Update

File c:\

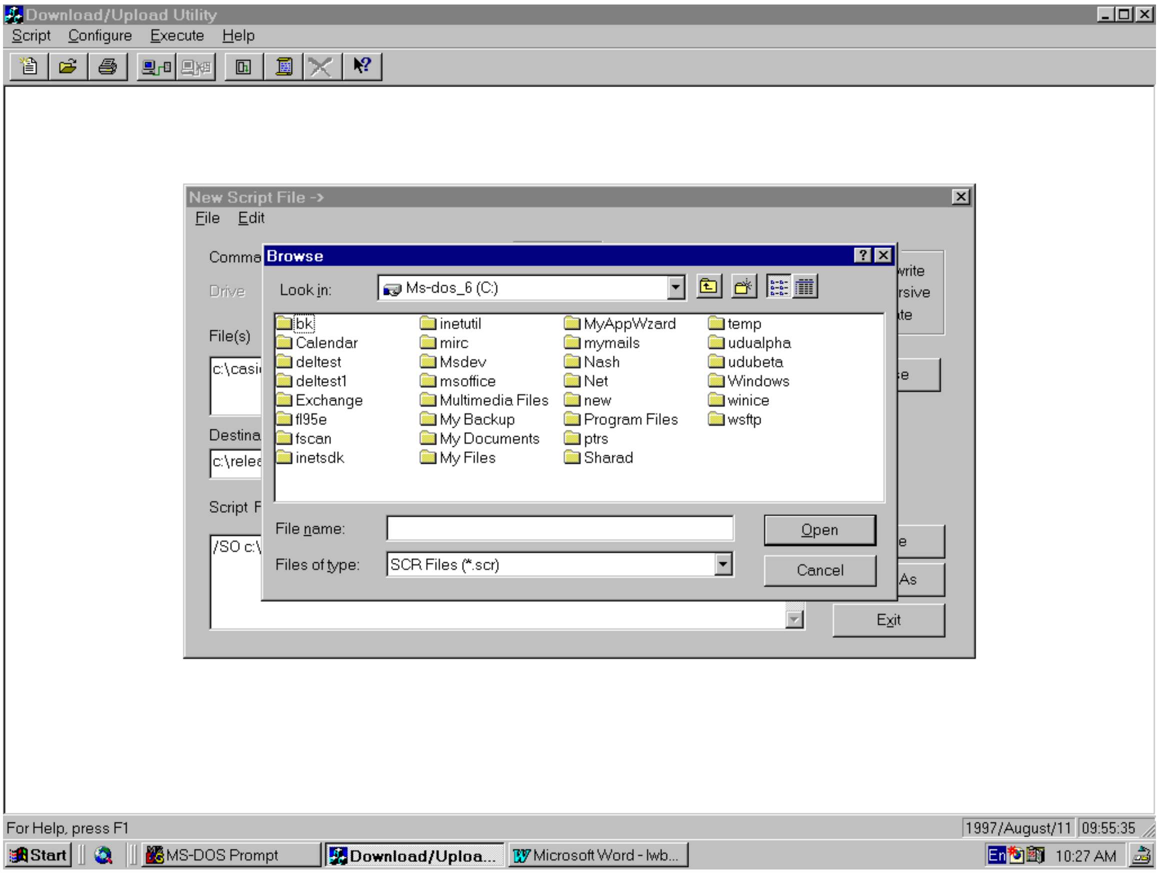
Destination Dir c:\release

Script File

For Help, press F1 1997/August/11 09:55:35

:

On selecting the Browse button, the following Directory Services dialog box is displayed:



Look in: Ms-dos_6 (C:)

File(s): c:\casi

Destina: c:\relee

Script F: /SO c:\

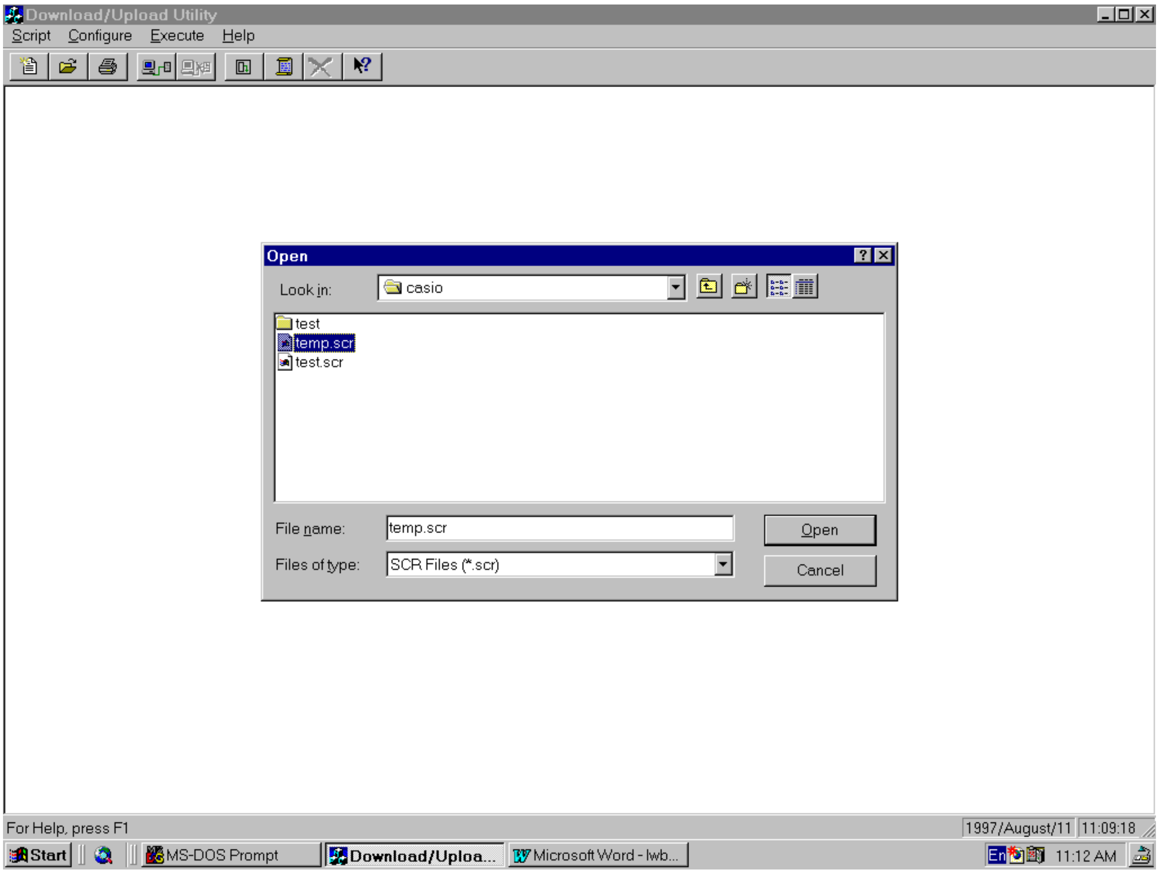
Files of type: SCR Files (*.scr)

For Help, press F1

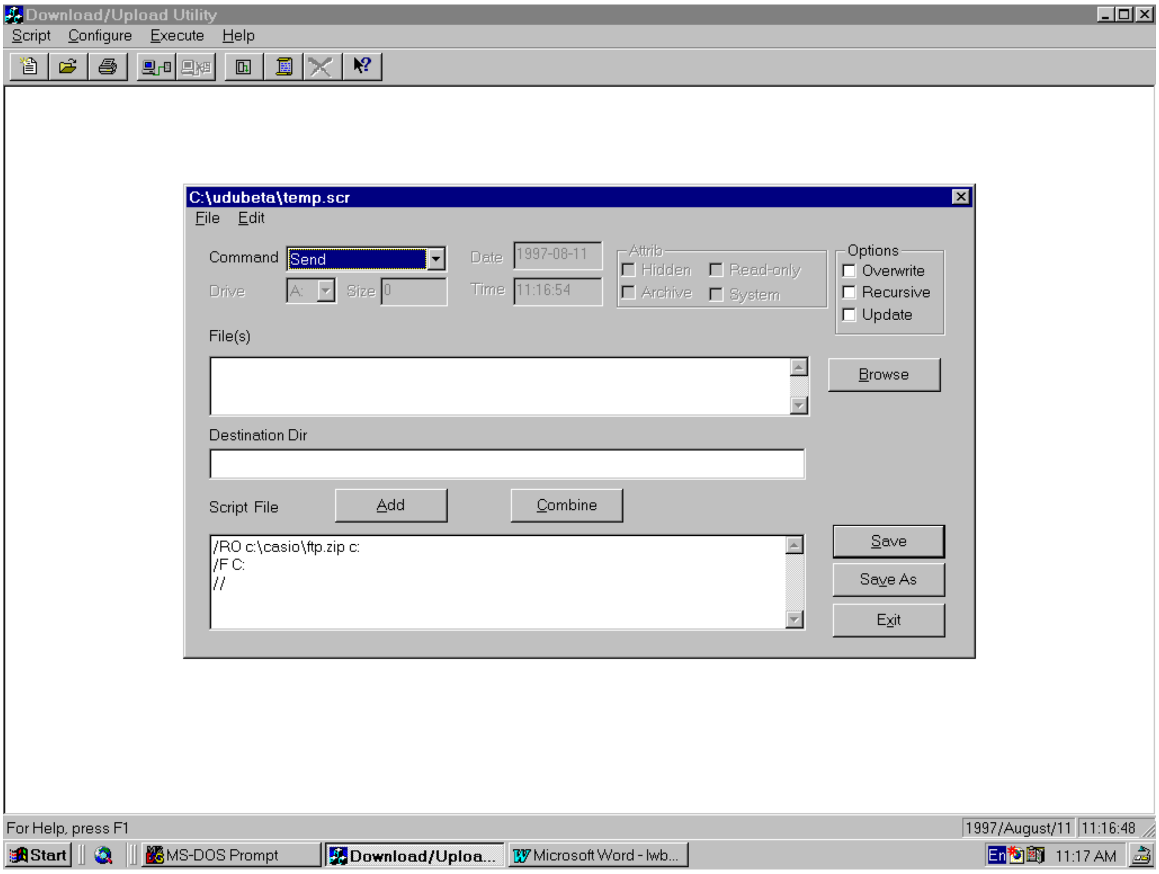
1997/August/11 09:55:35

11.2.2 Open

On selecting the Open sub-menu option, the following screen is displayed.



The following screen shows the display of the script file temp.scr :



Command Send

Drive A: Size 0

Date 1997-08-11

Time 11:16:54

Attrib
□ Hidden □ Read-only
□ Archive □ System

Options
□ Overwrite
□ Recursive
□ Update

File(s)

Destination Dir

Script File

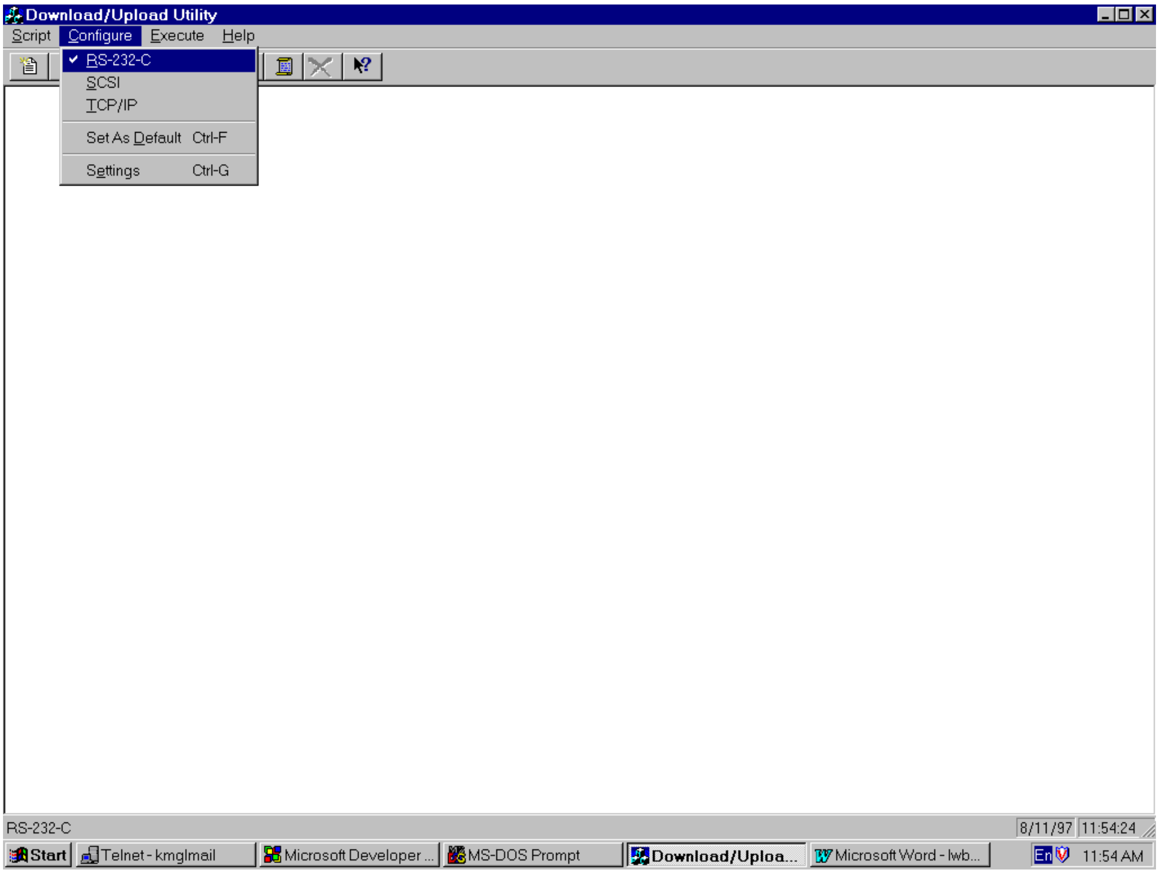
```
/RO c:\casio\ftp.zip c:
/F C:
//
```

