

# **HP FTAM/9000 Reference Manual**

**Edition 4**



**B1033-90004**  
**HP 9000 Networking**  
**E0597**

Printed in: U.S.A.

© Copyright 1997, Hewlett-Packard Company.

---

## Legal Notices

The information in this document is subject to change without notice.

*Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.* Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

**Warranty.** A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

**Restricted Rights Legend.** Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

HEWLETT-PACKARD COMPANY 3000 Hanover Street Palo Alto, California 94304 U.S.A.

Use of this manual and flexible disk(s) or tape cartridge(s) supplied for this pack is restricted to this product only. Additional copies of the programs may be made for security and back-up purposes only. Resale of the programs in their present form or with alterations, is expressly prohibited.

**Copyright Notices.** ©copyright 1983-97 Hewlett-Packard Company, all rights reserved.

Reproduction, adaptation, or translation of this document without prior written permission is prohibited, except as allowed under the copyright laws.

©copyright 1979, 1980, 1983, 1985-93 Regents of the University of California

This software is based in part on the Fourth Berkeley Software Distribution under license from the Regents of the University of California.

©copyright 1980, 1984, 1986 Novell, Inc.

©copyright 1986-1992 Sun Microsystems, Inc.

©copyright 1985-86, 1988 Massachusetts Institute of Technology.

©copyright 1989-93 The Open Software Foundation, Inc.

©copyright 1986 Digital Equipment Corporation.

©copyright 1990 Motorola, Inc.

©copyright 1990, 1991, 1992 Cornell University

©copyright 1989-1991 The University of Maryland

©copyright 1988 Carnegie Mellon University

**Trademark Notices** UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

X Window System is a trademark of the Massachusetts Institute of Technology.

MS-DOS and Microsoft are U.S. registered trademarks of Microsoft Corporation.

OSF/Motif is a trademark of the Open Software Foundation, Inc. in the U.S. and other countries.



---

## **Printing History**

The manual printing date and part number indicate its current edition. The printing date will change when a new edition is printed. Minor changes may be made at reprint without changing the printing date. The manual part number will change when extensive changes are made.

Manual updates may be issued between editions to correct errors or document product changes. To ensure that you receive the updated or new editions, you should subscribe to the appropriate product support service. See your HP sales representative for details.

First Edition: April 1991 (HP-UX Release 8.0)

Second Edition: November 1992 (HP-UX Release 9.0)

Third Edition: January 1995 (HP-UX Release 10.0)

Fourth Edition: May 1997 (HP-UX Release 10.3)



---

## Preface

### Purpose and Scope

The purpose of this manual is to help you troubleshoot errors returned in your HP FTAM/9000 applications. This manual lists the errors, probable causes, and corrective actions. Additionally, this manual includes the HP FTAM/ 9000 reference pages.

---

#### NOTE

This manual lists FTAM error and reference information. If you are learning FTAM and need more comprehensive explanations, refer to the HP FTAM/9000 Programmer's Guide.

FTAM/9000 runs on HP OTS/9000, an HP network product that provides a lower-level OSI protocol “stack,” in conjunction with an 802.3 network link.

### Audience

This manual is for application programmers who are familiar with the C programming language and who need to correct errors in their applications. This manual is also for programmers who need to consult the FTAM/9000 reference pages (which are copies of the online man pages).

### MAP 3.0 Interface Note

Users should note that the FTAM/9000 programmatic interface is taken from the MAP 3.0 specification for FTAM. The MAP 3.0 FTAM specification includes error codes and other details, which consequently appear in the programmatic interface to FTAM/9000. This document contains numerous references to MAP-based details found in FTAM/9000.

## Organization

- Chapter 1**     **FTAM return\_codes**This chapter contains a list of FTAM errors returned in the `inout_dcb->result_code.return_code` field. Possible causes and recovery actions are also listed.
- Chapter 2**     **HP vendor\_codes**This chapter contains a list of HP-specific errors returned in the `inout_dcb->result_code.vendor_code` field. Possible causes and recovery actions are also listed.
- Chapter 3**     **Diagnostic error\_ids**This chapter contains a list of diagnostic values returned in the `inout_dcb->diagnostic->error_id` field. Possible causes and recovery actions are also listed.
- Chapter 4**     **Event Management Errors**This chapter contains a list of values returned when calling `em_wait()`, `em_fdmemory()`, and `em_gperror()`. These errors return in the `result->return_code` and `result->vendor_code` fields. Possible causes and recovery actions are also listed.
- Chapter 5**     **HP FTAM/9000 PICS**This chapter contains information on how to obtain a copy of the Protocol Implementation Conformance Statement (PICS) for the HP FTAM/9000 product. Use this information in conjunction with other vendor's PICS to determine how the various responders communicate.
- Chapter 6**     **Installation Filesets**This chapter lists the filesets created during FTAM/9000 installation.
- Chapter 7**     **HP FTAM/9000 Reference Pages**This chapter lists the HP FTAM/9000 reference pages (man pages) that are available online.



## Documentation Guide

### For More Information

### Read

Installing and  
Configuring HP  
FTAM/9000

*Installing and Administering HP  
FTAM/9000 (B1033-90034)*

Troubleshooting HP  
FTAM/9000

*OSI Troubleshooting Guide (32070-90020)*

FTAM Programming

*HP FTAM/9000 Programmer's Guide  
(B1033- 90014)*

FTAM Protocol  
Specifications

*ISO 8571, Information Processing Systems –  
Open Systems Interconnection – File  
Transfer, Access and Management*

International  
Standards ISO 8571

*ISO 8571, Information Processing Systems –  
Open Systems Interconnection – File  
Transfer, Access and Management*

MAP 3.0 Interface  
Specifications

*MAP 3.0 Application Interface Specification*

NBS Phase III

*Implementation Agreements for Open  
Systems Interconnection Protocols from the  
NBS Workshop for Implementors of Open  
Systems Interconnection*

ACSE

*ISO 8649, Information Processing Systems –  
Open Systems Interconnection – Service  
Definition for the Association Control Service  
Element*



---

# Contents

<b>1. FTAM return_codes</b>	
inout_dcb->result.return_code . . . . .	15
<b>2. HP vendor_codes</b>	
inout_dcb->result.vendor_code . . . . .	55
<b>3. Diagnostic error_ids</b>	
inout_dcb->diagnostic->error_id . . . . .	69
<b>4. Event Management Errors</b>	
EM return_codes . . . . .	83
EM vendor_codes . . . . .	86
result->return_code . . . . .	86
<b>5. HP FTAM/9000 PICS</b>	
<b>6. Installation Filesets</b>	
Files Created During Software Installation. . . . .	92
<b>7. HP FTAM/9000 Reference Pages</b>	

---

## **Contents**

---

# **1 FTAM return\_codes**

This chapter lists the `inout_dcb->result.return_codes`. The initiator and responder detect FTAM errors and return them in the user program as integer defined constants.

## FTAM return\_codes

- The `mapftam.h` file contains defined constants for FTAM return\_codes.
- To obtain a printable character string for the error, call `ft_gperror()`.

Refer to the “Handling Errors” chapter in the *HP FTAM/9000 Programmer's Guide* for information on checking for and handling errors.

---

## inout\_dcb->result.return\_code

MESSAGE	FTE000_SUCCESS
CAUSE	Successful function execution (no error).
ACTION	Informative message; no action required.
MESSAGE	FTE001_INV_EVENT_NAME
CAUSE	The <i>return_event_name</i> is a negative value and therefore, invalid.
ACTION	Call the function with a valid, positive <i>return_event_name</i> .
MESSAGE	FTE002_DUP_EVENT_NAME
CAUSE	You previously made an asynchronous call using the same value for <i>return_event_name</i> .
ACTION	Call the function asynchronously with a <i>return_event_name</i> that is not already in use.
MESSAGE	FTE003_ABORT_IND_RCVD
CAUSE	An abort indication was received on the <i>connection_id</i> specified in the call.
ACTION	Call <i>ft_ireceive()</i> to obtain abort information. Perform error recovery based on the information returned in either the <i>inout_dcb-&gt;info.pabort_info</i> or <i>inout_dcb-&gt;info.aabort_info</i> field. If the error still occurs, refer to the <i>OSI Troubleshooting Guide</i> .

FTAM return\_codes  
inout\_dcb->result.return\_code

---

MESSAGE	FTE004_BUFFER_OVERFLOW
CAUSE	The <i>inout_dcb</i> specified is not large enough to hold the output of the call.
ACTION	You have two options: <ul style="list-style-type: none"><li>• Call the function with the address of a NULL <i>inout_dcb</i>, thus requesting the interface to dynamically allocate the output area.</li><li>• Call the function with a larger buffer.</li></ul>
MESSAGE	FTE005_INV_CONN_ID
CAUSE	The <i>connection_id</i> from <i>ft_connect()</i> is no longer active or <i>ft_connect()</i> did not return the <i>connection_id</i> .
ACTION	Call the function with a valid, active <i>connection_id</i> returned from an <i>ft_connect()</i> request.
MESSAGE	FTE006_NETWORK_UNAVAIL
CAUSE	All contact with <i>ftam_init</i> was lost either because it was killed or exited.
ACTION	Call <i>ft_aeactivation()</i> to activate another <i>ftam_init</i> . If the error still occurs, terminate the application and refer to the <i>OSI Troubleshooting Guide</i> .
MESSAGE	FTE007_NO_REQ_RESOURCES
CAUSE	Current resources are insufficient to honor the request; virtual memory is exhausted.
ACTION	Check your memory management. Free any unused resources (e.g., memory), call <i>ft_dfcb()</i> to free DCBs that are no longer needed, and call the function again.



---

MESSAGE	FTE008_NO_CON_RESOURCES
CAUSE	No connection resources are available; generated on <i>ft_sdata()</i> when connection resources are temporarily exhausted.
ACTION	Call <i>ft_nwcleared()</i> ; when this request is noted (the resources are freed), call <i>ft_sdata()</i> again.
MESSAGE	FTE009_RELEASE_REQ_ISSUED
CAUSE	You called a function on a connection that had an un-noted release request.
ACTION	Call <i>ft_connect()</i> to establish a new connection and continue processing. When calling functions, always use an active <i>connection_id</i> returned from <i>ft_connect()</i> .
MESSAGE	FTE010_MY_DIR_NAME_UNKNOWN
CAUSE	The <i>my_dir_name</i> associated with <i>ftam_init</i> is unknown. Either it does not exist, or you incorrectly typed the name.
ACTION	Call <i>ft_aeactivation()</i> with a configured <i>my_dir_name</i> .
MESSAGE	FTE011_AE_INVOC_EXHAUSTED
CAUSE	You reached the maximum number of allowable AE invocations between the user program and <i>ftam_init</i> .
ACTION	You have three options. <ul style="list-style-type: none"><li>• Write programs to use fewer AE invocations.</li><li>• Finish processing on AEs and deactivate them.</li><li>• Reconfigure the system to support more AE invocations.</li></ul>

FTAM return\_codes  
inout\_dcb->result.return\_code

---

MESSAGE	FTE013_BAD_AE_LABEL
CAUSE	The <i>ae_label</i> was deactivated or not returned from an <i>ft_aeactivation()</i> request.
ACTION	Call the function with a valid, active <i>ae_label</i> returned from an <i>ft_aeactivation()</i> request.
MESSAGE	FTE015_FUNCTIONING_CONNECT
CAUSE	You called <i>ft_aedeactivation()</i> when <i>ftam_init</i> had one or more open connections.
ACTION	You have two options. Finish processing on the connection, <ul style="list-style-type: none"><li>• Call <i>ft_rrequest()</i> to release the connection, and call <i>ft_aedeactivation()</i> to deactivate <i>ftam_init</i>.</li><li>• Call <i>ft_aereset()</i> to abruptly abort all open connections and then call <i>ft_aedeactivation()</i> to deactivate <i>ftam_init</i></li></ul>
MESSAGE	FTE016_LACK_TARGET_INFO
CAUSE	Insufficient target address information; either the <i>called_dir_name</i> or <i>input_dcb-&gt;called_presentation_address</i> parameter is missing from <i>ft_connect()</i> . You must provide one or both. EXAMPLE: You called <i>ft_connect()</i> with a NULL <i>called_dir_name</i> and 0 in the <i>input_dcb-&gt;called_presentation_address.n_nsaps</i> parameter.
ACTION	Call <i>ft_connect()</i> with a valid <i>called_dir_name</i> or <i>input_dcb-&gt;called_presentation_address</i> parameter. If you provide both, <i>input_dcb-&gt;called_presentation_address</i> has precedence.

