

**FRONTDOOR**  
**Developer's Kit**  
**R e f e r e n c e**

**\* PRE-RELEASE \***

# FRONTDOOR

Developer's Kit

Reference

Copyright © 1997 Definite Solutions HB; All rights reserved.

---

All Definite Solutions products and trademarks are trademarks or registered trademarks of Definite Solutions HB; with the exception of FrontDoor which is a registered trademark of Joaquim Homrighausen. Other brands and product names are trademarks or registered trademarks of their respective holders.

## LEGAL NOTICE

The FrontDoor Developer's Kit and information attached hereto, hereafter referred to as "FDDEV", is provided "as is", without warranty of any kind or fitness for a particular purpose, either expressed or implied, all of which are hereby explicitly disclaimed. Definite Solutions HB only guarantees that FDDEV will occupy disk space.

The entire risk as to the quality and performance of FDDEV is with you. Should FDDEV prove defective or incorrect, you assume the entire cost of all necessary servicing, repair, and/or correction. In no event shall Definite Solutions HB, Joaquim Homrighausen, or Mats Wallin be liable to the you or anyone else for any damages or costs, including, but not limited to, any lost profits, lost savings, lost income, lost information, loss of the right to use FDDEV, or other incidental or consequential damages arising out of the use or inability to use FDDEV.

Additional statements by agents, employees, distributors, and dealers of Definite Solutions HB do not constitute warranties by Definite Solutions HB and do not bind Definite Solutions.

The information contained in this document was assembled from source code and various data definitions from the FrontDoor E-Mail system. Definite Solutions HB does not guarantee the accuracy of this information and it is subject to change without further notice.

Definite Solutions HB will attempt to keep this information as up-to-date as possible. If you decide to write an application with explicit support for FrontDoor, feel free to use this information as long as you understand that you're on your own. We do not guarantee any form of explicit support for this package. It is intended to be used by programmers and developers.

FDDEV may be published and distributed to other people as long as no part of it is modified by any means, this includes translation to any other language (technical or social), and as long as no charges are applied (including but not limited to trading). FDDEV information may not be used to reverse engineer FrontDoor, or any related utility, nor may it be used to commercially compete with Definite Solutions HB.

<p>This publication is protected by international copyright laws and treaty provisions. It may only be distributed and used in accordance with those laws and treaty provisions.</p>
--

**DEF-E-FDDEV230-D-970100-EP**

Produced in Sweden.

# TABLE OF CONTENTS

<b>1. INTRODUCTION .....</b>	<b>5</b>
Version notation.....	5
Program notation.....	5
Application recommendations.....	5
Files.....	6
Unix-style dates/timestamps .....	6
CRC values .....	6
Look and feel .....	6
SETUP.FD.....	7
<b>2. ENVIRONMENT VARIABLES .....</b>	<b>8</b>
<b>3. FILES.....</b>	<b>10</b>
Reserved space.....	16
<b>4. MACROS .....</b>	<b>17</b>
The basics.....	17
Internal macros.....	17
<b>5. THE LOG FILE.....</b>	<b>20</b>
Separator line .....	21
<b>6. SEMAPHORE FILES .....</b>	<b>22</b>
FDRESCAN.NOW .....	26
Inter-process Communications Area (ICA).....	27
<b>7. THE STATIC QUEUE (STQ) .....</b>	<b>29</b>
Advantages.....	29

Limitations .....	29
Basic rules .....	29
The file format.....	30
Forcing rescans with care.....	35
Rescanning the STQ.....	35
Accessing the file .....	35
Other semaphores.....	36
Error recovery .....	36
Alternative method of deleting entries .....	37
Notes to Conference Mail application authors .....	37
<b>8. EXTERNAL INTERFACE.....</b>	<b>38</b>
<b>9. EXTERNAL REQUEST PROCESSORS.....</b>	<b>39</b>
The request file (=R and =Q) format.....	42
<b>10. SERVICE REQUESTS .....</b>	<b>44</b>
<b>11. MESSAGES.....</b>	<b>48</b>
Fax cover messages.....	48
Locked messages.....	48
The LOCAL message status.....	48
.MSG file format notes.....	48
<b>12. MULTI-LINE AWARENESS.....</b>	<b>49</b>
<b>13. CONTACT INFORMATION.....</b>	<b>50</b>
<b>14. INDEX .....</b>	<b>51</b>

# 1. INTRODUCTION

!!!

November 21, 1997

PRE-RELEASE \* FrontDoor Developer's Kit 2.30 \* PRE-RELEASE

No information in this document is guaranteed to be final. Information in this document is not an indication of supported or new features in the upcoming release of FrontDoor. If you find errors or you think something is missing from one or more files, please do not hesitate to contact us.

!!!

REVISION: 2 (PRE-RELEASE) /// Date: **1997-11-21**

This document covers FrontDoor versions 2.30.mL (MultiLine) and 2.25.sL (SingleLine).

The FrontDoor environment can be controlled by means of external files ("semaphores"), command-line parameters, environment variables, and data written to specific memory locations, in addition to the information used from FrontDoor's various configuration files.

The basic concept is that "external parameters" (those mentioned above, with the exception of configuration files) overrides the settings specified in the configuration files.

## Version notation

The following notations are used to indicate where features are supported differently by various versions of the software:

[ML]	Multi-line version
[SL]	Single-line version (Shareware)

## Program notation

FD	FrontDoor Mailer
FM	FrontDoor Editor
FDNC	FrontDoor Nodelist Compiler/Manager
FrontDoor	The FrontDoor package
FDSETUP	The FrontDoor package main configuration utility

## Application recommendations

It is recommended that applications written with FrontDoor in mind, support at least the following conventions:

- Country-dependant date and time formats. That is, call the OS to fetch information about how dates and time should be formatted and separated.

- When file I/O is performed, applications should open and create the files in the minimum required file mode. That is, if you only need to read from a file, open it as "ReadOnly" and not "ReadWrite".
- When file I/O is performed, applications shouldn't "hog" the file if it has been opened in a sharing mode that may deny other applications access to it. That is, fetch or store the information you need and then close the file.

## Files

File	PASCAL.RAR
Contents description	A collection of include (.INC) files for use with Borland Pascal 7.0 or above.
File	C.RAR
Contents description	A collection of include (.H) files for use with most ANSI C and C++ compilers.
File	FDDEV230.PDF
Contents description	This file.

## Unix-style dates/timestamps

A UNIX-style timestamp, as used by FrontDoor, is a 32-bit integer containing the number of seconds since midnight (local time), January 1, 1970.

Some compiler/library manufacturers have a different idea of what a UNIX-style timestamp is (1968, 1969, etc) – it is a **requirement** that your compiler/library uses the same format as FrontDoor if you intend for your application to interface with FrontDoor. Note that many C/C++ compiler run-time libraries do not return the actual local time when the time() function is used

## CRC values

CRC (Cyclic Redundancy Check) values are used frequently in the FrontDoor environment to ensure data integrity, etc. The CRC-32 values present in many configuration files is calculated identically to the one used in the Zmodem file-transfer protocol (with -1L being the initial feed). How you generate a CRC value is beyond the scope of this documentation.

## Look and feel

A number of applications have a similar interface to that of FDSETUP, the FrontDoor main configuration Utility. Requests for permission to use the same user interface style are received quite frequently. The stock reply has been, and still is, "Yes, provided the following is included in the program and documentation's credit section: **'Original visual interface concept designed by Joaquim Homrighausen. Used with permission.'**".

## SETUP.FD

SETUP.FD is the main configuration file in the FrontDoor environment. Applications that need to access FrontDoor's files should first locate the correct copy of SETUP.FD, read it, verify it, and then use the SYSTEM path defined in the \_SHARED structure to locate other FrontDoor files. You should not assume that the directory where SETUP.FD is located is the same as the SYSTEM path in SETUP.FD.

If the FD environment variable ("FD") is set, FrontDoor will use the path pointed to by it to open and retrieve the information in the SETUP.FD file. If the FD environment variable is not set, FrontDoor will assume that SETUP.FD is in the current directory. If SETUP.FD cannot be found, FD, FM, and FDNC will terminate immediately. If FDSETUP cannot locate SETUP.FD (using the above logic), it asks the user if a new SETUP.FD should be created (either in the directory pointed to by the FD environment variable, or in the current directory).

## 2. ENVIRONMENT VARIABLES

FrontDoor uses several environment variables, most notably, the FD environment variable. Most of the following environment variables are not required for running FrontDoor.

