

HI.CommunicationEngine
FTPサーバ
リファレンスマニュアル

ご注意

1. 本製品(ソフトウェア製品及びその関連ソフトウェア製品を含む。以下、同じ。)の使用に際しては、「外国為替及び外国貿易法」等、技術輸出に関する日本及び関連諸国の関係法規の遵守が必要となります。
2. 弊社は、本製品の使用に際しては、弊社もしくは第三者の特許権、著作権、商標権、その他の知的所有権等の権利に関し、別途、個別の契約書等(マニュアルの記載を含む。以下、同じ。)にて弊社による明示的な許諾がある場合を除き、その保証または実施権の許諾を行うものではありません。また本製品を使用したことにより第三者の知的所有権等の権利に関わる問題が生じた場合、弊社はその責を負いませんので予めご了承ください。
3. 本製品およびその仕様、またはマニュアルに記載されている事柄については、将来、事前の予告なしに変更することがありますので、最終的な設計、ご購入、ご使用に際しましては、事前に最新の製品規格または仕様書(マニュアルを含む)をご確認ください。
4. 本製品の使用(マニュアル記載事項に基づくものも含む)により直接または間接に生ずるいかなる損害についても、弊社は一切の責任を負いません。また、本製品の配布に使用される搭載機器や媒体が原因の損害に対しましても、弊社は一切の責任を負いません。
5. 本製品を、宇宙、航空、原子力、燃焼制御、運輸、交通、各種安全装置、ライフサポート関連の医療機器等のように、特別な品質・信頼性が要求され、その故障や誤動作が直接人命を脅かしたり、人体に危害を及ぼす恐れのある用途向けには使用できません。お客様の用途がこれに該当するかどうか疑問のある場合には、事前に弊社営業担当迄ご相談をお願い致します。
6. 本製品を使用してお客様のシステム製品を設計される際には、通常予測される故障発生率、故障モードをご考慮の上、本製品の動作が原因での事故、その他の拡大損害を生じないようにフェールセーフ等の十分なシステム上の対策を講じて頂きますようお願い致します。
7. 本製品およびマニュアルの著作権は弊社が所有しております。お客様は、弊社から提供された本製品を、別途、個別の契約書等にて定める場合を除き、いかなる場合においても全体的または部分的に複製・解析・改変することはできないものとします。
8. お客様は、別途、個別の契約書等にて定める場合を除き、本製品のマニュアルの一部または全部を無断で使用、複製することはできません。
9. 弊社は、本製品を1台のコンピュータで使用する権利をお客様に対してのみ許諾します。よって、本製品を第三者へ譲渡、貸与、賃借することは許諾しないものとします。但し、別途、個別の契約書等にて定められる場合はその条件に従います。
10. 本製品をはじめ弊社半導体およびその関連製品についてのお問い合わせ、ご相談は弊社営業担当迄お願い致します。

μ ITRON は、Micro Industrial TRON の略称です。TRON は、The Realtime Operating system Nucleus の略称です。

その他、本書で登場するシステム名、製品名は各社の登録商標または商標です。

はじめに

このマニュアルは、HI.CommunicationEngine TCP/IPマネージャ上で動作するTCP/IPネットワークアプリケーション「HI.CommunicationEngine FTPサーバ」について説明します。

HI.CommunicationEngine FTPサーバは、TCP/IPマネージャを経由し、ネットワーク上の端末（FTPクライアント）からの接続要求を受けて、ファイルの転送とファイルを管理する機能を提供します。このリファレンスマニュアルではFTPサーバのサービスコールとその使い方および関連事項を説明します。TCP/IPマネージャについては関連マニュアルを参照してください。

【関連マニュアル】

- HI.CommunicationEngineTCP/IPマネージャリファレンスマニュアル
- 使用する μ ITRON仕様OS のユーザーズマニュアル

目次

1. 概要	1
1.1 FTPの機能概要	1
1.2 サービスタスクの実現	2
1.3 サービスコールのしくみ.....	3
1.4 使用するリソース.....	3
1.5 サポートするFTPサーバ機能.....	4
1.6 FTPサーバ サービスコールの種類.....	6
2. FTPサーバ サービスコールの使用方法	7
2.1 FTPサーバの基本フロー.....	7
2.2 FTPサーバ サービスポイントの状態遷移図.....	9
3. FTPサーバ サービスコール	10
3.1 初期化サービスコール	11
3.1.1 <i>FTD_init</i> FTPサーバのサービスを初期化する.....	11
3.2 基本サービスコール.....	12
3.2.1 <i>FTD_start</i> FTPサーバのサービスを開始する.....	12
3.2.2 <i>FTD_stop</i> FTPサーバのサービスを終了する.....	15
3.2.3 <i>FTD_acp_srv</i> FTPサーバを接続待ち状態にする.....	16
3.2.4 <i>FTD_sht_svp</i> サービスポイントの接続を切断する.....	18
3.2.5 <i>FTD_put_dat</i> FTPクライアントにデータを送信する.....	19
3.2.6 <i>FTD_get_dat</i> FTPクライアントからデータを受信する.....	20
3.2.7 <i>FTD_set_opt</i> サービスポイントの動作モードを設定する.....	21
3.2.8 <i>FTD_get_opt</i> サービスポイントの状態を取得する.....	26
3.2.9 <i>FTD_can_svp</i> ペンディングしている処理をキャンセルする.....	28
3.2.10 <i>FTD_evt_end</i> FTPサーバイベントの処理完了を通知する.....	29
4. コールバック	30
4.1 コールバックルーチン	31
4.1.1 <i>FTD_callback</i> FTPサーバイベントをユーザに通知する.....	31
4.2 コールバックイベント	32
(1) <i>TEV_FTD_CONN</i> FTPクライアントの接続イベント.....	33
(2) <i>TEV_FTD_DISC</i> FTPクライアントの切断イベント.....	34
(3) <i>TEV_FTD_QUIT</i> FTPクライアントの切断要求イベント.....	35
(4) <i>TEV_FTD_USER</i> ユーザ名の識別要求イベント.....	36
(5) <i>TEV_FTD_PASS</i> パスワードの識別要求イベント.....	37
(6) <i>TEV_FTD_ACCT</i> ユーザアカウントの識別要求イベント.....	38
(7) <i>TEV_FTD_CWD</i> カレントディレクトリの変更要求イベント.....	39
(8) <i>TEV_FTD_CDUP</i> 親ディレクトリへの変更要求イベント.....	40
(9) <i>TEV_FTD_RMD</i> ディレクトリの削除要求イベント.....	41
(10) <i>TEV_FTD_MKD</i> ディレクトリの作成要求イベント.....	42
(11) <i>TEV_FTD_PWD</i> カレントディレクトリの送信要求イベント.....	43
(12) <i>TEV_FTD_LIST</i> ファイル一覧の送信要求イベント.....	44
(13) <i>TEV_FTD_NLST</i> 詳細なファイル一覧の送信要求イベント.....	45
(14) <i>TEV_FTD_DELE</i> ファイルの削除要求イベント.....	46
(15) <i>TEV_FTD_RNFR</i> ファイル名の変更要求イベント (変更元ファイル名).....	47
(16) <i>TEV_FTD_RNTO</i> ファイル名の変更要求イベント (変更後ファイル名).....	48
(17) <i>TEV_FTD_RETR</i> ファイルの送信要求イベント.....	49

(18)	TEV_FTD_STOR	新規ファイルへの書き込み要求イベント	50
(19)	TEV_FTD_APPE	既存ファイルへの書き込み要求イベント	52
(20)	TEV_FTD_STOU	ユニークファイルへの書き込み要求イベント	54
(21)	TEV_FTD_ALLO	ファイル書き込みスペースの確保要求イベント	55
(22)	TEV_FTD_SITE	サーバ依存コマンドの受信イベント	56
(23)	TEV_FTD_STAT	サーバの状態送信要求イベント.....	57
(24)	TEV_FTD_HELP	サーバのコマンドヘルプ送信要求イベント	58
(25)	TEV_FTD_ABOR	コマンドの中断イベント	60
(26)	TEV_FTD_DPCN	データ接続の接続開始イベント	61
(27)	TEV_FTD_ERROR	エラーイベント	62

5. FTPサーバ レスポンスコード.....63

図表目次

図1.1	FTP構成	1
図1.2	FTPサーバ サービスタスクの概念	2
図1.3	各サービスコールとサービスタスクとの通信	3
図2.1	サービスポイントIDの概念	8
図2.2	ユーザプログラムの基本フロー	8
図2.3	サービスポイント状態遷移図	9
表1.1	サポートするFTPサーバ機能	4
表1.2	サポートするFTPコマンド	5
表1.3	FTPサーバ サービスコール一覧	6
表4.1	FTPサーバ イベント一覧	32
表5.1	FTPサーバ レスポンスコード一覧	63

1. 概要

1.1 FTP の機能概要

FTP (ファイル転送プロトコル) はリモートマシンにログインし、ファイルの転送とファイルを管理する機能を提供します。この機能は異なる OS 間におけるファイル操作の非互換性を取り除く為です。ログインに際し、認証を取得 (正式なログイン名、パスワード名が必要) しなければなりません。TCP プロトコルを使用する為、高い信頼性を持っており、大容量のファイル転送に向いています。

FTP は FTP サーバと FTP クライアントで構成され、2つの TCP コネクションを必要とします。1つはサーバ/クライアント間で制御コマンドのやり取りを行う為の制御用の TCP コネクションで、もう1つがデータ転送時にサーバ/クライアント間でデータ転送に使用される TCP コネクションです。2つの TCP コネクションはそれぞれ異なるポート番号によって接続されます。

FTP クライアントが FTP サーバに対してコマンドを送ると、FTP サーバはそのコマンドに対する処理を行い、その処理結果をレスポンスコードとレスポンスメッセージという形式で応答します。コマンドによってはデータ転送が発生し、クライアント、サーバ間でデータ転送が行われます。

FTP で使用できるデータタイプには ASCII とバイナリがあり、ASCII タイプでは転送されたデータがローカル環境のコード体系に自動変換されます。

本「HI.CommunicationEngine FTP サーバ」は、FTP サーバの機能を提供します。ディスク等に対してファイルアクセスを行う場合は、別途ファイルシステムが必要となります。本 FTP サーバにはサンプルソフトとして簡易ファイルシステムを付属しており、RAM を使用した FTP サーバの動作確認が可能です。