---

MESSAGE	FTE017_CONTEXT_NAME_MISSING
CAUSE	The context_name is missing. EXAMPLE: You called <i>ft_connect()</i> with a 0 (zero) in the <i>input_dcb-&gt;context_name.length</i> field.
ACTION	Call the function with the following FTAM <i>context_name</i> : 1 0 8571 1 1.
MESSAGE	FTE018_CALLED_DIR_NAME_NO_ADDR
CAUSE	On <i>ft_connect()</i> , the called_dir_name for the FTAM responder has no associated presentation address. Either <i>called_dir_name</i> is not configured on your network, or you incorrectly typed the name.
ACTION	Call <i>ft_connect()</i> with a configured <i>called_dir_name</i> that identifies the FTAM responder to which you are trying to connect.
MESSAGE	FTE019_CONNECT_EXHAUSTED
CAUSE	You reached the maximum number of allowable connections between ftam_init and the responder.
ACTION	You have three options. <ul style="list-style-type: none"><li>• Activate another ftam_init and call <i>ft_connect()</i> again.</li><li>• Finish processing on some of the open connections and re-use those connections.</li><li>• Finish processing on some of the open connections, call <i>ft_rrequest()</i> or <i>ft_abort()</i> to release or abort the connection, and call <i>ft_connect()</i> again.</li></ul>

FTAM return\_codes  
inout\_dcb->result.return\_code

---

MESSAGE	FTE021_USER_ABORTED_CONN
CAUSE	You attempted processing on a connection on which <i>ft_abort()</i> was requested, but not noted.
ACTION	Open another connection and call the function again. Attempt processing only on active connections.
MESSAGE	FTE024_USER_RELEASED_CONN
CAUSE	You attempted processing on a connection on which an <i>ft_rrequest()</i> was successfully completed.
ACTION	Call <i>ft_connect()</i> to establish a connection and perform the appropriate processing. Attempt processing only on active connections.
MESSAGE	FTE025_PROTOCOL_ERROR
CAUSE	A function call violated an FTAM protocol in one of the following ways. <ul style="list-style-type: none"><li>• You called a function outside the current regime.</li><li>• You called an illegal function within a group. EXAMPLE: You called <i>ft_rdata()</i> after receiving a data end indication.</li></ul>
ACTION	Ensure you are in the correct regime for calling the function. Call only functions that are allowed within the group.
MESSAGE	FTE026_INVALID_INPUT_DCB
CAUSE	Invalid <i>input_dcb</i> ; you passed a NULL pointer to a function with mandatory parameters in the <i>input_dcb</i> .
ACTION	You have two options. Set all mandatory <i>input_dcb</i> parameters and call the function again. Call <i>ft_didcb()</i> to automatically initialize the <i>input_dcb</i> .

---

MESSAGE	FTE027_INVALID_INOUT_DCB
CAUSE	You set the <i>inout_dcb</i> parameter address to NULL and passed this parameter to an FTAM function call.
ACTION	You have two options. <ul style="list-style-type: none"><li>• Pass a valid, non-NULL pointer to the <i>inout_dcb</i> pointer.</li><li>• Call <i>ft_didcb()</i> to set the <i>inout_dcb</i>, passing an <i>additional_size</i> large enough to hold all possible <i>inout_dcb</i> data. Pass the address of the pointer returned by <i>ft_didcb()</i> to the FTAM function.</li></ul>
MESSAGE	FTE028_NO_MIN_OUT_SPACE
CAUSE	You did not provide the minimum, required <i>inout_dcb</i> space. The <i>inout_dcb-&gt;size</i> must be large enough to return the size and result structure (12 bytes).
ACTION	You have two options. <ul style="list-style-type: none"><li>• Call the function with the address of a NULL <i>inout_dcb</i> buffer, indicating the interface should dynamically initialize the DCB.</li><li>• Call the function with a larger <i>inout_dcb</i> buffer (at least 12 bytes).</li></ul>
MESSAGE	FTE029_AABORT_IND_RCVD
CAUSE	A user abort indication arrived before the responder confirmed <i>ft_connect()</i> .
ACTION	Refer to the <i>OSI Troubleshooting Guide</i> .

FTAM return\_codes  
inout\_dcb->result.return\_code

---

**MESSAGE** FTE030\_APABORT\_IND\_RCVD  
**CAUSE** A provider abort indication arrived before the responder confirmed *ft\_connect()*.  
**ACTION** Refer to the troubleshooting manual for your protocol stack.

**MESSAGE** FTE031\_INVALID\_BUF\_PTR  
**CAUSE** Invalid buffer pointer; you provided a NULL pointer instead of a valid address. **EXAMPLE:** You called *ft\_connect()* and passed a NULL output *connection\_id* pointer.  
**ACTION** Call the function with a valid, non-NULL pointer.

**MESSAGE** FTE032\_IPC\_ERROR  
**CAUSE** General IPC error; communication between the user program and *ftam\_init* is lost.  
**ACTION** Refer to the troubleshooting *OSI Troubleshooting Guide*.

**MESSAGE** FTE034\_PROVIDER\_INIT\_ERROR  
**CAUSE** Cannot initialize *ftam\_init*.  
**ACTION** Refer to the *OSI Troubleshooting Guide*.

---

MESSAGE	FTE035_CONFIRMATION_FAILED
CAUSE	The FTAM responder detected an error and sent a negative confirmation to <i>ftam_init</i> .
ACTION	Examine the following <i>inout_dcb</i> fields for additional error information, and take action accordingly. <ul style="list-style-type: none"><li>• <i>result.vendor_code</i></li><li>• <i>diagnostic-&gt;error_id</i>•<i>diagnostic-&gt;further_details</i></li><li>• <i>action_result</i></li><li>• <i>state_result</i></li></ul>
MESSAGE	FTE036_INV_PRES_ADDRESS
CAUSE	The maximum number of selectors or nsaps was exceeded. EXAMPLE: You called <i>ft_connect()</i> and set the <i>input_dcb-&gt;called_presentation_address.n_nsaps</i> field to a value greater than 8.
ACTION	Call <i>ft_connect()</i> with valid fields within the presentation address ( <i>struct P_address</i> ). Max. <i>psap_selector</i> length = 16 octets Max. <i>ssap_selector</i> length = 16 octets Max. <i>tsap_selector</i> length = 32 octets Max. <i>nsap</i> length = 20 octets Max. number of <i>n_nsaps</i> = 8
MESSAGE	FTE037_SYSTEM_STATE_ERROR
CAUSE	Internal inconsistencies occurred.
ACTION	Refer to the <i>OSI Troubleshooting Guide</i> .
MESSAGE	FTE038_DIRECTORY_ERROR
CAUSE	Unable to access directory information.
ACTION	Refer to the <i>OSI Troubleshooting Guide</i> .

FTAM return\_codes  
inout\_dcb->result.return\_code

---

MESSAGE	FTE040_INV_MY_DIRNAME
CAUSE	The <i>my_dir_name</i> structure syntax is incorrect on <i>ft_aeactivation()</i> . EXAMPLE: You specified <i>my_dir_name</i> with a negative number of rdns.
ACTION	Call <i>ft_aeactivation()</i> with a syntactically correct <i>my_dir_name</i> . The <i>n</i> field in struct <i>Dir_dn</i> or struct <i>Dir_rdn</i> must be a positive value. If you provide an <i>attr_id</i> and <i>attr_value</i> in struct <i>Dir_ava</i> , each field must have a length greater than zero and the pointer set to a non-NULL value. If you do not provide these fields, the <i>length</i> must be zero.
MESSAGE	FTE041_INV_MY_AE_TITLE_OPT
CAUSE	Invalid <i>my_ae_title_option</i> parameter for <i>ft_aeactivation()</i> ; the specified value is not within the <i>Ae_title_option</i> enumeration.
ACTION	Call <i>ft_aeactivation()</i> with a syntactically correct <i>my_ae_title_option</i> .



---

MESSAGE	FTE042_INV_MY_AE_TITLE
CAUSE	Invalid <i>my_ae_title</i> parameter on <i>ft_aeactivation()</i> ; syntax is incorrect. EXAMPLE: You called <i>ft_aeactivation()</i> and specified <i>User_object_id_</i> option as the <i>my_ae_title_option</i> . You then passed in an invalid struct <i>Object_id</i> as the <i>ae_object_id</i> .
ACTION	Call <i>ft_aeactivation()</i> with a syntactically correct <i>my_ae_title</i> . <ul style="list-style-type: none"><li>• If the <i>ae_title</i> is a directory distinguished name (i.e., you supplied <i>ae_dir_dn</i>), the <i>n</i> field in struct <i>Dir_dn</i> or struct <i>Dir_rdn</i> must be a positive value. If you provide an <i>attr_id</i> and <i>attr_value</i> in struct <i>Dir_ava</i>, each field must have a length greater than zero and the pointer set to a non-NULL value.</li><li>• If the <i>ae_title</i> is an object ID (i.e., you supplied <i>ae_object_id</i>), the length must be greater than zero and the pointer must be a non-NULL value.</li></ul>
MESSAGE	FTE043_INV_CALLED_DIR_NAME
CAUSE	Invalid <i>called_dir_name</i> parameter on <i>ft_connect()</i> ; syntax is incorrect. EXAMPLE: You specified <i>called_dir_name</i> with a negative number of avas.
ACTION	Call <i>ft_connect()</i> with a syntactically correct <i>called_dir_name</i> . The <i>n</i> field in struct <i>Dir_dn</i> or struct <i>Dir_rdn</i> must be a positive value. If you provide an <i>attr_id</i> and <i>attr_value</i> in struct <i>Dir_ava</i> , each field must have a length greater than zero and the pointer set to a non-NULL value. If you do not provide these fields, the length must be zero.

FTAM return\_codes  
inout\_dcb->result.return\_code

---

MESSAGE	FTE044_INV_CALLED_AE_T_OPT
CAUSE	Invalid <i>called_ae_title_option</i> parameter; syntax is incorrect; the specified value is not within the <i>Ae_title_option</i> enumeration.
ACTION	Call <i>ft_connect()</i> with a syntactically correct <i>called_ae_title_option</i> .
MESSAGE	FTE045_INV_CALLED_AE_TITLE
CAUSE	Invalid <i>called_ae_title</i> parameter; syntax is incorrect. EXAMPLE: You called <i>ft_connect()</i> and specified <i>User_object_id_option</i> as the <i>called_ae_title_option</i> . You then passed an invalid struct <i>Object_id</i> as the <i>ae_object_id</i> .
ACTION	Call <i>ft_connect()</i> with a syntactically correct <i>my_ae_title</i> . <ul style="list-style-type: none"><li>• If the <i>ae_title</i> is a directory distinguished name (i.e., you supplied <i>ae_dir_dn</i>), the <i>n</i> field in struct <i>Dir_dn</i> or struct <i>Dir_rdn</i> must be a positive value. If you provide an <i>attr_id</i> and <i>attr_value</i> in struct <i>Dir_ava</i>, each field must have a length greater than zero and the pointer set to a non-NULL value.</li><li>• If the <i>ae_title</i> is an object ID (i.e., you supplied <i>ae_object_id</i>), the length must be greater than zero and the pointer must be a non-NULL value.</li></ul>
MESSAGE	FTE090_INV_DYNAMIC_MEM_PTR
CAUSE	You passed a pointer to <i>ft_fdmemory()</i> that was not allocated by <i>ft_gperror()</i> .
ACTION	Only pass pointers allocated by <i>ft_gperror()</i> to <i>ft_fdmemory()</i> .

---

MESSAGE	FTE092_INV_ADDITIONAL_SIZE
CAUSE	The cause may be one of the following. <ul style="list-style-type: none"><li>• You requested an <i>additional_size</i> for an <i>input_dcb</i>.</li><li>• You requested an <i>additional_size</i> for an <i>inout_dcb</i> that caused an allocation failure.</li></ul>
ACTION	<ul style="list-style-type: none"><li>• Call <i>ft_didcb()</i> with an <i>additional_size</i> for an <i>inout_dcb</i> only.</li><li>• Call <i>ft_didcb()</i> with a smaller <i>additional_size</i> for the <i>inout_dcb</i>.</li></ul>
MESSAGE	FTE094_INVALID_DCB_POINTER
CAUSE	You passed a pointer to <i>ft_dfdbc()</i> that was not allocated by <i>ft_didcb()</i> . EXAMPLE: You called <i>malloc()</i> to allocate memory for an <i>Ft_fcopy_out_dcb</i> and then called <i>ft_dfdbc()</i> to free the pointer.
ACTION	If you called <i>ft_didcb()</i> to allocate the pointer, call <i>ft_dfdbc()</i> to free it. If you called <i>malloc()</i> to allocate the pointer, call <i>free()</i> to free it.
MESSAGE	FTE095_NWC_REVOKED
CAUSE	The <i>ft_nwcleared()</i> request was revoked; you called a function (on a specified <i>connection_id</i> ) that had an un-noted <i>ft_nwcleared()</i> request. You will not know when the resource is free.
ACTION	Informative message; no action required.
MESSAGE	FTE096_INVALID_DCB_TYPE
CAUSE	The <i>dcb_type</i> requested in the call to <i>ft_didcb()</i> is nonexistent.
ACTION	Call <i>ft_didcb()</i> with a known <i>dcb_type</i> .