Name	FD
Versions	All
Description	Specifies where the main FrontDoor configuration file, SETUP.FD, is located. The current directory is the default directory if this environment variable has not been defined.

Name	FDOVR
Versions	[ML]
Description	Overrides where FD looks for its overlay file (FD.OVR). FD will by default look for this file in the directory where FD.EXE is located.

Name	FMOVR
Versions	[ML]
Description	Overrides where FM looks for its overlay file (FM.OVR). FM will by default look for this file in the directory where FM.EXE is located.

Name	POINTNET
Versions	None
Description	To force the FD and FM to automatically remap "fake" point addresses that are constructed using the Zone:Net/Node to true point addresses using the Zone:Net/Node.Point construct. <b>This is no longer supported.</b>

Name	TASK
Versions	[ML]
Description	Allows FrontDoor to properly support multi-line operations including avoiding filename collisions, separate configurations, etc. The minimum value for TASK is zero (0) and the maximum is 255. <b>If this environment variable has not been defined, FD will terminate.</b>

Name	TZUTC
Versions	All
Description	<p>To establish a specific system's geographical location relative to UTC (GMT) which is 0000. The format of the TZUTC data is as follows:</p> <p style="text-align: center;">[ +   - ]hhmm</p> <p>where '+' (plus) is optional and assumed if '-' (minus) is not present. For systems located east of UTC0000, a positive value is expected, for systems located west of UTC0000 a negative value is expected. E.g.</p> <p style="text-align: center;">SET TZUTC=0200</p> <p>is the correct setting for Central Europe during summer time. The information specified in the TZUTC variable is added to local messages written with FM as "<sup>a</sup>TZUTC: &lt;data&gt;". FD puts it into its (EMSI) session negotiation packet to aid time synchronization and other services.</p>
Name	FDOPT
Versions	All
Description	<p>Contains options overriding configuration settings. Most options supported in the FDOPT variable are also supported on the command-line to the various programs. The available settings for FDOPT are listed in detail in the FrontDoor documentation.</p>

## 3. FILES

A list of the files used in the FrontDoor environment follows.

File	SETUP.FD
Location	See above
Version notes	None
Definition(s)	CTL.H and CTL.INC
Contents description	Global system information for the FrontDoor environment. It must be present before any of the programs will operate.

File	MDCFGnnn.FD
Location	SYSTEM directory
Version notes	[ML]
Definition(s)	LINECFG.H and LINECFG.INC
Contents description	Task specific modem configuration data. This file allows multiple mailers (FD) to share the same SETUP.FD and still override the modem configuration for different tasks.  If the current task is zero or MDCFGnnn.FD does not exist, FD uses the modem configuration data located in SETUP.FD.

File	EVENTnnn.FD
Location	SYSTEM directory
Version notes	EVENT.FD in [SL] (see below)
Definition(s)	EVENT.H and EVENT.INC
Contents description	FD's operating schedule (mail and external event definitions).  [ML] If the current task is zero or EVENTnnn.FD does not exist, FD uses EVENT.FD.

File	ROUTE <sub>nnn</sub> .FD
Location	SYSTEM directory
Version notes	ROUTE.FD in [SL] (see below)
Definition(s)	See FrontDoor documentation
Contents description	FD's mail routing definitions. This file is tokenized into a binary file by FD. The resulting file is named ROUTE <sub>nnn</sub> .FD@. [ML] If the current task is zero or ROUTE <sub>nnn</sub> .FD does not exist, FD uses ROUTE.FD (though it is still tokenized into ROUTE <sub>nnn</sub> .FD@).

File	ROUTE <sub>nnn</sub> .FD@
Location	PACKETS directory
Version notes	ROUTE.FD@ in [SL] (see below)
Definition(s)	None
Contents description	FD's mail routing definitions, tokenized from ROUTE <sub>nnn</sub> .FD or ROUTE.FD. If this file does not exist or its source file has changed, FD will automatically recreate the tokenized file. [ML] If FD cannot locate ROUTE <sub>nnn</sub> .FD@, it will first search for ROUTE <sub>nnn</sub> .FD and then ROUTE.FD. If the file could be located, it is automatically tokenized into ROUTE <sub>nnn</sub> .FD@.

File	OUTBOUND.HIS
Location	SYSTEM directory
Version notes	OUTB <sub>nnn</sub> .HIS in [ML] (for non-zero tasks) if NOSHAREOH is present on the command-line or in FDOPT.
Definition(s)	MAILHIST.H and MAILHIST.INC
Contents description	FD's outbound call history. The maximum age of the entries in this file is determined by a setting in SETUP.FD (" <i>Days to keep mail history</i> "). The file is automatically repacked by FD after the first call following 0000 (midnight).

File	INBOUND.HIS
Location	SYSTEM directory
Version notes	INBnnn.HIS in [ML] (for non-zero tasks) if NOSHAREIH is present on the command-line or in FDOPT.
Definition(s)	MAILHIST.H and MAILHIST.INC
Contents description	FD's inbound call history. The maximum age of the entries in this file is determined by a setting in SETUP.FD (" <i>Days to keep mail history</i> "). The file is automatically repacked by FD after the first call following 0000 (midnight).

File	LASTCnnn.FD
Location	SYSTEM directory
Version notes	LASTCALL.FD in [SL]
Definition(s)	LASTCALL.H and LASTCALL.INC
Contents description	FD's recent activity and most recent inbound and outbound call information.

File	NODIAL.FD
Location	SYSTEM directory
Version notes	None
Definition(s)	UNDIAL.H and UNDIAL.INC
Contents description	FD's undialable database.

File	MODEMnnn.FD
Location	SYSTEM directory
Version notes	MODEM.FD in [SL]
Definition(s)	MODEM.H and MODEM.INC
Contents description	Modem settings overrides for specific systems, nodelist flags, or baudrates. [ML] If the current task is zero or MODEMnnn.FD does not exist, FD uses MODEM.FD.

File	PASSWORD.FD
Location	SYSTEM directory
Version notes	None
Definition(s)	SESSION.H and SESSION.INC
Contents description	Session level passwords and specific session settings for remote systems.

File	REQUEST.FD
Location	SYSTEM directory
Version notes	None
Definition(s)	REQUEST.H and REQUEST.INC
Contents description	Password definitions for file and update requests (and service requests since they are defined through the file request interface).

File	TERMPROF.FD
Location	SYSTEM directory
Version notes	None
Definition(s)	TERMINAL.H and TERMINAL.INC
Contents description	The Terminal's user profile sets for IEMSI (Interactive EMSI) negotiated terminal sessions. The default user profile is stored in SETUP.FD.

File	TERMKEYS.FD
Location	SYSTEM directory
Version notes	None
Definition(s)	TERMINAL.H and TERMINAL.INC
Contents description	The Terminal's macro key sets. The primary macro key set (0) is stored in SETUP.FD.

File	TERMPHON.FD
Location	SYSTEM directory
Version notes	None
Definition(s)	TERMINAL.H and TERMINAL.INC
Contents description	The Terminal's internal telephone directory.

File	FOLDER.FD
Location	SYSTEM directory
Version notes	None
Definition(s)	FOLDER.H and FOLDER.INC
Contents description	FM's message folder list.

File	NAMES.FD
Location	SYSTEM directory
Version notes	None
Definition(s)	See FrontDoor documentation
Contents description	A list of system wide target addresses and name macros.

File	FDNODE.CTL
Location	NODELIST directory
Version notes	None
Definition(s)	See FrontDoor documentation
Contents description	Nodelist compiler control statements, telephone number translation directives, and cost assignments for different geographical areas.

File	Nodelist Database
Location	NODELIST directory
Version notes	None
Definition(s)	NODELIST.H and NODELIST.INC
Contents description	Various nodelist database files.

File	ACTMONDB.FD
Location	SYSTEM directory
Version notes	[SL] will only write to/update the zero task records.
Definition(s)	FDAMONDF.H and FDAMONDF.INC
Contents description	The FrontDoor activity database. This file is frequently updated by FD to indicate what it is presently doing. The program FDAMADD is included in the FrontDoor package which allows information to be written to the activity database.

File	FILQUEUE.FD
Location	SYSTEM directory
Version notes	None
Definition(s)	FDSTQ.H and FDSTQ.INC
Contents description	The static queue (STQ). This file is a single-file database containing instructions for FD about files to send, systems to call, etc. For applications that primarily use FD to deliver data and distribute files to other systems, the STQ is the recommended interface (as opposed to creating messages with file attachments in the System NetMail Folder).