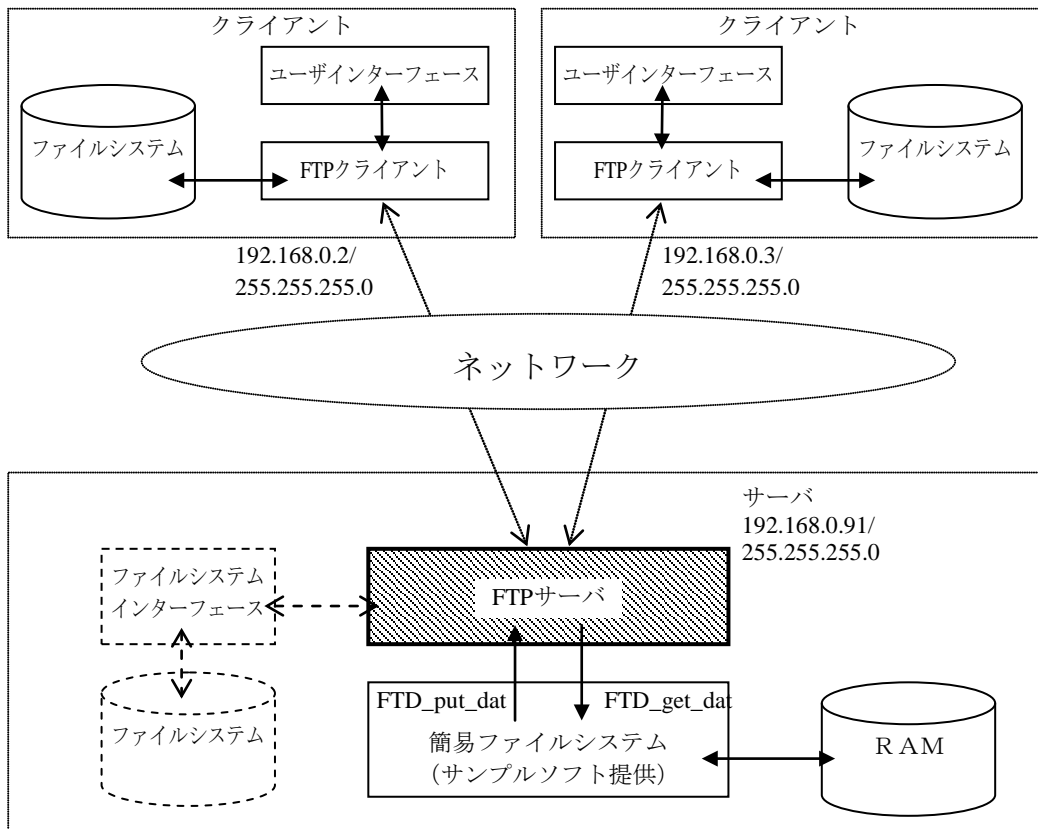


図1.1 FTP構成

1.2 サービスタスクの実現

本FTPサーバのサービスは「タスク」として実現します。図1.2にFTPサーバのサービスタスクの概念を示します。

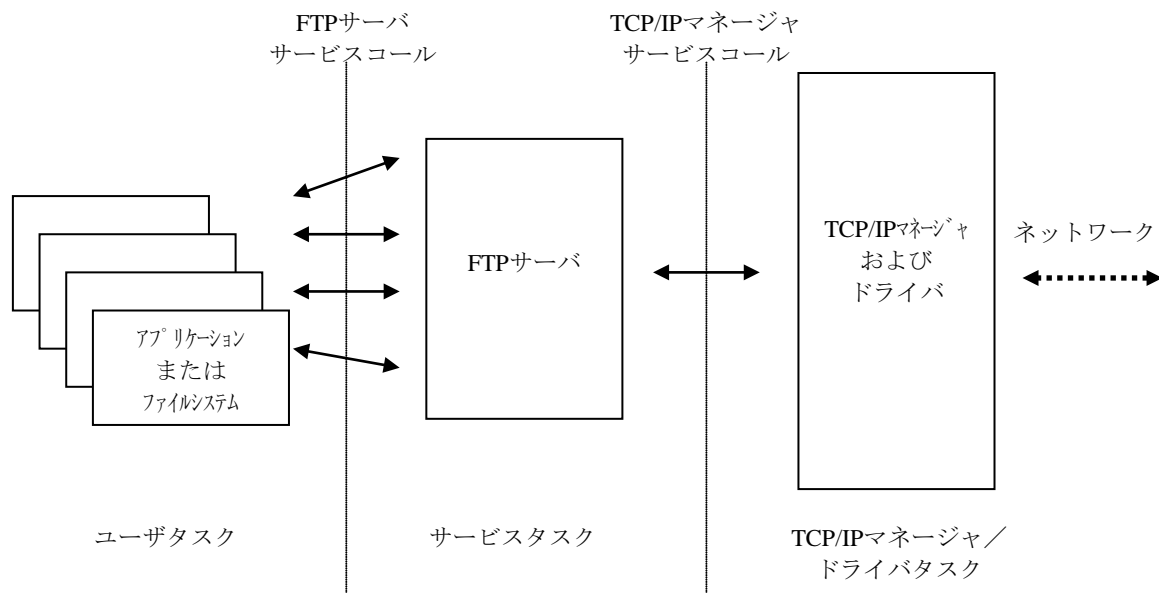


図1.2 FTPサーバ サービスタスクの概念

1.3 サービスコールのしくみ

各サービスコールは拡張サービスコール形式で呼び出されます（FTD_initのみサブルーチン形式）。各サービスコールとサービスタスクとは μ ITRON仕様OSのメールボックス、およびイベントフラグ機能を利用してデータのやり取りを行います。

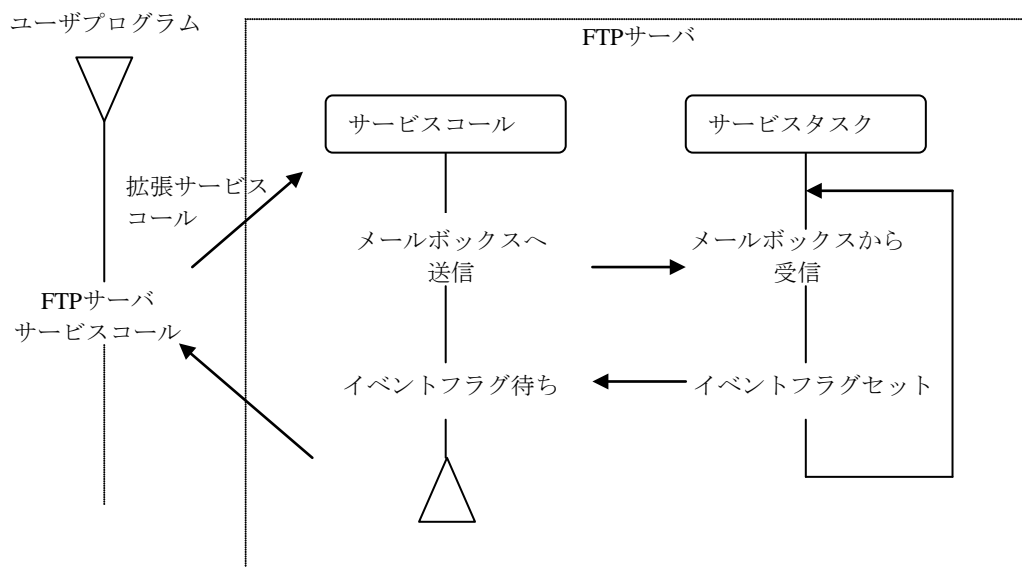


図1.3 各サービスコールとサービスタスクとの通信

1.4 使用するリソース

本FTPサーバでは μ ITRON仕様OSのリソースのうち、1タスク、1メールボックス、1イベントフラグ、および10個の拡張サービスコールを使用します。また、TCP/IPマネージャのリソースのうち、最低2つのTCP通信端点、および最低1つのTCP受付口を使用します。各サービスコールはポーリング、およびタイムアウト待ちをサポートしますが、ノンブロッキング（コールバックによる処理完了通知）はサポートしません。

1.5 サポートする FTP サーバ機能

本 FTP サーバがサポートする FTP 機能を表 1.1 に、FTP コマンドを表 1.2 に示します。
サポート有無の欄に○印があるものがサポートする機能で、×印は未サポートの機能です。

(注) 本 FTP サーバは、RFC959 定義に準拠した FTP クライアントにて動作します。RFC959 定義に準拠しない FTP クライアントでは正常な動作は保証しません。

表1.1 サポートするFTPサーバ機能

項目	機能	説明	サポート有無
データタイプ	ASCII (AN)	ASCII (Non-Print) デフォルトの書式	○
	ASCII (AT)	ASCII (Telnet 書式制御) 書式	×
	ASCII (AC)	ASCII (フォートラン改行制御) 書式	×
	EBCDIC	EBCDIC 形式	×
	IMAGE	イメージ形式	○
	LOCAL 8	論理バイト長 8 ビットのデータ	○
	LOCAL m	論理バイト長 m ビットのデータ	×
転送モード	ストリームモード	ストリームモード	○
	ブロックモード	ブロックモード	×
	圧縮モード	圧縮モード	×
データ構造	ファイル構造	内部的な構造をもたない 1 バイト単位の並びのファイル	○
	レコード構造	シーケンシャルレコード構造のファイル	×
	ページ構造	ページインデックス、ページサイズ、データから構成されるファイル	×
その他	タイムアウト設定	アイドルタイムアウトの指定	○
	Telnet オプション	Telnet オプションの処理	○
	ファイル転送の再開	ストリームモードでのバイトオフセット指定によるファイル転送の再開	○
	リスタートマーカ	ブロックモードでのリスタートマーカによるファイル転送の再開	×
	マルチアドレス	複数の IP アドレスによる接続待ち	×

表1.2 サポートするFTPコマンド

項目	機能	説明	サポート有無
FTP コマンド	USER	ユーザ名を識別する	○
	PASS	パスワードを識別する	○
	ACCT	ユーザアカウントを識別する	○
	CWD/XCWD	カレントディレクトリを変更する	○
	CDUP/XCUP	親ディレクトリへ移動する	○
	SMNT	異なるファイルシステムのデータ構造をマウントする	×
	REIN	制御用のコネクションを残したまま、ユーザ名などのパラメータを初期化する	×
	QUIT	FTPのコネクションを終了する	○
	PORT	クライアント側のデータ転送用コネクションのポート番号を通知する（受動接続）	○
	PASV	サーバ側にデータ転送用コネクションの作成を要求する（能動接続）	○
	TYPE	データタイプを指定する	○
	STRU	ファイルの構造を指定する	○
	MODE	ファイルの転送モードを指定する	○
	RETR	サーバにデータ転送用コネクションを通してファイルを転送するように要求する	○
	STOR	サーバにデータ転送用コネクションを通して送るデータを新規ファイルに書き込むように要求する	○
	STOU	STORと同様であるが、ユニークファイルに書き込む	○
	APPE	サーバにデータ転送用コネクションを通して送るデータを既存ファイルに追加して書き込むように要求する	○
	ALLO	サーバにディスクスペースを確保するように要求する	○
	REST	ファイル転送を再開するように要求する	○
	RNFR	ファイル名変更元を指定する	○
	RNTO	ファイル名変更先を指定する	○
	ABOR	実行中のFTPコマンドを中止する	○
	DELE	ファイルを削除する	○
	RMD/XRMD	ディレクトリを削除する	○
	MKD/XMKD	ディレクトリを作成する	○
	PWD/XPWD	カレント・ディレクトリの送信を要求する	○
	LIST	ディレクトリ一覧の送信を要求する	○
	NLST	詳細なディレクトリ一覧の送信を要求する	○
	SITE	サーバ実装依存のコマンドを送信する	○
	STAT	サーバのOSタイプを取得する	○
	SYST	サーバの状態を要求する	○
	HELP	コマンドの説明を表示する	○
NOOP	何もしない	○	