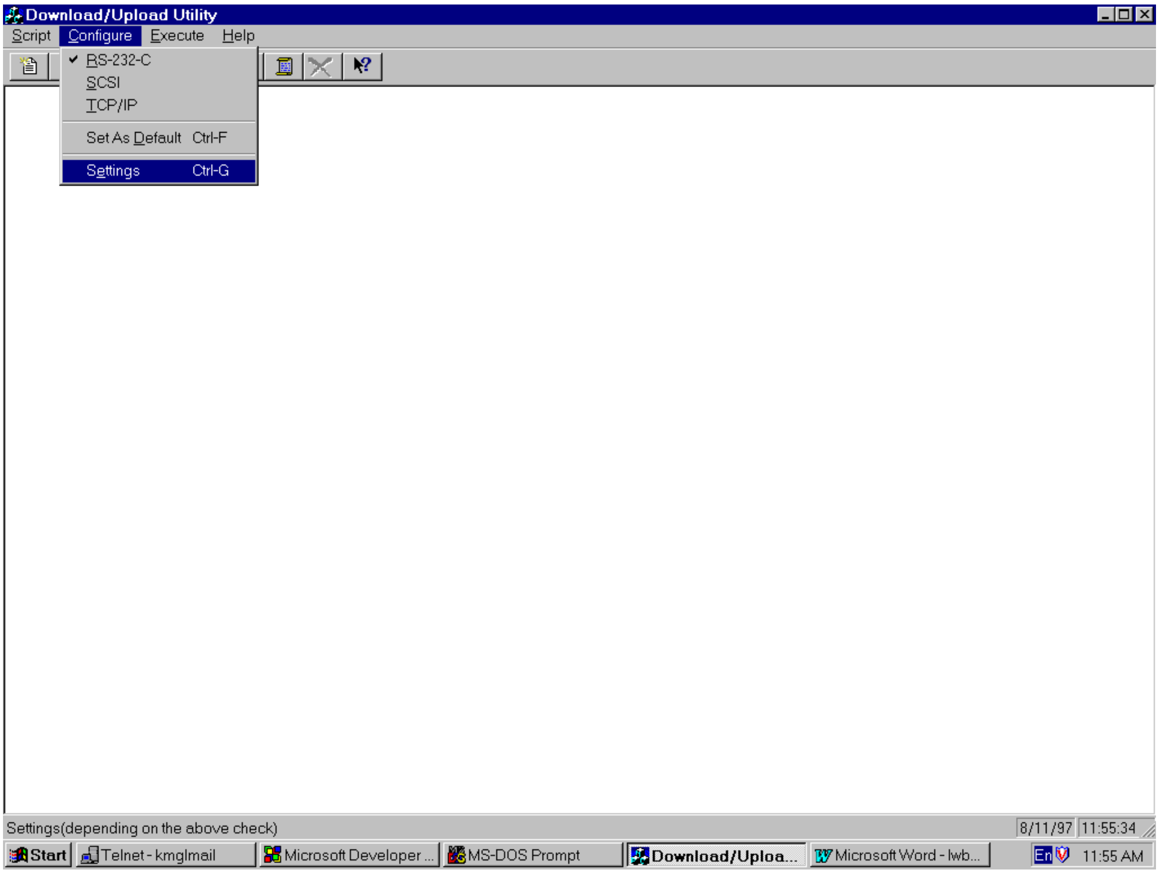
11.3 Configure Set Menu

This menu enables the user to configure the communication interface with the I/O Box. This menu has the following pull down menus:

1. RS-232C
2. SCSI
3. TCP/IP (currently not available)
4. Set As Default
5. Settings

The first three pull down menus enable the selection of one of the two interfaces - RS-232C and SCSI. A check mark appears against the selected interface. If the selection of TCP/IP is disabled on the pull-down menu, this indicates that the interface is not available in the current version of the Upload/Download utility. The following figure shows the RS-232C interface selected, and the Settings option is about to be selected.





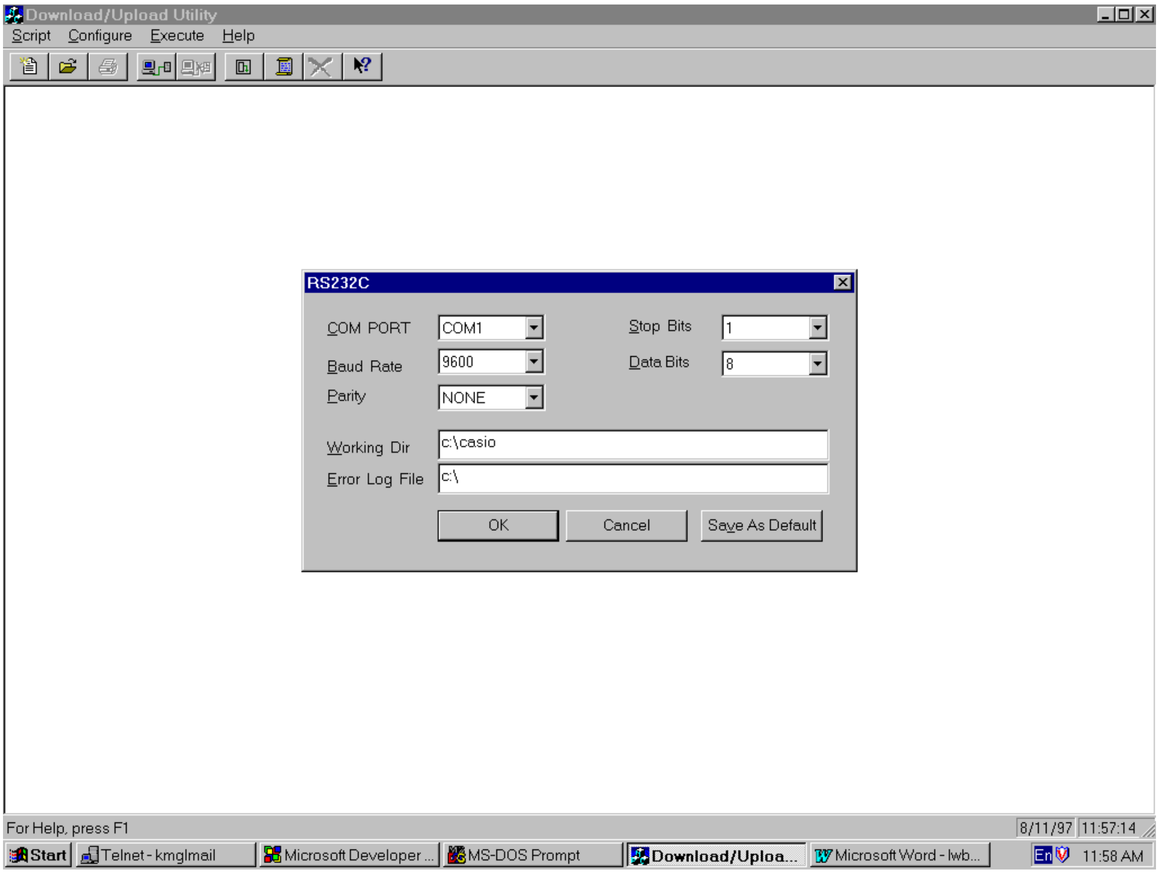
Settings(depending on the above check)

11.3.1 Communication Set Screen (RS-232C)

If the SETTINGS option is chosen, the following dialog box is displayed. The dialog box contents depends upon the interface selected (RS-232C or SCSI).

If the options are chosen and the OK button is clicked, the settings are saved to the Configuration File (DEVICE.INI). If the Save As Default button is selected, the RS-232C interface with the defined settings are saved as the default interface and settings for the utility.

The Working Directory and the name of the Error Log file also have to be specified using this configuration dialog box.
Clicking on the Cancel button cancels any selections/changes and reverts to the earlier settings.



COM PORT: COM1

Baud Rate: 9600

Parity: NONE

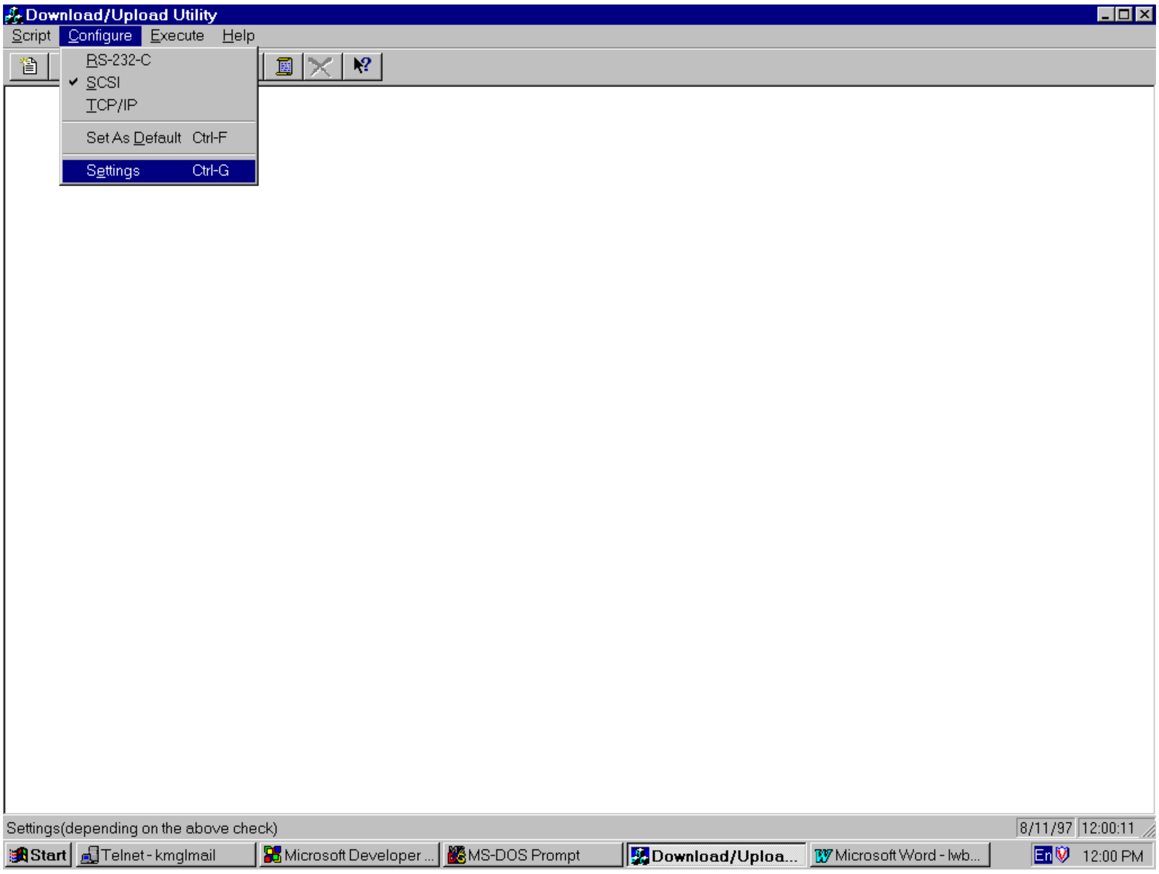
Stop Bits: 1

Data Bits: 8

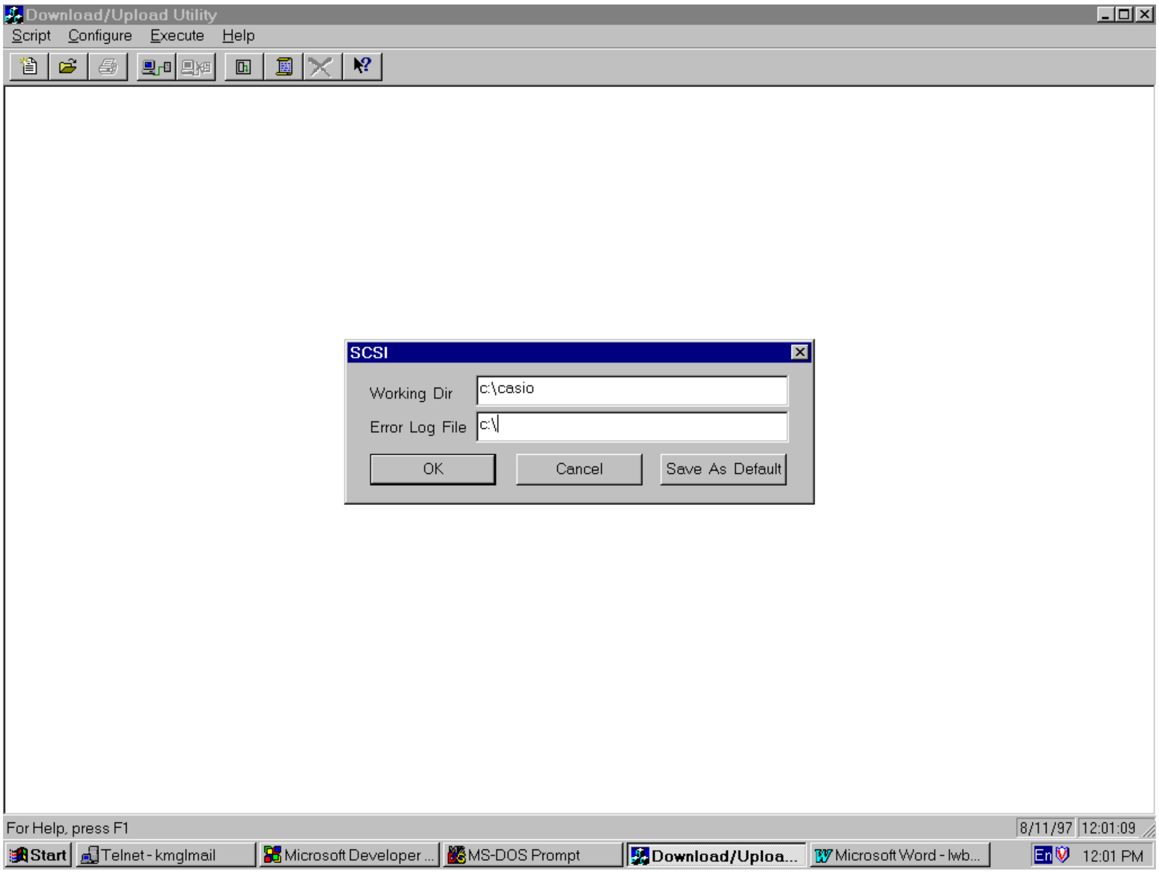
Working Dir: c:\casio

Error Log File: c:\

The following two screens show the selection of SCSI interface and the corresponding dialog boxes for the setting.