FTAM return\_codes  
inout\_dcb->result.return\_code

---

MESSAGE	FTE097_NO_SPACE_AVAILABLE
CAUSE	No memory available.
ACTION	Check your memory management. Free any unused resources (e.g., memory), call <i>ft_dfpcb()</i> to free DCBs that are no longer needed, and call the function again.
MESSAGE	FTE098_INVALID_RETURN_CODE
CAUSE	The <i>result-&gt;return_code</i> passed to <i>ft_gperror()</i> is invalid or non-existent; it has no associated error string.
ACTION	Call <i>ft_gperror()</i> with a valid, defined <i>return_code</i> .
MESSAGE	FTE099_NO_RESOURCE_OUTSTANDING
CAUSE	No resource outstanding; you called <i>ft_nwcleared()</i> without previously receiving the <i>FTE008_NO_CON_RESOURCES</i> error. The <i>ft_nwcleared()</i> function is valid only if an <i>FTE008_NO_CON_RESOURCES</i> error returns on an <i>ft_sdata()</i> function.
ACTION	Call <i>ft_nwcleared()</i> only when you receive a <i>FTE008_NO_CON_RESOURCES</i> error.
MESSAGE	FTE101_BUFF_TOO_BIG
CAUSE	Buffer holding user data to send ( <i>ft_sdata()</i> ) is too large for implementation (more than 7K or 14 <i>data_units</i> ).
ACTION	Call multiple <i>ft_sdata()</i> functions with smaller or fewer buffers, or as smaller number of <i>data_units</i> .

---

MESSAGE	FTE102_BUFF_TOO_SMALL
CAUSE	The <i>inout_dcb</i> on an <i>ft_rdata()</i> request is not large enough to hold the output of the call or the <i>des_requested</i> parameter is greater than 12.
ACTION	Exit the Data Transfer regime to the File Open regime. Call <i>ft_read()</i> and <i>ft_rdata()</i> . Ensure all <i>inout_dcbs</i> are large enough to hold all possible data or let the interface allocate the <i>inout_dcb</i> .
MESSAGE	FTE104_CONFIRM_NOT_RCVD
CAUSE	The responder did not confirm this grouped request; an error on a previous request in the group was detected, and the end group confirm was immediately generated. EXAMPLE: You called the following sequence of functions: <i>ft_bgroup()</i> , <i>ft_select()</i> , <i>ft_open()</i> , and <i>ft_egroup()</i> . After these passed across the network, the responder detected an error on <i>ft_select()</i> . The <i>ft_open()</i> was not processed, and the <i>ft_egroup()</i> confirm was sent. This error would return on <i>ft_open()</i> .
ACTION	The error probably occurred on a previously confirmed request within the group. Examine the following <i>inout_dcb</i> fields for previously confirmed requests and take action accordingly. <i>result.vendor_codediagnostic-&gt;error_iddiagnostic-&gt;further_detailsaction_resultstate_result</i> EXAMPLE: Examine the <i>inout_dcb</i> for <i>ft_select()</i> to determine the cause of the error.

FTAM return\_codes  
inout\_dcb->result.return\_code

---

MESSAGE	FTE105_CONT_TYPE_UNSupport
CAUSE	The specified contents type is not supported (i.e., is not an FTAM-1, FTAM-2, FTAM-3, or NBS-9 document type). EXAMPLE: You called <i>ft_open()</i> with a <i>contents_type.contents_form</i> of <i>FT_ABS_SYN_CON_SET_PAIR_FORM</i> .
ACTION	Call the function with a supported <i>contents_type</i> . For all functions requiring a <i>contents_type</i> , ensure <i>contents_type.contents_form</i> = <i>FT_DOCUMENT_TYPE</i> and <i>contents_type.contents_info.document.name</i> indicates FTAM-1, FTAM-2, FTAM-3, or NBS-9. For <i>ft_fopen()</i> and <i>ft_open()</i> , <i>contents_type.contents_form</i> can be <i>FT_CONTENTS_UNKNOWN</i> .
MESSAGE	FTE112_GROUP_NOT_OPEN
CAUSE	Grouping was not open on the connection. You called <i>ft_egroup()</i> before calling <i>ft_bgroup()</i> .
ACTION	Call <i>ft_bgroup()</i> on a connection before calling <i>ft_egroup()</i> .
MESSAGE	FTE113_GROUP_ALREADY_OPEN
CAUSE	Grouping was already open on the connection. You called <i>ft_bgroup()</i> on a connection on which you already called <i>ft_bgroup()</i> .
ACTION	Close the group by calling <i>ft_egroup()</i> . Open another group with an <i>ft_bgroup()</i> request.

---

MESSAGE	FTE114_INV_ACCESS_CNTL
CAUSE	Invalid bits are set in the <i>action_list</i> field of struct <i>Ft_access_control_element</i> . EXAMPLE: You set Bit (1) in the parameter, but only Bits (8) to (15) are valid for the <i>action_list</i> .
ACTION	Correctly set the <i>FT_FA_XXX</i> bits and call the function again.
MESSAGE	FTE115_INV_ACCTL_ID
CAUSE	Invalid identity field of struct <i>Ft_access_control_element</i> ; syntax is incorrect.
ACTION	Ensure characters in the parameter are part of the graphic string character set. These ordinal values are 32 to 126 and 160 to 255. If used within escape sequences, the values 27, 142, and 143 are also available.
MESSAGE	FTE116_INV_ACCTL_PASSWDS
CAUSE	Invalid password in <i>access_passwords</i> field of struct <i>Ft_access_control_element</i> ; syntax is incorrect.
ACTION	Call the function with a valid <i>Ft_single_file_pw</i> . A password is invalid if the length field is greater than zero and the pointer is NULL or if the length is greater than 65535.
MESSAGE	FTE118_INV_ACCOUNT
CAUSE	Invalid account parameter; syntax is incorrect.
ACTION	Ensure characters in the parameter are part of the graphic string character set. These ordinal values are 32 to 126 and 160 to 255. If used within escape sequences, the values 27, 142, and 143 are also available.

FTAM return\_codes  
inout\_dcb->result.return\_code

---

**MESSAGE** FTE119\_INV\_ACCS\_CONTEXT

**CAUSE** Invalid *access\_context* parameter for *ft\_read()*; the specified value is not within the *Ft\_access\_context* enumeration.

**ACTION** Call *ft\_read()* with a valid *access\_context*.

  

**MESSAGE** FTE120\_INV\_ACTION\_RESULT

**CAUSE** Invalid *action\_result* parameter for *ft\_abort()*, *ft\_edata()*, *ft\_cancel()*, or *ft\_rcancel()*; the specified value is not within the *Ft\_action\_result* enumeration.

**ACTION** Call the function with a valid *action\_result*.

  

**MESSAGE** FTE121\_INV\_DIRNAME

**CAUSE** Invalid *dirname* parameter for *ft\_frattributes()*, *ft\_fcattributes()*, or *ft\_fdelete()*; syntax is incorrect. **EXAMPLE:** You specified *dirname* with a negative number of *avas*.

**ACTION** Call the function with a syntactically correct *dir\_name*. The *n* field in struct *Dir\_dn* or struct *Dir\_rdn* must be a positive value. If you provide an *attr\_id* and *attr\_value* in struct *Dir\_ava*, each field must have a length greater than zero and the pointer set to a non-NULL value. If you do not provide these fields, the length must be zero.



---

MESSAGE	FTE122_INV_ATTR_FILENAME
CAUSE	Invalid filename parameter for <i>ft_cattributes()</i> or <i>ft_fcattributes()</i> ; syntax is incorrect.
ACTION	Ensure characters in the parameter are part of the graphic string character set. These ordinal values are 32 to 126 and 160 to 255. If used within escape sequences, the values 27, 142, and 143 are also available.
MESSAGE	FTE125_INV_CONCUR_CNTL
CAUSE	One of the <i>Ft_concurrency_control</i> fields contains an invalid file lock. The file lock is not in the <i>Ft_file_lock</i> enumeration.
ACTION	Call the function with a valid file lock in each <i>concurrency_control</i> field.
MESSAGE	FTE127_INV_CREATE_ID
CAUSE	Invalid <i>identity_of_creator</i> parameter; syntax is incorrect.
ACTION	Ensure characters in the parameter are part of the graphic string character set. These ordinal values are 32 to 126 and 160 to 255. If used within escape sequences, the values 27, 142, and 143 are also available.
MESSAGE	FTE128_INV_DELETE_ACT
CAUSE	Invalid <i>delete_action</i> parameter for <i>ft_fclose()</i> ; the specified value is not within the <i>Ft_delete_action</i> enumeration.
ACTION	Call <i>ft_fclose()</i> with a valid <i>delete_action</i> .

FTAM return\_codes  
inout\_dcb->result.return\_code

---

MESSAGE	FTE129_INV_DEST_ACCOUNT
CAUSE	Invalid <i>input_dcb-&gt;dest_account</i> parameter for <i>ft_fmove()</i> or <i>ft_fcopy()</i> ; syntax is incorrect.
ACTION	Ensure characters in the parameter are part of the graphic string character set. These ordinal values are 32 to 126 and 160 to 255. If used within escape sequences, the values 27, 142, and 143 are also available.
MESSAGE	FTE130_INV_DEST_DIRNAME
CAUSE	Invalid <i>destination_dirname</i> parameter on <i>ft_fmove()</i> or <i>ft_fcopy()</i> ; syntax is incorrect. EXAMPLE: You called <i>ft_fmove()</i> and specified <i>destination_dirname</i> with a negative number of rdns.
ACTION	Call the function with a syntactically correct <i>destination_dirname</i> . The <i>n</i> field in struct <i>Dir_dn</i> or struct <i>Dir_rdn</i> must be a positive value. If you provide an <i>attr_id</i> and <i>attr_value</i> in struct <i>Dir_ava</i> , each field must have a length greater than zero and the pointer set to a non-NULL value. If you do not provide these fields, the length must be zero.
MESSAGE	FTE131_INV_DEST_FNAME
CAUSE	Invalid <i>destination_filename</i> parameter; syntax is incorrect.
ACTION	Ensure characters in the parameter are part of the graphic string character set. These ordinal values are 32 to 126 and 160 to 255. If used within escape sequences, the values 27, 142, and 143 are also available.

---

MESSAGE	FTE132_INV_DEST_INIT_ID
CAUSE	Invalid <i>input_dcb-&gt;dest_init_id</i> parameter for <i>ft_fmove()</i> or <i>ft_fcopy()</i> ; syntax is incorrect.
ACTION	Ensure characters in the parameter are part of the graphic string character set. These ordinal values are 32 to 126 and 160 to 255. If used within escape sequences, the values 27, 142, and 143 are also available.
MESSAGE	FTE133_INV_DIAG_STRUCT
CAUSE	Invalid struct <i>Ft_diagnostic</i> structure; syntax is incorrect.
ACTION	Call the function with syntactically correct fields within struct <i>Ft_diagnostic</i> . Ensure you set all required fields. <ul style="list-style-type: none"><li>• The <i>error_source</i> and <i>error_observer</i> must be in the <i>Ft_entity_ref</i> enumeration.</li><li>• The <i>diag_type</i> must be in the <i>Ft_diag_type</i> enumeration.</li><li>• The <i>further_details</i> characters must be part of the graphic string character set. These ordinal values are 32 to 126 and 160 to 255. If used within escape sequences, the values 27, 142, and 143 are also available.</li></ul>
MESSAGE	FTE135_INV_FILENAME
CAUSE	Invalid filename parameter; syntax is incorrect.
ACTION	Ensure characters in the parameter are part of the graphic string character set. These ordinal values are 32 to 126 and 160 to 255. If used within escape sequences, the values 27, 142, and 143 are also available.

FTAM return\_codes  
inout\_dcb->result.return\_code

---

MESSAGE	FTE136_INV_FILE_STATUS
CAUSE	Invalid <i>file_status</i> parameter for <i>ft_create()</i> or <i>ft_fopen()</i> ; the specified value is not within the <i>Ft_file_status</i> enumeration.
ACTION	Call the function with a valid <i>file_status</i> .
MESSAGE	FTE137_INV_INITIATOR_ID
CAUSE	Invalid <i>init_id</i> or <i>initiator_identity</i> parameter; syntax is incorrect.
ACTION	Ensure characters in the parameter are part of the graphic string character set. These ordinal values are 32 to 126 and 160 to 255. If used within escape sequences, the values 27, 142, and 143 are also available.
MESSAGE	FTE138_INV_LEGAL_QUAL
CAUSE	Invalid <i>legal_qualification</i> parameter; syntax is incorrect.
ACTION	Ensure characters in the parameter are part of the graphic string character set. These ordinal values are 32 to 126 and 160 to 255. If used within escape sequences, the values 27, 142, and 143 are also available.
MESSAGE	FTE142_INV_PERM_ACTS
CAUSE	The bits for the <i>FT_PA_XXX</i> defined constants are not set correctly in the <i>permitted_actions</i> parameter. EXAMPLE: You set Bit (1) in the parameter, but only Bits (5) to (15) are valid.
ACTION	Correctly set the <i>FT_PA_XXX</i> bits and call the function again.