1.6 FTPサーバ サービスコールの種類

表1.3 FTPサーバ サービスコール一覧

項番	サービスコール名称	説明
1	FTD_init	FTPサーバのサービスを初期化する
2	FTD_start	FTPサーバのサービスを開始する
3	FTD_stop	FTPサーバのサービスを終了する
4	FTD_acp_srv	FTPサーバを接続待ちにする
5	FTD_sht_svp	サービスポイントの接続を切断する
6	FTD_put_dat	FTPクライアントにデータを送信する
7	FTD_get_dat	FTPクライアントからデータを受信する
8	FTD_set_opt	サービスポイントの動作モードを設定する
9	FTD_get_opt	サービスポイントの状態を取得する
10	FTD_can_svp	ペンディングしている処理をキャンセルする
11	<i>FTD_callback</i> (仮称)	FTPサーバのイベントをユーザに通知する (ユーザ実装ルーチン)
12	FTD_evt_end	FTPサーバのイベント処理完了を通知する

2. FTP サーバ サービスコールの使用方法

FTP サーバのサービスコールを使用するには HI.CommunicationEngine TCP/IP マネージャ HI.CommunicationEngine 共通タイマライブラリ、およびドライバプログラムを動作可能な状態にする必要があります。TCP/IP マネージャおよびドライバプログラムの使用方法に関しては、「HI.CommunicationEngineTCP/IP マネージャリファレンスマニュアル」、および各ドライバプログラムのマニュアルを参照してください。タイマライブラリに関しては、「FTP サーバ構築マニュアル」を参照してください。

FTP サーバを使用するユーザプログラムでは次のヘッダファイルをインクルードしてください。

```
#include "tcpip03.h"          /* TCP/IP マネージャ定義ファイル */
#include "ftpd03.h"          /* FTP サーバ定義ファイル */
```

一般的な FTP サーバの使用方法を以下に説明します。

2.1 FTP サーバの基本フロー

- (1) 最初に FTD_init をコールする必要があります。FTD_init はシステムリセットの際などに一度のみコールします。FTD_init にて FTP サーバの各サービスコールが μ ITRON 仕様 OS の拡張サービスコールとして登録されます (図 2.2 ①)。
- (2) FTD_start をコールすると、FTP サーバが使用するタスク、メモリ等のリソースが確保され、各サービスコールが使用可能な状態となります (図 2.2 ②)。
- (3) FTD_acp_srv をコールすることでサービスポイントがクライアントの接続待ち状態となります (図 2.2 ③)。
- (4) FTP クライアントが接続されるとコールバックルーチンによりクライアント接続イベントが通知されます。この時、コールバックのパラメータによりサービスポイント ID がユーザに渡されます。FTP サーバでは接続されたクライアントをサービスポイント ID により区別します。これによって、同時に複数のクライアントからの接続を受け、それぞれに対し別々のサービスを提供することが可能です (図 2.1を参照)。
- (5) クライアントからファイルの送信要求があると、コールバックルーチンによりファイル送信イベントが通知されます。要求されたファイルのデータをクライアントに送信するには FTD_put_dat を使用します (図 2.2 ⑤)。FTD_put_dat はすべてのデータを送信するまで繰り返しコールすることができます。ただし、コールバックルーチン内から FTD_put_dat をコールすることはできません。送信データの終端ではデータ長に 0 を指定して FTD_put_dat をコールする必要があります。
- (6) クライアントからファイルの書き込み要求があると、コールバックルーチンによりファイル受信イベントが通知されます。要求されたファイルのデータを受信するには FTD_get_dat を使用します (図 2.2 ⑥)。FTD_get_dat はすべてのデータを受信するまで繰り返しコールすることができます。ただし、コールバックルーチン内から FTD_get_dat をコールすることはできません。すべてのデータを受信すると FTD_get_dat のリターン値は 0 となります。
- (7) クライアントから接続切断要求があると、コールバックルーチンによりクライアントの切断イベントが通知されます。クライアントとの接続を強制切断するには FTD_sht_svp をコールします (図 2.2 ⑦)。切断されたサービスポイント ID はフリーとなり、以降のクライアントの接続にて再び使用されます。
- (8) FTP サーバを停止する場合は、FTD_stop をコールします。FTP サーバが使用するタスク、メモ

リ等のリソースが解放されます。FTPサーバを停止した後であれば、再度 FTD_start をコールすることができます (図 2.2 ⑨)。

- (9) FTPサーバのイベントはすべてコールバックによりユーザーに通知されます。イベント処理が完了したら FTD_evt_end をコールして FTPサーバに完了通知を行う必要があります。(図 2.2 ④)。