File	
Location	
Version notes	
Definition(s)	
Contents description	

## Reserved space

Fields marked **Reserved**, **Reserverat**, **Not Used**, or **!Unused!** are reserved storage fields and may not be used for your own purposes. This is for your own good. The obvious situation would be if a new version of FrontDoor is released that uses that space and you don't have time to release an updated version of your utility.

## 4. MACROS

Throughout the FrontDoor environment, a special macro construct can be used in string fields. These macros are supported in user interaction as well as in configuration files. It is essential that a FrontDoor utility program support this macro construct. The macro syntax is as follows:

`#[macro]`

where “macro” is the name of the macro, e.g. “FD”, “USERNAME”, etc. In most cases, these macros are used to refer to data external to FrontDoor, i.e. the contents of an environment variable. There are, however, situations where a macro needs to be resolved according to some FrontDoor-specific logics.

### The basics

When `#[macro]` is encountered, a list of internal macros (described below) is first scanned. If no match was made in the internal macro list, the environment variables are scanned for a match. If no match was made with an environment variable, the `#[macro]` text should be left intact. When scanning for a match, the text comparison should be case insensitive.

It should be noted that recursive macros are supported. That is, `#[macro1]` may expand to a string containing the literal text `#[macro2]`. Thus, when scanning a string for macros, it is important to include the macro expansion in further processing. An application does **not** have to support a construct that results in infinite recursion, e.g. `#[macro1]` expands to the literal text `#[macro1]`, as this is considered an error.

### Internal macros

The following macros are supported internally by FrontDoor (note that configuration data may include macros - the log file field in `SETUP.FD` could for example be defined as `#[username]FD#[task].LOG`). Unless otherwise noted (by “[Optional]”), it is recommended that your application supports these macros.

Macro	LOGFILE [Optional]
Description	The name of the log file used by FD.
Notes	This is only supported by FD.

Macro	TASK
Description	The current task number (0-255).
Notes	For [SL], this always expands to a zero (0).

Macro	ROUTEFD [Optional]
Description	The full pathname of the (route) file used by FD to create its tokenized ROUTE <sub>nnn</sub> .FD@ route file, e.g. ROUTE.FD, ROUTE1.FD, etc. If the route file for a specific task does not exist, FD expands this to <system path>\ROUTE.FD.
Notes	This is only supported by FD.
Macro	SYPATH
Description	The SYSTEM directory as defined in SETUP.FD.
Notes	None
Macro	INFILES [Optional]
Description	The path used to receive files during non-secure mail sessions.
Notes	This is only supported by FD.
Macro	INSECFILES [Optional]
Description	The path used to receive files during secure mail sessions. If no secure path has been specified in SETUP.FD, this expands to the path used for non-secure sessions (see INFILES above).
Notes	This is only supported by FD.
Macro	YYYY
Description	The current year (e.g. "1997").
Notes	None
Macro	YY
Description	The current year without the century portion (e.g. "97"). For the year 2001, this expands to "01", i.e. left-padded with a zero.
Notes	None
Macro	MM
Description	The current month (1-12). For months 1-9, this expands to "0<month>", i.e. left-padded with a zero.
Notes	None

Macro	DD
Description	The current day (1-31). For days 1-9, this expands to "0<day>", i.e. left-padded with a zero.
Notes	None

Macro	PADTASK
Description	The current task number (0-255). For tasks 0-9, this expands to "00<task>", for tasks 10-99, this expands to "0<task>" – i.e. left-padded with one or two zeros.
Notes	For [SL], this always expands to three zeros (000).

Macro	HEXTASK
Description	The current task number (0-255) expressed in hexadecimal notation. The case of hexadecimal letters (A-F) is not important. This macro always expands to two characters.
Notes	For [SL], this always expands to two zeros (00).

Macro	SEMPATH
Description	The path used as the "semaphore path". If no specific semaphore path has been specified in SETUP.FD, SEMPATH expands to the same as SYSPATH.
Notes	For [SL], this always expands to the same as SYSPATH.

Macro	
Description	
Notes	

## 5. THE LOG FILE

FrontDoor and TosScan, as well as several other programs uses the same logging format. All lines in a FrontDoor log file must be terminated by a <CR><LF> pair (ASCII 13, ASCII 10). Below is a sample of three different programs all using the same log format:

```

----- Sun 17 Aug 97, FD 2.30.mL; Task=0
+ 17.27.21 Event 0-@ >>> (NoUsers:2h57m, NoReq:4h17m)
  17.27.21 27 active bundles (27 on hold)
= 17.50.44 RING
= 17.50.44 DFRS: FM:00931478550103 TO:0855630102
= 17.50.48 RING
= 17.50.53 CONNECT 64000/ARQ/X75/LAPB
+ 17.50.55 Slash #1 - Venray (NL), 2:512/1
~ 17.50.55   AKA: 2:512/5, 2:512/1000, 255:3031/0, 255:3031/1
~ 17.50.55 SysOp: jeroen van de leur
~ 17.50.55 Using: FrontDoor 2.30.mL/AE000064
: 17.50.55 RTime: 17 Aug 97, 17.47.37 (0100)
: 17.50.55 LTime: 17 Aug 97, 17.50.53 (0200)
~ 17.50.55 Site: http://www.slash.nl
~ 17.50.55 Flags: CM,XA,V34+,VFC,V32T,H16,V42B,U,X75,Z19
~ 17.50.55 Phone: +31 478 550103
- 17.50.55 sType: EMSI/MD5
- 17.50.55 Password protected mail session
* 17.50.57 Rcvd C:\FD\FILES\SECURE\ABCDEFGH.SA0; 1769b, 1769 CPS
* 17.51.02 Sent C:\FD\FILES\OUT\IJKLMNOP.FR0; 25674b, 6418 CPS
  17.51.02 Truncated sent file C:\FD\FILES\OUT\IJKLMNOP.FR0
+ 17.51.07 Mail transfer completed
$ 17.51.07 From 2:512/1, 0.14, 0 (2744 CPS)
- 17.51.08 Mail received, ErrorLevel=255; exiting

----- Sun 16 Nov 97, FDRPR 2.00; Task=1
+ 18.15.47 John Doe, Driver Divers - Stockholm, 255:3046/391
% 18.15.47  FREQ: OS2DRVR.RAR
% 18.15.47  FREQ: W95DRVR.RAR
% 18.15.47  FREQ: LINXDRVR.RAR
  18.15.49 Updating download counters
% 18.15.49 Found C:\FD\FILES\DRIVERS\OS2DRVR.RAR (Area #61)
% 18.15.49 Found C:\FD\FILES\DRIVERS\W95DRVR.RAR (Area #61)
% 18.15.49 Found C:\FD\FILES\DRIVERS\LINXDRVR.RAR (Area #61)

----- Mon 17 Nov 97, O/T-Track+ 2.66
= 13.03.13 Processing path C:\FD\MSG\NET\
| 13.03.14 All done

```

The time format used in the log file is always in 24 hour format (with 010000 written as <SPACE>0<SEP>00<SEP>00), but the separators (':' for USA, '.' for Sweden, etc.) should be taken from the OS country structure. The century of the year is never written to the log.

Please use a format identical to the above samples. There are several log analyzers out there for FrontDoor. Also, please don't use any of the 'symbols' (first character on the line) that are used by FrontDoor or TosScan as that could confuse some log analyzers.

## Separator line

The separator or “dash” line is created whenever FD is started. If you want to insert information about which program is running, do it like the samples above show. The format is as follows:

- Eight dashes (ASCII 45) followed by two spaces.
- The abbreviated name of the current weekday (Mon, Tue, Wed, Thu, Fri, Sat, Sun) followed by a space.
- The current day of the month (1-31) followed by a space. If the current day is less than ten (10), insert a zero before the day (e.g. “09” for 9).
- The abbreviated name of the current month (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec) followed by a space.
- The last two digits of the current year. If the last two digits of the current year is less than ten (10), insert zeros to pad (e.g. 2000 is “00”, 2001 is “01”).
- If you want to add a short program identifier, add a comma (‘,’) followed by the program’s identifier. If you want to add a task indicator after the program identifier, add a semi-colon (‘;’) followed by the text “<SPACE>Task=nnn” where “nnn” is the task number 0-255.