MESSAGE	FTE143_INV_PROC_MODE
CAUSE	Invalid bits are set in the <i>processing_mode</i> parameter. EXAMPLE: You set Bit (1) in the parameter, but only Bits (11) to (15) are valid.
ACTION	Call <i>ft_open()</i> with valid bits set in the <i>processing_mode</i> parameter.
MESSAGE	FTE148_INV_REQ_ACCESS
CAUSE	Invalid bits are set in the <i>requested_access</i> parameter for <i>ft_create()</i> , <i>ft_fopen()</i> , or <i>ft_select()</i> . EXAMPLE: You set Bit (1) in the parameter, but only Bits (8) to (15) are valid for the <i>requested_access</i> .
ACTION	Correctly set the <i>FT_FA_XXX</i> bits for the <i>requested_access</i> parameter and call the function again.
MESSAGE	FTE149_INV_SRC_ACCOUNT
CAUSE	Invalid <i>input_dcb-&gt;source_account</i> parameter for <i>ft_fcopy()</i> or <i>ft_fmmove()</i> .
ACTION	Ensure characters in the parameter are part of the graphic string character set. These ordinal values are 32 to 126 and 160 to 255. If used within escape sequences, the values 27, 142, and 143 are also available.

FTAM return\_codes  
inout\_dcb->result.return\_code

---

MESSAGE	FTE150_INV_SRC_DIRNAME
CAUSE	Invalid <i>source_dirname</i> parameter for <i>ft_copy()</i> or <i>ft_move()</i> ; syntax is incorrect. EXAMPLE: You specified <i>source_dirname</i> with a negative number of rdns.
ACTION	Call the function with a syntactically correct <i>source_dirname</i> . The <i>n</i> field in struct <i>Dir_dn</i> or struct <i>Dir_rdn</i> must be a positive value. If you provide an <i>attr_id</i> and <i>attr_value</i> in struct <i>Dir_ava</i> , each field must have a length greater than zero and the pointer set to a non-NULL value. If you do not provide these fields, the length must be zero.
MESSAGE	FTE151_INV_SRC_FNAME
CAUSE	Invalid <i>source_filename</i> parameter; syntax is incorrect.
ACTION	Ensure characters in the parameter are part of the graphic string character set. These ordinal values are 32 to 126 and 160 to 255. If used within escape sequences, the values 27, 142, and 143 are also available.
MESSAGE	FTE152_INV_SRC_INIT_ID
CAUSE	Invalid <i>input_dcb-&gt;source_init_id</i> parameter for <i>ft_copy()</i> or <i>ft_move()</i> ; syntax is incorrect.
ACTION	Ensure characters in the parameter are part of the graphic string character set. These ordinal values are 32 to 126 and 160 to 255. If used within escape sequences, the values 27, 142, and 143 are also available.

---

MESSAGE	FTE153_INV_STORE_ACCOUNT
CAUSE	Invalid <i>storage_account</i> parameter; syntax is incorrect.
ACTION	Ensure characters in the parameter are part of the graphic string character set. These ordinal values are 32 to 126 and 160 to 255. If used within escape sequences, the values 27, 142, and 143 are also available.
MESSAGE	FTE154_INV_STRUCT_ID
CAUSE	Invalid <i>structure_id</i> parameter; the specified value is not within the <i>Ft_structure_id</i> enumeration.
ACTION	Call <i>ft_sdata()</i> with a valid <i>structure_id</i> .
MESSAGE	FTE155_NO_SYNCH_IN_GROUP
CAUSE	You made a synchronous call within a group; you called a function with a synchronous <i>return_event_name</i> while grouping was open on the specified connection. EXAMPLE: You called <i>ft_bgroup()</i> on a connection and then called <i>ft_select()</i> synchronously on that same connection.
ACTION	Call the function with an asynchronous <i>return_event_name</i> .
MESSAGE	FTE156_INV_FUNC_UNITS
CAUSE	The bits for the <i>FT_FU_XXX</i> defined constants are not set correctly in the <i>input_dcb-&gt;connect_in_info.functional_units</i> parameter for <i>ft_connect()</i> . EXAMPLE: You set Bit (14) in the parameter, but only Bits (5) to (13) are valid.
ACTION	Correctly set the <i>FT_FU_XXX</i> bits for the <i>input_dcb-&gt;connect_in_info.functional_units</i> parameter and call <i>ft_connect()</i> again.

FTAM return\_codes  
inout\_dcb->result.return\_code

---

**MESSAGE** FTE157\_NULL\_BUFF\_PTR  
**CAUSE** Pointer to data buffer is NULL. EXAMPLE: You called *ft\_sdata()* and passed a NULL pointer in for the *data\_unit*.  
**ACTION** Call *ft\_sdata()* with a non-NULL buffer pointer.

**MESSAGE** FTE158\_INV\_CONT\_TYPE  
**CAUSE** Invalid *contents\_type.contents\_info.document.name* parameter; invalid struct *Octet\_string*.  
**ACTION** Call the function with a valid *Ft\_contents\_type* structure. The struct *Octet\_string* is invalid if the length field is greater than zero and the pointer is NULL or if the length is greater than 65535.

**MESSAGE** FTE159\_INV\_FADU\_ID  
**CAUSE** Invalid *fadu\_identity.fadu\_form* or *fadu\_identity.fadu\_info* parameter for *ft\_read()*, *ft\_write()*, *ft\_erase()*, or *ft\_locate()*.  
**ACTION** Call the function with *fadu\_identity.fadu\_form* equal to *FT\_FADU\_LOCATION*. Additionally, the *fadu\_identity.fadu\_info* must be a *fadu\_location* that is in the *Ft\_fadu\_location* enumeration.

**MESSAGE** FTE160\_INV\_FADU\_OP  
**CAUSE** Invalid *fadu\_operation* parameter for *ft\_write()*; the specified value is not within the *Ft\_fadu\_operation* enumeration.  
**ACTION** Call *ft\_write()* with a valid *fadu\_operation*.



---

MESSAGE	FTE161_NULL_FILENAME
CAUSE	You passed a null pointer to filename on a function that requires filename to have a non-NULL value.
ACTION	Call the function with a valid, non-NULL filename.
MESSAGE	TE162_NULL_SRC_FNAME
CAUSE	You called <i>ft_fcopy()</i> or <i>ft_fmove()</i> with a NULL <i>source_filename</i> .
ACTION	Call the function with a valid, non-NULL <i>source_filename</i> .
MESSAGE	FTE163_SAME_FILE
CAUSE	Duplicate filenames when source and destination are identical; you called <i>ft_fcopy()</i> or <i>ft_fmove()</i> , but specified the same source and destination file-store and the same source and destination file.
ACTION	Call the function using different source and destination files.
MESSAGE	FTE166_UNKN_ATTR_GROUP
CAUSE	The bits for the <i>FT_AG_XXX</i> defined constants are not set correctly in the <i>input_dcb-&gt;connect_in_info.attribute_groups</i> parameter for <i>ft_connect()</i> . EXAMPLE: You set Bit (1) in the <i>attribute_group</i> field, but only Bits (13) to (15) are valid.
ACTION	Correctly set the <i>FT_AG_XXX</i> bits for the <i>input_dcb-&gt;connect_in_info.attribute_groups</i> parameter and call <i>ft_connect()</i> again.

FTAM return\_codes  
inout\_dcb->result.return\_code

---

MESSAGE	FTE167_UNKN_SERV_CLASS
CAUSE	Unknown <i>input_dcb-&gt;connect_in_info.service_class</i> parameter; the bits for the <i>FT_SC_XXX</i> defined constants were either invalid or not set. EXAMPLE: You set Bit (1) in the <i>input_dcb-&gt;connect_in_info.service_class</i> parameter, but only Bits (11) to (15) are valid.
ACTION	Call <i>ft_connect()</i> with a valid <i>FT_SC_XXX</i> bits set for the <i>input_dcb-&gt;connect_in_info.service_class</i> parameter.
MESSAGE	FTE169_INV_PRIVATE_USE
CAUSE	Invalid <i>private_use</i> parameter; invalid struct <i>Octet_string</i> .
ACTION	Call the function with a valid <i>private_use</i> . The struct <i>Octet_string</i> is invalid if the length field is greater than zero and the pointer is NULL or if the length is greater than 65535.
MESSAGE	FTE170_INV_FUTURE_FILESIZE
CAUSE	The specified <i>future_filesize</i> parameter is less than zero.
ACTION	Call the function with a positive <i>future_filesize</i> .
MESSAGE	FTE171_INV_FILE_AVAILABILITY
CAUSE	Invalid <i>file_availability</i> parameter; the specified value is not within the <i>Ft_file_availability</i> enumeration.
ACTION	Call the function with a valid <i>file_availability</i> .

---

MESSAGE	FTE172_INV_THRESHOLD
CAUSE	The threshold parameter for <i>ft_bgroup()</i> exceeds the maximum number of calls allowed within a group.
ACTION	Call <i>ft_bgroup()</i> with a valid threshold value. The threshold cannot be zero or greater than five.
MESSAGE	FTE173_INV_FTQOS
CAUSE	Invalid <i>input_dcb-&gt;connect_in_info.quality_of_service</i> parameter; the specified value is not within the <i>Ft_qos</i> enumeration.
ACTION	Call <i>ft_connect()</i> with a valid <i>input_dcb-&gt;connect_in_info.quality_of_service</i> .
MESSAGE	FTE174_ATTRIB_UNMODIFIABLE
CAUSE	A bit is set in the attributes mask that cannot be modified or set for the specified call. EXAMPLE: You called <i>ft_cattributes()</i> and specified the <i>contents_type</i> as an attribute you want to change, but <i>contents_type</i> is not modifiable
ACTION	Call the function with valid mask bits set. Ensure you can change the specified attribute for the given function.
MESSAGE	FTE175_FCANCEL_IND_RECEIVED
CAUSE	While in the process of sending ( <i>ft_sdata()</i> ) or receiving data ( <i>ft_rdata()</i> ), the responder detected an error and sent a cancel indication to the <i>ftam_init</i> .
ACTION	Examine the cancel indication data and respond by calling <i>ft_rcancel()</i> .

FTAM return\_codes  
inout\_dcb->result.return\_code

---

MESSAGE	FTE176_INV_OVERWRITE
CAUSE	Invalid <i>input_dcb-&gt;overwrite</i> parameter for <i>ft_fcopy()</i> or <i>ft_fmove()</i> ; the specified value is not within the <i>Ft_delete_overwrite</i> enumeration.
ACTION	Call the function with a valid <i>input_dcb-&gt;overwrite</i> .
MESSAGE	FTE177_INV_CLASS
CAUSE	Invalid class parameter; the specified value is not within the <i>Ft_class</i> enumeration.
ACTION	Call the function with a valid class.
MESSAGE	FTE178_INV_STRING_SIGN
CAUSE	Invalid <i>string_significance</i> parameter; the specified value is not within the <i>Ft_string_significance</i> enumeration.
ACTION	Call the function with a valid <i>string_significance</i> .
MESSAGE	FTE181_INV_DE_PRIM_TYPE
CAUSE	Invalid <i>data_unit-&gt;data</i> . <i>data_element-&gt;prim_type</i> parameter for <i>ft_sdata()</i> ; the <i>data_unit-&gt;data</i> . <i>data_element-&gt;prim_type</i> specified in the <i>Ft_data_element</i> structure is nonexistent. The <i>prim_type</i> must be in the <i>Ft_prim_type</i> enumeration.
ACTION	Call <i>ft_sdata()</i> with a valid <i>data_unit-&gt;data</i> . <i>data_element-&gt;prim_type</i> .

---

MESSAGE	FTE182_INV_DE_PRIMITIVE
CAUSE	Invalid <i>data_unit-&gt;data.data_element-&gt;primitive</i> for <i>ft_sdata()</i> ; <i>octet_string</i> syntax is incorrect. EXAMPLE: You specified <i>FT_DE_OCTET_STRING</i> as the <i>data_unit-&gt;data.data_element-&gt; prim_type</i> and then specified an invalid <i>octet_string</i> .
ACTION	Call <i>ft_sdata()</i> with a valid <i>data_unit-&gt;data.data_element-&gt; primitive</i> . The <i>octet_string</i> is invalid if the length field is greater than zero and the pointer is NULL or if the length is greater than 65535.
MESSAGE	FTE183_INV_READ_PASSWD
CAUSE	Invalid read password; syntax is incorrect.
ACTION	Call the function with a valid <i>Ft_single_file_pw</i> . A password is invalid if the length field is greater than zero and the pointer is NULL or if the length is greater than 65535.
MESSAGE	FTE184_INV_INSERT_PASSWD
CAUSE	Invalid insert password; syntax is incorrect.
ACTION	Call the function with a valid <i>Ft_single_file_pw</i> . A password is invalid if the length field is greater than zero and the pointer is NULL or if the length is greater than 65535.
MESSAGE	FTE185_INV_REPLACE_PASSWD
CAUSE	Invalid replace password; syntax is incorrect.
ACTION	Call the function with a valid <i>Ft_single_file_pw</i> . A password is invalid if the length is greater than zero and the pointer non- NULL or if the length is greater than 65535.