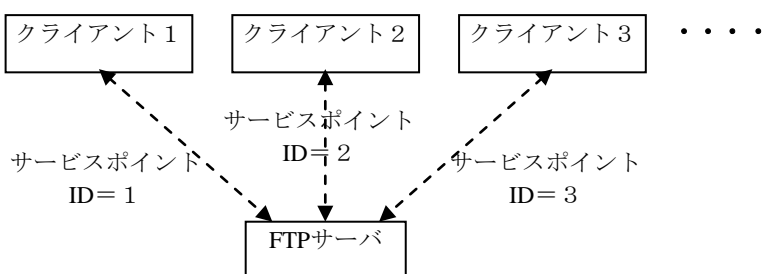


図2.1 サービスポイントIDの概念

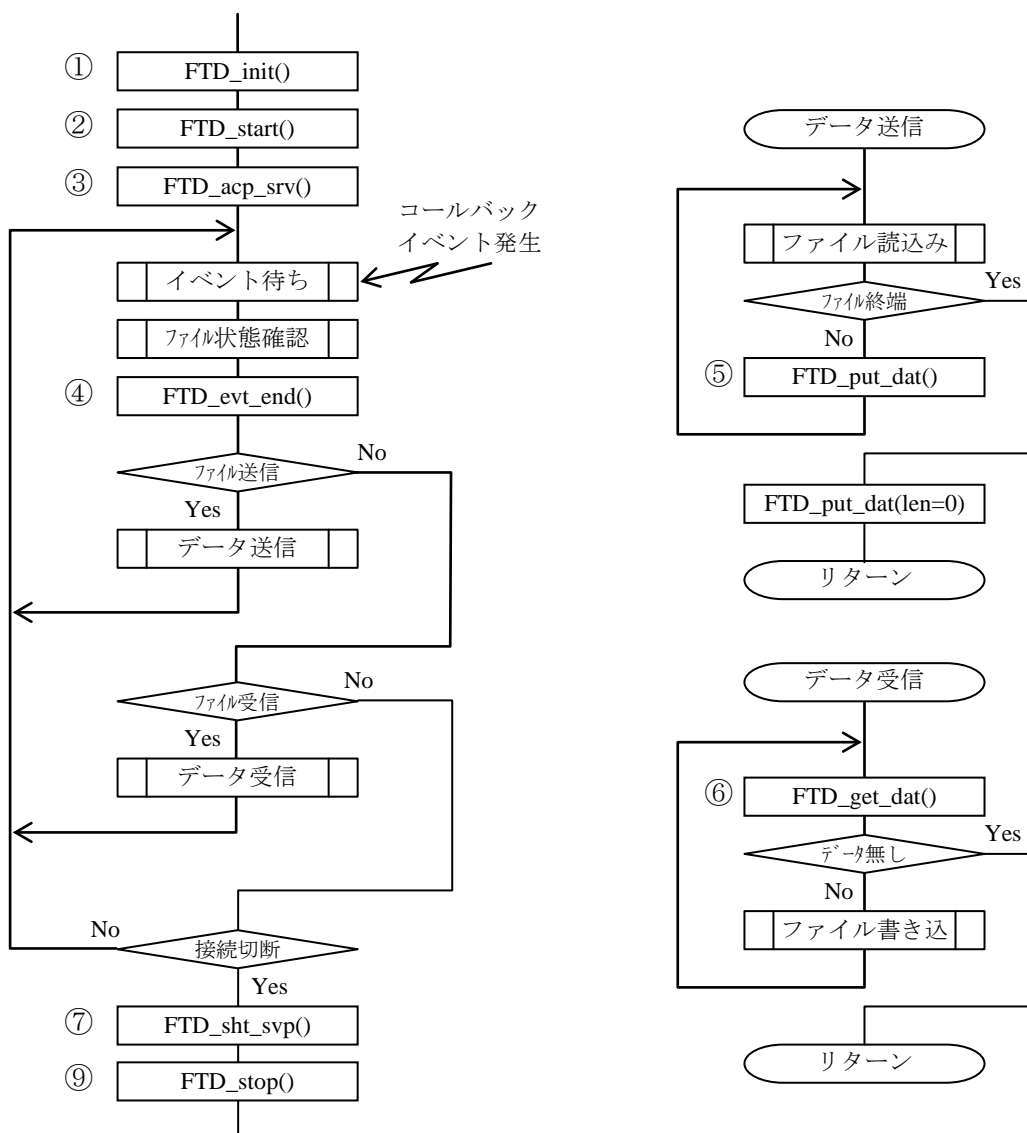


図2.2 ユーザープログラムの基本フロー

2.2 FTP サーバ サービスポイントの状態遷移図

FTP サーバのサービスポイントの状態遷移図を図 2.3 に示します。

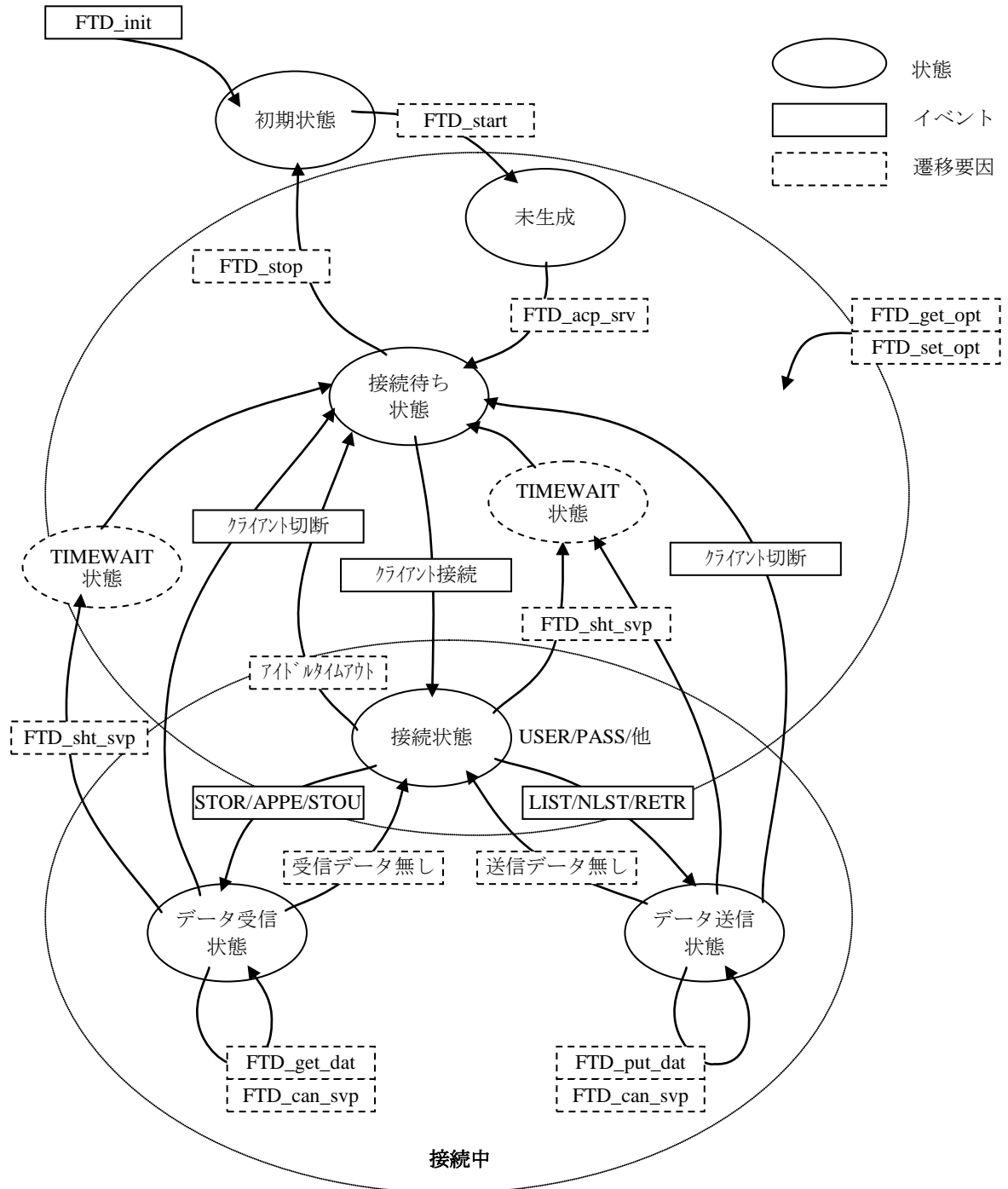


図2.3 サービスポイント状態遷移図

3. FTP サーバ サービスコール

各サービスコールについての説明は次の形式で行います。

No.	サービスコール名	機能	【発行可能なシステム状態*1】
C言語インタフェース			
サービスコール呼出し形式			
パラメータ			
型	パラメータ	パラメータの意味	
・	・	・	
・	・	・	
・	・	・	
リターンパラメータ			
型	パラメータ	パラメータの意味	
・	・	・	
・	・	・	
パケットの構造			
リターン値/エラーコード			
リターン値またはニモニック	リターン値またはエラーコードの意味		
・	・		
・	・		
・	・		
解 説			
・・・・・・・・・・・・・・・・			

論理式では以下の記号を使用します

[A] : Aを省略できることを示します。

(A||B) : AまたはBのどちらか一方を指定できることを示します。

(A|B) : AとBの両方を指定できることを示します。

*1発行可能なシステム状態を以下のアルファベットで示します

T: タスク実行状態

D: ディスパッチ禁止状態

L: CPUロック状態

I: 非タスク部実行状態

なお、各状態の詳細は各ITRON仕様OSのユーザーズマニュアルを参照してください。

発行可能なシステム状態以外の状態でサービスコールを発行した場合、システムの正常な動作は保証されません。

3.1 初期化サービスコール

3.1.1 FTD_init FTP サーバのサービスを初期化する

【T/D/I】

C 言語インタフェース

```
void FTD_init ( void );
```

パラメータ

なし

リターンパラメータ

なし

リターン値/エラーコード

なし

解 説

FTPサーバで使用する内部変数を初期化し、各サービスコールを μ ITRON仕様OSの拡張サービスコールに登録します。

FTPサーバのサービスコールを使用する前に、一度のみ本サービスコールを呼び出す必要があります。

拡張サービスコールの機能コード不足等で拡張サービスコールが登録できなかった場合でもエラーコードは返りません。この場合、FTD_startをコールすると拡張サービスコールが未定義であるエラーコード(E_RSFN)、またはシステムエラー(E_SYS)が返されます。

3.2 基本サービスコール

3.2.1 FTD_start FTP サーバのサービスを開始する

【T】

C 言語インタフェース

```
ER ercd = FTD_start ( T_FTD_STA *pk_sta );
```

パラメータ

T_FTD_STA	*pk_sta;	リソース情報の先頭アドレス
-----------	----------	---------------

リターンパラメータ

ER	ercd	リターン値またはエラーコード
W	pk_sta->memlen	使用するワークメモリの長さ
W	pk_sta->wbuflen	使用するウィンドウバッファ用メモリの長さ
H	pk_sta->perrno	発生したエラーの詳細を示す番号

パケットの構造

```
typedef struct{
    T_FTD_RSC    resource;    リソース情報
    T_FTD_OSC    osconf;     OSコンフィグレーション情報
    H            perrno;     発生したエラーの詳細を示す番号
} T_FTD_STA;

typedef struct{
    ID            maxsvpid;    最大サービスポイントID
    UW           *memadr;     使用するワークメモリの先頭アドレス
    W            memlen;     使用可能なワークメモリの長さ
    UH           *wbufadr;    ウィンドウバッファ用メモリの先頭アドレス (省略可)
    W            wbuflen;    使用可能なウィンドウバッファ用メモリの長さ (省略可)
    INT          csbufsz;     制御コネクションで使用する送信ウィンドウバッファのサイズ
    INT          crbufsz;     制御コネクションで使用する受信ウィンドウバッファのサイズ
    INT          dsbufsz;     データコネクションで使用する送信ウィンドウバッファのサイズ
    INT          drbufsz;     データコネクションで使用する受信ウィンドウバッファのサイズ
} T_FTD_RSC;

typedef struct{
    ATR          tskatr;      FTPサーバのタスク属性
    ID           tskid;       FTPサーバのタスクID番号
    PRI          tskpri;      FTPサーバのタスク優先度
    VP           stkadr;      FTPサーバタスクスタックアドレス
    INT          stksz;       FTPサーバのタスクスタックサイズ
    ID           mbxid;       FTPサーバのメールボックスID番号
    ID           evfid;       FTPサーバのイベントフラグID番号
} T_FTD_OSC;
```

リターン値/エラーコード

E_OK	正常終了
E_PAR	パラメータエラー (pk_staおよびmemadrが0または4の倍数以外、maxsvpid ≤ 0、maxsvpid > 10、wbufadrが奇数、各ウィンドウバッファのサイズが2047以下または奇数)
E_NOMEM	メモリ不足 (必要なメモリが確保できない)
E_OBJ	オブジェクト状態不正 (すでにサービスが開始している)
E_SYS	システムエラー (OSのリソース確保に失敗)

解 説

FTPサーバのサービスを開始します。

本サービスコールにてFTPサーバのタスク生成、およびメールボックス、イベントフラグの生成が行われます。

maxsvpidには使用するサービスポイントIDの最大値を指定します。サービスポイントIDは同時に接続を許可するクライアントの数だけ必要となります。(注1)

FTPサーバでは、必要となるワークメモリを**memadr**で指定されたメモリアドレスから切り出して使用します。切り出したワークメモリの長さを**memlen**に返します。最大サービスポイントIDに比例して使用するワークメモリが多くなります。

必要となるワークメモリのサイズは以下の計算式により算出することができます。

$$\text{ワークメモリサイズ} = 1452 \times \text{maxsvpid} + 400 \quad (\text{byte})$$

wbufadr、**wbuflen**にはウィンドウバッファに使用するメモリアドレス、および長さを指定します。必要となるウィンドウバッファを**wbufadr**で指定されたメモリアドレスから切り出して使用します。切り出したウィンドウバッファの長さを**wbuflen**に返します。最大サービスポイントIDに比例して使用するウィンドウバッファが多くなります。

wbufadr、**wbuflen**に0を指定した場合は**memadr**に指定されたワークメモリを使用します。

ワークメモリは、**FTD_stop**にて解放されます。解放前にワークメモリの内容を書き換えた場合の動作は保証されません。

csbufsz、**crbufsz**には、FTPの制御コネクションにて使用する送信用ウィンドウバッファのサイズ、および受信用ウィンドウバッファのサイズを指定します。**csbufsz**、**crbufsz**は共に2048バイト以上を指定してください。推奨値は2048バイト以上です。

dsbufsz、**drbufsz**には、FTPのデータコネクションにて使用する送信用ウィンドウバッファのサイズ、および受信用ウィンドウバッファのサイズを指定します。**dsbufsz**、**drbufsz**は共に2048バイト以上を指定してください。推奨値は8192バイト以上です。

必要となるウィンドウバッファのサイズは以下の計算式により算出することができます。

$$\text{ウィンドウバッファサイズ} = (\text{csbufsz} + \text{crbufsz} + \text{dsbufsz} + \text{drbufsz}) \times \text{maxsvpid} \quad (\text{byte})$$

tskatr、**tskid**、**tskpri**、**stkadr**、**stksz**には、FTPサーバのタスク属性、タスクID番号、タスク優先度、タスクスタックアドレス、タスクスタックサイズを指定します。**tskid**に0を指定した場合は、OSのID番号自動割付け機能により、ID番号を割り付けます。**stkadr**にNULLを指定した場合は、タスクスタック領域はHI7000/4シリーズのダイナミックスタック領域から割り付けられます。V2.00より前のバージョンのHI7000/4シリーズを使用する場合は、必ずNULLを指定してください。タスクスタックサイズに関しては「スタックサイズ算出マニュアル」を参照してください。

mbxid、**evfid**には、FTPサーバが使用するメールボックスのID番号、およびイベントフラグのID番号を指定します。**mbxid**、**evfid**に0を指定した場合は、OSのID番号自動割付け機能により、ID番号を割り付けます。

指定した**memlen**、**wbuflen**が必要なメモリサイズに満たない場合は、エラーコードとして**E_NOMEM**を返し、必要なメモリサイズを**memlen**、**wbuflen**に返します。

指定したパラメータが不正の場合は、エラーコードとして**E_PAR**を返します。

OSのリソース確保に失敗した場合は、エラーコードとして**E_SYS**を返します。

FTD_initコール時に拡張サービスコールの登録に失敗した場合、およびタイマライブラリが動作していない場合には、エラーコードとして**E_SYS**を返します。タイマライブラリに関しては、「FTPサーバ HI7000シリーズ用構築マニュアル」の「4. HI.CommunicationEngine共通タイマライブラリ」の項を参照してください。

すでにFTPサーバが開始している状態で本サービスコールをコールした場合、エラーコードとして**E_OBJ**を返します。

リターン値がE_PAR、またはE_SYSの場合、発生したエラーの詳細を示す番号をpernoに格納します。pernoに格納される番号の種類を以下に示します。ただし、pk_staが不正なアドレスの場合、pernoには格納しません。