## 6. SEMAPHORE FILES

FrontDoor supports and uses a number of semaphore files to indicate a specific state of the program, and to force the program to perform a specific task. Applications external to FrontDoor can take advantage of these files to control the behavior of FrontDoor and to monitor its activities.

For [ML], a separate "semaphore path" can be configured, which specifies where FrontDoor should look for and create the files. If the semaphore path is not specified, the SYSTEM directory is used. The SYSTEM directory is always used by [SL] as the semaphore path.

When creating or updating semaphore files, it is recommended that you first attempt to open the file, and if successful, simply change the file's timestamp, rather than using a CREAT() call. This has several advantages, specially under environments that track deleted files even if they are of zero length.

To create a zero byte semaphore file from the OS prompt, use:

```
REM><semaphore_filename>
```

File	FDALIVE.nnn
Versions	All
Description	Indicates that FD is operating and active. This file is touched at least once every 60 seconds by FD, even when it is frozen. The file is removed when FD terminates or a user invokes an external program (including OS shells) from within FD. It is not removed when the Terminal is in use.

File	FDINTERM.nnn
Versions	All
Description	Indicates that FD is currently in Terminal (Emulator) mode. This file is removed once the user leaves the Terminal.

File	FMALIVE.nnn
Versions	All
Description	Indicates that FM is operating and active. This file is touched at least once every 60 seconds by FM, even when it is frozen. The file is removed when FM terminates or a user invokes an external program (including OS shells) from within FM.

File	FDFREEZE.nnn
Versions	All
Description	<p>Forces a specific task to freeze. FD scans for this file periodically and if detected, it closes all open files, except its overlay file, and lowers DTR. The FOSSIL driver is also deinitialized.</p> <p>FD will not honor this file while a mail session is in progress. It will, however, act upon it as soon as it has completed the mail session. This semaphore is not checked for when the Terminal is active.</p>
File	FDFREEZE.ALL
Versions	All
Description	This semaphore has the same function as above, with the exception that it forces all active mailers to freeze instead of a specific mailer.
File	PDFROZEN.nnn
Versions	All
Description	Indicates that a specific mailer is in a frozen state.
File	FMFREEZE.nnn
Versions	All
Description	Same as FDFREEZE.nnn, but affects FM instead of FD.
File	FMFREEZE.ALL
Versions	All
Description	Same as FDFREEZE.ALL, but affects FM instead of FD.
File	FMFROZEN.nnn
Versions	All
Description	Same as PDFROZEN.nnn, but for FM.
File	FDCANSES.nnn
Versions	All
Description	Forces FD to terminate the call/session in progress. If FD is currently sending or receiving a file, the call will not be terminated until the file has been transmitted/received. Once the call has been terminated, this file is removed by FD.

File	FDRENUM.nnn
Versions	All
Description	<p>Indicates that the System NetMail Folder is being renumbered (or otherwise updated) by an application. No applications may modify messages in the System NetMail Folder if this semaphore is present.</p> <p>FM creates this file (FDRENUM.nnn) when it is renumbering the System NetMail Folder. How FD reacts to this semaphore depends on what it is currently doing.</p> <p>If FD is processing the System NetMail Folder, it will act as if the FDFREEZE.nnn semaphore was present. Otherwise, it will prevent any mail sessions from taking place (i.e. ignore inbound calls and not place any outbound calls) as well as any rescans; until the file has been removed.</p>
File	FDSCAN.nnn
Versions	All
Description	<p>Indicates that FD is currently processing the System NetMail Folder. No applications may modify messages in the System NetMail Folder if this semaphore is present unless it is guaranteed that they will not be included by FD in a packet, i.e. Sent, Received, or Locked messages.</p>
File	FDINEXIT.nnn
Versions	All
Description	<p>Indicates that FD has dropped to an external program such as a BBS while a call is in progress. This file is used by FD when it "recycles" and compared with the timestamp of rescan semaphores, etc. to determine if any changes have been made to the System NetMail Folder while the external program was executing.</p>
File	FDNOEXIT.NOW
Versions	All
Description	<p>Prevents FD from triggering an exit when mail has been received.</p>
File	FDNOUSER.NOW
Versions	All
Description	<p>Prevents interactive (non-mail) callers from being passed on to a BBS or similar program, even if FD has been configured to allow interactive access.</p>

File	FDNOSCAN.NOW
Versions	All
Description	Prevents FD from rescanning the System NetMail Folder.
File	FDRESCAN.NOW
Versions	All
Description	Forces FD and FM to rescan the System NetMail Folder (unless FDNOSCAN.NOW is present). See below for additional notes regarding this semaphore.
File	ROUTEnnn.FD
Versions	All
Description	While not a semaphore file per se, FD will monitor the timestamp of the route file and if a change is detected, it has the same effect as if FDRESCAN.NOW would have been updated (see above). Additionally, if a change of the file's timestamp is detected, FD will tokenize it into ROUTEnnn.FD@ (see "Files").
File	FDXITnnn.eee
Versions	All
Description	Forces FD (with task number NNN) to terminate with an errorlevel indicated by EEE (0-255). If a mail session is in progress, FD will complete it prior to exiting. The semaphore is removed as soon as FD has acted upon it.
File	FDXITANY.eee
Versions	All
Description	Similar to the FDXITnnn.eee semaphore with the exception that no task number is specified. The first mailer to detect this semaphore will act upon it and remove the semaphore once it has done so.
File	FDINMAIL.NEW
Versions	All
Description	Similar to the FDXITANY.eee semaphore with the exception that it does not specify a task number nor errorlevel. The first mailer to detect this semaphore will act as if it has received mail in an inbound session. The semaphore is removed as soon as FD has acted upon it.

File	FDNOCALL.nnn
Versions	All
Description	Prevents FD from placing outbound calls.

File	FMRESCAN.NOW
Versions	None
Description	<b>This is no longer supported</b> (see “FDRESCAN.NOW”, below).

File	FMNEWNET.nnn
Versions	All
Description	Indicates that FM has created one or more messages in the System NetMail Folder.

File	FMNEWLOC.nnn
Versions	All
Description	Indicates that FM has created one or more messages in a Folder with Local status.

File	FMNEWCNF.nnn
Versions	All
Description	Indicates that FM has created one or more messages in a Folder with Conference status.

File	FDQPACK.nnn
Versions	All
Description	Indicates that FD is currently packing the static queue (STQ).

File	
Versions	
Description	

## FDRESCAN.NOW

This semaphore has several purposes and the result of its presence differs depending on its location. If FD detects a change to this semaphore in its “semaphore path”, it will rescan the System NetMail Folder, as well as the STQ, and process outbound mail.

This semaphore is also used to signal a change in .MSG-type folders. FD will create/update this semaphore in the directory of the System NetMail Folder when new mail has been unpacked after a mail session.

If FM detects a change to this semaphore in the folder (.MSG-style) that is currently active, it will rebuild its list of messages. FM does not monitor this file in the “semaphore path”.

## Inter-process Communications Area (ICA)

If FrontDoor has been loaded with USEICA present on the command-line or FDOPT, it supports an alternative method for communicating specific changes. Because the use of the ICA causes problems on some computers, it is disabled by default.

The ICA can be used by TSR (Terminate and Stay Resident) programs to detect changes and to signal changes to FrontDoor. The ICA is located in low BIOS memory at location 0000:04F0 (hexadecimal, segment:offset notation).

All strings must be preceded by an ASCII 6 character. Once an application has acted upon a string in the ICA, the application will clear the applicable portion of the ICA.

The following strings are supported/used:

### FM

String	<ASCII_6>FDSCAN
Description	Written by FM when it normally writes FDRESCAN.NOW to the directory of the System NetMail Folder.

String	<ASCII_6>FMSCAN
Description	If detected when the System NetMail Folder is the active folder, FM will rebuild its message list for that folder.

### FD

String	<ASCII_6>FDXITnnn.eee
Description	See FDXITnnn.eee semaphore.

String	<ASCII_6>FDXITANY.eee
Description	See FDXITANY.eee semaphore.

String	<ASCII_6>FDSCAN
Description	See FDRESCAN.NOW semaphore.

String	<ASCII_6>FMSCAN
Description	Written by FD when it normally writes FDRESCAN.NOW to the directory of the System NetMail Folder..

String	<ASCII_6>FDINMAIL
Description	See FDINMAIL.NEW semaphore.

String	<ASCII_6>FDXITANY.eee
Description	See FDXITANY.eee semaphore.