FTAM return\_codes  
inout\_dcb->result.return\_code

---

**MESSAGE** FTE186\_INV\_EXTEND\_PASSWD  
**CAUSE** Invalid extend password; syntax is incorrect.  
**ACTION** Call the function with a valid *Ft\_single\_file\_pw*. A password is invalid if the length field is greater than zero and the pointer is NULL or if the length is greater than 65535.

**MESSAGE** FTE187\_INV\_ERASE\_PASSWD  
**CAUSE** Invalid erase password; syntax is incorrect.  
**ACTION** Call the function with a valid *Ft\_single\_file\_pw*. A password is invalid if the length field is greater than zero and the pointer is NULL or if the length is greater than 65535.

**MESSAGE** FTE188\_INV\_READATTR\_PASSWD  
**CAUSE** Invalid *read\_attribute* password; syntax is incorrect.  
**ACTION** Call the function with a valid *Ft\_single\_file\_pw*. A password is invalid if the length field is greater than zero and the pointer is NULL or if the length is greater than 65535.

**MESSAGE** FTE189\_INV\_CHANGEATTR\_PASSWD  
**CAUSE** Invalid *change\_attribute* password; syntax is incorrect.  
**ACTION** Call the function with a valid *Ft\_single\_file\_pw*. A password is invalid if the length field is greater than zero and the pointer is NULL or if the length is greater than 65535.

---

MESSAGE FTE190\_INV\_DELETE\_PASSWD  
CAUSE Invalid *delete\_file* password; syntax is incorrect.  
ACTION Call the function with a valid *Ft\_single\_file\_pw*. A password is invalid if the length field is greater than zero and the pointer is NULL or if the length is greater than 65535.

MESSAGE FTE191\_INV\_FILESTORE\_PASSWD  
CAUSE Invalid *filestore\_pw* password; syntax is incorrect.  
ACTION Call the function with a valid *Ft\_single\_file\_pw*. A password is invalid if the length field is greater than zero and the pointer is NULL or if the length is greater than 65535.

MESSAGE FTE192\_INV\_CREATE\_PASSWD  
CAUSE Invalid *create\_file\_pw* password; syntax is incorrect.  
ACTION Call the function with a valid *Ft\_single\_file\_pw*. A password is invalid if the length field is greater than zero and the pointer is NULL or if the length is greater than 65535.

MESSAGE FTE193\_INV\_SRCFVS\_PASSWD  
CAUSE Invalid *source\_filestore\_pw* password; syntax is incorrect.  
ACTION Call the function with a valid *Ft\_single\_file\_pw*. A password is invalid if the length field is greater than zero and the pointer is NULL or if the length is greater than 65535.

FTAM return\_codes  
inout\_dcb->result.return\_code

---

MESSAGE FTE194\_INV\_DESTFS\_PASSWD  
CAUSE Invalid *dest\_filestore\_pw* password; syntax is incorrect.  
ACTION Call the function with a valid *Ft\_single\_file\_pw*. A password is invalid if the length field is greater than zero and the pointer is NULL or if the length is greater than 65535.

MESSAGE FTE196\_DIRNAME\_NO\_ADDR  
CAUSE The *dirname* parameter for *ft\_fdelete()*, *ft\_frattributes()*, or *ft\_fcattributes()* has no associated presentation address. Either it is not configured for your network, or you incorrectly typed the name.  
ACTION Call the function with a configured *dirname* that identifies the desired FTAM responder.

MESSAGE FTE197\_DEST\_DIRNAME\_NO\_ADDR  
CAUSE The *destination\_dirname* parameter for *ft\_fcopy()* or *ft\_fmove()* has no associated presentation address. Either it is not configured for your network, or you incorrectly typed the name.  
ACTION Call the function with a configured *destination\_dirname* that identifies the desired FTAM responder.

MESSAGE FTE198\_SRC\_DIRNAME\_NO\_ADDR  
CAUSE The *source\_dirname* parameter for *ft\_fcopy()* or *ft\_fmove()* has no associated presentation address. Either it is not configured for your network, or you incorrectly typed the name.  
ACTION Call the function with a configured *source\_dirname* that identifies the desired FTAM responder.



**MESSAGE** FTE199\_INV\_DU\_NODE\_DESC  
**CAUSE** You called *ft\_sdata()* with an invalid *node\_descriptor.node\_name* of type struct *Octet\_string*.  
**ACTION** Call *ft\_sdata()* with a valid *node\_descriptor*. The struct *Octet\_string* is invalid if the length field is greater than zero and the pointer is NULL or if the length is greater than 65535.

**MESSAGE** FTE200\_USR\_CANC\_DATA\_TRAN  
**CAUSE** You canceled the data transfer. This message returns on *ft\_rdata()* when you call *ft\_cancel()*, a positive confirmation returns, and *ft\_rdata()* requests are pending.  
**ACTION** Informative message; no action required.

**MESSAGE** FTE201\_INV\_SRC\_CONCUR\_CNTL  
**CAUSE** The *input\_dcb->src\_concur\_cntl* parameter is invalid for *ft\_fcopy()* or *ft\_fmove()*.  
**ACTION** Call the function with a valid file lock in each *input\_dcb->src\_concur\_cntl* field.

**MESSAGE** FTE202\_INV\_DEST\_CONCUR\_CNTL  
**CAUSE** The *input\_dcb->dest\_concur\_cntl* parameter is invalid for *ft\_fcopy()* or *ft\_fmove()*.  
**ACTION** Call the function with a valid file lock in each *input\_dcb->dest\_concur\_cntl* field.

FTAM return\_codes  
inout\_dcb->result.return\_code

---

**MESSAGE** FTE501\_ABNORMAL\_TERM  
**CAUSE** One of the connections on a HLCF call was terminated abnormally.  
**ACTION** Refer to the *OSI Troubleshooting Guide*.

**MESSAGE** FTE502\_LOC\_RES\_UNAVAIL  
**CAUSE** Required resources are not obtainable for a HLCF call; generated when all connections for the given *ftam\_init* are exhausted; returns on HLCF calls.  
**ACTION** Refer to the *OSI Troubleshooting Guide*.

**MESSAGE** FTE503\_REM\_SYS\_UNSUITABLE  
**CAUSE** A remote system negotiated below the minimal *functional\_units* required to complete the request; returns on HLCF calls.  
**ACTION** Refer to the *OSI Troubleshooting Guide*.

**MESSAGE** FTE504\_TRANSFER\_CANCEL  
**CAUSE** The responder canceled the data transfer on a HLCF call.  
**ACTION** Refer to the *OSI Troubleshooting Guide*.

**MESSAGE** FTE505\_TRANSFER\_DEST\_ABORT  
**CAUSE** The responder on the destination node aborted the data transfer on a HLCF call.  
**ACTION** Refer to the *OSI Troubleshooting Guide*.

---

MESSAGE	FTE506_TRANSFER_ERROR
CAUSE	The responder ended the data transfer on a HLCF call; indicates an unsuccessful end data indication arrived at one of the nodes.
ACTION	Refer to the <i>OSI Troubleshooting Guide</i> .
MESSAGE	FTE507_TRANSFER_SRC_ABORT
CAUSE	The responder on the source node aborted the data transfer on a HLCF call.
ACTION	Refer to the <i>OSI Troubleshooting Guide</i> .

FTAM return\_codes  
inout\_dcb->result.return\_code

---

## **2** **HP vendor\_codes**

This chapter contains a list of HP-specific errors returned in the `inout_dcb->result.vendor_code` field. The `vendor_codes` reflect HP-specific error information; you do not receive errors specific to other vendors in this field.

HP vendor\_codes

- The `mapftam.h` file contains defined constants for HP vendor\_codes.
- To receive a printable character string, call `ft_gperror()`.

Refer to the “Handling Errors” chapter in the *HP FTAM/9000 Programmer's Guide* for information on checking for and handling errors.

---

## **inout\_dcb->result.vendor\_code**

MESSAGE	FTV000_NO_ADDL_INFO
CAUSE	No additional information is available
ACTION	Informative message; no action required.
MESSAGE	FTV101_UNABLE_TO_RCV_IPC_CON
CAUSE	Unable to establish interprocess communication (IPC) connection with <i>ftam_init</i> .
ACTION	Collect relevant error information and contact your HP support representative.
MESSAGE	FTV102_CANNOT_KILL_INITIATOR
CAUSE	Unable to deactivate <i>ftam_init</i> .
ACTION	The <i>kill()</i> function failed on <i>ftam_init</i> . Check the log file for errors logged by <i>ftam_init</i> . Find <i>ftam_init</i> 's process ID (PID) and kill the <i>ftam_init</i> manually.
MESSAGE	FTV103_INTERNAL_ERROR
CAUSE	An internal error occurred.
ACTION	Check the log file for logged error messages. Collect relevant error information and contact your HP support representative.

HP vendor\_codes  
inout\_dcb->result.vendor\_code

---

MESSAGE	FTV104_CANNOT_EXEC_INITIATOR
CAUSE	Cannot execute <i>ftam_init</i>
ACTION	<ul style="list-style-type: none"><li>• Check the log file for errors logged by <i>ftam_init</i>.</li><li>• Consult HP-UX system reference manuals for action corresponding to <i>errno</i>.</li><li>• Collect relevant error information and contact your HP support representative.</li></ul>
MESSAGE	FTV105_AE_INVOC_TABLE_FULL
CAUSE	Application Entity (AE) invocation table is full.
ACTION	Verify that applications are shutting down invocations when finished. You cannot establish new AE invocation for presentation addressed that are not currently active. When AE invocations terminate, entries will become available.
MESSAGE	FTV106_AE_LABEL_TABLE_FULL
CAUSE	The maximum number of <i>ftam_inits</i> for a single process was exceeded.
ACTION	Deactivate all <i>ftam_inits</i> that are no longer in use.
MESSAGE	FTV107_SEM_ERROR
CAUSE	Semaphore error.
ACTION	<ul style="list-style-type: none"><li>• Check the log file.</li><li>• Consult HP-UX system reference manuals for action corresponding to <i>errno</i>.</li><li>• Collect relevant error information and contact your HP support representative.</li></ul>



---

**MESSAGE** FTV108\_SHMEM\_ERROR  
**CAUSE** Shared memory error.  
**ACTION**

- Check the log file for errors logged by `ftam_init`.
- Consult HP-UX system reference manuals for action corresponding to *errno*.
- Collect relevant error information and contact your HP support representative.

**MESSAGE** FTV109\_BAD\_IPC\_MSG  
**CAUSE** Bad interprocess message.  
**ACTION** Check the log file for logged error messages. Collect relevant error information and contact your HP support representative.

**MESSAGE** FTV110\_VIRTUAL\_MEMORY\_EXHAUSTED  
**CAUSE** Current resources are insufficient to honor request; virtual memory is exhausted.  
**ACTION** Check your memory management. Free any unused resources (e.g., memory), call `ft_dfcb()` to free initialized unused *dcbs*, and call the function again.

**MESSAGE** FTV111\_CID\_TABLE\_FULL  
**CAUSE** *Connection\_id* table is full.  
**ACTION** Check the log file for logged error messages. Free some connection entries by releasing or aborting connections.

**MESSAGE** FTV112\_CONFIG\_ERROR  
**CAUSE** Configuration error.  
**ACTION** Check and correct the OSI Configuration database.

HP vendor\_codes  
inout\_dcb->result.vendor\_code

---

**MESSAGE** FTV113\_NO\_AE\_INVOC\_ENTRY  
**CAUSE** The AE invocation entry for this presentation address is missing.  
**ACTION** Collect relevant error information and contact your HP support representative.

**MESSAGE** FTV201\_EM\_EMPTY\_EVENT\_EXP  
**CAUSE** Event management internal error.  
**ACTION** Check the log file for logged error messages. Collect relevant error information and contact your HP support representative.

**MESSAGE** FTV202\_EM\_EVENT\_MAX\_EXCD  
**CAUSE** Event management error; maximum number of events was exceeded.  
**ACTION** Free some memory resources and call the function again. If the error still occurs, collect relevant error information and contact your HP support representative.

**MESSAGE** FTV203\_EM\_INTERNAL\_ERROR  
**CAUSE** Event management internal error.  
**ACTION** Check the log file for logged error messages. Collect relevant error information and contact your HP support representative.

---

<b>MESSAGE</b>	FTV301_ILLEGAL_CALL_IN_GROUP
<b>CAUSE</b>	You called an illegal function within a group. You requested an illegal call while grouping was open on a connection.
<b>ACTION</b>	Call only valid functions within a group.
<b>MESSAGE</b>	FTV302_CANNOT_INIT_DCB_TABLE
<b>CAUSE</b>	Unable to initialize DCB table.
<b>ACTION</b>	Call the function again. If the error still occurs, collect relevant error information and contact your HP support representative.
<b>MESSAGE</b>	FTV303_CANNOT_ADD_TO_DCB_TABLE
<b>CAUSE</b>	Unable to add to DCB table.
<b>ACTION</b>	Free some memory resources and call the function again. If the error still occurs, collect relevant error information and contact your HP support representative.
<b>MESSAGE</b>	FTV304_CANNOT_INIT_ICS
<b>CAUSE</b>	The FTAM library cannot initialize the initial configuration store (ICS).
<b>ACTION</b>	Check and correct the OSI Configuration database.
<b>MESSAGE</b>	FTV305_CANNOT_ACCESS_ICS
<b>CAUSE</b>	The FTAM library cannot access the ICS.
<b>ACTION</b>	Check and correct the OSI Configuration database.