リターン値がE_PARの場合：

perno	エラー内容
1	maxsvpidが11以上、または0以下
2	memadrが0または4の倍数以外
3	wbufadrが奇数
4	csbufsz、crbufsz、dsbufsz、drbufszのいずれかが2047以下または2の倍数以外
9	その他のエラー

リターン値がE_SYSの場合：

perno	エラー内容
1	メールボックスの生成に失敗
2	イベントフラグの生成に失敗
3	タスクの生成に失敗
4	タスクの起動に失敗
5	FTD_initにて拡張サービスコールの登録に失敗
6	タイマライブラリが動作していない
9	その他のエラー

- (1) TCP/IPマネージャの最大TCP通信端点数、最大TCP受付口数、およびTCP/IPマネージャを使用している他プログラムのTCP通信端点、TCP受付口の状況によって作成可能なサービスポイントが最大サービスポイントIDより少なくなる場合があります。FTPサーバでは1サービスポイントにつき、2つのTCP通信端点、1つのTCP受付口を使用します。これらが不足した場合は制御コネクションまたはデータコネクションが接続できなくなります。

3.2.2 FTD_stop

FTP サーバのサービスを終了する

【T】

C 言語インタフェース

```
ER ercd = FTD_stop ( void );
```

パラメータ

なし

リターンパラメータ

ER	ercd	リターン値またはエラーコード
----	------	----------------

リターン値/エラーコード

E_OK	正常終了
------	------

解 説

FTPサーバのサービスを終了し、FTD_startで切り出したワークメモリを解放します。また、FTPサーバのタスク、およびメールボックス、イベントフラグを削除します。

クライアントがまだ接続されている場合はすべて強制切断されます。すべてのサービスポイントに対し、ペンディング中の処理がある場合は強制終了します。強制終了されたサービスコールは、エラーコードとしてE_RLWAIを返します。

3.2.3 FTD_acp_srv FTP サーバを接続待ち状態にする

【T】

C 言語インタフェース

```
ER ercd = FTD_acp_srv ( T_FTD_ACP *pk_acp );
```

パラメータ

T_FTD_ACP	*pk_acp	FTPサーバ接続情報の先頭アドレス
-----------	---------	-------------------

リターンパラメータ

ER	ercd	リターン値またはエラーコード
H	pk_acp->permo	発生したエラーの詳細を示す番号

パケットの構造

```
typedef struct{
    UW    myaddr;      FTPサーバのIPアドレス
    UH    myportc;    FTPサーバの制御コネクションのポート番号 (0の場合、
                    デフォルトポート番号21を使用)
    UH    myportd;    FTPサーバのデータコネクションのポート番号 (0の場合、
                    デフォルトポート番号20を使用)
    FP    callback;   イベント通知コールバックルーチンのアドレス
    H     permo;      発生したエラーの詳細を示す番号
} T_FTD_ACP;
```

リターン値/エラーコード

E_OK	正常終了
E_PAR	パラメータエラー (pk_acpが4の倍数以外、IPアドレスが0xffffffffまたは動作中のIPアドレスと異なる、callbackが0または2の倍数以外、myportcとmyportdが同一番号)
E_NOID	ID番号不足 (TCP通信端点IDが不足、またはTCP受付口IDが不足)
E_OBJ	オブジェクト状態エラー (既に接続待ちになっている、またはポート番号既使用)

解説

FTPサーバを接続待ち状態にします。

クライアントが接続されると、接続を通知するコールバックルーチンがコールされ、その際にサービスポイントIDがパラメータとして渡されます。一度FTD_acp_srvをコールするとFTD_stopがコールされるか、または同時に接続されたクライアントの数が最大サービスポイントIDに達するまでクライアントの接続を受け付けます。(注1)

クライアントが接続されると、クライアントが接続したことを通知するコールバックルーチンがコールされます。コールバックルーチンの詳細に関しては「4 コールバック」の項を参照してください。

myaddrにはFTPサーバが使用するIPアドレスを32ビット (4バイト) で指定します。

myaddrに0を指定した場合は、TCP/IPマネージャに登録されている自IPアドレスが使用されます。

myportc、myportdにはFTPサーバが使用する制御コネクションのポート番号、およびデータコネクションのポート番号を指定します。myportcが0の場合はデフォルトポート番号21を、myportdが0の場合はデフォルトポート番号20を使用します。

指定したパラメータが不正の場合は、エラーコードとしてE_PARを返します。

FTPサーバのIPアドレスに指定可能なアドレスはTCP/IPマネージャに登録した自IPアドレスのみです。それ以外のIPアドレスを指定した場合は、エラーコードとしてE_PARを返します。

TCP/IPマネージャのTCP通信端点IDに空きがない場合、またはTCP/IPマネージャのTCP受付口が確保できなかった場合は、エラーコードとしてE_NOIDを返します。

すでに接続待ちとなっているサービスポイントに対して本サービスコールを発行した場合、エラーコードとしてE_OBJを返します。

リターン値がE_PARの場合、発生したエラーの詳細を示す番号をpermoに格納します。permoに格納される番号の種類を以下に示します。ただし、pk_acpが不正なアドレスの場合、permoには格納しません。

perno	エラー内容
1	callbackが0または2の倍数以外
2	myaddrがブロードキャスト、または動作中のIPアドレスと異なる
3	myportcとmyportdが同一番号
9	その他のエラー

(注1) TCP/IPマネージャの最大通信端点数、およびTCP/IPマネージャを使用している他プログラムの通信端点の使用状況によって同時に接続できるクライアントの数が最大サービスポイントIDより少なくなる場合があります。

3.2.4 FTD_sht_svp サービスポイントの接続を切断する

【T】

C言語インタフェース

```
ER ercd = FTD_sht_svp ( ID svpid , TMO tmout );
```

パラメータ

ID	svpid	サービスポイントID
TMO	tmout	タイムアウト値

リターンパラメータ

ER	ercd	リターン値またはエラーコード
----	------	----------------

リターン値/エラーコード

E_OK	正常終了
E_PAR	パラメータエラー (tmout < -1)
E_ID	不正ID番号 (サービスポイントIDは無効)
E_NOEXS	オブジェクト未生成 (サービスポイントIDは使用されていない)
E_OBJ	オブジェクト状態エラー (接続状態でない、またはデータコネクションが接続中)
E_TMOUT	ポーリング失敗またはタイムアウト
E_RLWAI	処理のキャンセル、待ち状態の強制解除

解 説

接続されているクライアントを切断します。
接続切断が完了した場合、サービスポイントは削除されます。

svpidには切断するサービスポイントIDを指定します。

tmoutにはクライアントからの応答待ち時間を指定します。tmoutに正の値を指定した場合は待ち時間(単位はOSの単位時間と同じ)、TMO_POL(0)を指定した場合はポーリング、TMO_FEVR(-1)を指定した場合は永久待ちとなります。

ポーリング指定、および待ち時間指定によりタイムアウトが発生した場合は、RST送信による強制切断を行い、リターン値としてE_TMOUTを返します。

指定したパラメータが不正の場合は、エラーコードとしてE_PARを返します。

無効なサービスポイントIDを指定した場合は、エラーコードとしてE_IDを返します。

接続状態以外のサービスポイントに対して本サービスコールを発行した場合は、エラーコードとしてE_OBJを返します。

ポーリング指定の場合でも、他のサービスコールがペンディングしている場合、ペンディングが終了するまで切断処理は行われず、待ち状態となります。

3.2.5 FTD_put_dat FTP クライアントにデータを送信する

【T】

C 言語インタフェース

```
ER_UINT ercd = FTD_put_dat ( ID svpid , VP data , INT len , TMO tmout );
```

パラメータ

ID	svpid	サービスポイントID
VP	data	送信するデータの先頭アドレス
INT	len	送信したいデータの長さ
TMO	tmout	タイムアウト値

リターンパラメータ

ER_UINT	ercd	リターン値またはエラーコード
---------	------	----------------

リターン値/エラーコード

正の値	正常終了 (送信バッファに入れたデータの長さ)
E_PAR	パラメータエラー (dataが0、len<0、tmout<-1)
E_ID	不正ID番号 (サービスポイントIDは無効)
E_NOEXS	オブジェクト未生成 (サービスポイントIDは使用されていない)
E_OBJ	オブジェクト状態不正 (データ送信可能状態でない)
E_TMOUT	ポーリング失敗またはタイムアウト
E_RLWAI	処理のキャンセル、待ち状態の強制解除
EV_CLSDAT	データコネクションが切断された

解 説

dataで指定されたアドレスから長さlenのデータをクライアントに送信します。

全てのデータが送信できたら、lenに0を指定してコールすることで、データ送信を終了します。

1回のコールで指定したデータ全てが送信されるとは限りません。全てのデータが送信されるまで本関数を繰り返しコールしてください。

送信バッファに空きがある場合、送信バッファに入れたデータの長さを返します。送信バッファに空きがない場合は待ち状態となります。

サービスポイントがASCIIモードの場合、送信データの改行文字はNTV形式に自動変換されます。

ファイルデータおよびディレクトリリスト等、すべてのデータの送信が完了したら、lenに0を指定して本サービスコールをコールしてください。

svpidには送信に使用するサービスポイントIDを指定します。

tmoutには送信バッファが空くまでの待ち時間を指定します。tmoutに正の値を指定した場合は待ち時間(単位はOSの単位時間と同じ)、TMO_POL(0)を指定した場合はポーリング、TMO_FEVR(-1)を指定した場合は永久待ちとなります。

送信バッファに空きが無い状態にて、ポーリング指定、および待ち時間指定によりタイムアウトが発生した場合、データは送信されず、リターン値としてE_TMOUTを返します。lenに0を指定した場合にタイムアウトが発生した場合は、データコネクションを強制切断し、リターン値としてE_TMOUTを返します。

指定したパラメータが不正の場合は、エラーコードとしてE_PARを返します。

無効なサービスポイントIDを指定した場合は、エラーコードとしてE_IDを返します。

接続状態以外のサービスポイント、またはコールバックルーチン内にて本サービスコールを発行した場合は、エラーコードとしてE_OBJを返します。

制御用のTCPコネクションが強制切断された場合は0が返り、サービスポイントが削除されます。制御用のTCPコネクションが切断された場合、サービスポイントは削除されます。

データ転送用のTCPコネクションが強制切断された場合は、エラーコードとしてEV_CLSDATを返します。この場合、データの送信は継続できません。lenに0を指定してデータ送信を終了してください。

3.2.6 FTD_get_dat FTP クライアントからデータを受信する

【T】

C 言語インタフェース

```
ER_UINT ercd = FTD_get_dat ( ID svpid , VP data, INT len , TMO tmout );
```

パラメータ

ID	svpid	サービスポイントID
VP	data	受信データを格納する領域の先頭アドレス
INT	len	受信データを格納する領域の長さ
TMO	tmout	タイムアウト値