## 7. THE STATIC QUEUE (STQ)

FrontDoor 2.25/2.30 introduces a static transmission queue database, or in short, "the static queue" - hereafter "STQ". The STQ is basically a single file, variable-length record, database containing zero or more "action" entries.

Because of the single-file approach, variable-length record size and use of binary digit fields where appropriate, the STQ can hold a large number of entries without severe performance penalties (many more than, for example, the .MSG-format System NetMail folder). The STQ furthermore simplifies the sending and requesting of files.

### Advantages

To say that the advantages of using the STQ over traditional-style NetMail messages with file attachments are endless is perhaps to exaggerate, but if used properly, the STQ can expand a system's capabilities and flexibility beyond most people's imagination.

### Limitations

The two major limitations (depending on how you look at it) of the STQ are:

- No routing is possible of entries in the STQ. All addresses will be used as specified. In the event that a non-existent, or non-dialable address is specified for a non-held entry, FD will simply create a bundle in its bundle list with hold status.
- Message text cannot be easily transmitted by using the STQ. It is, of course, possible to simply create a .PKT (mail packet) file and add an entry referencing this .PKT file to the STQ.

The two above limitations are "as designed" and do not, in our opinion, constitute practical limitations.

### Basic rules

- All applications must open the STQ in **ReadWrite** (or **ReadOnly**) **DenyNone** mode. No other modes are acceptable. There is one exception to this which is explained later.
- If, for some reason, an application creates a new STQ by using `creat()`, `creatnew()`, or similar functions – it must be closed and re-opened as described above immediately after writing the necessary fixed-size header (described below).
- All entries must be added (appended) to the end of the STQ. Re-using entries marked as "deleted" is not permitted. If, for some reason, the size of an entry is modified, the old entry must be marked as "deleted" and the new entry added to the end of the STQ.
- When adding entries, the entire entry must be written in a single I/O operation (application to operating system layer). When modifying the status of an entry, the four-byte flags field must be written in a single I/O operation.

- All timestamps in the STQ are UNIX-style timestamps as described at the beginning of this document.
- All strings are stored with a single length byte, followed by zero or more bytes. The maximum length of a string is 256 characters including the length byte. An empty string is simply stored as a <NUL> character.
- With one exception (described below), it is not permitted to add a reference (entry) to a file to be transmitted unless and until the specific file is actually available for transmission.
- The FDRESCAN.NOW semaphore file (in the “semaphore path”) is used to signal changes to the STQ. The FDRESCAN.NOW semaphore in the System NetMail Folder should not be updated if the application only modifies the STQ.

## The file format

The STQ is stored in a file called FILQUEUE.FD in the SYSTEM directory. It consist of one fixed-length binary header and zero or more variable-length records. The fixed-length record is 1024 bytes (1 Kb).

### Fixed-length binary file header

The STQ contains one fixed-length binary header from offset 0 (zero) to 1023, the length is 1024 bytes (1 Kb). The format of this header is:

```
char
  qHdr[22]="FrontDoor File Queue<EOF><NUL>"
unsigned short int
  Revision=0x0100;
unsigned long
  TimeCreated,
  TimePacked,
  ReservedLong,
  PackRecovery;
char
  ReservedData[];
```

**TimeCreated** and **TimePacked** are both UNIX-style timestamps as described at the beginning of this document. **TimePacked** should be set to zero when the STQ is first created. **ReservedData[]** is simply an array of bytes to fill out the block to 1024 bytes. FD will clear the entire **ReservedData[]** field with NULs each time the STQ is compressed/packed.

The **PackRecovery** field must contain zero (0) for an application to be allowed to add, modify, or in any way process the STQ. If it contains any other value, the STQ is not in working condition (see “Error recovery”).

### Variable-length records

Each variable-length record has the format:

```
unsigned short int
    EntryLen;
unsigned long
    EntryTime,
    Flags,
    TimeStamp;
char
    Address[],
    Filename[],
    TFA_or_Password[];
```

```
.. or ..
```

```
<EL><ETET><FFFF><TSTS><Address><Filename><TFA>
```

## EntryLen

An unsigned short (16-bit) integer holding the size of the data that follows. It does not include the **EntryLen** field itself. If an entire record is parsed (that is, all required fields are parsed), but less data than EntryLen bytes has been extracted, the remaining number of bytes until the next entry should be extracted and ignored.

## EntryTime

Time when entry was added (UNIX-style).

## Flags

Miscellaneous flags (see below).

## Timestamp

Timestamp (UNIX-style) used if specific flags are set (see below).

## Address

Unless otherwise specified in the FLAGS field, this is a FidoNet-style network address in the format Zone:Net/Node[.Point], where [.Point] is omitted if zero. With the exception of .Point, no fields are optional. Note that when creating the string that goes into this field, all components of the address must be considered unsigned (i.e. a binary value of ffff hex must be expressed as 65535).

Examples of some valid addresses:

255:3046/0  
2:201/330  
65535:65535/65535.65535

## Filename

If the entry's FLAGS field indicate the need for a filename or directory name, this field contains such information, otherwise this field contains a single length character (<NUL>). For a plain *file to be transmitted* entry, this field may contain wildcards.

## TFA/Password

For files with the TFA option set in the FLAGS field, this field contains the filename that will be transmitted to the remote in place of the name used locally. A typical example of use for the TFA option is a Conference Mail processor that needs to store local ARCmail files with one name and transmit them as something else.

For files with the FREQ option set in the FLAGS field, this field contains the password to transmit to the system from the file(s) are to be requested. No space characters (ASCII 32) may be stored in FREQ passwords.

## Flags

Please refer to FDSTQ.H and/or FDSTQ.INC for the binary representation of these flags.

## FQflgKFS

Remove file after successful transmission after checking queue and System NetMail Folder that no other attaches for the same file exist. If this flag is used, FILENAME must not contain wildcards.

## FQflgKFSNoCheck

Remove file after successful transmission. The file will be removed immediately after it has been (successfully) transmitted, no checking is done for other attaches referring to the same file(s).

## FQflgTFS

Similar to FQflgKFS, but instead of removing the file, FD will truncate it to zero-length.

## FQflgTFSNoCheck

Identical to FQflgKFSNoCheck, but instead of being removed, the file(s) will be truncated to zero-length. This flag is ideally suited for Conference Mail processors.

## FQflgIsARCmail

Indicates that the file contains ARCmail (compressed mail).

## **FQflgSendStart**

Indicates that FD has begun transmission of the file. This flag is not cleared if the session fails. It is intended to be used by applications that should not add or modify file data if transmission of the file has been initiated.

## **FQflgSendAfter**

Prevents entry from being processed until the current time is equal to or greater than the time specified in the `TIMESTAMP` field.

## **FQflgKeepExpired**

Indicates that if the entry has expired (`FQflgSendUntil`), it should be kept in the STQ when the file is compressed/packed.

## **FQflgSendUntil**

The entry is considered active until the current time is equal to or greater than the time specified in the `TIMESTAMP` field. Once an entry has expired, it will be removed by FD when the STQ is compressed/packed (unless the above `FQflgKeepExpired` flag is also set). Files referenced by the entry will be immediately removed or truncated if they have either of the `FQflgKFSNoCheck`, `FQflgTFSNoCheck`, or `FQflgIsSpool` flags set. For entries with `FQflgKFS` or `FQflgTFS`, a check is first made to see that no other entries refer to the specified files (in which case the files are not removed).

## **FQflgIsHold**

Entry is on hold and does not affect, nor will it be processed during, outbound calls.

## **FQflgIsCrash**

Entry has crash status and will be treated as a message with crash status to the specified system.

## **FQflgIsIMM**

Entry has immediate status and will be treated as a message with immediate status to the specified system (i.e. all event and routing restrictions will be ignored).

## **FQflgNoPickup**

Entry will not be processed during inbound sessions.

## FQflgIsSpool

FILENAME contains a path and filemask, e.g. C:\FD\SPOOL\PNT-1\\*.\*, indicating a spool directory. This flag implies the FQflgIsFile and FQflgIsKFSNoCheck flags. When FD scans the STQ, entries with FQflgIsSpool status will be ignored unless one or more files matching the specified filemask exist. FQflgIsSpool entries are not automatically removed by FD unless they have expired (see FQflgSendUntil above). An entry with the FQflgIsSpool flag set will result in FD sending all files matching the filemask specified in FILENAME.

## FQflgIsFREQ

FILENAME contains filemask to request from remote system. If FILENAME includes a path, an update request is performed.