Settings(depending on the above check)



Working Dir c:\casio

Error Log File c:\

For Help, press F1

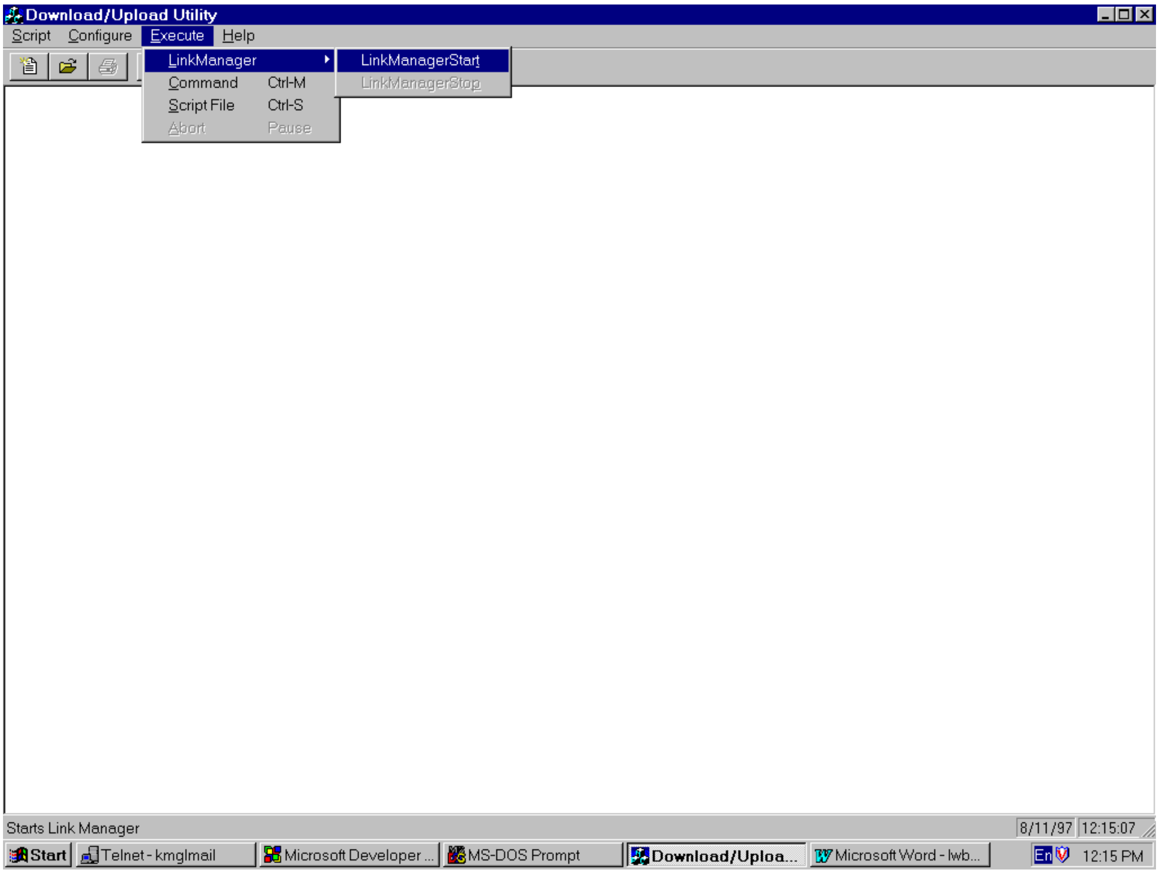
11.4 Execute Menu

This menu has the following pull down sub-menus:

1. LinkManager
2. Command
3. Script
4. Abort

The LinkManager has two sub menus -- LinkManagerStart and LinkManagerStop.

The following screen shows the selection of the Link Manager sub-menu. When the LinkManager --> LinkManagerStart is clicked, the LinkManager is started up.



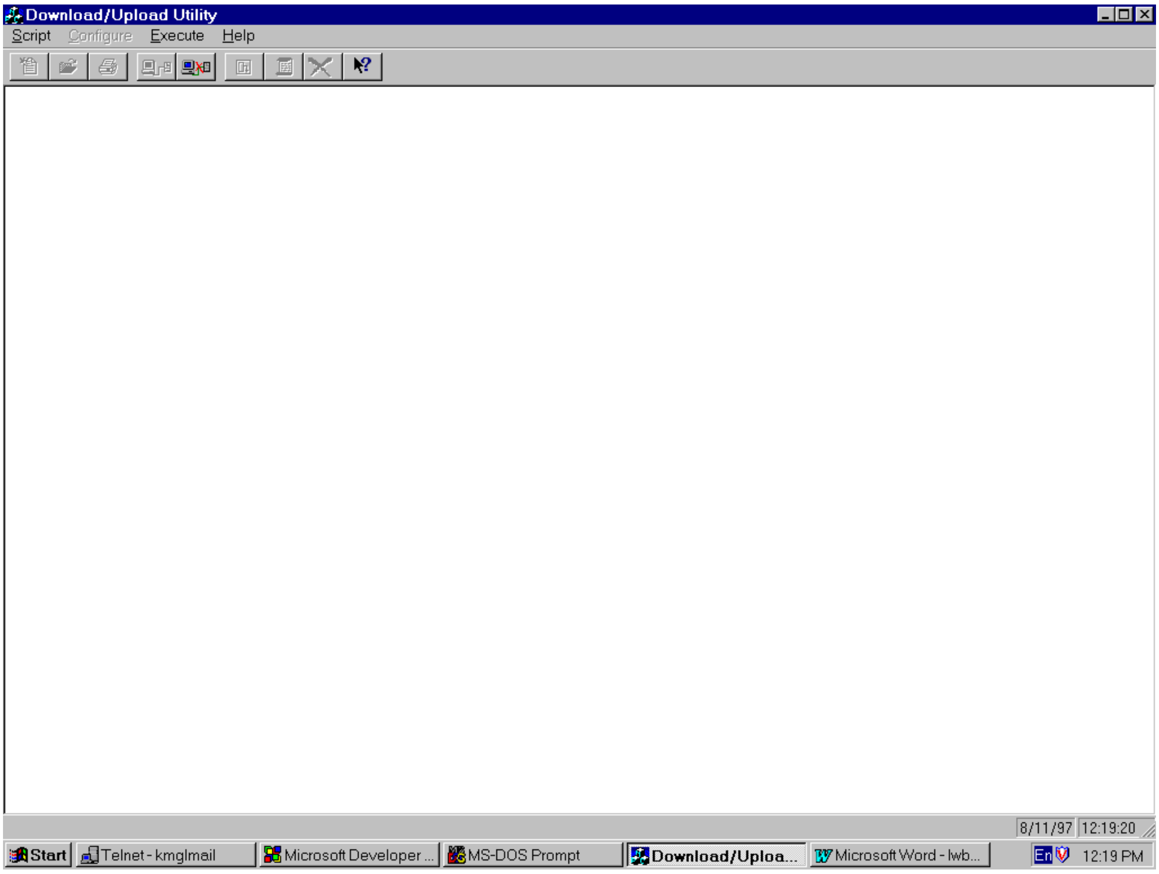
Also, the Configure menu is disabled, thus, preventing the user from editing the Configuration File (DEVICE.INI) when the File Transfer protocol utility is active. Also, at this stage, the New and Open sub-menus of the Script menu are disabled (not shown in the screen). This prevents the user from editing the Script file or creating a script file when a session is in progress (protocol utility is active).

Thus, all editing functions are disabled when the protocol utility is active.

11.4.1 Server Mode

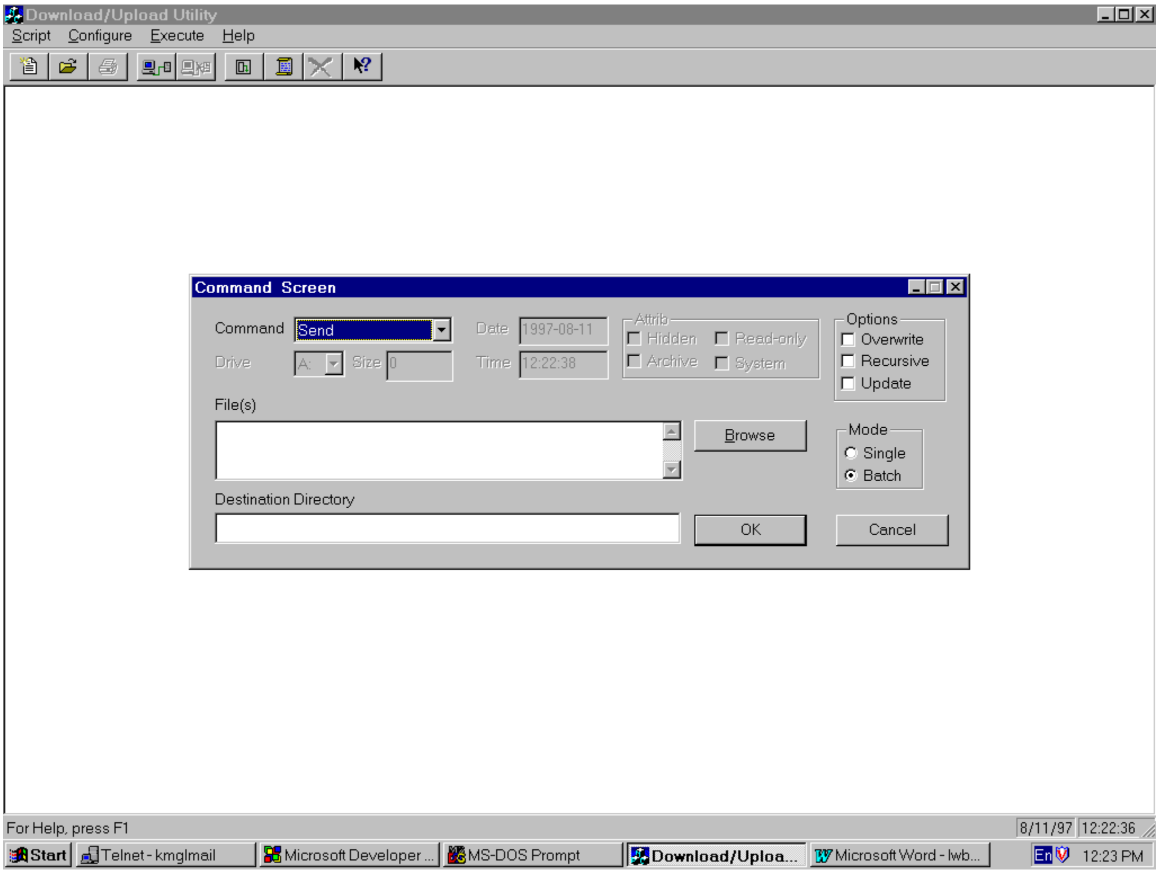
The LinkManagerSTART sub-menu is used for starting up the File Transfer Protocol program. On selecting this option, the communications are initialized, and the File Transfer utility enters the Server Mode, where it listens at the selected communication interface (RS-232C serial port, SCSI port) for incoming connection requests from the I/O Box.

If a connection request is received, communication with the I/O Box is established, and the utility enters the Server Mode. In this mode, it executes any command received from the I/O Box (or from IT-2000).
Once this option is selected, it is prevented by being selected again by disabling the menu.



11.4.2 Command Execution

The Command sub-menu leads to the following Command Execution Wizard screen.
The following screen shows the Command Execution Wizard displayed with the RS-232C interface selected from the Configure menu.



Command: Send
Drive: A: Size: 0
Date: 1997-08-11
Time: 12:22:38
Attrib: □ Hidden □ Read-only □ Archive □ System
Options: □ Overwrite □ Recursive □ Update
File(s):
Destination Directory:
Mode: ○ Single ● Batch

For Help, press F1   8/11/97   12:22:36

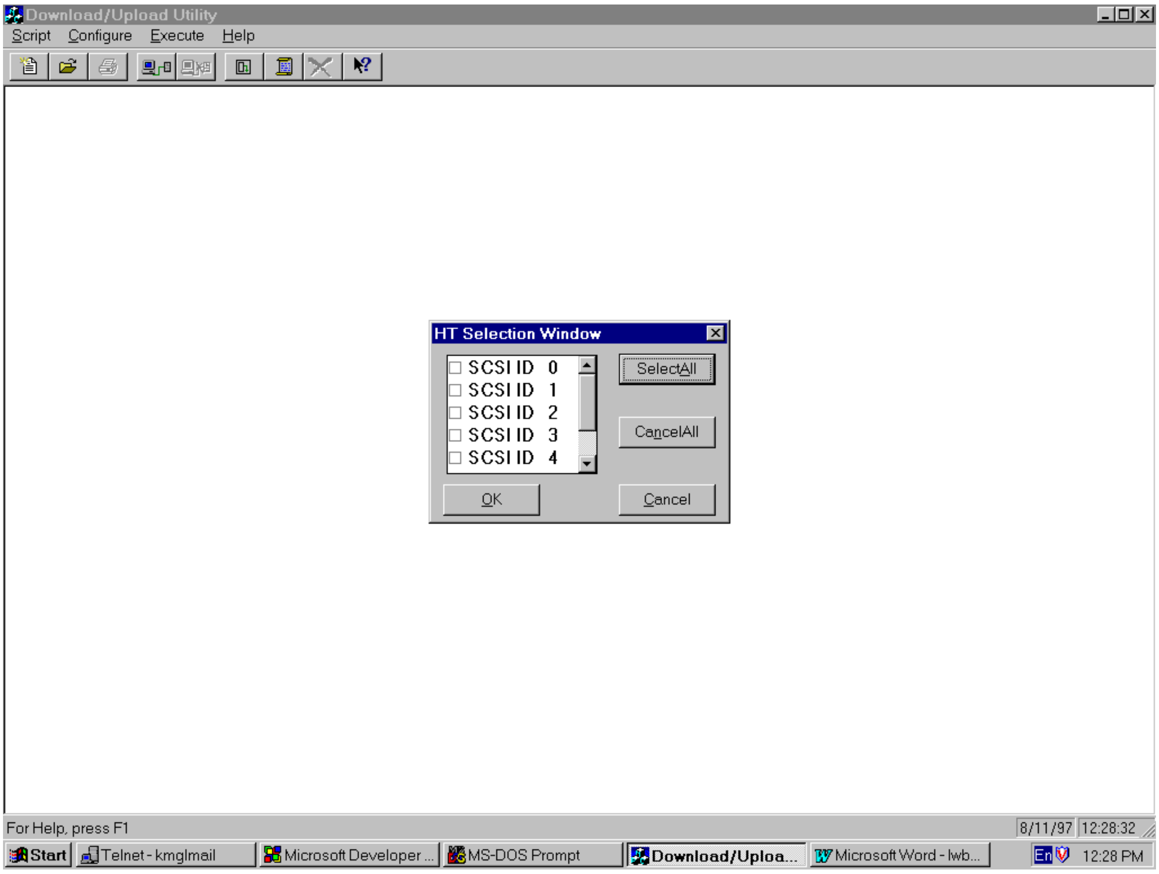
For each command, the valid options are only enabled, while all invalid options are grayed out or disabled.