リターンパラメータ

ER_UINT	ercd	リターン値またはエラーコード
---------	------	----------------

リターン値/エラーコード

0以上	正常終了 (取得したデータのサイズ)
E_PAR	パラメータエラー (dataが0、len<0、tmout<-1)
E_ID	不正ID番号 (サービスポイントIDは無効)
E_NOEXS	オブジェクト未生成 (サービスポイントIDは使用されていない)
E_OBJ	オブジェクト状態不正 (データ受信可能状態でない)
E_TMOUT	ポーリング失敗またはタイムアウト
E_RLWAI	処理のキャンセル、待ち状態の強制解除
EV_CLSDAT	データ接続が切断された

解 説

クライアントから送られたデータを受信し、dataで指定されたアドレスに格納します。受信データが無くなるか受信したデータの長さがlenに達するまでデータを取得し、リターン値として取得したデータの長さを返します。受信データが無い場合は待ち状態となります。データをすべて受信した場合、リターン値として0を返します。

全てのデータを受信する前にlenに0を指定してコールした場合、データの受信を強制終了します。

サービスポイントがASCIIモードの場合、受信データの改行文字はFTPサーバのテキストファイル形式に自動変換されます。

svpidには受信に使用するサービスポイントIDを指定します。

tmoutにはクライアントからの受信待ち時間を指定します。tmoutに正の値を指定した場合は待ち時間(単位はOSの単位時間と同じ)、TMO_POL(0)を指定した場合はポーリング、TMO_FEVR(-1)を指定した場合は永久待ちとなります。

受信データが無い状態にて、ポーリング指定、および待ち時間指定によりタイムアウトが発生した場合、リターン値としてE_TMOUTを返します。

指定したパラメータが不正の場合は、エラーコードとしてE_PARを返します。

無効なサービスポイントIDを指定した場合は、エラーコードとしてE_IDを返します。

接続状態以外のサービスポイント、またはコールバックルーチン内にて本サービスコールを発行した場合は、エラーコードとしてE_OBJを返します。

制御用のTCP接続が強制切断された場合は0が返り、サービスポイントが削除されます。制御用のTCP接続が切断された場合、サービスポイントは削除されます。

データ転送用のTCP接続が強制切断された場合は、エラーコードとしてEV_CLSDATを返します。この場合、データの受信は継続できません。

3.2.7 FTD_set_opt サービスポイントの動作モードを設定する

【T】

C 言語インタフェース

```
ER ercd = FTD_set_opt ( ID svpid , INT optname , VP optval );
```

パラメータ

ID	svpid	サービスポイントID
INT	optname	設定する動作モードの種類
VP	optval	設定内容を格納した領域の先頭アドレス

リターンパラメータ

ER	ercd	リターン値またはエラーコード
----	------	----------------

リターン値/エラーコード

E_OK	正常終了
E_PAR	パラメータエラー (optnameの値が不正、optvalが4の倍数以外)
E_ID	不正ID番号 (サービスポイントIDは無効)
E_NOEXS	オブジェクト未生成 (サービスポイントIDは使用されていない)
E_OBJ	オブジェクト状態不正 (設定可能状態でない)

解 説

サービスポイントの動作モードを変更します。

svpidには変更するサービスポイントIDを指定します。svpidに0を指定した場合は、FTPサーバのデフォルト設定を変更します。(デフォルト設定の変更の場合、すでに接続されているサービスポイントには影響しません。)

optnameには、FTD_MOD_CPORT (制御コネクションの動作モードの設定)、FTD_MOD_DPORT (データコネクションの動作モードの設定)、FTD_MOD_ASC (テキストファイル形式の設定)、FTD_MOD_TOUT (アイドルタイムアウトの設定) が指定できます。

指定したパラメータが不正の場合は、エラーコードとしてE_PARを返します。

無効なサービスポイントIDを指定した場合は、エラーコードとしてE_IDを返します。

接続状態のサービスポイント、またはコールバックルーチン内にて本サービスコールを発行した場合は、エラーコードとしてE_OBJを返します。

(1) optname = FTD_MOD_CPORT(1) (制御コネクションの動作モードの設定)

制御コネクションの動作モードを指定します。デフォルト設定では、cepmode = TMD_TIMWDS | TMD_CLSNFFS、keepime = 120秒、kretrytime = 30秒、kretrycnt = 4 回、sockTIMEWAIT = 1 (秒) および sockMSLはTCP/IPマネージャの構築情報としてtcpipConfTbl に設定したtcpMSL (最大セグメント寿命) を基準に、sockMSL = tcpMSL (秒) となっています。

svpidには0のみ指定できます。(FTPサーバのデフォルト設定を変更します。すでに接続されているサービスポイントには影響しません。)

optvalに指定するパケットの構造 :

```
typedef struct{
    UW      cepmode;      通信端点制御オプション
    H      keepime;      キープアライブパケット開始タイマ (秒)
    H      kretrytime;    キープアライブパケットのリトライ間隔 (秒)
    H      kretrycnt;    キープアライブパケットのリトライ回数
    H      sockMSL;      最大セグメント生存時間タイマ (秒)
    H      sockTIMEWAIT; TIMEWAIT (時間待機) 状態タイマ (秒)
    UH     optTYPE;      オプションタイプ ( 0、1 または 2 )
    H      minRTO;      最小セグメント再送時間タイマ (秒)
    H      maxRTO;      最大セグメント再送時間タイマ (秒)
}
```

UB	sockTOS;	Type Of Service値
UB	sockTTL;	Time To Live値
H	synRTO;	SYNセグメント再送時間タイマ初期値 (秒)
H	datRTO;	データセグメント再送時間タイマ初期値 (秒)
UH	reserve;	システム予約 (0 固定)

}T_FTD_MODSVP;

optTYPEには、オプション指定構造体のタイプ情報を設定します。
optTYPE = 0のときは、optTYPE以降の構造がないものとみなされます。
optTYPE = 1のときは、optTYPE以降のminRTO, maxRTO, sockTOS, sockTTLがあるものとみなされます。
optTYPE = 2のときは、optTYPE以降のminRTO, maxRTO, sockTOS, sockTTL, synRTO, datRTO, reserveがあるものとみなされます。

通信端点制御オプション(cepmode)には、次の値を設定します。

ビット0,1:シリーウィンドウ回避機能

TMD_ACKAUTO (H'00000000)	独自制御
TMD_ACKNAGLE (H'00000001)	Nagleアルゴリズムによる制御
TMD_ACKNODLY (H'00000002)	遅延制御なし

ビット2:キープアライブ機能

TMD_KEEPOFF (H'00000000)	キープアライブ機能を使用しない
TMD_KEEPPON (H'00000004)	キープアライブ機能を使用する

ビット3:

TMD_DEFRTTO (H'00000000)	デフォルトのRTOを使用する
TMD_REQRTTO (H'00000008)	指定したRTOを使用する

ビット4:

TMD_DEFTOS (H'00000000)	デフォルトのTOSを使用する
TMD_REQTOS (H'00000010)	指定したTOSを使用する

ビット5:

TMD_DEFMTTL (H'00000000)	デフォルトのTTLを使用する
TMD_REQMTTL (H'00000020)	指定したTTLを使用する

ビット8:

TMD_TIMWEN (H'00000000)	TIMEWAIT状態有効モード
TMD_TIMWDS (H'00000100)	TIMEWAIT状態禁止モード

ビット9:

TMD_CLSNFDL (H'00000000)	クローズ通知遅延モード
TMD_CLSNFFS (H'00000200)	クローズ通知即時モード

ビット10:

TMD_SYNRTODEF (H'00000000)	現在のSYN再送時間初期値を使用する
TMD_SYNRTOREQ (H'00000400)	指定したSYN再送時間初期値を使用する

ビット11:

TMD_DATRTODEF (H'00000000)	現在のデータ再送時間初期値を使用する
TMD_DATRTOREQ (H'00000800)	指定したデータ再送時間初期値を使用する

ビット12:

TMD_RTOSCLSEC (H'00000000)	RTOスケール秒基準モード
TMD_RTOSCLCYC (H'00001000)	RTOスケール周期タイマイベント基準モード

ビット13:

TMD_SLOWON (H'00000000)	スロースタート機能有効モード
TMD_SLOWOFF (H'00002000)	スロースタート機能無効モード

cepmodeのビット8が0 (TIMEWAIT状態有効モード) の場合は、FINWAIT-2状態やCLOSING状態からCLOSED状態に移行する場合はTIMEWAIT状態を経由します。

cepmodeのビット8が1 (TIMEWAIT状態禁止モード) の場合は、FINWAIT-2状態やCLOSING状態からCLOSED状態に移行する場合にTIMEWAIT状態を経由しません。

cepmodeのビット9が0 (クローズ通知遅延モード) の場合は、TCP/IPマネージャはCLOSED状態に移行後0.5秒経過後にクローズを認識します。

cepmodeのビット9が1 (クローズ通知即時モード) の場合は、TCP/IPマネージャはCLOSED状態に移行

後直ちにクローズを認識します。

cepmodeのビット10が0 (SYN再送時間デフォルトモード) の場合は、現在の値を使用します。通信端点生成直後のSYN再送時間の初期値にはデフォルトの値として3秒を使用します。

cepmodeのビット10が1 (SYN再送時間指定モード) の場合は、SYN再送時間の初期値にはsynRTOに指定した値を使用します。

cepmodeのビット11が0 (セグメント再送時間デフォルトモード) の場合は、現在の値を使用します。通信端点生成直後のセグメント再送時間の初期値にはデフォルトの値として3秒を使用します。

cepmodeのビット11が1 (セグメント再送時間指定モード) の場合は、セグメント再送時間の初期値にはdatRTOに指定した値を使用します。

cepmodeのビット12が0 (RTOスケール秒基準モード) の場合は、セグメント再送時間の単位として秒が用いられます。

cepmodeのビット12が1 (RTOスケール周期タイマイベント基準モード) の場合は、セグメント再送時間の単位としてタイマイベント周期が用いられます (例えば10ミリ秒毎にタイマイベントが発生する場合、単位は10ミリ秒となります)。RTOスケール周期タイマイベント基準モードを指定した場合は、ビット3、ビット10、ビット11を1にし、minRTO、maxRTO、synRTO、datRTOを指定してください。これらのビットが0の場合は、エラーコードとしてE_PARを返します。

cepmodeのビット13が0 (スロースタート機能有効モード) の場合は、受信確認 (ACK) 無しに送信できるパケットの個数をコネクション直後に1とし、受信確認を受け取るごとに増加させます。そのため、送信のパフォーマンスはコネクション直後は低速ですが、受信確認 (ACK) を受け取るごとに加速します。尚、送信できるパケットの個数は、送信ウィンドウバッファサイズ/MSSを超える事はありません。

cepmodeのビット13が1 (スロースタート機能無効モード) の場合は、前述のスロースタート機能を使わないため、コネクション直後からある程度の送信速度が得られますが、ネットワーク上で輻輳が発生する場合は、このモードは使用しないでください。

keepetimeに0を指定すると、以前に設定された (または初期値) のキープアライブパケット開始タイマをそのまま使用します (書き換えしません)。

kretrytimeに0を指定すると、以前に設定された (または初期値) のキープアライブパケット繰返しタイマをそのまま使用します (書き換えしません)。

sockMSLに0を指定すると、以前に設定された (または初期値) の最大セグメント寿命タイマをそのまま使用します (書き換えしません)。

sockTIMEWAITに0を指定すると、以前に設定された (または初期値) のTIME-WAIT (時間待機: TCPSTAT_TIMEWAIT) 状態タイマをそのまま使用します (書き換えしません)。