HP vendor\_codes  
inout\_dcb->result.vendor\_code

---

MESSAGE	FTV306_CANNOT_DECODE_DDN
CAUSE	The FTAM library cannot decode the directory distinguished name to a struct <i>Dir_dn</i> .
ACTION	Check the log file for logged error messages and recovery actions. If necessary, refer to the <i>OSI Troubleshooting Guide</i> .
MESSAGE	FTV307_INCOMPATIBLE_STACK
CAUSE	Your application was linked with a version of <i>libmapftam.a</i> which is not compatible with the underlying OSI network software (or “stack”).
ACTION	Re-link the application with the version of <i>libmapftam.a</i> on the system experiencing the error.
MESSAGE	FTV308_STACK_NOT_INITIALIZED
CAUSE	The underlying OSI network software (or “stack”) has not been started.
ACTION	Start the underlying OSI network software.
MESSAGE	FTV401_ERROR_IN_GROUP_CAUSED_STATE_FAILURE
CAUSE	After making a grouped request, one of the requests within the threshold set for the group failed; therefore, the regime did not change.
ACTION	Check the <i>action_result</i> , <i>state_result</i> , and diagnostic in the <i>inout_dcb</i> to determine which request failed and why. If the error still occurs, collect relevant error information and contact your HP support representative.

---

MESSAGE	FTV402_FUNC_UNIT_NEGOTIATED_DOWN
CAUSE	The responder does not support all the <i>functional_units</i> requested on the <i>ft_connect()</i> request.
ACTION	Informative message; no action required. Check to see which <i>functional_units</i> the responder supports. Attempt the appropriate action using low level calls if the <i>functional_units</i> supported allow it.
MESSAGE	FTV403_ABORT_BEFORE_CON_CNF_REQ
CAUSE	An abort indication returned on a connection before the responder confirmed the request.
ACTION	Check the <i>inout_dcb-&gt;diagnostic</i> to determine the reason for the abort and call the function again.
MESSAGE	FTV404_NEG_CNF_ON_CON_REQ
CAUSE	The responder sent a negative confirmation on an <i>ft_connect()</i> request.
ACTION	Check Association Control Service Element (ACSE) errors and <i>inout_dcb-&gt;diagnostic</i> to determine the reason for the rejection and call the function again.
MESSAGE	FTV405_HLCF_ABORT_IND_RCVD
CAUSE	An abort indication was received on a high level context free (HLCF) connection.
ACTION	Check the <i>inout_dcb-&gt;diagnostic</i> to determine the reason for the abort and call the function again.

HP vendor\_codes  
`inout_dcb->result.vendor_code`

---

<b>MESSAGE</b>	FTV406_HLCF_SRC_ABORT_IND_RCVD
<b>CAUSE</b>	An abort indication was received on an HLCF source connection.
<b>ACTION</b>	Check the <i>inout_dcb-&gt;diagnostic</i> to determine the reason for the abort and call the function again.
<b>MESSAGE</b>	FTV407_HLCF_DEST_ABORT_IND_RCVD
<b>CAUSE</b>	An abort indication was received on an HLCF destination connection.
<b>ACTION</b>	Check the <i>inout_dcb-&gt;diagnostic</i> ; then call the function again.
<b>MESSAGE</b>	FTV408_EXTRA_RDATA_REQ_AT_TRANS_END
<b>CAUSE</b>	Extra <i>ft_rdata()</i> request remained after a <i>DATA_END_IND</i> indication.
<b>ACTION</b>	Informative message; no action required.
<b>MESSAGE</b>	FTV409_NEG_SELECT_CNF_ON_HLCF_REQ
<b>CAUSE</b>	The responder sent a negative confirmation on the select request within a HLCF call.
<b>ACTION</b>	Check the <i>inout_dcb-&gt;diagnostic</i> to determine the reason for the failure and call the function again.
<b>MESSAGE</b>	FTV410_NEG_CREATE_CNF_ON_HLCF_REQ
<b>CAUSE</b>	The responder sent a negative confirmation on the create request within a HLCF call.
<b>ACTION</b>	Check the <i>inout_dcb-&gt;diagnostic</i> to determine the reason for the failure and call the function again.

---

MESSAGE	FTV411_NEG_OPEN_CNF_ON_HLCF_REQ
CAUSE	The responder sent a negative confirmation on the open request within a HLCF call.
ACTION	Check the <i>inout_dcb-&gt;diagnostic</i> to determine the reason for the failure and call the function again.
MESSAGE	FTV412_NEG_RATTR_CNF_ON_HLCF_REQ
CAUSE	The responder sent a negative confirmation on the read attributes request within a HLCF call.
ACTION	Check the <i>inout_dcb-&gt;diagnostic</i> to determine the reason for the failure and call the function again.
MESSAGE	FTV413_NEG_CATTR_CNF_ON_HLCF_REQ
CAUSE	The responder sent a negative confirmation on the change attributes request within a HLCF call.
ACTION	Check the <i>inout_dcb-&gt;diagnostic</i> to determine the reason for the failure and call the function again.
MESSAGE	FTV414_NEG_EDATA_IND_ON_HLCF_REQ
CAUSE	Negative data end indication on an HLCF call.
ACTION	Check the <i>inout_dcb-&gt;diagnostic</i> to determine the reason for the failure and call the function again.
MESSAGE	FTV415_NEG_ETRANS_CNF_ON_HLCF_REQ
CAUSE	The responder sent a negative confirmation on the transfer end request within a HLCF call.
ACTION	Check the <i>inout_dcb-&gt;diagnostic</i> to determine the reason for the failure and call the function again.

HP vendor\_codes  
inout\_dcb->result.vendor\_code

---

MESSAGE	FTV416_NEG_CLOSE_CNF_ON_HLCF_REQ
CAUSE	The responder sent a negative confirmation on the close request within a HLCF call.
ACTION	Check the <i>inout_dcb-&gt;diagnostic</i> to determine the reason for the failure and call the function again.
MESSAGE	FTV417_NEG_DELETE_CNF_ON_HLCF_REQ
CAUSE	The responder sent a negative confirmation on the delete request within a HLCF call.
ACTION	Check the <i>inout_dcb-&gt;diagnostic</i> to determine the reason for the failure and call the function again.
MESSAGE	FTV418_NEG_DESELECT_CNF_ON_HLCF_REQ
CAUSE	The responder sent a negative confirmation on the deselect request within a HLCF call.
ACTION	Check the <i>inout_dcb-&gt;diagnostic</i> to determine the reason for the failure and call the function again.
MESSAGE	FTV419_NEG_RELEASE_CNF_ON_HLCF_REQ
CAUSE	The responder sent a negative confirmation on the release request within a HLCF call.
ACTION	Check the <i>inout_dcb-&gt;diagnostic</i> to determine the reason for the failure and call the function again.
MESSAGE	FTV420_HLCF_STATE_ERROR
CAUSE	HLCF state error.
ACTION	Check the log file. Collect relevant error information; then contact your HP support representative.



---

**MESSAGE** FTV421\_NOT\_IN\_DATA\_TRANS\_PHASE

**CAUSE** You called *ft\_rdata()* while in a regime other than data transfer.

**ACTION** Call *ft\_rdata()* in the Data Transfer regime only after an *ft\_read()* request and before the transferring of data ends.

HP vendor\_codes  
inout\_dcb->result.vendor\_code

---

## **3 Diagnostic error\_ids**

This chapter contains a list of diagnostic values returned in the *inout\_dcb->diagnostic->error\_id* field. These errors are generated in the FTAM protocol machine (FPM) and the Virtual Filestore (VFS), not in the interface. This chapter includes only those *diagnostic->error\_ids*

Diagnostic error\_ids

returned by HP initiators and responders; it may or may not include *diagnostics* returned by other vendors. The causes and recovery actions listed are specific to HP's implementation.

- The *f\_error.h* file contains defined constants for all FTAM *diagnostic->error\_ids*.
- To receive a printable character string, review the information in *inout\_dcb->diagnostic->further\_details*.

Refer to the “Handling Errors” chapter in the *HP FTAM/9000 Programmer's Guide* for information on checking for and handling errors.

---

## inout\_dcb->diagnostic->error\_id

- |    |         |   |
|----|---------|---|
| 3  | MESSAGE | F_FTAM_MGT_PROBLEM  |
|    | CAUSE   | FTAM management problem.  |
|    | ACTION  | Collect relevant error information and contact your HP support representative.  |
| 8  | MESSAGE | F_SUBSEQ_ERR  |
|    | CAUSE   | This diagnostic returns on grouped functions if the threshold is not met. If the threshold number of functions is not processed before a <i>state_result</i> failure is detected, a negative re-sponse is made for the group. All grouped functions after the <i>ft_bgroup()</i> response and before the <i>state_result</i> failure will have an <i>action_result</i> of failure and will have a <i>diagnostic-&gt;error_id</i> of <i>F_SUBSEQ_ERR</i> . |
|    | ACTION  | Check <i>inout_dcb-&gt;diagnostic</i> to determine which function caused the group to fail. Refer to the appropriate section in this manual for specific recovery action guidelines.  |
| 10 | MESSAGE | F_ACCESS_VIOLATES_VFS_SEC   |
|    | CAUSE   | The access control element for this file or directory is violated.  |
|    | ACTION  | Check the access control and correct it.  |

Diagnostic error\_ids  
inout\_dcb->diagnostic->error\_id

---

11	MESSAGE	F_ACCESS_VIOLATES_LOCAL_SEC
	CAUSE	Could not access the local real file or directory due to UNIX permission problem.
	ACTION	Check the permissions of the file and directory.
1007	MESSAGE	F_FTAM_PROTO_ERR
	CAUSE	FTAM protocol error; may be one of the following reasons. <ul style="list-style-type: none"><li>• The parameter values could not be encoded.</li><li>• You issued the calls in an incorrect order.</li></ul>
	ACTION	Use the appropriate corrective action, as follows. <ul style="list-style-type: none"><li>• Call the function with valid parameters.</li><li>• Call the functions in the correct order.</li></ul>
1009	MESSAGE	F_FTAM_PROTO_ERR_FUNCU
	CAUSE	FTAM protocol error. You issued a request using a <i>functional_unit</i> that was not negotiated between <i>ftam_init</i> and the responder on an <i>ft_connect()</i> request.
	ACTION	Call <i>ft_connect()</i> using all required <i>functional_units</i> , which are based on the <i>service_classes</i> specified.

---

1011	MESSAGE	F_LWR_LAYER_FAIL
	CAUSE	A layer below FTAM (e.g., Presentation or Session layer) caused or reported an error. Usually this error is indicated by an abort indication; FTAM returns abort information on this diagnostic after calling <i>ft_ireceive()</i> .
	ACTION	Execute the <i>netfmt</i> utility to obtain log information and correct the error as described in the log message. If needed, collect relevant error information and contact your HP support representative.
1015	MESSAGE	F_ILLEGAL_GRP_SEQ
	CAUSE	The grouping sequence is illegal. Depending on the <i>service_class</i> negotiated, only certain sequences of PDUs can be in a grouped request.
	ACTION	Call the grouped functions in a valid sequence. Refer to the <i>HP FTAM/9000 Programmer's Guide</i> for acceptable sequences when grouping functions.
2008	MESSAGE	F_ASSOC_MGT
	CAUSE	Association management; returns when an internal data structure is not present or incorrect.
	ACTION	Ensure the <i>ft_aeactivation()</i> data structures are present and correctly set. You have two options. <ul style="list-style-type: none"><li>• Terminate the call.</li><li>• Reactivate the <i>ftam_init</i> and try to re-establish the connection.</li></ul>

Diagnostic error\_ids  
inout\_dcb->diagnostic->error\_id

---

2015	MESSAGE	F_INIT_ID_UNACCEPT
	CAUSE	The login name (initiator identity) is unacceptable on an <i>ft_connect()</i> request. Either the name does not exist or your incorrectly entered it.
	ACTION	Call the function with a valid initiator identity (i.e., an existing HP-UX login name).
2020	MESSAGE	F_INVALID_FS_PASSWD
	CAUSE	The <i>filestore_pw</i> is invalid.
	ACTION	Call the function with a valid <i>filestore_pw</i> . The <i>filestore_pw</i> is the HP-UX password associated with the initiator identity that is the HP-UX login.
3000	MESSAGE	F_FILE_NOT_FOUND
	CAUSE	The filename specified on <i>ft_select()</i> does not exist or you entered it incorrectly.
	ACTION	Call <i>ft_select()</i> with a correct and existing filename.
3002	MESSAGE	F_INIT_ATTRIB_NOT_POSSIBLECAUSE
	CAUSE	The initial attributes do not allow the request to be completed. You did not supply the filename, <i>contents_type</i> , or <i>permitted_actions</i> on a request.
	ACTION	Call the function with a filename, <i>contents_type</i> , or <i>permitted_actions</i> as needed.