Option of Single Shot Mode or Batch Mode

The Command Execution Wizard Screen gives the user the options to execute each command once (Single Shot Mode) or as a batch (many times - Batch Mode) by proper selection of the Mode option.
If the Batch Mode is selected, the command is executed repeatedly until the user selects the ABORT TRANSFER sub-menu on the EXECUTE menu.
The default mode is the Batch Mode.

If the interface selected is SCSI, then the following HT Selection Dialog is displayed. The user can then select the IDs of the HTs to which the command is to be sent. On pressing OK, the Command Execution Wizard as shown above is displayed, and the command entered from the screen will be sent to the selected HT IDs.

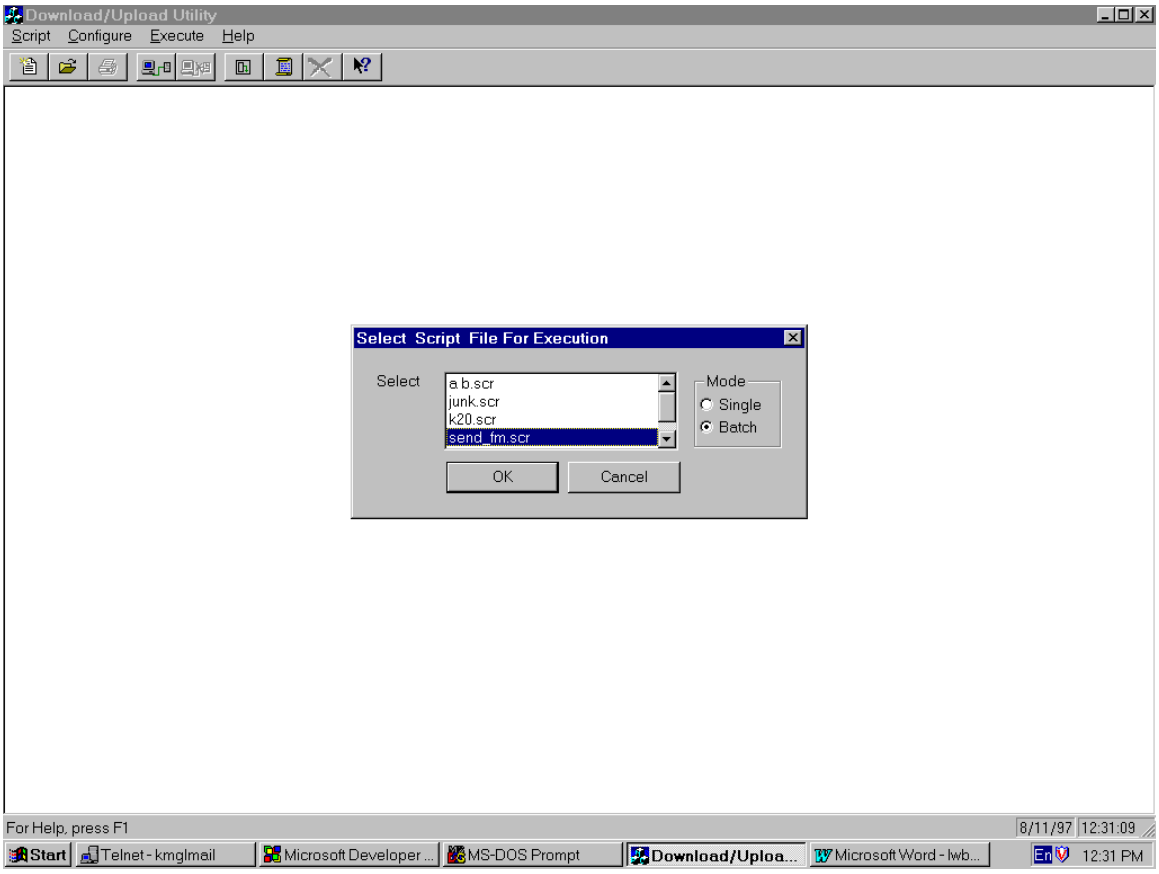
If Select All option is chosen, all IDs are selected, and if Cancel All is chosen, all selected IDs are canceled.



11.4.3 Script Execution

The following screen shows the dialog box displayed when the user selects the Script pull down menu from the Execute menu and the interface selected is RS-232C. On pressing OK, the selected Script File is executed. The user has the option to use the batch or One shot mode.

If the interface selected is SCSI, the HT Selection Dialog appears as shown above, before the following screen is displayed.



Select Script File For Execution

Select
a b.scr
junk.scr
k20.scr
send_fm.scr

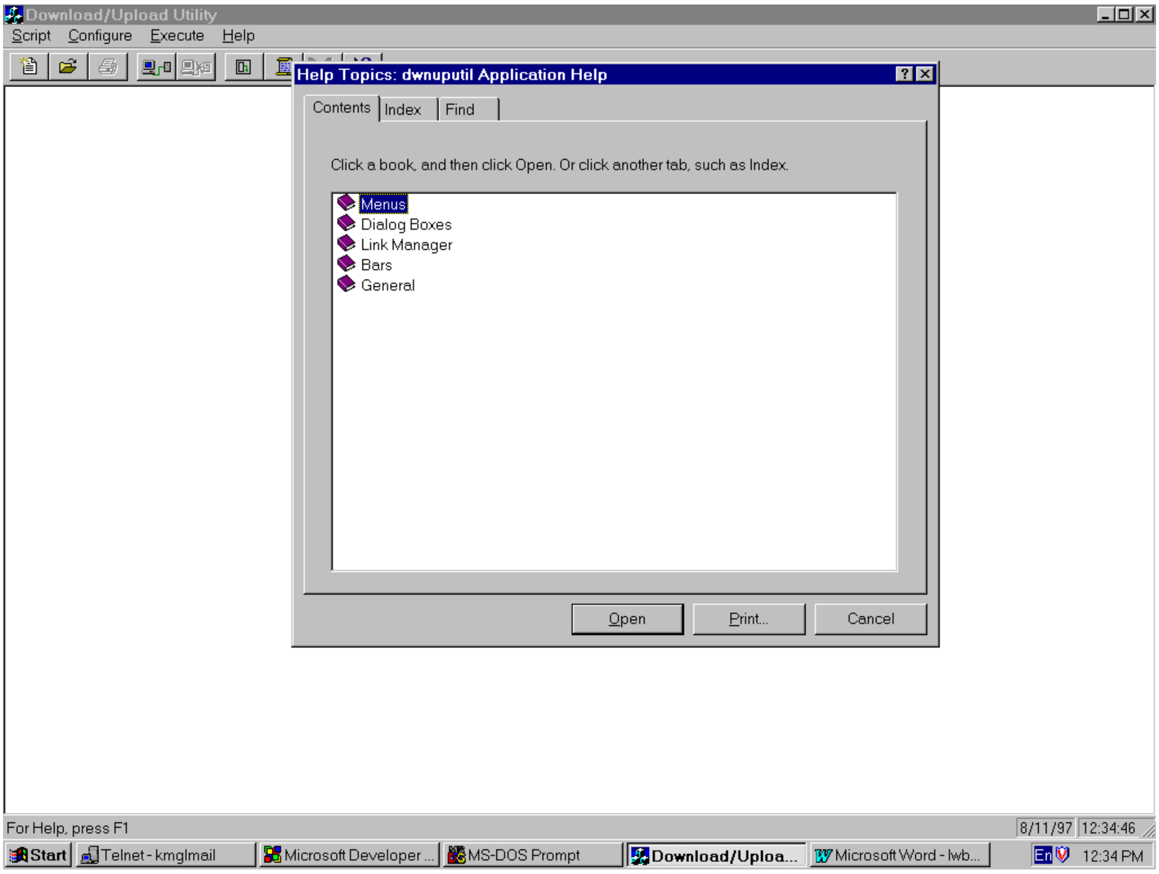
Mode
○ Single
● Batch

11.4.4 Abort

The Abort sub-menu ends a session, when selected. If the interface selected is RS-232C, then only one session exists. But if the interface selected is SCSI, then an HT selection dialog is displayed, with all the active session IDs enabled. The user can then select the sessions that have to be aborted.

11.5 Help

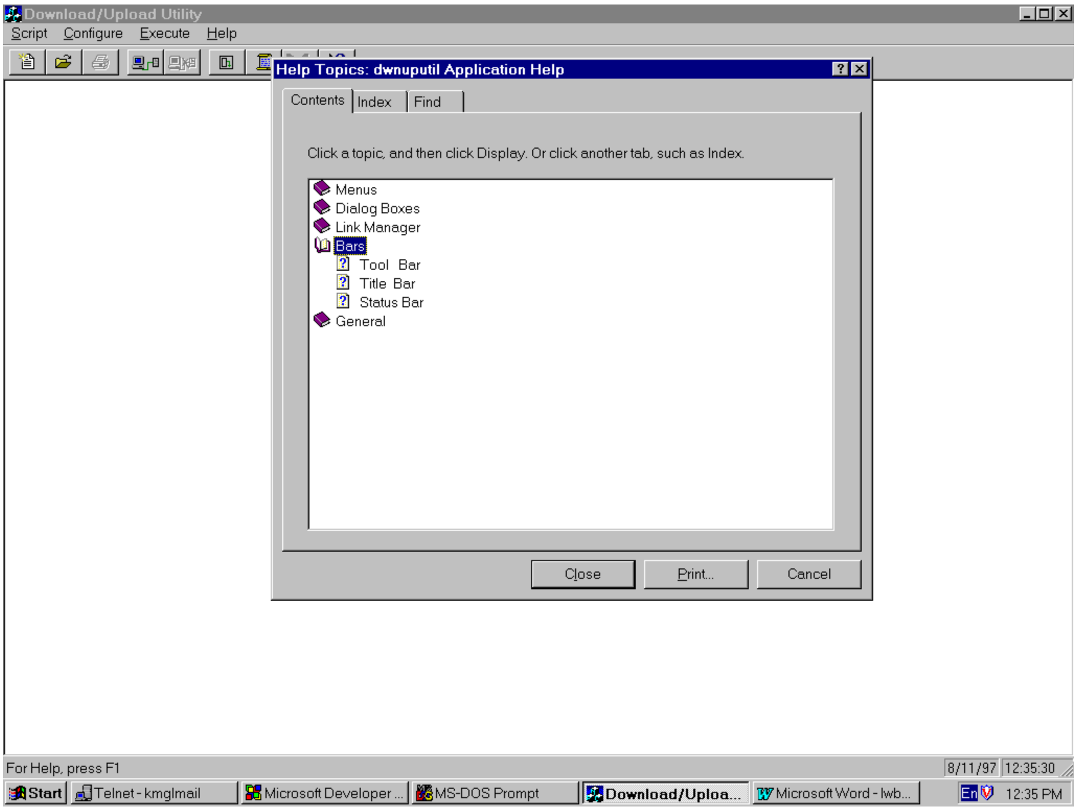
The following screen shows a typical Help screen which appears when the Help menu is selected. The user can search for help on any of the Upload/Download utility functions from this screen.



Click a book, and then click Open. Or click another tab, such as Index.

For Help, press F1

8/11/97 12:34:46



Click a topic, and then click Display. Or click another tab, such as Index.

12. Status and Error Message Display (GUI)

The Upload/Download utility screens have the facility to display the status and error message of File Transmission and Reception. The following functions are supported:

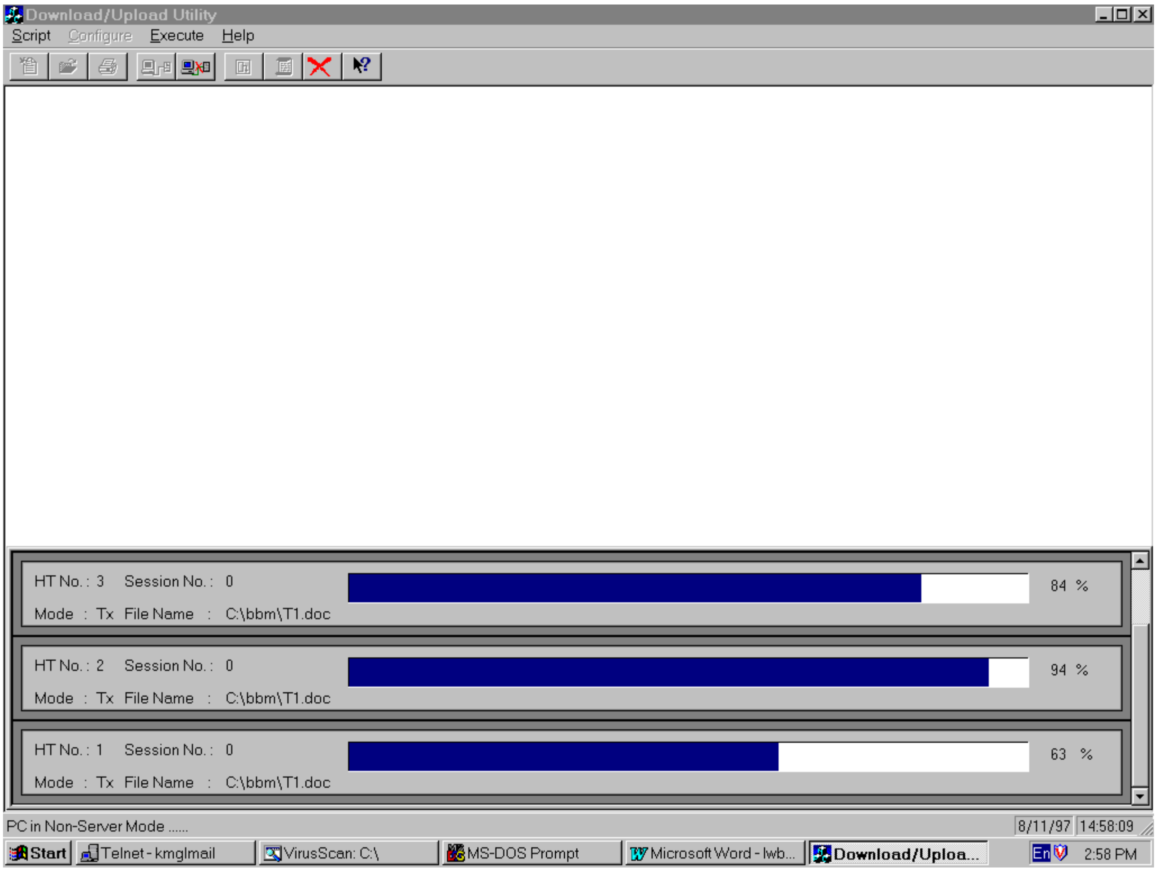
12.1 Status Display

Display of progress bar for File Transmission and Reception. This display contains the following information:

1. Session No.
2. File Name with full path name
3. Tx or Rx (Send/Receive))
4. Progress bar
5. Percentage of file transmitted/received (range : 0 to 100%)

In case of multiple sessions, a maximum of 3 progress bars are displayed on the screen , and the other bars can be seen by using the scroll bar.