ただし、sockMSLに0以外の値を指定し、sockTIMEWAITに0を指定すると、sockTIMEWAITにsockMSL×2の値を自動で設定します。

minRTOとmaxRTOは、cepmodeにTMD_REQRTOがセットされているときだけ有効です。セグメント再送時間の計算結果がminRTOより小さくなる場合はminRTOを、maxRTOより大きくなる場合はmaxRTOをセグメント再送時間として使用します。

minRTOはmaxRTO以下でなければなりません。minRTOがmaxRTOより大きい場合は、エラーコードとしてE_PARを返します。

また、maxRTOはsockMSL以下でなければなりません。maxRTOがsockMSLより大きい場合は、エラーコードとしてE_PARを返します。

なお、minRTOに0を指定した場合は、最小セグメント再送時間を0.5秒に設定します。

(実際の時間は、設定値マイナス0.1秒から設定値の間の時間になります)

sockTOSは、cepmodeにTMD_REQTOSがセットされているときだけ有効です。sockTOSに指定したTOS値を、送信するIPパケットヘッダに設定します。

sockTTLは、cepmodeにTMD_REQTTLがセットされているときだけ有効です。sockTTLに指定したTTL値を、送信するIPパケットヘッダに設定します。ただし、sockTTLに0を指定した場合は、TMD_REQTTLオフとして扱い、デフォルトのTTLを、送信するIPパケットヘッダに設定します。

TOSおよびTTLのデフォルト値としては、man_ip_startもしくはman_ip_creadrで指定したT_IP_INFのdefaulttosとdefaultttlの値を使用します。

synRTOは、cepemodeにSYNRTREQがセットされているときだけ有効です。

尚、synRTOはminRTO以上でsockMSL以下でなければなりません。synRTOに指定した値がこの範囲に無い場合は、エラーコードとしてE_PARを返します。

datRTOは、cepemodeにDATRTREQがセットされているときだけ有効です。

尚、datRTOはminRTO以上でmaxRTO以下でなければなりません。datRTOに指定した値がこの範囲に無い場合は、エラーコードとしてE_PARを返します。

reserveはシステム予約領域です。必ず0を設定してください。

(2) optname = FTD_MOD_DPORT(2) (データ接続の動作モードの設定)

データ接続の動作モードを指定します。デフォルト設定では、cepemode = TMD_TIMWDS | TMD_CLSNFFS、keepstime = 120秒、kretrytime = 30秒、kretrycnt = 4回、sockTIMEWAIT = 1 (秒) および sockMSLはTCP/IPマネージャの構築情報としてtcpipConfTbl に設定したtcpMSL (最大セグメント寿命) を基準に、sockMSL = tcpMSL (秒) となっています。

svpidには0のみ指定できます。(FTPサーバのデフォルト設定を変更します。すでに接続されているサービスポイントには影響しません。)

optvalに指定するパケットの構造に関しては「(1) 制御接続の動作モードの設定」を参照してください。

(3) optname = FTD_MOD_ASC(3) (テキストファイル形式の設定)

サービスポイントのテキストファイル形式を設定します。デフォルト設定では FTD_MOD_CRLF となっています。

FTPにてASCIIモードでデータ転送を行う場合、データ中の改行文字は送信側でサービスポイントのテキストファイル形式からNTV形式に変換され、受信側でNTV形式からサービスポイントのテキストファイル形式に自動変換されます。

svpidに0を指定した場合、FTPサーバのデフォルト設定を変更します。(すでに接続されているサービスポイントには影響しません。)

optvalに指定するパケットの構造：

```
typedef struct{
    UW      mode;          ASCIIモードの設定
}T_FTD_MODASC;
```

modeには次の値が指定できます。

FTD_MOD_CRLF (H'00000001)	サービスポイントの改行文字をCR+LFに設定
FTD_MOD_LF (H'00000002)	サービスポイントの改行文字をLFに設定
FTD_MOD_CR (H'00000004)	サービスポイントの改行文字をCRに設定

(4) optname = FTD_MOD_TOUT(4) (アイドルタイムアウトの設定)

サービスポイントのアイドルタイムアウト時間を設定します。デフォルト設定では0 (タイムアウト無し) となっています。

アイドルタイムアウトに指定された時間の間、クライアントからの操作、またはファイルデータの送受信がない場合は、自動的に接続を強制切断します。

svpidに0を指定した場合、FTPサーバのデフォルト設定を変更します。(すでに接続されているサービスポイントには影響しません。)

optvalに指定するパケットの構造：

```
typedef struct{
    UW      tmout;        アイドル時のタイムアウト時間 (単位：OSの単位時間)
```

```
}T_FTD_MODTOUT;
```

tmoutに0を指定した場合は、タイムアウトによる接続切断は行いません。

3.2.8 FTD_get_opt サービスポイントの状態を取得する

【T】

C言語インタフェース

```
ER ercd = FTD_get_opt ( ID svpid , INT optname , VP optval );
```

パラメータ

ID	svpid	サービスポイントID
INT	optname	取得情報の種類
VP	optval	情報を格納する領域の先頭アドレス

リターンパラメータ

ER	ercd	リターン値またはエラーコード
----	------	----------------

リターン値/エラーコード

正の値	正常終了 (取得情報の長さ)
E_PAR	パラメータエラー (optnameが不正、optvalが4の倍数以外)
E_ID	不正ID番号 (サービスポイントIDは無効)
E_NOEXS	オブジェクト未生成 (サービスポイントIDは使用されていない)

解 説

サービスポイントの現在の状態を取得します。

svpidには取得するサービスポイントIDを指定します。

optnameには、FTD_MOD_SRVINF(1) (FTPサーバの状態取得)、FTD_MOD_SVPINF(2) (サービスポイントの状態取得) が指定できます。

指定したパラメータが不正の場合は、エラーコードとしてE_PARを返します。

無効なサービスポイントIDを指定した場合は、エラーコードとしてE_IDを返します。

(1) optname = FTD_MOD_SRVINF(1) (FTPサーバの状態取得)

FTPサーバの現在の状態を取得します。

svpidには0のみ指定可能です。

optvalに格納されるパケットの構造:

```
typedef struct{
    UW    svrst;           FTPサーバの状態
    ID    repid;          接続待ちとなっているTCP受付口のID
    ID    cepid;          接続待ちとなっているTCP通信端点のID
    T_IPV4EP c_myaddr;    サーバのIPアドレスおよびポート番号 (制御コネクション)
    T_IPV4EP d_myaddr;    サーバのIPアドレスおよびポート番号 (データコネクション)
} T_FTD_SRVINF;

typedef struct{
    UW    ipaddr;         IPアドレス
    UH    portno;        ポート番号
} T_IPV4EP;
```

svrstには、次の値を返します。

FTD_SRV_ACCEPT (H'00000001)	接続可 (新規クライアントは接続可能)
FTD_SRV_NOACCEPT (H'00000000)	接続不可 (TCP通信端点IDまたはサービスポイントIDに空きが無い場合、新規クライアントは接続不可)

repid、cepidにはFTPサーバが接続待ちで使用しているTCP/IPマネージャのTCP受付口ID、およびTCP通信端点IDを返します。FTPサーバが接続待ち状態でない場合、0を返します。

(2) optname = FTD_MOD_SVPINF(2) (サービスポイントの状態取得)
 svpstで指定されたサービスポイントの現在の状態を取得します。

optvalに格納されるパケットの構造：

```
typedef struct{
    UW      svpst;           サービスポイントの状態
    ID      ccepid;         制御コネクションで使用しているTCP通信端点のID
    ID      drepid;        データコネクションで使用しているTCP受付口のID
    ID      dcepid;        データコネクションで使用しているTCP通信端点のID
    T_IPV4EP c_dstaddr;     クライアントのIPアドレスおよびポート番号 (制御コネクション)
    T_IPV4EP d_dstaddr;     クライアントのIPアドレスおよびポート番号 (データコネクション)
} T_FTD_SVPINF;

typedef struct{
    UW      ipaddr;         IPアドレス
    UH      portno;        ポート番号
} T_IPV4EP;
```

svpstには、次の値を返します。

TEV_FTD_NOEVT (H'00000000)	イベントなし
TEV_FTD_CONN (H' 00001201)	接続イベント中
TEV_FTD_QUIT (H' 00001203)	接続切断イベント中
TEV_FTD_USER (H' 00001204)	ユーザ名の識別要求イベント中
TEV_FTD_PASS (H' 00001205)	パスワードの識別要求イベント中
TEV_FTD_ACCT (H' 00001206)	ユーザアカウントの識別要求イベント中
TEV_FTD_CWD (H' 00001207)	カレントディレクトリの変更要求イベント中
TEV_FTD_CDUP (H' 00001208)	親ディレクトリへの変更要求イベント中
TEV_FTD_RMD (H' 00001209)	ディレクトリの削除要求イベント中
TEV_FTD_MKD (H' 0000120a)	ディレクトリの作成要求イベント中
TEV_FTD_PWD (H' 0000120b)	カレントディレクトリの送信要求イベント中
TEV_FTD_LIST (H' 0000120c)	ファイル一覧の送信要求イベント中
TEV_FTD_NLST (H' 0000120d)	詳細なファイル一覧の送信要求イベント中
TEV_FTD_DELE (H' 0000120e)	ファイルの削除要求イベント中
TEV_FTD_RNFR (H' 0000120f)	ファイル名の変更要求イベント中 (変更元ファイル名)
TEV_FTD_RNTO (H' 00001210)	ファイル名の変更要求イベント中 (変更後ファイル名)
TEV_FTD_RETR (H' 00001211)	ファイルの送信要求イベント中
TEV_FTD_STOR (H' 00001212)	新規ファイルへの書き込み要求イベント中
TEV_FTD_APPE (H' 00001213)	既存ファイルへの追加書き込み要求イベント中
TEV_FTD_STOU (H' 00001214)	ユニークファイルへの書き込み要求イベント中
TEV_FTD_ALLO (H' 00001215)	ディスクスペース確保要求イベント中
TEV_FTD_SITE (H' 00001216)	サーバ依存コマンド受信イベント中

ccepidには制御コネクションで使用しているTCP/IPマネージャのTCP通信端点IDを返します。

drepidにはデータコネクションの接続 (パッシブ接続時) で使用しているTCP/IPマネージャのTCP受付口を返します。データコネクションがパッシブ接続待ち以外の場合は、0を返します。

dcepidにはデータコネクションで使用しているTCP/IPマネージャのTCP通信端点IDを返します。データコネクションが接続待ち、または接続状態でない場合、0を返します。