## FQflgIsFile

FILENAME contains filemask to transmit to remote system.

## FQflgTFA (or "FQflgPassword")

This flag is only valid for entries with FQflgIsFile or FQflgIsFREQ status. For FQflgIsFile entries, this field contains the filename which FD should transmit to the remote (*Transmit File As..*). The filename specified in TFA is transmitted to the remote as is (no case or path conversion). For FQflgIsFREQ entries, this field contains the password that should be transmitted to the remote together with the filemask to be requested. If no password is required for a FQflgIsFREQ entry, the FQflgTFA flag should not be set.

## FQflgHidden

No special treatment with the exception that entries with this flag present should not be visible in interactive browsers/queue manipulation utilities. In FD, the STQ manager will not display entries with this flag present.

## FQflgLocked

Entry is locked and will be ignored (but not deleted).

## FQflgDeleted

Entry has been deleted and will be ignored (and removed when the file is compressed/packed).

## Poll entries

An application can create an "empty" entry with nothing but status flags associated with it. Such empty entries can be used to force FD into placing an outbound call, i.e. making a "poll" of the remote system. An empty entry's **Filename** and **TFA\_or\_Password** fields contain a single <NUL> character each.

## Forcing rescans with care

When an application adds or modifies entries in the STQ, care should be taken regarding the updating of the FDRESCAN.NOW semaphore in the “semaphore path”. When adding an entry with the FQflgIsHold flag set, no rescan should be signalled; the same applies when adding an entry with the FQflgLocked flag set (since it cannot, by definition, go anywhere).

## Rescanning the STQ

Each time FD has come to the end of its current list of files to send, it will scan the STQ for newly added entries as well as process spool entries. If new files are encountered, it will create a new list of files to send and then repeat this procedure until no more files remain to be sent or an error occurs.

## Accessing the file

One additional file is used for the STQ. The file is FILQUEUE.LCK and is located in the “semaphore path”.

## Opening the file

If, for any reason, an application fails to open the file queue and the return code is "Access Denied" (EACCES) or "File not found" (ENOFILE), it should take different action depending on whether or not the lock semaphore (FILQUEUE.LCK) file exist.

If the lock semaphore does not exist and the error code was EACCES, the application should wait a short period of time (1-5 seconds) before trying to open the STQ again. In the event of further failures with the same error code, the application may make more attempts, or abort with a fatal error.

If the lock semaphore does not exist and the error code was ENOFILE, the application should create FILQUEUE.FD in SYSTEM directory, write a properly formatted header to it, close the file, and reopen it in the correct access mode.

If the lock semaphore does exist, the application should wait a short period of time (1-10 seconds) before trying to open the STQ again. When the lock semaphore is present, it is typically an indication that the STQ is being compressed/packed by FD.

## Adding new entries

To add new entries to the file, the application needs to open the static queue, acquire a write lock, seek to the end of the file, write the data, and release the write lock – in that order. To modify the flags field of an entry, no write lock needs to be acquired.

To acquire a write lock, the application should call creatnew() to create the FILQUEUE.LCK semaphore. If successful, the file may be closed, or kept open for the duration of the operation. Either way, it must be removed once the application has completed its I/O operation(s) on the static queue. A write lock may never be applied for more ten (10) seconds unless the STQ is being compressed/packed.

If the `FILQUEUE.LCK` semaphore already exist, the application should wait a short period of time (1-5) seconds before trying again. If `creatnew()` is not available, the application must use a function that fails if the file already exist – the function must not modify the file in any way.

## Deadlock

Even though short timer settings are specified above, an application should allow for some flexibility in handling file access to the queue when no write lock can be acquired. If an application needs to have write access to the queue and one cannot be acquired, an alternative to aborting does exist.

While waiting to acquire the write lock, an application should set a timer to 60 seconds. This is to be considered the `DeadlockTimer`. Once every five seconds, the application should attempt to fetch the timestamp of the lock semaphore. If it has been updated since the last fetch, the `DeadlockTimer` should be reset to 60 seconds. If the `DeadlockTimer` expires, the application should remove the lock semaphore and attempt to create it. If successful, the application has acquired a write lock.

Once the write lock has been acquired, the timestamp of the lock semaphore should be updated at least once every ten seconds to allow other applications to utilize the deadlock detection method described above. Note that in many operating environments, simply setting the timestamp, or setting the timestamp and issuing a commit call does not update the directory entry - in those environments it is usually required to update the timestamp and then close the file again.

How long an application is willing to wait before aborting is entirely up to the programmer.

## Other semaphores

When FD is compressing/packing the queue, it will check for an `FDINSESS.*` semaphore, and if none exist, create the semaphore `FDQPACK.nnn`.

After it has successfully created the semaphore, it will once again check for an `FDINSESS.*` semaphore, if none exist, packing will commence, otherwise the `FDQPACK.nnn` semaphore is removed and the program continues to operate as usual.

When FrontDoor checks its semaphores, it will "freeze" when the `FDQPACK.nnn` semaphore is detected, thus effectively preventing it from answering the phone and processing outbound calls. `FDQPACK.nnn` files where `.nnn` is its "own" task number are ignored and used for a special purpose (see "Error recovery").

## Error recovery

When FD is compressing/packing the queue, it will copy active entries to a temporary file, `FILQPACK.TMP` (SYSTEM directory), as well as expired entries with `KFS/TFS` status (for later checking). Before it starts this operation, it updates the `TimePacked` field to Midnight of the current date.

Once the temporary file has been successfully written, it is closed, and the PackRecovery field in the queue header set to ffffffff hex. The temporary file is then reopened and copied back into the STQ, effectively overwriting the existing queue. If the temporary file cannot be successfully written, it is not considered an error - it is simply removed, and packing is left until the next day (since the TimePacked field has now been updated).

Unless an error occurs, the queue is then truncated at the last written position, the PackRecovery field is set to zero (0), and the temporary file is removed, along with the FDQPACK.nnn semaphore. If an error occurs while copying the contents of the temporary file to the STQ, the static queue is truncated immediately following the fixed header (if possible), and FD is terminated with a fatal error.

If FD, upon startup, detects that the queue has been left in an unfinished state (i.e. it finds its "own" FDQPACK.nnn semaphore), it will execute the compression/pack function. If that function detects that the PackRecovery field is anything but zero (0), it will simply copy the temporary file back to the queue - if the field is zero, it will proceed to compress/pack the queue as usual.

## Alternative method of deleting entries

To remove an entry from the STQ, an application need only change the status of an entry to include the FQflgDeleted flag. When removing entries with the FQflgKFS or FQflgTFS flags set, however, some extra work must normally be performed; i.e. the application needs to check whether or not the file is present in a still-active entry, and should also check the System NetMail Folder for file attach messages referring to the file(s). The code required to do this is not complicated, but since FD needs to do this internally, an alternative exist for applications that need to remove entries with either of the forementioned flags present.

Simply add the FQflgHidden and FQflgSendUntil flags to the entry's status and set the TimeStamp field to zero - this will instruct FD that the entry has expired and should be removed at the next opportunity (when the STQ maintenance function is automatically invoked). It should also be noted that the FQflgSendAfter and FQflgKeepExpired flags must be removed, if present, from the entry's status. Adding the FQflgHidden flag ensures that the entry is not visible in the FD STQ manager, accessible from the main screen.

## Notes to Conference Mail application authors

It should be noted that entries with a FQflgSendAfter or a FQflgSendUntil flag set, are treated as inactive by FD if they have either expired or not yet become active. The obvious pitfall in this situation is an application that keeps adding data to an "ARCmail" file after it has, in effect, expired and thus will not be sent by FD.

When a Conference Mail application is scanning the STQ to determine if it should add mail to an existing file, or create a new file and add a new entry to the STQ, it should be able to handle the FQflgSendStart flag. Once FD has begun transmission of a file, it sets this flag. If this flag is encountered when the Conference Mail application scans for existing mail, the entry should be ignored and if no other applicable entries (without this flag) are encountered, a new entry should be added to the STQ. Note that the presence of this flag does not mean that the file associated with the entry can be removed.

## 8. EXTERNAL INTERFACE

A number of files are created "on the fly" by FD. The contents of these files are intended to be used by external applications interfacing with FD.