The following screen shows the progress bar display during file transmission:



HT No. : 3 Session No. : 0 84 %
Mode : Tx File Name : C:\bbm\T1.doc

HT No. : 2 Session No. : 0 94 %
Mode : Tx File Name : C:\bbm\T1.doc

HT No. : 1 Session No. : 0 63 %
Mode : Tx File Name : C:\bbm\T1.doc

PC in Non-Server Mode 8/11/97 14:58:09

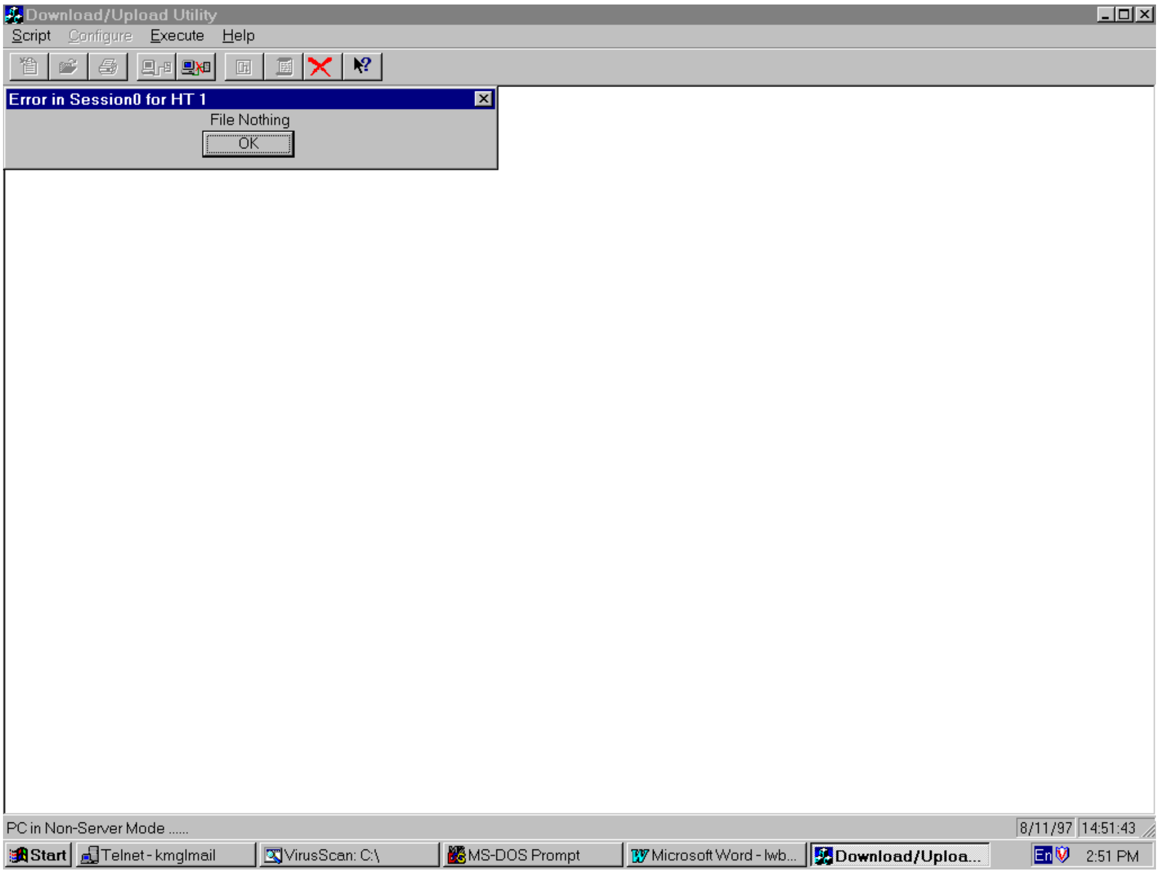
12.2 Error Message Display

Error Message display in the form of an Error Message Dialog Box.
A maximum of 8 error message boxes can be displayed on the screen at any time. If more than seven errors occur, the eighth message box will display the following message:

> “More Errors Exist... See Error Log File for details! “

The Error Message Display boxes will remain on the screen until the user acknowledges the message by clicking on the OK button. The user has to press/click OK once for each Error Message Display box.

The following screen shows the display of an error message dialog box upon occurrence of an error during file transmission.



Error in Session0 for HT 1

File Nothing

PC in Non-Server Mode ......

8/11/97 14:51:43

13. Syntax of Script File Commands

This section describes the syntax of the script file commands. Script files shall have the extension .SCR, and they shall be stored in the Working Directory as defined in the Configuration file DEVICE.INI

```
 <SCRIPT FILE> ::= <COMMANDS>
        <COMMANDS> ::= <COMMANDS>  <COMMAND> | null
        <COMMAND> ::= '/' <CMDBODY> <LS>
        <CMDBODY> ::=   <APPEND>
                     | <CHILD_PROC>
                     | <DELETE>
                     | <FORMAT>
                     | <BEEP>
                     | <RENAME>
                     | <RECEIVE>
                     | <SEND>
                     | <TIME_ADJUST>
                     | <END_SESSION>

        <APPEND> ::= <APPEND_CMD> <APD_OPTION> <SP> <PATHNAME_PAIR>
        <CHILD_PROC> ::= <CHILD_PROC_CMD> <SP> <CMD_PARAMETER>
        <FORMAT> ::= <FORMAT_CMD> <SP> <DRIVE>
        <BEEP> ::= <BEEP_CMD>
        <RENAME> ::= <RENAME_CMD> <SP> <PATHNAME_PAIR>
        <RECEIVE> ::= <RECEIVE_CMD> <OPTIONS> <SP> <PATHNAME_LIST>
        <SEND> ::= <SEND_CMD> <OPTIONS> <SP> <PATHNAME_LIST>
        <PRINT> ::= <PRINT_CMD> <SP> <STRING>
        <TIME_ADJUST> ::= <TIME_CMD> <SP> <TIME_VALUE>
        <END_SESSION> ::= <END_CMD> <PARAM>

        <APPEND_CMD> ::= 'A'
        <CHILD_PROC_CMD> ::= 'C'
        <FORMAT_CMD> ::= 'F'
        <BEEP_CMD> ::= 'B'
        <RENAME_CMD> ::= 'N'
        <RECEIVE_CMD> ::= 'R'
        <SEND_CMD> ::= 'S'
        <TIME_CMD> ::= 'T'
        <PRINT_CMD> ::= 'P'
```

<END_CMD> ::= '/'

```
<PATHNAME_PAIR> ::= <PATHNAME> <DELM> <PATHNAME>
<CMD_PARAMETER> ::= <CMD_NAME> <STRING>
<CMD_NAME> ::= <PATHNAME>
<PATHNAME_LIST> ::=<PATHNAME><DELM> <PATHNAME_LIST> | <PATHNAME>
<DRIVE> ::= <DRIVE_LETTER> ':'
<TIME_VALUE> ::= <DATE> <TIME>
<OPTIONS> ::= <OPTIONS> <OPTION> | NULL
<OPTION> ::= <RECURSIVE_OPTION> | <UPDATE_OPTION>
<RECURSIVE_OPTION> ::= 'R'
<UPDATE_OPTION> ::= 'U'
<APND_OPTION> ::= 'S' | 'R'
<STRING> ::= '"' <CHARS> '"'
<DELM> :: = <SP>
<LS> ::= CR | <SP>
<SP> ::= <SP> SP | SP
<PARAM> ::= <SP> <NUMBER>
```

Notes:

<RECURSIVE_OPTION> is for sending or receiving files and directories and sub-directories.

<UPDATE_OPTION> is for sending or receiving updated files only.

<CHILD_PROC> is for executing child processes .

<PRINT> is for printing string to display.

<PATHNAME> follows the MS-DOS definition. **<PATHNAME>** means directory or file name.

14. Configuration Set File

A typical Configuration Set File (DEVICE.INI) will be as follows:

```
IF=RS232C
ErrorFile=c:\lmwin\uerr.log
WorkFile=c:\lmwin
INFORMATION=CCJ
[RS-232C]
Port=COM1
Baud=38400
Parity=None
Stop=1
Data=8
[SCSI]
ScanTime=3
[ETHERNET]
IPADDR=1.1.1.1
EthernetAddr=2.2.2.2
```

15. Format of LMWIN.INI

The following parameters are to be used by the LMWIN.INI. These parameters are read in when start up. Setting screen is not available for each parameter. In case the file is not existed or corrupted, the default values are valid.

Definition\Default	Range	Description
[CODE]		
-1=F501		Error return value of child processor (fixed currently)
[MODE]		
MINIMIZE=OFF	ON	Execute with the minimum.
	OFF	Execute with Window size.
SERVER=OFF	ON	Server mode is set at start up.
	OFF	Communication mode is set at start up.
1SHOT=OFF	ON	Complete connection by executing once.
	OFF	Start up by the batch mode.
DUPLICATE=OFF	ON	Allow startup of multiple EXEs.
	OFF	Not allow startup of multiple EXEs.
COMMAND_ TIMEOUT=	0	Timeout during the session is infinit.
	1-3600	Timeout during the session is 1 second to 3600 seconds (= 1 hour). Any value greater than 3600 is automatically set to 3600.
CHILD_THROUGH	ON	After start up of the child-process execution, the next command is executed.
	OFF	After start up of the child-process execution, the system waits until an application program is finished.

16. SCSI Interface

Since the higher-level commands operate in the same way as with the RS-232C Interface (version 1.0), this chapter describes operations that are specific to the SCSI Interface (version 2.0 or later version).

16.1 Setup Conditions

For the detail, refer to the "IT-2000 I/O Box Installation Manual".

17. API for Application Development

This chapter describes the File Transfer Application Programming Interface for the Windows 95 version of the Upload/Download utility.

17.1 Environment

Hardware Environment

IBM compatible PC AT

CPU	: 486DX or Pentium
Clock	: 100 MHz (minimum)
Memory Capacity	: 16 Mbytes RAM or more

Development Environment

Windows 95
Microsoft Visual C++ Ver 4.2 (English version)

Execution Environment

Windows 95 (English version)

17.2 Supplied Items

1. LMWINAPI.LIB	FTP API Library
2. LMWINAPI.H	Include File for FTP API.
3. DEVICE.INI	Configuration file. This file contains the communication interface specifications. It has to be present in the execution directory.
4. DRIVER32.DLL	These files (4 to 8) are necessary always for the LMWIN communication.They must be resided in folder of an application program.
5. HFC32.DLL	
6. LMAN32.DLL	
7. LMWINAPI.DLL	
8. SCSIDRV.DLL	

18. API Specifications

18.1 Send API

```
INT16 LMWIN_Send(char *File1...);
INT16 LMWIN_Send(char OptionFlag,char *File1,...);
```

18.2 Receive API

```
INT16 LMWIN_Receive(char *File1,...);
INT16 LMWIN_Receive(char OptionFlag,char *Str,...);
```

18.3 Append API

```
INT16 LMWIN_Append(const char *Src, const char *Dest);
INT16 LMWIN_Append(const char Option, const char *Src, const char *Dest);
```

18.4 Delete API

```
INT16 LMWIN_Delete_Files(char *Str,...);
```

18.5 Move API

```
INT16 LMWIN_Move_File(const char *OldName, const char *NewName);
```

18.6 Other API

```
INT16 LMWIN_Set_DateTime(char *Date, char *Time);
INT16 LMWIN_Set_DateTime(char *DateRTime);
INT16 LMWIN_Display_String(const char *Str);
INT16 LMWIN_Buzzer();
INT16 LMWIN_Get_File_Info(const char *FileName, FILEINFO *File);
INT16 LMWIN_Set_File_Info(const char *File, char AttribFlag, UINT32 Size, char
                                                             *Date, char *Time);
INT16 LMWIN_Set_File_Info(const char *File);
INT16 LMWIN_Set_File_Info(const char *File, char AttribFlag);
INT16 LMWIN_Set_File_Info(const char *File, char AttribFlag, UINT32 Size, char
                                                                  *DateRTime);
INT16 LMWIN_Get_Disk_Info(const char Drive, DISKINFO *Disk);
INT16 LMWIN_End_Indication();
INT16 LMWIN_End_Indication(INT16 Err_Code);
INT16 LMWIN_Exec_ScriptFile(const char *ScriptFile);
```

NOTE: Non-zero return code indicates an error for all APIs.