---

3004	MESSAGE	F_NON_EXISTENT_FILE
	CAUSE	The shadow file for the filename specified on <i>ft_select()</i> does not exist.
	ACTION	Remove the existing FTAM data file and recreate it using FTAM functions; doing so recreates the shadow file.
3005	MESSAGE	F_FILE_ALREADY_EXISTS
	CAUSE	You called <i>ft_create()</i> with <i>file_status</i> set to <i>FT_NEW</i> , and a file by that same name already exists in the filestore.
	ACTION	Call <i>ft_create()</i> with a different filename.
3007	MESSAGE	F_CANNOT_DELETE_FILE
	CAUSE	On an <i>ft_select()</i> , <i>ft_create()</i> , or <i>ft_fopen()</i> function, you selected a file without setting the <i>FT_FA_DELETE_FILE</i> bit in the <i>requested_access</i> field.
	ACTION	Set <i>FT_FA_DELETE_FILE</i> to ON (1) and call the function again.

Diagnostic error\_ids  
inout\_dcb->diagnostic->error\_id

---

3010	MESSAGE	F_CONC_CTL_NOT_POSSIBLE
	CAUSE	<p>You cannot currently access the file for one of the following reasons</p> <ul style="list-style-type: none"><li>• Another application is accessing the file and enum <i>Ft_file_lock</i> in <i>Ft_concurrency_control</i> is set such that you cannot concurrently access the file.</li><li>• The <i>requested_access</i> and <i>concurrency_control</i> parameters are not compatible.</li></ul> <p>EXAMPLES: The <i>requested_access</i> for an action is not set and you request a <i>concurrency_control</i> other than <i>FT_NOT_REQUIRED</i> or <i>FT_NO_ACCESS</i>. Someone has the file selected with the <i>FT_FA_INSERT</i> action set to <i>FT_EXCLUSIVE</i> and you request an <i>FT_FA_INSERT</i> action of <i>FT_SHARED</i>.</p>
	ACTION	<p>Call the function again after the file is closed by the other application. If you still cannot access the file, ensure your <i>concurrency_control</i> parameter is consistent with your <i>requested_access</i> parameter.</p>

---

3016	MESSAGE	F_ACCESS_CTL_INCONSISTENT
	CAUSE	<p>This diagnostic returns on <i>ft_select()</i>, <i>ft_create()</i>, and HLCF calls. The cause may be one of the following reasons.</p> <ul style="list-style-type: none"><li>• The bits set in the <i>FT_AC_XXX</i> defined constants prohibit your requested access to the file.</li><li>• The <i>requested_access</i>, <i>conc_access</i>, or <i>file_passwords</i> values do not match the access control element corresponding to your initiator identity.</li></ul> <p>EXAMPLE: Passwords for <i>ft_select()</i> do not match those in the <i>Ft_access_control_element</i> corresponding to the current initiator identity.</p>
	ACTION	<p>Change the file attributes so that the access control element corresponding to your initiator identity allows you to perform the necessary actions (<i>requested_access</i>, <i>conc_access</i>, <i>file_passwords</i>).</p>
3020	MESSAGE	F_CREATE_SLCTD_EXIST_FILE
	CAUSE	<p>On an <i>ft_create()</i> call, the file already exists and the <i>file_status</i> is set to <i>FT_OLD</i>.</p>
	ACTION	<p>Informative message; no action required.</p>
3023	MESSAGE	F_CREATE_OVRRD_NOT_POSSIBLE
	CAUSE	<p>A file exists without a corresponding shadow file, and the <i>file_status</i> is set to <i>FT_NEW</i>, <i>FT_OLD</i>, or <i>FT_REPLACE_CONTENTS</i>.</p>
	ACTION	<p>Either remove the data file and try again or set <i>file_status</i> to <i>FT_RECREATE</i>.</p>

Diagnostic error\_ids  
inout\_dcb->diagnostic->error\_id

---

3028	MESSAGE	F_REQ_ACC_VIOL_PERM_ACT
	CAUSE	The bits set for the <i>requested_access FT_FA_XXX</i> defined constants are not a subset of the bits set for the <i>permitted_actions FT_PA_XXX</i> defined constants.
	ACTION	Correctly set the <i>FT_FA_XXX</i> bits or call the function with a valid <i>requested_access</i> . The <i>requested_access</i> must be a subset of the <i>permitted_actions</i> values on <i>ft_select()</i> or <i>ft_create()</i> .
4001	MESSAGE	F_ATTRIB_CANNOT_READ
	CAUSE	You cannot read the file attributes because when you selected the file, you did not set <i>requested_access</i> to include <i>FT_FA_READ_ATTRIBUTE</i> .
	ACTION	Deselect the file and select it again with the <i>requested_access</i> set to include <i>FT_FA_READ_ATTRIBUTE</i> .

---

4002	MESSAGE	F_ATTRIB_CANNOT_CHNG
	CAUSE	<p>You cannot change the attribute for one of the following reasons.</p> <ul style="list-style-type: none"><li>• When you selected the file, you did not set <i>requested_access</i> to include <i>FT_FA_CHANGE_ATTRIBUTE</i>.</li><li>• You tried to add a duplicate access control element for an existing initiator identity.</li><li>• You tried to change a file's name to the name of a file that already exists.</li></ul>
	ACTION	<p>Use the appropriate corrective action, as follows.</p> <ul style="list-style-type: none"><li>• Deselect the file and select it again with <i>requested_access</i> set to include <i>FT_FA_CHANGE_ATTRIBUTE</i>.</li><li>• Remove the access control element for the existing initiator identity and insert one for the initiator identity you want.</li><li>• Change the name of the file to one that does not exist.</li></ul>
5002	MESSAGE	F_BAD_FADU_TYPE_ERR
	CAUSE	<p>The <i>structure_id</i> describing the type of FADU information is not correct.</p>
	ACTION	<p>Set the <i>structure_id</i> to a valid value. Use <i>FT_DATA_UNIT</i> for all document types and <i>FT_NODE_DESC</i> for FTAM-2 document types only.</p>

Diagnostic error\_ids  
inout\_dcb->diagnostic->error\_id

---

5014	MESSAGE	F_DATA_ELEMENT_TYPE
	CAUSE	The <i>prim_type</i> field in struct <i>Ft_data_element</i> is not set correctly.
	ACTION	Set <i>prim_type</i> to a valid value. For FTAM-1 document types, <i>prim_type</i> must be <i>IA5_STRING</i> or <i>GENERAL_STRING</i> . For FTAM-2 document types, <i>prim_type</i> must be <i>GRAPHIC_STRING</i> . For FTAM-3 document types, <i>prim_type</i> must be <i>OCTET_STRING</i> . For INTAP-1 document types, <i>prim_type</i> must be <i>RECORD_END</i> or <i>RECORD_CONT</i> .
5017	MESSAGE	F_OPR_INCONSISTENT
	CAUSE	The operation is inconsistent. Causes may include the following cases. <ul style="list-style-type: none"><li>• You called <i>ft_read()</i> though the <i>processing_mode</i> on <i>ft_open()</i> did not have the <i>FT_PM_READ</i> bit set to ON (1).</li><li>• You called <i>ft_write()</i> though the <i>processing_mode</i> on <i>ft_open()</i> did not have <i>FT_PM_INSERT</i>, <i>FT_PM_REPLACE</i>, or <i>FT_PM_EXTEND</i> bit set ON (1), depending on the document type.</li></ul>
	ACTION	Correctly set the <i>FT_PM_XXX</i> bits on the <i>processing_mode</i> before calling <i>ft_open()</i> .

---

5023	MESSAGE	F_PROC_MD_INCONSISTENT
	CAUSE	The bits set for the <i>processing_mode FT_FA_XXX</i> defined constants are not a subset of the bits set for the <i>requested_access FT_PA_XXX</i> defined constants.
	ACTION	Correctly set the <i>FT_FA_XXX</i> bits or call <i>ft_open()</i> with a valid <i>processing_mode</i> . The <i>processing_mode</i> must be a subset of the <i>requested_access</i> values on <i>ft_select()</i> or <i>ft_create()</i> .
5024	MESSAGE	F_ACCESS_CNTXT_NOT_AVAIL
	CAUSE	The <i>access_context</i> is not available. Given a particular document type, a VFS implementation may not support certain <i>access_contexts</i> . EXAMPLE: You tried accessing an FTAM-2 file with an <i>access_context</i> of HA (Hierarchical All).
	ACTION	Use an <i>access_context</i> value that is consistent with the file document type.
5026	MESSAGE	F_BAD_WRITE
	CAUSE	On an <i>ft_write()</i> function, the <i>Ft_fadu_location</i> in <i>Ft_fadu_identity</i> points to a non-existent FADU; therefore, you cannot write to the file (FTAM-1, FTAM-3, and INTAP-1) or FADU (FTAM-2).
	ACTION	Call <i>ft_sdata()</i> with a valid <i>fadu_identity</i> . For FTAM-1, FTAM-3, and INTAP-1 document types, <i>fadu_identity</i> must be <i>FT_FIRST</i> . For FTAM-2 document types, <i>fadu_identity</i> must be <i>FT_BEGIN</i> .

Diagnostic error\_ids  
inout\_dcb->diagnostic->error\_id

---

5027	MESSAGE	F_BAD_READ
	CAUSE	The <i>fadu_identity</i> on an <i>ft_read()</i> request points to a non-existent FADU; therefore, you cannot read the file (FTAM-1, FTAM-3, and INTAP-1) or FADU (FTAM-2).
	ACTION	Call <i>ft_read()</i> with a valid <i>fadu_identity</i> . For FTAM-1, FTAM-3, and INTAP-1 document types, <i>fadu_identity</i> must be <i>FT_FIRST</i> . For FTAM-2 document types, <i>fadu_identity</i> must be <i>FT_BEGIN</i> .
5031	MESSAGE	F_LCL_FAIL_DVC_FAIL
	CAUSE	Local device failure. An I/O error occurred when trying to write the file via an HP-UX system call. For example, you may be out of disc space.
	ACTION	If needed, create disc space on the real file system (responder) where the write failure occurred. Otherwise, collect relevant error information and contact your HP support representative.



---

## **4 Event Management Errors**

This chapter contains a list of values that may be returned when calling *em\_wait()*, *em\_fdmemory()*, and *em\_gperror()*. These errors return in the *result->return\_code* and *result->vendor\_code* fields. To receive a printable character string for returned errors, call *em\_gperror()*.

## Event Management Errors

Refer to the “Handling Errors” chapter in the *HP FTAM/9000 Programmer's Guide* for information on checking for and handling errors.

---

## EM return\_codes

This section contains a list of values that may be returned in the *result->return\_code* field when calling *em\_wait()*, *em\_fdmemory()*, and *em\_gperror()*.

---

### NOTE

The error EME031\_INVALID\_BUF\_PTR returns only as a function return value if the result parameter is set to a NULL pointer (i.e., it does not return in the *return\_code* or *vendor\_code*).

---

MESSAGE SUCCESS

CAUSE Successful function execution (no error).

ACTION Informative message; no action required.

MESSAGE EME002\_EXP\_EMPTY

CAUSE No events were posted; therefore, there is no reason to wait.

ACTION Call a function asynchronously before calling *em\_wait()*.

MESSAGE EME004\_TIME\_INV

CAUSE The timeout parameter is invalid.

ACTION Call *em\_wait()* with a valid timeout parameter. Valid timeout values are as follows. Parameter timeout  
Period -1 – Indefinite wait 0 – Return immediately  
with results 1 to 32767 – Wait up to this time period  
(tenths of seconds)

## Event Management Errors

### EM return\_codes

---

MESSAGE	EME005_TIMEOUT
CAUSE	None of the posted events were noted within the time allotted by the timeout parameter.
ACTION	Call <i>em_wait()</i> again to receive the noted events, possibly with a longer timeout parameter.
MESSAGE	EME031_INVALID_BUF_PTR
CAUSE	Invalid buffer pointer; you provided a NULL pointer instead of a valid address for one of the parameters. The <i>result-&gt;vendor_code</i> indicates the <i>return_event_name</i> is invalid. Only the function return value reflects the error if the result parameter is invalid.
ACTION	Set the <i>return_event_name</i> and result parameters to valid, non- NULL values.
MESSAGE	EME032_IPC_ERROR
CAUSE	Your program could not access the <i>ftam_init</i> processing the <i>return_event_name</i> .
ACTION	Refer to the <i>result-&gt;vendor_code</i> for the actual IPC error value. Call <i>em_gperror()</i> to translate the value into a message and refer to the log file to determine why you could not access <i>ftam_init</i> . If you are unable to correct the error, refer to the <i>OSI Troubleshooting Guide</i> .
MESSAGE	EME090_INV_DYNAMIC_MEM_PTR
CAUSE	The <i>memory_pointer</i> you specified is not the address of dynamic memory allocated by the interface.
ACTION	Only call <i>em_fdmemory()</i> with the address of dynamic memory allocated by <i>em_gperror()</i> .