File	CALLER.nnn
Location	SYSTEM directory
Definition(s)	CALLER.H and CALLER.INC
Contents description	Holds information about the call in progress. The FD creates this file when it has dropped to an external program such as a BBS or when one of the External Mail strings triggers and exit. It also creates this file when a mail call has been initiated (inbound or outbound).
File	???????.FDS
Versions	SYSTEM directory
Definition(s)	SREQ.H and SREQ.INC
Contents description	Holds information about the remote system during a mail session. This file is created when FD invokes an external request processor (see "External request processors") and when a service request is processed (see "Service requests").
File	???????.`nn
Versions	SYSTEM directory
Definition(s)	N/A
Contents description	<p>Indicates that FD is presently dialing or in session with the remote system who's address is indicated by ???????. The task number of the mailer that created the file is indicated by the last two characters in the filename in hexadecimal notation.</p> <p>The base filename (???????) is created by generating a string of the remote system's address, e.g. 255:3046/0 or 2:201/330.1 and then calculating a CRC-32 on the string. The resulting 32-bit CRC is then used as the filename (hexadecimal notation).</p> <p>When dialing a system, FD creates only one "busy semaphore", since it does not know which other addresses (AKAs) remote system will present. As soon as the connection has been established and system information has been exchanged, FD creates one busy semaphore for each address presented by the remote. If FD detects an existing busy semaphore for the system it is trying to dial, it will abort the dial attempt.</p> <p>For inbound sessions, FD will wait for any busy semaphores matching the remote system's AKAs to be removed. The maximum time FD will wait is 21 seconds. If they are still present after 21 seconds, FD will terminate the session. This allows Conference Mail applications to safely add data to mail archives that are destined for a remote system calling the local system, thus preventing corrupt archives.</p>

## 9. EXTERNAL REQUEST PROCESSORS

The External Request Processor ("ERP") concept allows the user to configure FD to pass control to an external application to handle all file requests received from a remote system. The possibilities for an application like this are many, not to say endless, including but not limited to fast CD-ROM searches, supporting proprietary BBS file database formats such as those used by Maximus and RemoteAccess. The ERP concept works much like service requests (see "Service requests").

To configure FD for an ERP, the user simply specifies an application that should be called and then enables the use of it.

When FD receives a file request, it creates a list of the requested files and then spawns the ERP. When the ERP returns, FD scans another list containing the files to send to the remote system (if any) and what to do with the files after they have been sent.

There are, however, a few things an ERP may **not** do:

- Remove any files that are currently held open by FD.
- Communicate with the COM port in use by FD via the FOSSIL driver.
- Allocate memory (including LIM/EMS and XMS memory) and not properly deallocate it.
- Modify or delete messages in the System NetMail Folder (creating new messages is OK).
- Modify or delete .PKT and .FDC files in FD's packet directory.

An ERP should return control to FD inside of 35-40 seconds if at all possible. It is a requirement that the ERP be able to:

- Handle critical errors, such as disk failures, gracefully. That is, terminate without locking up the system.
- Because the user may or may not choose to enable swapping of FD's program image to disk or EMS/XMS memory during execution of the ERP, it is advisable that the ERP keep memory usage to a minimum.
- FD will process the received file requests prior to invoking the ERP to handle Service Requests. That is, the ERP can safely assume that all remaining files are definite file or alias requests and does not have to handle Service Request definitions.
- If any of the requests fails, it is the ERP's responsibility to inform the remote system of the reason for the failure. The best method to do this is to create an FSC-39 compatible (FDPKT.H) mail packet and place it as the first file to send in the response file. The contents of the file defined in FDSETUP (if any) as the "Message" file, should be included in the message text of the notification.

The following macros, in no specific order, can be specified on the command-line to the ERP in FDSETUP. Note that the =L, =N, =O, =I, and =K macros should be placed within double quotes ("") since it's quite possible that they contain OS control characters, such as '>', '<', and '|'.

E.g. MyProg /Operator="=0".

Parameter	=A
Expansion	The primary network address of the remote system. This is expressed as z:n/n[.p], where [.p] is omitted if it's zero.
Parameter	=B
Expansion	The current DCE speed. E.g. 16800, 28800, 33600, 64000, 128000, etc.
Parameter	=C
Expansion	Cost per minute to call the remote system. This is 0 (zero) for inbound sessions.
Parameter	=D
Expansion	The string INBOUND or OUTBOUND depending on the initial flow of the session. INBOUND means that the remote system called this system, OUTBOUND means that this system initiated the session.
Parameter	=L
Expansion	The remote system's location. Spaces are replaced by the underscore (_) character.
Parameter	=N
Expansion	The remote system's site name. Spaces are replaced by the underscore (_) character.
Parameter	=O (this is not a zero)
Expansion	The name of the remote system's primary operator (SysOp). Spaces are replaced by the underscore (_) character.
Parameter	=X
Expansion	The string SECURE or UNSECURE depending on whether or not the session is protected with a session level password.
Parameter	=Z
Expansion	The type of the session, i.e. WAZOO for FTS-6 (YooHoo) sessions, EMSI for FSC-56, or OTHER for all other types of sessions.

Parameter	=F
Expansion	The full pathname of the file containing "Remote Site Info". The contents of this file is described in SREQ.H and SREQ.INC. This file is automatically removed by FD when the session has completed. See also "=M" (below).
Parameter	=T
Expansion	The full pathname to the file FD expects to contain files to send when the ERP returns control to it. This is also referred to as "the request response file". The format of this file is defined below. This file is automatically removed by FD when the session has completed. See also "=E" (below).
Parameter	=R
Expansion	The full pathname of the file containing the requested files (described below). This is also referred to as "the request file". This file is <b>not</b> automatically removed by FD. It is the ERP's responsibility to do so. See also "=Q" (below)
Parameter	=G
Expansion	The number of minutes until the next event that does not allow interactive access. If the next event that does not allow interactive access is not known, this expands to 10080.
Parameter	=H
Expansion	The number of minutes until the next event that does not allow file requests. If the next event that does not allow file requests is not known, this expands to 10080.
Parameter	=I
Expansion	This can expand to either "N/A" or the data following the rign signal (DFRS), i.e. CALLER*ID information. Spaces are replaced by the underscore (_) character.
Parameter	=J
Expansion	The remote system's telephone number. Spaces are replaced by the underscore (_) character.
Parameter	=K
Expansion	The remote system's nodelist flags. Spaces are replaced by the underscore (_) character.

Parameter	=M
Expansion	The name (without path) of the file containing "Remote Site Info". The contents of this file is described in SREQ.H and SREQ.INC. This file is automatically removed by FD when the session has completed. See also "=F" (above).
Parameter	=W
Expansion	The remote system's nodelist status.
Parameter	=E
Expansion	The name (without path) of the file FD expects to contain files to send when the ERP returns control to it. This is also referred to as "the request response file". The format of this file is defined below. This file is automatically removed by FD when the session has completed. See also "=T" (above).
Parameter	=Q
Expansion	The name (without path) of the file containing the requested files (described below). This is also referred to as "the request file". This file is <b>not</b> automatically removed by FD. It is the ERP's responsibility to do so. See also "=R" (above)
Parameter	=Y
Expansion	The current task number (0-255).
Parameter	=V
Expansion	The remote mailer's product information. The remote mailer software presented a serial number, it is concatenated to this string after a slash has been appended. E.g. "FrontDoor_2.30.mL/AE000001". Spaces are replaced by the underscore (_) character.

## The request file (=R and =Q) format

The format of the file containing the requested files is as follows:

```
<filename>[ <space>!<password>][ <space><+><time>]<CR>
```

where <filename> is any valid DOS filemask, including wildcards. The <time> field is a UNIX-style timestamp in decimal notation, indicating that the remote system only wants files that are more recent than the specified time (some software allows the <+> to be replaced by a <-> to indicate that all match files with a timestamp less or equal to the specified time should be sent, FD, however, does not support this).

The ERP is responsible for honoring the values found in SETUP.FD for the maximum number of files to send to the remote system during a file request session.

### The request response file (=T and =E) format

The format of the file FD expects to contain files to send when the ERP terminates is as follows:

```
<symbol><filename><CR><LF>  
..  
..  
<symbol><filename><CR><LF>
```

where <symbol> indicates what FD should do with the file after it has been transmitted. A question mark (?) means that FD should retain the file if it was not successfully transmitted and delete it if it was. A plus (+) means that the file should be retained regardless of the result of the transmission. A minus (-) means that the file should be removed regardless of the result of the transmission.