19. API Function Descriptions

19.1 Send API

The Send API is used to send files from the Host PC to the IT-2000 using the File Transfer Protocol. This API has the following parameters, some of which are optional:

1. Full path of the Source File to be sent
2. Full path of the Destination Directory
3. Options (Overwrite, Update and Recursive)

The followings are the two methods of usage:

`INT16 LMWIN_Send(char *File1...);`

Parameters : File name (s), Destination Directory, NULL
Return Value : Error code from FTP (see Chapter 7.1 "Error Codes” for the details.)

Examples

```
1) LMWIN_Send (”c:\\casio\\*.bat”,”c:\\casio\\”,NULL);
2) LMWIN_Send (“c:\\autoexec.bat”,”c:\\casio\\*.bat”,”c:\\casio\\”,NULL);
3) LMWIN_Send (“c:\\autoexec.bat”,”c:\\casio\\*.bat”,”b:\\casio\\”,NULL);
```

`INT16 LMWIN_Send(char OptionFlag,char *File1,...);`

Parameters : Option Flag, File Name (s), Destination Directory, NULL where Option Flag is a combination of the following :
NOOPTION, OVERWRITE, UPDATE and RECURSIVE
Return Value : Error code from FTP (see Chapter 7.1 "Error Codes” for the details.)

Examples

```
1) LMWIN_Send (RECURSIVE|OVERWRITE,”c:\\casio\\*.bat”,”c:\\casio\\”,NULL);
2) LMWIN_Send (UPDATE,“c:\\autoexec.bat”,”c:\\casio\\*.bat”,”c:\\casio\\”,NULL);
3) LMWIN_Send (RECURSIVE|OVERWRITE,”c:\\casio\\*.bat”,”b:\\casio\\”,NULL);
```

19.2 Receive API

The Receive API is used to receive files from the Host PC to the IT-2000 using the File Transfer Protocol. This API has the following parameters, some of which are optional:

1. Full path of the Source File to be received
2. Full path of the Directory into which the received files are to be stored
3. Options (Overwrite, Update and Recursive)

The followings are the two methods of usage:

`INT16 LMWIN_Receive(char *File1,...);`

Parameters : File Name (s), Destination Directory, NULL
Return Value : Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

Examples

```
1) LMWIN_Receive (”c:\\casio\\*.bat”,”c:\\casio\\”,NULL);
2) LMWIN_Receive (“c:\\autoexec.bat”,”c:\\casio\\*.bat”,”c:\\casio\\”,NULL);
3) LMWIN_Receive (“b:\\autoexec.bat”,”b:\\casio\\*.bat”,”c:\\casio\\”,NULL);
```

INT16 LMWIN_Receive(char OptionFlag,char *Str,...);

Parameters : OptionFlag, FileName(s), Destination Directory, NULL where
OptionFlag is a combination of the following :
NOOPTION, OVERWRITE, UPDATE and RECURSIVE

Return Value : Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

Examples

```
1) LMWIN_Receive (RECURSIVE|OVERWRITE,”c:\\casio\\*.bat”,”c:\\casio\\”,NULL);
2) LMWIN_ReceiveSend (UPDATE,“c:\\autoexec.bat”,”c:\\casio\\*.bat”, ”c:\\casio\\”,NULL);
3) LMWIN_Receive (RECURSIVE|OVERWRITE,”b:\\casio\\*.bat”,”c:\\casio\\”,NULL);
```

19.3 Append API

The Append API is used between the Host PC and the IT-2000 for the following operations:
1. Send a file from the Host PC and append it to a specified file on the IT-2000.
2. Receive a file from the IT-2000 and append it to a specified file on the Host PC.

The followings are the two methods of usage:

INT16 LMWIN_Append(const char *Src, const char *Dest);

Parameters : Source File, Destination File
Default is taken as Append Send.

Return Value : Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

Example

```
LMWIN_Append( “c:\\autoexec.bat”,”c:\\casio\\autoecxec.bat”);
```

Here, “c:\\autoexec.bat” is sent by the Host PC and appended to “c:\\casio\\autoexec.bat” on the IT-2000.

INT16 LMWIN_Append(const char Option, const char *Src, const char *Dest);

Parameters : Option, Source File, Destination File where Option is either ‘R’ or ‘S”.

Return Value : Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

Example

```
1) LMWIN_Append (‘r’ , “c:\\autoexec.bat”,”c:\\casio\\autoecxec.bat”);
```

Here, “c:\\autoexec.bat” is received from the IT-2000 and appended to “c:\\casio\\autoexec.bat” on the Host PC.

```
2) LMWIN_Append (‘s’ , “c:\\autoexec.bat”,”c:\\casio\\autoecxec.bat”);
```

Here, “c:\\autoexec.bat” is sent by the Host PC and appended to “c:\\casio\\autoexec.bat” on the IT-2000.

19.4 Delete API

The delete API is used to delete files on the IT-2000. It accepts the full path name (s) of the files on the IT-2000 as parameters.

```
INT16 LMWIN_Delete_Files(char *Str,...);
```

Parameters : FileName(s), NULL
Return Value : Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

Examples

```
1) LMWIN_Delete_Files (“c:\\autoexec.bat”,NULL);
2) LMWIN_Delete_Files (“c:\\autoexec.bat”,”c:\\casio\\autoecxec.bat”,NULL);
```

```
INT16 LMWIN_Delete(char OptionFlag,char *File1,...);
```

Parameters : OptionFlag, FileName(s), Destination Directory, NULL where
OptionFlag is acombination of the following :
NOOPTION, OVERWRITE, and RECURSIVE
Return Value : Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

Examples

```
1)LMWIN_Delete(RECURSIVE|OVERWRITE,”c:\\casio\\autoexec.bat”,”c:\\casio\\autoexec.
  bat”, NULL);
2)LMWIN_Delete(RECURSIVE|OVERWRITE,”b:\\casio\\autoexec.bat”,”b:\\casio\\autoexec.
  bat” , NULL);
```

19.5 Move API

The Move API is used to move/rename files on the IT-2000. It accepts the full path names of the Source and Destination files on the IT-2000 as parameters.

```
INT16 LMWIN_Move_File(const char *OldName, const char *NewName);
```

Parameters : OldName, NewName
Return Value : Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

Example

```
LMWIN_Move_File(“c:\\autoexec.bat”,”c:\\casio\\autoexec.bat”);
LMWIN_Move_File(“b:\\autoexec.bat”,”b:\\casio\\autoexec.bat”);
```

19.6 Other API

This section describes the other API supported :

```
INT16 LMWIN_Set_DateTime(char *Date, char *Time);
```

This usage of the Set Date and Time API is used to set both date and time. When used, it sets the date and time on the IT-2000 with the specified values of date and time.
This API does not check the validity of the date and time specified.

Parameters : Date Value, Time Value
Return Value : Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

Example

```
LMWIN_Set_DateTime( “1998-12-12”,”12:12:12”);
```

```
INT16 LMWIN_Set_DateTime(char *DateRTime);
```

This usage of the Set Date and Time API is used to set either date or time. When used, it sets the date or time on the IT-2000 with the specified values of date or time.
This API does not check the validity of the date or time specified

Parameters : Date Value or Time Value
Return Value : Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

Example

```
1) LMWIN_Set_DateTime( “1998-12-12”);
2) LMWIN_Set_DateTime(”12:12:12”);
```

```
INT16 LMWIN_Display_String(const char *Str);
```

This API is used to send a string of characters to the IT-2000 using the FTP API.

Parameters : String to be displayed
Return Value : Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

Example

```
LMWIN_Display_String(“String to be displayed”);
```

```
INT16 LMWIN_Buzzer();
```

This API is used to activate the buzzer on the IT-2000. It sends merely a command to the IT-2000, and the return code only indicates if the command was correctly received by the IT-2000.
It does not guarantee buzzer activation.

Parameters : None
Return Value : Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

Example

```
LMWIN_Buzzer();
```

INT16 LMWIN_Get_File_Info(const char *FileName, FILEINFO *File);

This API fetches the file information (attributes, size, date and time of creation) of the specified file on the IT-2000. Only one file can be specified at a time..

Parameters : FileName, Pointer to FILEINFO struct.
The FILEINFO structure is updated with the file information.

FILEINFO structures member is the following form.

Data Type	Member	Contents
UINT32	size	file size
UINT32	date	date
UCHAR	time[3]	time
UCHAR	attrib	file attribute
char	path[1024]	file path

Return Value : Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

Example

```
1) LMWIN_Get_File_Info(“c:\\autoexec.bat”,&FileInfo);
2) LMWIN_Get_File_Info(“b:\\autoexec.bat”,&FileInfo);
```

INT16 LMWIN_Set_File_Info(const char *File, char AttribFlag, UINT32 Size, char *Date, char *Time);

This API sends the file Information of the specified file to the IT-2000. The different methods of usage are as described :

Parameters : FileName,AtribFlag,FileSize,DateValue,Timevalue where AttribFlag is a combinationof READ_ONLY, HIDDEN SYSTEM, ARCHIVE.NORMAL.
Return Value : Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

Example

```
LMWIN_Set_File_Info(“c:\\autoexec.bat”, HIDDEN | READ_ONLY,100,“1998-12-12”,”12:12:12”);
```

INT16 LMWIN_Set_File_Info(const char *File);

Parameters : FileName
Return Value : Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

Example

```
LMWIN_Set_File_Info(“c:\\autoexec.bat”);
```

INT16 LMWIN_Set_File_Info(const char *File, char AttribFlag);

Parameters : FileName ,AttribFlag where AttribFlag is a combination of READ_ONLY, HIDDEN SYSTEM, ARCHIVE.NORMAL
Return Value : Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

Example

```
LMWIN_Set_File_Info(“c:\\autoexec.bat”, HIDDEN | READ_ONLY);
```

INT16 LMWIN_Set_File_Info(const char *File, char AttribFlag, UINT32 Size, char *DateRTime);

Parameters : FileName,AtribFlag,FileSize,DateValue or Timevalue where AttribFlag is a combination of READ_ONLY, HIDDEN SYSTEM, ARCHIVE.NORMAL.
Return Value : Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

Example

```
1) LMWIN_Set_File_Info(“c:\\autoexec.bat”, HIDDEN | READ_ONLY,100,“1998-12-12”);
2) LMWIN_Set_File_Info(“c:\\autoexec.bat”, HIDDEN | READ_ONLY,100,”12:12:12”);
```

INT16 LMWIN_Get_Disk_Info(const char Drive, DISKINFO *Disk);

This API fetches the disk information of the specified drive on the IT-2000.

Parameters : DriveLetter ,Pointer to DISKINFO struct.
The DISKINFO Structure is updated with the disk information
Return Value : Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

DISKINFO structures member is the following form.

Data Type	Member	contents
UINT32	Capacity	disk Capacity
UINT32	FreeSpace	disk FreeSpace
UCHAR	Condition	disk Condition

Example

```
LMWIN_Get_Disk_Info(‘c’, &DiskInfo);
```

INT16 LMWIN_End_Indication();

This API sends the End Indication command to the IT-2000.

Parameters : None
Return Value : Error code from HFC Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

Example

```
LMWIN_End_Indication();
```

INT16 LMWIN_End_Indication(INT16 Err_Code);

This API sends the End Indication command to the IT-2000 with a specific ending code.
Details of ending codes are specified in the FTP specification document.

Parameters : Ending code
Return Value : Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

Example

```
LMWIN_End_Indication( 20);
```

```
INT16 LMWIN_Exec_ScriptFile(const char *ScriptFile);
```

This API will execute a script file containing the File Transfer utility commands.
A script file of the FTP utility will have the extension.SCR.

Parameters	: Script file name
Return Value	: Error code from FTP (see Chapter 7.1 "Error Codes“ for the details.)

Example

```
LMWIN_Exec_ScriptFile(“test.scr”);
```

20. File Check Utility

20.1 Development Concept

Because of careless mistakes by a program installer during the installation of system and application programs, often the installation error is brought to light because system errors occurred during operation by the user.

Below are some of the actual problems that occurred.

(1) On an installation that extended over multiple cards, the postal number dictionary (POST.DIC) was not installed. Subsequently, when a business application tried to convert a postal number, an error occurred because the dictionary was not present.

(2) On an installation for updating the driver a compatibility fault often occurred. It occurred because the driver had not been updated and this fact was not checked at all.

(3) When the application program was transferred for installation, one of the transfer packets was omitted, resulting in an error at subsequent execution of the application program.

To prevent the above problems this File Check utility is created.

20.2 Purpose

The File Check utility is used, by both SE and user's system section, to ensure that installation is properly completed. This will prevent the above described problems from occurring. Moreover, problem such as (3) above, can now be avoided because the specification of the file transfer protocol has been improved.

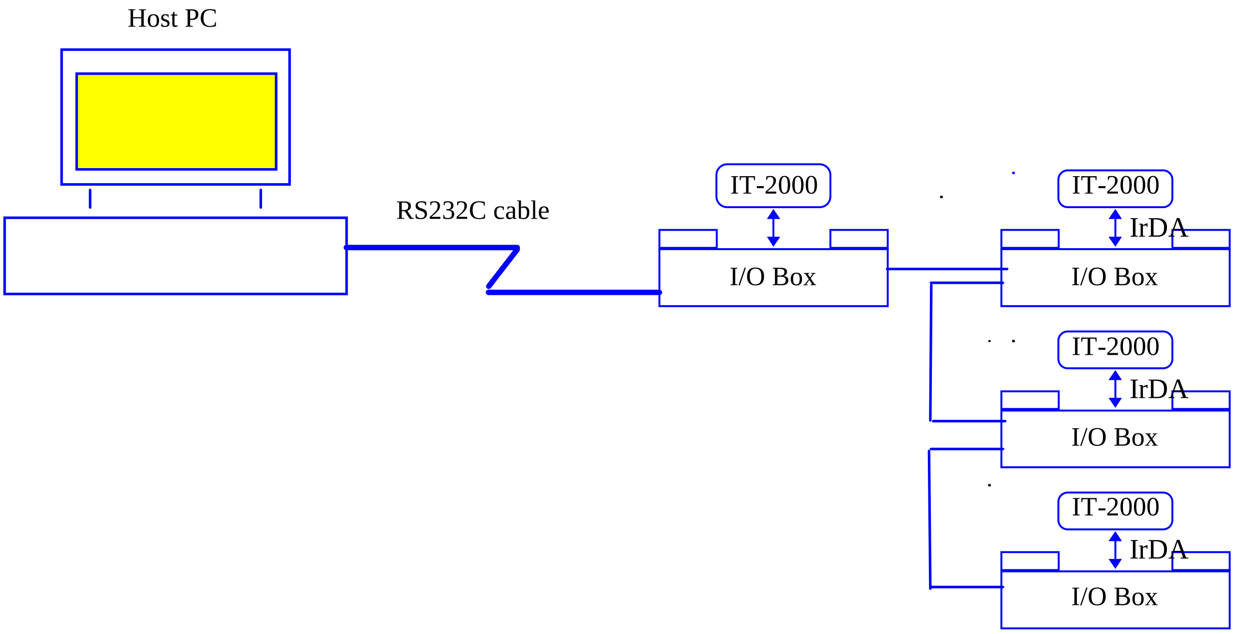
20.3 Scope

This File Check utility is capable of finding any installation error irrespective of the file transfer method used for installation.
The transfer method is the way the file is transferred between the host PC and IT-2000 (HT), and between IT-2000s.
The term "Host PC" includes a personal computer (PC) and any IT-2000 to emulate as PC.