MESSAGE	EME097_NO_SPACE_AVAILABLE
CAUSE	The interface could not allocate sufficient space for the <i>return_string</i> and <i>vendor_string</i> ; the system ran out of memory.
ACTION	Free any allocated dynamic memory that is no longer needed and call the function again. EXAMPLE: You made multiple calls to <i>em_gperror()</i> requesting that the interface allocate memory; however, you did not call <i>em_fdmemory()</i> to free the memory after <i>em_gperror()</i> completed.
MESSAGE	EME098_INVALID_RETURN_CODE
CAUSE	The <i>input_results</i> structure contains a <i>return_code</i> that does not exist.
ACTION	Only call <i>em_gperror()</i> with an <i>input_result-&gt;return_code</i> that was returned by the interface.

## EM vendor\_codes

This section contains a list of values that may be returned in the *result->vendor\_code* field when calling *em\_wait()*, *em\_fdmemory()*, and *em\_gperror()*.

### result->return\_code

MESSAGE	EMV000_NO_CODE_SUPPLIED
CAUSE	No additional vendor-specific information is available; the <i>return_code</i> provides sufficient information.
ACTION	Informative message; no action required.
MESSAGE	EMV002_PARAM_EVENT_NAME
CAUSE	The <i>return_event_name</i> buffer pointer is set to NULL.
ACTION	Set the <i>return_event_name</i> parameter to a valid, non-NULL value.
MESSAGE	EMV003_PARAM_IN_RESULT
CAUSE	The <i>input_result</i> buffer pointer is set to NULL.
ACTION	Set the <i>input_result</i> parameter to a valid, non-NULL value.
MESSAGE	EMV004_PARAM_RET_STRING
CAUSE	The <i>return_string</i> buffer pointer is set to NULL.
ACTION	Set the <i>return_string</i> parameter to a valid, non-NULL value.

---

MESSAGE	EMV005_PARAM_VEN_STRING
CAUSE	The <i>vendor_string</i> buffer pointer is set to NULL.
ACTION	Set the <i>vendor_string</i> parameter to a valid, non-NULL value.

Event Management Errors  
**EM vendor\_codes**



---

# **5 HP FTAM/9000 PICS**

For a copy of the Protocol Implementation Conformance Statement (PICS) for the HP FTAM/9000 product, please contact your HP support representative.

HP FTAM/9000 PICS

---

## **6** **Installation Filesets**

The files listed in this chapter are installed in the indicated directories at the time you install FTAM. Installing the reference page (man page) files is recommended, but omitting them will not operationally affect FTAM.

---

## Files Created During Software Installation

<b>Files in fileset FTAM</b>	<b>Function</b>
/opt/ftam/bin/ftam	Interactive interface executable
/opt/ftam/bin/fcattr	fcattr command
/opt/ftam/bin/fcp	fcp command
/opt/ftam/bin/fdel	fdel command
/opt/ftam/bin/fls	fls command
/opt/ftam/bin/fmv	fmv command
/opt/ftam/bin/fchdoc	Utility to change FTAM attributes of a local file
/opt/ftam/sbin/osiinitshm	OTS startup utility for FTAM
/opt/ftam/sbin/osirmshm	OTS startup utility for FTAM
/opt/ftam/sbin/osistat	OTS startup utility for FTAM
/opt/ftam/sbin/id.conf	OTS startup utility for FTAM
/opt/ftam/lbin/ftam_resp	FTAM responder daemon process
/opt/ftam/lbin/ftam_init	FTAM initiator service provider process
/opt/ftam/include/map.h	Include file for common MAP library
/opt/ftam/include/mapftam.h	Programmatic interface structure definitions
/opt/ftam/include/f_error.h	Programmatic interface error definitions
/opt/ftam/lib/libmap.a	MAP library for FTAM
/opt/ftam/lib/llib-lmap	lint library for FTAM programmatic interface

<b>Files in fileset FTAM</b>	<b>Function</b>
/opt/ftam/lib/llib-lmap.ln	lint library for FTAM programmatic interface
/opt/ftam/lib/libmapftam.a	Programmatic interface library definitions
/opt/ftam/lib/llib-lmapftam	lint library for FTAM programmatic interface
/opt/ftam/lib/llib-lmapftam.ln	lint library for FTAM programmatic interface
/opt/ftam/lib/nls/C/acse.cat	Network tracing and logging NLS catalog
/opt/ftam/lib/nls/C/cm.cat	Network tracing and logging NLS catalog
/opt/ftam/lib/nls/C/fchdoc.cat	Message catalog used by fchdoc command
/opt/ftam/lib/nls/C/fmt124f.cat	Network tracing and logging NLS catalog
/opt/ftam/lib/nls/C/ftam_cmds.cat	Message catalog used by FTAM user interface commands
/opt/ftam/lib/nls/C/ftamfmt.cat	Network tracing and logging NLS catalog
/opt/ftam/lib/nls/C/hps.cat	Network tracing and logging NLS catalog
/opt/ftam/lib/nls/C/mapftam.cat	Message catalog used by ft_gperror() and FTAM user interface commands
/opt/ftam/lib/nls/C/mapem.cat	NLS catalog for em_gperror()
/opt/ftam/lib/nls/C/osistat.cat	NLS catalog for osistat
/opt/ftam/lib/nls/C/shm.cat	NLS catalog for shared memory manager
/opt/ftam/shlib/libmap.sl	MAP shared library
/opt/ftam/shlib/libmapftam.sl	FTAM shared library
/opt/ftam/shlib/libfmtftam.sl	Network tracing and logging subformatter
/opt/ftam/shlib/libfmt116f.sl	Network tracing and logging subformatter
/opt/ftam/shlib/libfmt119f.sl	Network tracing and logging subformatter
/opt/ftam/shlib/libfmt121f.sl	Network tracing and logging subformatter

Installation Filesets  
Files Created During Software Installation

<b>Files in fileset FTAM</b>	<b>Function</b>
/opt/ftam/shlib/libfmt122f.sl	Network tracing and logging subformatter
/opt/ftam/shlib/libfmt124f.sl	Network tracing and logging subformatter
/opt/ftam/shlib/fmt124f.sl	Network tracing and logging subformatter
/opt/ftam/demos/Makefile	Makefile for FTAM demo
/opt/ftam/demos/README	README file for FTAM demo
/opt/ftam/demos/setup	Set up of sample file
/opt/ftam/demos/ftm_globs.h	Include file for demo programs
/opt/ftam/demos/ftm_conn.c	Demo for connection establishment
/opt/ftam/demos/ftm_dirnam.c	Source file for demo programs
/opt/ftam/demos/ftm_hlcopy.c	Demo file copy using high-level functions
/opt/ftam/demos/ftm_llcopy.c	Demo file copy using low-level functions
/opt/ftam/demos/ftm_parm.c	Source file for demo programs
/opt/ftam/demos/ftm_util.c	Source file for demo programs
/opt/ftam/demos/resp_api/Makefile	Source file for demo programs
/opt/ftam/demos/resp_api/test_resp	Source file for demo programs
/opt/ftam/demos/resp_api/test_resp1	Source file for demo programs
/opt/ftam/demos/resp_api/test_vsp	Source file for demo programs
/opt/ftam/demos/resp_api/ftm_util.c	Source file for demo programs
/opt/ftam/demos/resp_api/vsp.c	Source file for demo programs
/etc/opt/ftam/conf/ftam_conf	Miscellaneous FTAM configuration parameters
/etc/opt/ftam/conf/ftam_pw	Configuration file for FTAM password types

<b>Files in fileset OSIF-MAN</b>	<b>Function</b>
/opt/ftam/man/man1.Z/fcattr.1	Man page for indicated function
/opt/ftam/man/man1.Z/fchdoc.1	Man page for indicated function
/opt/ftam/man/man1.Z/fcp.1	Man page for indicated function
/opt/ftam/man/man1.Z/fdel.1	Man page for indicated function
/opt/ftam/man/man1.Z/fls.1	Man page for indicated function
/opt/ftam/man/man1.Z/ftam.1	Man page for indicated function
/opt/ftam/man/man1.Z/ftam_resp.1	Man page for indicated function
/opt/ftam/man/man3.Z/ft_abort.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_aeactiva.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_aedeacti.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_aereset.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_bgroup.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_cancel.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_cattribu.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_close.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_connect.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_create.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_delete.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_deselect.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_dfddb.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_didcb.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_edata.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_egroup.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_erase.3n	Man page for indicated function

Installation Filesets  
Files Created During Software Installation

<b>Files in fileset OSIF-MAN</b>	<b>Function</b>
/opt/ftam/man/man3.Z/ft_etranse.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_fcattrib.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_fclose.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_fcopy.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_fcopy_aet.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_fdelete.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_fdelete_aet.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_fdmemory.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_fmmove.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_fmmove_aet.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_fopen.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_frattrib.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_gperror.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_ireceive.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_locate.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_nwcleare.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_open.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_rattribu.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_rcancel.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_rdata.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_rdataqos.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_read.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_rrequest.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_sdata.3n	Man page for indicated function



<b>Files in fileset OSIF-MAN</b>	<b>Function</b>
/opt/ftam/man/man3.Z/ft_select.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ft_write.3n	Man page for indicated function
/opt/ftam/man/man3.Z/em_hp_select.3n	Man page for indicated function
/opt/ftam/man/man3.Z/em_wait.3n	Man page for indicated function
/opt/ftam/man/man3.Z/em_gperror.3n	Man page for indicated function
/opt/ftam/man/man3.Z/em_fdmemory.3n	Man page for indicated function
/opt/ftam/man/man3.Z/em_hp_sigio.3n	Man page for indicated function
/opt/ftam/man/man3.Z/ftam.3n	Man page for overview of programmatic FTAM
/opt/ftam/man/man4.Z/ftamrc.4	Man page for FTAM startup file

Installation Filesets  
**Files Created During Software Installation**



<b>Entry Name</b>	<b>Description</b>
em_fdmemory(3)	free dynamic memory allocated by Event Management
em_gperror(3)	translate an Event Management error to a string
em_hp_select(3)	wait for MAP 3.0 and non-MAP events
em_hp_sigio(3)	enable/disable signal notification for MAP 3.0
em_wait(3)	wait for an asynchronous MAP 3.0 event
fcattr(1)	change the attributes of an FTAM file
fcop(1)	copy an FTAM file
fdel(1)	remove an FTAM file
fls(1)	list an FTAM file or directory
ft_abort(3)	abort an FTAM connection
ft_aeactivation(3)	activate an FTAM initiator
ft_aedeactivation(3)	deactivate an FTAM initiator
ft_aereset(3)	reset an FTAM initiator
ft_bgroup(3)	begin a group of FTAM functions
ft_cancel(3)	cancel FTAM data transfer in progress
ft_cattributes(3)	change the file attributes of an FTAM file
ft_close(3)	close an FTAM file
ft_connect(3)	establish a connection with an FTAM responder
ft_create(3)	create an FTAM file
ft_delete(3)	delete the currently selected FTAM file
ft_deselect(3)	deselect the currently selected FTAM file
ft_dfddb(3)	free a dynamically initialized FTAM data control block
ft_didcb(3)	initialize an FTAM data control block

<b>Entry Name</b>	<b>Description</b>
ft_edata(3)	end a series of FTAM data primitives
ft_egroup(3)	end a group of FTAM functions
ft_erase(3)	erase all or part of an FTAM file
ft_ettransfer(3)	end an FTAM data transfer
ft_fattributes(3)	change the file attributes of an FTAM file
ft_fattributes_aet(3)	change the file attributes of an FTAM file
ft_fclose(3)	close and either deselect or delete an FTAM file
ft_fcopy(3)	copy an FTAM file
ft_fcopy_aet(3)	copy an FTAM file
ft_fdelete(3)	delete an FTAM file
ft_fdelete_aet(3)	delete an FTAM file
ft_fdmemory(3)	free dynamic memory allocated by FTAM
ft_fmove(3)	move an FTAM file
ft_fmove_aet(3)	move an FTAM file
ft_fopen(3)	select or create and then open an FTAM file
ft_frattributes(3)	read the file attributes of an FTAM file
ft_frattributes_aet(3)	read the file attributes of an FTAM file
ft_gperror(3)	translate an FTAM error to a printable string
ft_ireceive(3)	receive an FTAM abort indication
ft_locate(3)	locate a specific part of an FTAM file
ft_nwcleared(3)	note when outstanding FTAM resource is cleared
ft_open(3)	open an FTAM file
ft_rattributes(3)	read the file attributes of an FTAM field.
ft_rcancel(3)	respond to an FTAM cancel indication

<b>Entry Name</b>	<b>Description</b>
ft_rdata(3)	receive a block of FTAM data
ft_read(3)	request transfer of data from FTAM
ft_rrequest(3)	release an FTAM connection
ft_sdata(3)	send a block of FTAM data
ft_select(3)	select an FTAM file
ft_write(3)	request transfer of data to an FTAM file
ftam(1)	OSI file transfer, access and management program
ftamrc(3)	FTAM security file.