## 10. SERVICE REQUESTS

The Service Request interface is very similar to the ERP (described above). The same general rules apply in regards to memory allocation, termination, no FOSSIL I/O, etc. The intended usage is, however, somewhat different.

Service Requests are typically used to invoke programs that update databases, or sends specific files only after checking a database for specific information (product updates for example).

Service Requests are defined in the file request alias list. By placing a '>' character in front of the second item on the line defining the alias. E.g.

```
PRODUCTUPDATE >PRODUPD.EXE ^<symbol><filemask>
```

where "PRODUPD.EXE" is the program to execute, ^<symbol> defines what FD should do with the files matching <filemask> after the session has been completed (see above). If no ^<symbol> is specified, FD assumes that all files matching <filemask> should be removed after the session has been completed regardless of the transmission status of each file.

The following macros, in no specific order, can be specified on the command-line to the Service Request program in the alias definition. Note that the =L, =N, =O, =I, and =K macros should be placed within double quotes (") since it's quite possible that they contain OS control characters, such as '>', '<', and '|'.

E.g. MyProg /Operator="=O".

Parameter	=A
Expansion	The primary network address of the remote system. This is expressed as z:n/n[.p], where [.p] is omitted if it's zero.
Parameter	=B
Expansion	The current DCE speed. E.g. 16800, 28800, 33600, 64000, 128000, etc.
Parameter	=C
Expansion	Cost per minute to call the remote system. This is 0 (zero) for inbound sessions.
Parameter	=D
Expansion	The string INBOUND or OUTBOUND depending on the initial flow of the session. INBOUND means that the remote system called this system, OUTBOUND means that this system initiated the session.

Parameter	=L
Expansion	The remote system's location. Spaces are replaced by the underscore (_) character.
Parameter	=N
Expansion	The remote system's site name. Spaces are replaced by the underscore (_) character.
Parameter	=O (this is not a zero)
Expansion	The name of the remote system's primary operator (SysOp). Spaces are replaced by the underscore (_) character.
Parameter	=X
Expansion	The string SECURE or UNSECURE depending on whether or not the session is protected with a session level password.
Parameter	=Z
Expansion	The type of the session, i.e. WAZOO for FTS-6 (YooHoo) sessions, EMSI for FSC-56, or OTHER for all other types of sessions.
Parameter	=F
Expansion	The full pathname of the file containing "Remote Site Info". The contents of this file is described in SREQ.H and SREQ.INC. This file is automatically removed by FD when the session has completed. See also "=M" (below).
Parameter	=P
Expansion	The password specified by the remote system for the requested alias that invoked the Service Request. Spaces are replaced by the underscore (_) character.
Parameter	=S
Expansion	The requested alias.
Parameter	=U
Expansion	The timestamp specified by the remote system for the requested alias (update request) or the string "0".

Parameter	=G
Expansion	The number of minutes until the next event that does not allow interactive access. If the next event that does not allow interactive access is not known, this expands to 10080.
Parameter	=H
Expansion	The number of minutes until the next event that does not allow file requests. If the next event that does not allow file requests is not known, this expands to 10080.
Parameter	=I
Expansion	This can expand to either "N/A" or the data following the rign signal (DFRS), i.e. CALLER*ID information. Spaces are replaced by the underscore (_) character.
Parameter	=J
Expansion	The remote system's telephone number. Spaces are replaced by the underscore (_) character.
Parameter	=K
Expansion	The remote system's nodelist flags. Spaces are replaced by the underscore (_) character.
Parameter	=M
Expansion	The name (without path) of the file containing "Remote Site Info". The contents of this file is described in SREQ.H and SREQ.INC. This file is automatically removed by FD when the session has completed. See also "=F" (above).
Parameter	=W
Expansion	The remote system's nodelist status.
Parameter	=Y
Expansion	The current task number (0-255).

Parameter	=V
Expansion	The remote mailer's product information. The remote mailer software presented a serial number, it is concatenated to this string after a slash has been appended. E.g. "FrontDoor_2.30.mL/AE000001". Spaces are replaced by the underscore (_) character.

# 11. MESSAGES

## Fax cover messages

As of version 2.20, the MultiLine versions of FrontDoor are capable of receiving fax transmissions by using internal (for supported modems) code and external programs. The user can then view these documents from FM provided that an external fax document viewer is installed and that the cover message follows a certain format.

A cover message is little more than a message that has the local and file attach bits set, along with a ^aFLAGS line with the "FAX" status present. The name of the physical file that contains the actual fax document is listed in the message subject. Typically, a fax cover message also contains some information about the dimensions of the fax, from where it was received, etc.

## Locked messages

Messages in .MSG-style folders that have the read-only file attribute bit set are said to be "Locked", that is, FD will not attempt to send or otherwise process them, and FM will not allow the message to be edited, unless the user first unlocks the message. If the message text contains ^aFLAGS LOK information, FM will automatically set the file attribute to read-only for the .MSG file when the message is first displayed - that is, if the read-only isn't already set.

## The LOCAL message status

If a message in the System NetMail Folder appears to originate from the local system, FD requires that the Local message status bit is also present. If the Local status bit is not present in these messages, they will be ignored by FD.

The Local bit is also required to be present before FD honors status bits that may affect the type or routing of the message, including Crash, Direct, Immediate, File request, etc. FD automatically removes the Local bit when unpacking messages from mail packets (.PKT files).

## .MSG file format notes

FrontDoor uses the Opus (1.03) convention for .MSG file formats. That is, the binary "date\_arrived" and "date\_created" fields. If it cannot successfully parse any of those fields, it will use the ASCII date string.

## 12. MULTI-LINE AWARENESS

If you are designing an application that is to be used in conjunction with the MultiLine version of FrontDoor there are several rules your application must adhere to. Obvious things include opening files in the correct sharing mode and not keeping files open longer than absolutely necessary. Note that these rules apply even if you are writing applications for the SingleLine version of FrontDoor but intend to use them on a LAN or in a multi-tasking environment while FD or FM is active.

If FD is in a mail session, do not modify any messages that have been packed by FD into outgoing mail packets. Messages with Received or Sent status can safely be modified, moved, or deleted, since they are never packed by FD. Adding new messages to the System NetMail Folder is always safe.

If you modify or create new messages in the System NetMail Folder, always update the appropriate semaphores (see "Semaphore files").

Do not modify the nodelist database while a FD or FM is active. If you have frozen all active mailers and editors, you can safely recompile or update the nodelist database.

When you create new messages in the System NetMail Folder, use a file create call that fails if the specified filename already exists (e.g. creatnew).

If FD is in a mail session, do not modify, move, or delete files that may be attached to messages that have been packed by FD into outgoing mail packets.

## 13. CONTACT INFORMATION

Definite Solutions can be reached at:

**Postal address:** Definite Solutions HB  
Forskottsvagen 11  
S-12932 Hagersten  
**SWEDEN**

**Telephone:** +46 8 6456495      **(0800-1800 CET, Mon-Fri)**

**Fax:** +46 8 55630102

**BBS:** +46 8 55630102      (V.34 8N1, X.75 ISDN)  
+46 8 55630103      (V.34 8N1, X.75 ISDN)

**Internet:** [info@defsol.se](mailto:info@defsol.se)  
<http://www.defsol.se/>

# 14. INDEX

- A**
- Application recommendations ..... 5
- C**
- Command-line parameters
    - USEICA ..... 27
  - Contact information ..... 50
  - CRC ..... 6
- D**
- Definite Solutions ..... 50
- E**
- Environment variables 7, 8
  - ERP ..... 39
  - External interface ..... 38
  - External request processors
    - ..... 39
    - Parameters..... 40
- F**
- FDRESCAN.NOW . 25, 26
- Files ..... 10
- .MSG files ..... 48
- I**
- ICA..... 27
- L**
- Log file ..... 20
  - Separator line..... 21
  - Look and feel..... 6
- M**
- Messages ..... 48
  - Multi-line awareness..... 49
- N**
- Notations
    - Program ..... 5
    - Version ..... 5
- R**
- Reserved space ..... 16
- S**
- Semaphore files..... 22
  - Service requests ..... 44
  - SETUP.FD..... 7
  - Static queue.....*See* STQ
  - STQ..... 29
    - Accessing ..... 35
    - Basic rules..... 29
    - Binary header ..... 30
    - Conference Mail..... 37
    - Error recovery ..... 36
    - File format..... 30
    - Other semaphores..... 36
- T**
- Timestamps*See* UNIX-time
- U**
- UNIX-time..... 6