20.4 Terminology

The following abbreviations are used in these chapters (Chapter 17 to 26) to describe specific model name or software program.

PC (Host PC)	Personal computer (IBM PC/AT 100% compatible)
HT	IT-2000W
FCHK.EXE	File Check utility
LMWIN	Upload/Download utility
I/O Box	Master I/O Box (IT-2065IO-E)
I/O Box	Satellite I/O Box (IT-2060IO-E)



21. Operation Environment

The File Check utility must be used under the following operating environment.

(1) File transfer between Host PC and IT-2000.

Note :
The system configuration described on the previous page is only one example of PC-I/O Box connections and does not limit all the possibilities that the system has.

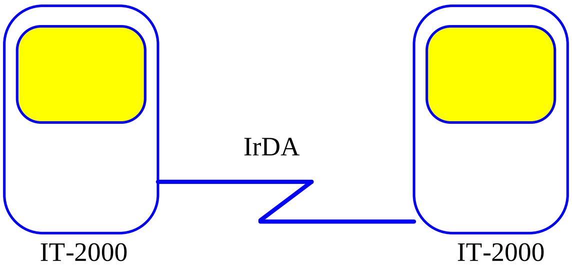
- Host PC

Operable model	IBM PC/AT compatible model
Operating System	MS-DOS (Ver.5.0 or later)
Program name	FCHK.EXE
Memory capacity	640 Kbytes or greater

- HT

Model	IT-2000W
Operating System	MS-DOS ver. 6.22
Program name	FCHK.EXE
Memory capacity	640 Kbytes or greater

(2) File transfer between IT-2000s



- HT

Operating model	IT-2000W
Operating system	MS-DOS ver. 6.22
Program name	FCHK.EXE

22. Functions

22.1 Operation Procedure

Use the File Check utility according to the following procedure.
Here, the procedure is described assuming that an application is installed from the PC to IT-2000.

(1) At the transmission side of PC (master side), set the file to be downloaded (copied).

(2) Specify the downloaded (copied) file using the script file, then direct the script file so that it transmits a list file (FCHK.LOG).

(3) Create the list file using the File Check utility on the PC, as follows :

```
FCHK  /G /SC <Script file name> <Destination directory name>
```

The FCHK.LOG file is created (FCHK.LOG file = list file) .

(4) At file transfer via LMWIN, also transfer the FCHK.LOG file to the partner station.
(The FCHK.LOG file will be copied at the same time the target file is copied.)

(5) Start up the File Check utility at the reception-station side (duplication side), and check that the transferred (copied) file and list file (FCHK.LOG) have been received (copied) properly.
If the check result is OK, this operation will terminate normally at error level 0, otherwise, it will terminate abnormally at an error level other than 0.

```
FCHK  /C A:\FCHK\FCHK.LOG
If error level 1 goto err
```

22.2 Detailed Functions

The File Check utility has the following functions :
(1) List file creating function
(2) List file comparing function

(1) List file creating function

Command option

```
FCHK /G [/Option] <file name list or Script file name> <Destination directory name>
          [FCHK.LOG File output Directory name]
```

Function

Specifies the file name to be transferred (copied) from the Host PC or IT-2000, and creates a list file that contains the list of all file names to be transferred (copied) and the check-sum data calculated for all the files. In addition, it creates the check-sum data of the list file.
The list and check-sum data creation process will be displayed on the screen.

The list file name created by this function is set to FCHK.LOG, and it will be located under the current directory, if the output destination is not specified.
In file transfer (file copy) the user should transfer (copy) the list file created by this function to the partner station (duplication side).

The following information should be included in the list file.

- File size
- Date and time of creation
- Transfer (copy) destination path (file) name
- Number of transferred (copied) files
- Check-sum data of all the transferred (copied) files
- Check-sum data of the list file

The check-sum for all the transferred (copied) files will be calculated in such a way that the data sections of all the transferred (copied) files are exclusive ORed for each double-word length.
For the check-sum of the list file, its contents will be successively added (ANDed) for each double-word length , and a value, which can offset this result to zero if it is added, will be employed as the check-sum data.

The check-sum data will be outputted as follows **:**

FILE_CHECKSUM=HHHHLLLL (HHHH: HiGHT-WORD / LLLL: LOW-WORD)
LIST_CHECKSUM=HHHHLLLL (HHHH: HiGHT-WORD / LLLL: LOW-WORD)

If an error occurs during the execution of this function, the existing list file (LOG file so far made at this point in time) will not be deleted provided that the error occurs during the creation of list file.
However, the existing list file will not be deleted because of errors that occur during the command parameter analysis process.

Options

/G	**:**	Creates a list file (FCHK.LOG). **(This is must)**
/SC	**:**	Specifies the script file name.

A file name that is specified by the parameter is the script file. FCHK.EXE will analyze the file name to be transferred against this script file, then a list file will be created.

/R : Specification of recursive call

All files under the directory of the file path name specified by the parameter will be assumed to be the objectives of the list file creation.
If the specified directory has sub-directories, the list file will be created from the file names added with the sub-directory names.
Even if this option is specified, specify the file names using their full path names in the file name list.
If this option is not specified, only the files that are specified in the file name list can be the objectives of the file list creation.

/AO : Additional output

If the FCHK.LOG file exists in the directory specified by [FCHK.LOG file output directory name], this
file will be appended with additional logs.
If it does not exist, a new FCHK.LOG file will be created.
However, if the specified directory does not exist, the program will be abnormally terminated.
(Error code **:** 0x13 Output path name for the specified list file is not found.)

Parameter

<File name list or Script file name >

To create the list file, describe the file name list of files to be transferred (copied).
Specify the file name at the transfer-side (copy-side) using its full path name. If multiple file names are specified at the transfer-side (copy-side), separate the path names with a double-colon ("::").
File names can be specified using a wild card.
If the /SC option is specified, specify the path name under which the script file exists. (For information about the specifications for the script file syntax, refer to Chapter 24 "Syntax Analysis of Script Files", and Chapter 13 "Syntax of Script File Commands".)

<Destination directory name>

Specify the directory name of the transmission destination (copy destination).
The directory name must include the drive name.
The delimiter character of the directory name should be " \ ".

Example:	b:\	(Specification of root directory)
	b:\PA\12KAI\	(Specification of sub-directory)
	b:\PA	(Incorrect specification)

NOTE: If the "/SC" option has been specified, this parameter will become invalid.

[FCHK.LOG File output directory name]

Specifies the output destination directory name of the FCHK.LOG file.
The directory name must include the drive name.
The delimiter character of the directory name should be " \ ".
If this parameter is not specified, the FCHK.LOG file will be created under the current directory.

Example:	b:\	(Specification of root directory)
	b:\PA\12KAI\	(Specification of sub-directory)
	b:\PA	(Incorrect specification)

<Restrictions>

1. The number of objective files of collecting logs is limited to 65,000.
   If log collection exceeds over 65,000 files, the operation terminates abnormally.

2. The file size of a script file should be a maximum of 32,000 bytes.
   If a script file is specified to contain more than 32,000 bytes, the operation terminates abnormally.

(2) List file comparing function

Command option

```
FCHK  /C <FCHK.LOG file path name>
```

Function

Performs the following verifications by comparing the file information of the files transferred (copied) from the partner station (master side) with the contents of the list file (FCHK.LOG), by calculating the check-sum data of the list file and comparing the result with that on the transferred list file, and by calculating the check-sum data of all the files transferred (copied) and comparing the result with the check-sum data in the list file.
The comparison process and results are displayed on the screen.
If the comparison results are correct, this operation will terminate normally at error level 0, which is used as the program termination code, otherwise, it will terminate abnormally at an error level other than 0.

The following file information should be compared.

- File size
- Date and time of creation
- Transfer (copy) destination path (file) name
- Number of transferred (copied) files
- Check-sum data of all the transferred (copied) files
- Check-sum data of the list file

The check-sum for all the transferred (copied) files will be calculated in such a way that the data sections of all the transferred (copied) files are exclusive ORed for each double-word length.
For the check-sum data of the list file, the contents will be successively added (ANDed) for each double-word length, and a value, which can offset this result to zero if it is added, will be employed as the check-sum data.

Option

Compares with the /C : list file (FCHK.LOG).

Parameter

<FCHK.LOG file path name>

Specifies the path name under which the list file (FCHK.LOG) exists.

<Restrictions>

There can be a maximum of 65,000 files compared.
If the value of " FILE_NO = " is greater than 65,000, operation terminates abnormally.

22.3 Format of List File

The list file created by the File Check utility should have the following format.

```
<FCHKLOG> :: = <FILENO> <FILEINFO> <FILECHECKSUM> <LISTCHECKSUM> null
<FILENO> :: = FILE_NO= <dec_num> <LS>
<FILEINFO> :: = <INFO> <LS>
<INFO> :: = <PATH> SP <SIZE> SP <DATE>
<LS> :: = CR
<FILECHECKSUM> :: = FILE_CHECKSUM= <hex_char> <LS>
<LISTCHECKSUM> :: = LIST_CHECKSUM= <hex_char> <LS>
<dec_num> :: = decimal number
<hex_char> :: = hexadecimal number represented in characters.
```

Example :

```
 FILE_NO=3
 A:\AP\MENU.EXE 12345 19960728-0630
 A:\CONFIG.SYS 1000 19960308-2058
 A:\AUTOEXEC.BAT 512 19960206-2340
 FILE_CHECKSUM=XXXXXXXX
 LIST_CHECKSUM=XXXXXXXX
```

22.4 Common Specifications to Each Function

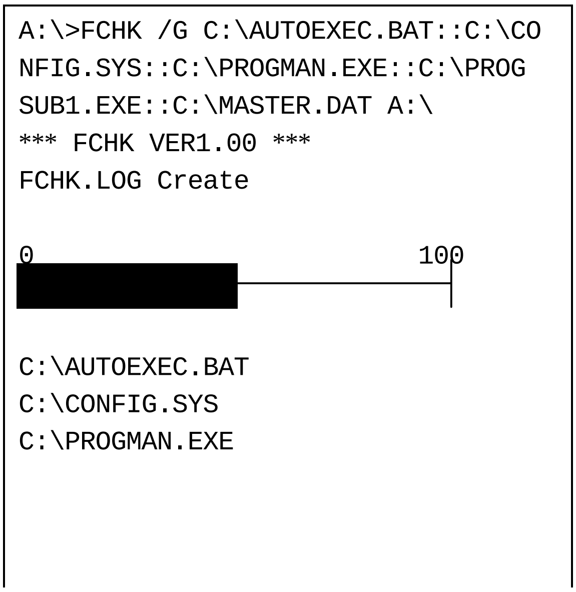
If "Ctrl+C" keys are pressed during the creation of list file or during the comparison of list file, the processing continues without interruption.

Note :
In some cases where the "Ctrl+C" (or STOP) keys are pressed, a " ^C " may appear in the display of the progress percentage data, and the data display may be duplicated.

23. Screen Images of Operations

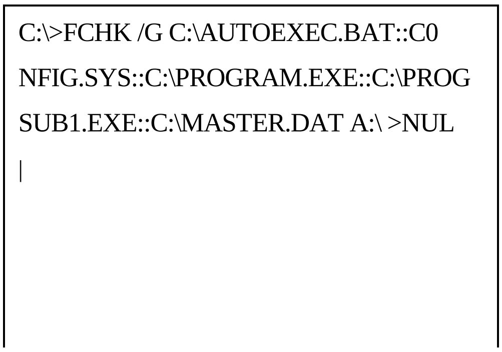
23.1 List File Creation

PC

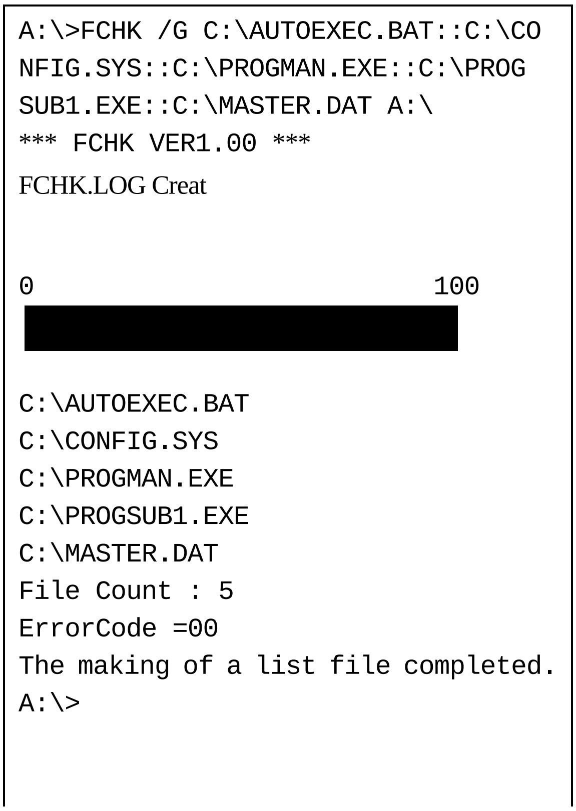


```
A:\>FCHK /G C:\AUTOEXEC.BAT::C:\CO
NFIG.SYS::C:\PROGMAN.EXE::C:\PROG
SUB1.EXE::C:\MASTER.DAT A:\
*** FCHK VER1.00 ***
FCHK.LOG Create

0                         100

C:\AUTOEXEC.BAT
C:\CONFIG.SYS
C:\PROGMAN.EXE
```

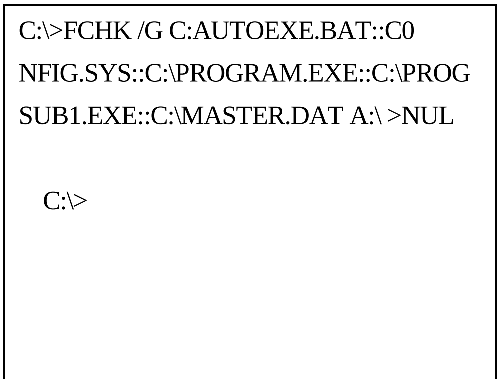
IT-2000



```
C:\>FCHK /G C:\AUTOEXEC.BAT::C0
NFIG.SYS::C:\PROGRAM.EXE::C:\PROG
SUB1.EXE::C:\MASTER.DAT A:\ >NUL
|
```



```
A:\>FCHK /G C:\AUTOEXEC.BAT::C:\CO
NFIG.SYS::C:\PROGMAN.EXE::C:\PROG
SUB1.EXE::C:\MASTER.DAT A:\
*** FCHK VER1.00 ***
FCHK.LOG Creat


0                           100

C:\AUTOEXEC.BAT
C:\CONFIG.SYS
C:\PROGMAN.EXE
C:\PROGSUB1.EXE
C:\MASTER.DAT
File Count : 5
ErrorCode =00
The making of a list file completed.
A:\>
```

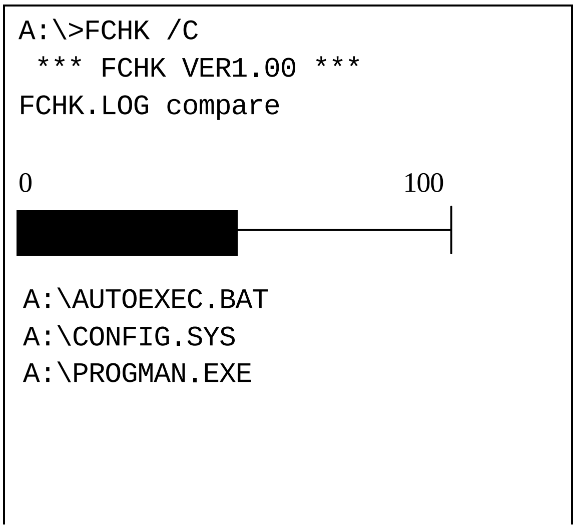


```
C:\>FCHK /G C:AUTOEXE.BAT::C0
NFIG.SYS::C:\PROGRAM.EXE::C:\PROG
SUB1.EXE::C:\MASTER.DAT A:\ >NUL

  C:\>
```

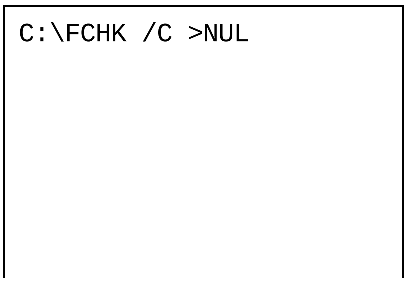
23.2 List File Check

PC

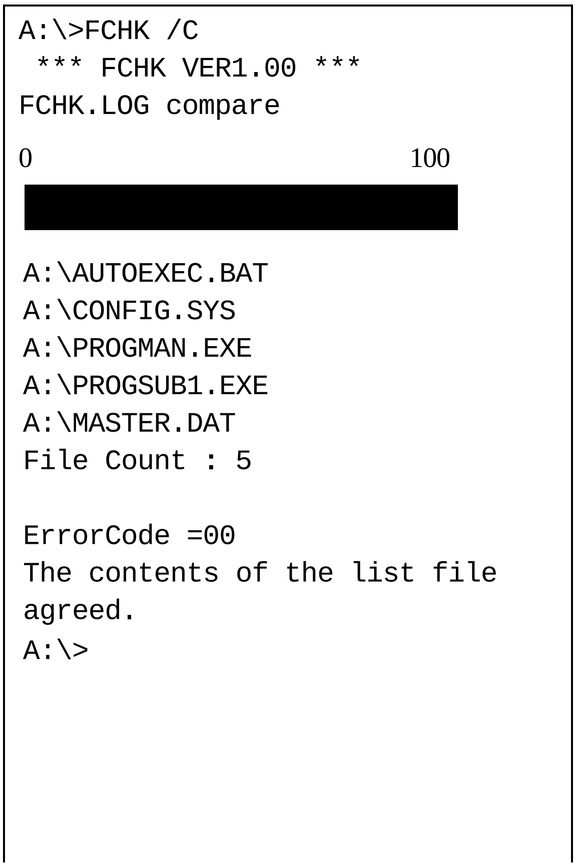
IT-2000



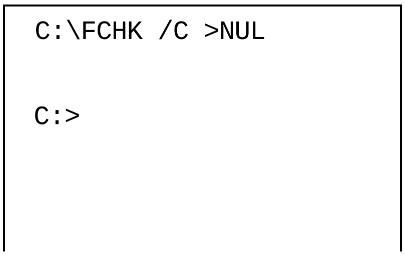
```
A:\>FCHK /C
 *** FCHK VER1.00 ***
FCHK.LOG compare

0                          100

 A:\AUTOEXEC.BAT
 A:\CONFIG.SYS
 A:\PROGMAN.EXE
```



```
C:\FCHK /C >NUL
```

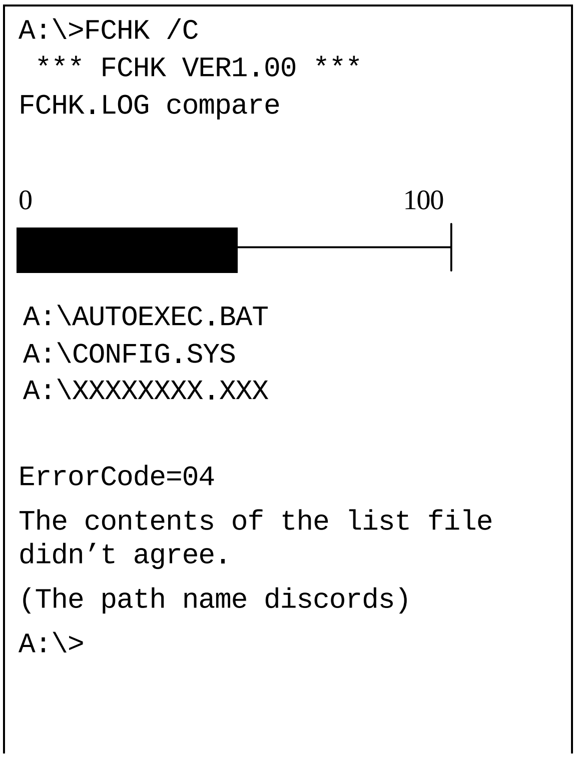


```
A:\>FCHK /C
 *** FCHK VER1.00 ***
FCHK.LOG compare
0                           100

 A:\AUTOEXEC.BAT
 A:\CONFIG.SYS
 A:\PROGMAN.EXE
 A:\PROGSUB1.EXE
 A:\MASTER.DAT
 File Count : 5

 ErrorCode =00
 The contents of the list file
 agreed.
 A:\>
```



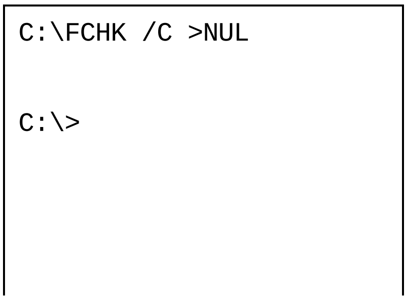
```
 C:\FCHK /C >NUL


 C:>
```

PC



```
A:\>FCHK /C
 *** FCHK VER1.00 ***
FCHK.LOG compare


0                              100

A:\AUTOEXEC.BAT
A:\CONFIG.SYS
A:\XXXXXXXX.XXX


ErrorCode=04
The contents of the list file
didn't agree.
(The path name discords)
A:\>
```

IT-2000



```
C:\FCHK /C >NUL


C:\>
```

24. Syntax Analysis of Script Files

If a script file name is specified when the list file creation function is executed, it will analyze the syntax of the script file, then create the list file.

The specifications of the script file syntax are given below.

```
<SCRIPT FILE> :: = <COMMANDS>
<COMMANDS> :: = <COMMANDS>  <COMMAND>  |  null
<COMMAND> :: =   □/□       <CMDBODY>  <LS>
<CMDBODY> :: =  <APPEND>
                 |  <CHILD_PROC>
                 |  <DELETE>
                 |  <FORMAT>
                 |  <BEEP>
                 |  <RENAME>
                 |  <RECEIVE>
                 |  <SEND>
                 |  <PRINT>
                 |  <TIME_ADJUST>
                 |  <END_SESSION>

<APPEND> :: = <APPEND_CMD> <APPEND_OPTION> <SP> <PATHNAME_PAIR>
<CHILD_PROC> :: = <CHILD_PROC_CMD> <SP> <CMD_PARAMETER>
<FORMAT> :: = <FORMAT_CMD> <SP> <DRIVE>
<BEEP> :: = <BEEP_CMD>
<RENAME> :: = <RENAME_CMD> <SP> <PATHNAME_PAIR>
<RECEIVE> :: = <RECEIVE_CMD> <OPTIONS> <SP> <PATHNAME_LIST>
<SEND> :: = <SEND_CMD> <OPTION> <SP> <PATHNAME_LIST>
<PRINT> :: = <PRINT_CMD> <SP> <STRING>
<TIME_ADJUST> :: = <TIME_CMD> <SP> <TIME_VALUE>
<END_SESSION> :: = <END_CMD> <PARAM>

<APPEND_CMD> :: = 'A'
<CHILD_PROC_CMD> :: = 'C'
<FORMAT_CMD> :: = 'F'
<BEEP_CMD> :: = 'B'
<RENAME_CMD> :: = 'N'
<RECEIVE_CMD> :: = 'R'
<SEND_CMD> :: = 'S'
<PRINT_CMD> :: = 'P'
<TIME_CMD> :: = 'T'
<END_CMD> :: = '/'

<PATHNAME_PAIR> :: = <PATHNAME> <DELM> <PATHNAME>
<CMD_PARAMETER> :: = <CMD_NAME> <STRING>
<CMD_NAME> :: = <PATHNAME>
<PATHNAME_LIST> :: = <PATHNAME> <DELM> <PATHNAME_LIST> | <PATHNAME>
<DRIVE> :: = <DRIVE_LETTER> ':'

<TIME_VALUE> :: = <DATE> <TIME>
```

```
<OPTIONS> :: = <OPTIONS> <OPTION> | null

<OPTION> :: = <RECURSIVE_OPTION> | <UPDATE_OPTION>
<RECURSIVE_OPTION> :: = 'R'
<UPDATE_OPTION> :: = 'U'
<APPEND_OPTION>::='S' | 'R'
<STRING> :: = ""<CHARS>""
<DELM> :: = <SP>
<LS> :: = CR | <SP>
<SP> :: = <SP> SP | SP
<PARAM> :: = <SP> <NUMBER>
```

This File Check utility searches <CMD_BODY> in the script file, and if it finds <CMD_BODY> followed by <SEND> (='S': See Note below, it judges <PATHNAME_LIST> that follows <SEND> as the destination path name and creates the list file (FCHK.LOG) .
Other <CMD_BODY> without being followed by <SEND> will be ignored.

Note : The following commands/options can be the objectives of creating the list file.

(1) "/S"
(2) "/SO"
(3) "/SR"
(4) "/SOR"
(5) "/SRO"

25. Installation Method of File Check Utility

25.1 For PC

The File Check utility for the PC will be installed as a DOS command.

25.2 For IT-2000

The File Check utility for IT-2000 will be installed as part of the IT-2000 Communication utility.
Since the installation to IT-2000 will be executed as a function, a file check (FCHK) function must exist.
This file check function must be supported by the user, however, it can be accessed either from the user application or the System Menu.
The following pages describe the interface specifications of the file check function.

Example of Batch File for Check on List File :

```
FCHK /C >NUL
IF ERRORLEVEL 1 GOTO ERROR
ECHO NORMAL END
GOTO END
: ERROR
ECHO ERROR END
: END
```

26. Error Messages/Codes in File Check Utility

The following messages can be given.

Code	Message	Meaning
00	The making of a list file completed. The contents of the list file agreed.	Normal termination
01	Specified path name not found.	File name specified by the list file does not exist.
02	The list file making error.	Physical error occurs during list file creation.
03	FCHK.LOG not found.	List file (FCHK.LOG) could not be found by the list file check.
04	The contents of the list file didn't agree.(The path name discords)	Verification result of list file checks is not matched. (No path name matched)
05	The contents of the list file didn't agree.(The size discords)	Verification result of list file checks is not matched (No size matched)
06	The contents of the list file didn't agree.(The date/the time discord)	Verification result of list file checks is not matched (No date/time matched)
07	The contents of the list file didn't agree. (All the file check-sum data discord).	Verification result of list file checks is not matched. (No all file check-sum matched)
08	The contents of the list file didn't agree. (The list file check-sum data discord)	Verification result of list file checks is not matched (No check-sum data of the list files matched)
09	A script file not found	Script file with the specified file name was not found.
0A	The script file syntax error	Specified script file includes a syntax error.
0B	The list file read error	Physical error occurs during list file check when the list file (FCHK.LOG) was being read.
0C	It is an unjust option.	Start-up option is illegal.
0D	The parameter error.	Specified parameter has an error.
10	Script file read error	Error occurs in the process of reading a script file.
11	It crossed limitation on the script file size.	The size of specified script file is 32,000 bytes or greater.
12	It crossed limitation on the number of the log files.	There are 65,000 files or more that can be used as the log files.
13	The output path name of the specified list file wasn't found out.	Output destination path name of the specified FCHK.LOG file was not found.
FF	The operation model is different. The version of DOS is different.	Executed under an earlier version of the DOS than ver 5.0.