3.2.9 FTD_can_svp ペンディングしている処理をキャンセルする

【T】

C言語インタフェース

```
ER ercd = FTD_can_svp(ID svpid, FN fnccd);
```

パラメータ

ID	svpid	サービスポイントID
FN	fnccd	キャンセルするサービスコールの機能コード

リターンパラメータ

ER	ercd	リターン値またはエラーコード
----	------	----------------

リターン値/エラーコード

E_OK	正常終了
E_PAR	パラメータエラー (fnccdが指定可能な機能コードでない)
E_ID	不正ID番号 (サービスポイントIDは無効)
E_NOEXS	オブジェクト未生成 (サービスポイントIDは使用されていない)
E_OBJ	オブジェクト状態エラー (fnccdで指定した処理はペンディングしていない)

解説

svpidで示されたサービスポイントに対し、fnccdで示されたペンディング中の処理をキャンセルします。キャンセルされたサービスコールは、エラーコードとしてE_RLWAIを返します。

キャンセル可能な処理のサービスコール名と、それを指定する機能コードは次の通りです。また、TFN_FTD_ALL(0)を指定すると、指定したサービスポイントでペンディングしているすべての処理をキャンセルします。

サービスコール名	機能コード
FTD_sht_svp	TFN_FTD_SHT_SVP(-0x1205)
FTD_put_dat	TFN_FTD_PUT_DAT(-0x1206)
FTD_get_dat	TFN_FTD_GET_DAT(-0x1207)
すべて	TFN_FTD_ALL(0)

3.2.10 FTD_evt_end FTP サーバイベントの処理完了を通知する

【T】

C 言語インタフェース

```
ER ercd = FTD_evt_end ( ID svpid , ER rtncd);
```

パラメータ

ID	svpid	サービスポイントID
ER	rtncd	イベント処理結果

リターンパラメータ

ER	ercd	リターン値またはエラーコード
----	------	----------------

リターン値/エラーコード

E_OK	正常終了
E_ID	不正ID番号 (サービスポイントIDは無効)
E_NOEXS	オブジェクト未生成 (サービスポイントIDは使用されていない)
E_OBJ	オブジェクト状態エラー (イベントはペンディングしていない)

解 説

コールバックにより通知されたイベント処理が完了、またはファイル転送の準備ができたことをFTPサーバに通知する場合に使用します。

svpidにはイベントの処理が完了したサービスポイントIDを指定します。

rtncdにはイベント処理結果を設定します。各イベントにより設定できる処理結果が異なりますので、詳細は「4.2 コールバックイベント」の項を参照してください。

処理結果の種類：

100～159：	イベント処理を開始、次の応答を待つ
200～259：	イベント処理正常終了
300～359：	追加情報が必要
400～459：	イベントは正常に完了しなかった
500～559：	イベントは実行できない

本サービスコールはコールバックルーチン内にて呼び出すことも可能です。FTD_evt_endがコールされるまで、同じサービスポイントの次のイベント (TEV_FTD_DISC、TEV_FTD_DPCN、TEV_FTD_ABOR以外) は発生しません。

イベントがペンディングしていない場合はE_OBJを返します。

4. コールバック

コールバックは、FTPサーバ内で発生した事象をアプリケーションプログラムに伝えるために用いられます。コールバックルーチンはユーザが作成し、そのアドレスのみがFTPサーバに渡されるため、モジュール名はユーザ任意となります。ここでは仮称として**FTD_callback**とします。

発生した事象に対する処理をコールバックルーチン内にて行うこともできますが、コールバックルーチンはFTPサーバのタスクレベルにてコールされます。そのためイベントに対する処理に時間がかかる場合、その間FTPサーバのタスクは動作できず、他の処理に遅延が発生してしまいます。コールバックルーチンではできるだけ早くリターンし、時間がかかる処理（ファイルシステムからのデータ読み込み／書き込み等）はユーザタスクにて処理するようにしてください。

コールバックルーチン内にて、**FTD_evt_end**以外のFTPサーバのサービスコールをコールすることはできません。コールバックルーチン内にて**FTD_evt_end**以外のサービスコールをコールした場合の動作は保証されません。

4.1 コールバックルーチン

4.1.1 *FTD_callback* FTP サーバイベントをユーザに通知する

【T】

C 言語インタフェース

```
ER ercd = callback ( ID svpid , UW evcd , VP parblk );
```

パラメータ

ID	svpid	サービスポイントID
UW	evcd	コールバックイベント番号
VP	parblk	パラメータを格納した領域の先頭アドレス

パケットの構造

各イベントの説明を参照

リターンパラメータ

ER	ercd	リターン値またはエラーコード (未使用)
----	------	----------------------

解 説

FTPサーバのコールバックイベントが発生した場合にコールされます。コールバックルーチンはFTPサーバのタスクレベルにてサブルーチンコールされます。

*svpid*にはイベントが発生したFTPサーバのサービスポイントIDが渡されます。

*evcd*にはイベントの内容を示すコールバックイベント番号が渡されます。

コールバックイベントの詳細については「4.2 コールバックイベント」の項を参照してください。

4.2 コールバックイベント

コールバックでユーザに通知されるコールバックイベントを説明します。

表4.1 FTPサーバイベント一覧

FTPサーバイベント番号	内容
TEV_FTD_CONN (0x1201)	FTPクライアントの接続イベント
TEV_FTD_DISC (0x1202)	FTPクライアントの切断イベント
TEV_FTD_QUIT (0x1203)	FTPクライアントの切断要求イベント
TEV_FTD_USER (0x1204)	ユーザ名の識別要求イベント
TEV_FTD_PASS (0x1205)	パスワードの識別要求イベント
TEV_FTD_ACCT (0x1206)	ユーザアカウントの識別要求イベント
TEV_FTD_CWD (0x1207)	カレントディレクトリの変更要求イベント
TEV_FTD_CDUP (0x1208)	親ディレクトリへの変更要求イベント
TEV_FTD_RMD (0x1209)	ディレクトリの削除要求イベント
TEV_FTD_MKD (0x120A)	ディレクトリの作成要求イベント
TEV_FTD_PWD (0x120B)	カレントディレクトリの送信要求イベント
TEV_FTD_LIST (0x120C)	ファイル一覧の送信要求イベント
TEV_FTD_NLST (0x120D)	詳細なファイル一覧の送信要求イベント
TEV_FTD_DELE (0x120E)	ファイルの削除要求イベント
TEV_FTD_RNFR (0x120F)	ファイル名の変更要求イベント (変更元ファイル名)
TEV_FTD_RNTO (0x1210)	ファイル名の変更要求イベント (変更後ファイル名)
TEV_FTD_RETR (0x1211)	ファイルの送信要求イベント
TEV_FTD_STOR (0x1212)	新規ファイルへの書き込み要求イベント
TEV_FTD_APPE (0x1213)	既存ファイルへの追加書き込み要求イベント
TEV_FTD_STOU (0x1214)	ユニークファイルへの書き込み要求イベント
TEV_FTD_ALLO (0x1215)	ディスクスペース確保要求イベント
TEV_FTD_SITE (0x1216)	サーバ依存コマンド受信イベント
TEV_FTD_STAT (0x1217)	サーバの状態送信要求イベント
TEV_FTD_HELP (0x1218)	サーバのコマンドヘルプ送信要求イベント
TEV_FTD_ABOR (0x1219)	コマンド中断イベント
TEV_FTD_DPCN (0x121A)	データ接続の接続開始イベント
TEV_FTD_ERROR (0x12FF)	エラーイベント

(1) TEV_FTD_CONN FTPクライアントの接続イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UW    ipaddr;      クライアントのIPアドレス
    UH    portno;     クライアントのポート番号
} T_IPV4EP;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントが接続したことを通知します。

svpidにはクライアントが接続したサービスポイントのIDが渡されます。

ipaddr、portnoには、接続したクライアントのIPアドレス、およびポート番号が渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。

FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_220 (220) 接続を許可する
FTD_RCD_421 (421) 接続を許可しない

クライアントの接続を許可する場合は、イベント処理結果としてFTD_RCD_220を指定してください。

接続を許可しない場合は、イベント処理結果としてFTD_RCD_421を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバの接続確立時のレスポンスには以下の種類があります。

220 Service ready for new user.

421 Service not available, closing control connection.

(2) TEV_FTD_DISC

FTP クライアントの切断イベント

パラメータブロックの構造 (parblk)

無し

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントとの接続が切断されたことを通知します。

svpidには切断されたサービスポイントのIDが渡されます。

本イベントの処理完了時には、FTD_evt_endをコールする必要はありません。

(3) TEV_FTD_QUIT

FTP クライアントの切断要求イベント

パラメータブロックの構造 (parblk)

無し

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントから切断要求があったことを通知します。

svpidには切断要求があったサービスポイントのIDが渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。

FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_221 (221) 接続を切断する

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのQUITコマンドのレスポンスには以下の種類があります。

221 Service closing control connection.

530 Not logged in.

(4) TEV_FTD_USER ユーザ名の識別要求イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UB      *p_name;      ユーザ名文字列が格納された領域の先頭アドレス
    INT     len;         ユーザ名文字列の長さ
} T_FTD_CBNAME;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントから認証の為のユーザ名を受信したことを通知します。

svpidにはユーザ名を受信したサービスポイントのIDが渡されます。

p_nameには受信したユーザ名文字列の先頭アドレスが渡されます。

lenには受信したユーザ名文字列の長さが渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。

FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_230 (230)	ユーザのログインを許可する
FTD_RCD_331 (331)	ログインにはパスワードが必要
FTD_RCD_332 (332)	ログインにはアカウント情報が必要
FTD_RCD_421 (421)	サービスは無効である

ユーザ名が正しいと判断し、ログインを許可する場合は、イベント処理結果としてFTD_RCD_230を指定してください。

パスワードを要求する場合は、イベント処理結果としてFTD_RCD_331を指定してください。

ログインを許可しない場合は、イベント処理結果としてFTD_RCD_421を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのUSERコマンドのレスポンスには以下の種類があります。

```
230 User logged in, proceed.
501 Syntax error in parameters or arguments.
421 Service not available, closing control connection.
331 User name okay, need password.
332 Need account for login.
```

(5) TEV_FTD_PASS パスワードの識別要求イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UB      *p_name;      パスワード文字列が格納された領域の先頭アドレス
    INT     len;         パスワード文字列文字列の長さ
} T_FTD_CBNAME;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントから認証の為のパスワードを受信したことを通知します。

svpidにはパスワードを受信したサービスポイントのIDが渡されます。
p_nameには受信したパスワード文字列の先頭アドレスが渡されます。
lenには受信したパスワード文字列の長さが渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。
FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_230 (230)	ユーザのログインを許可する
FTD_RCD_202 (202)	パスワードは不要、ログインを許可する
FTD_RCD_332 (332)	ログインにはアカウント情報が必要
FTD_RCD_421 (421)	サービスは無効である

パスワードが正しいと判断し、ログインを許可する場合は、イベント処理結果としてFTD_RCD_230を指定してください。

ログインを許可しない場合は、イベント処理結果としてFTD_RCD_421を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのPASSコマンドのレスポンスには以下の種類があります。

```
230 User logged in, proceed.
202 Command not implemented, superfluous at this site.
501 Syntax error in parameters or arguments.
503 Bad sequence of commands.
421 Service not available, closing control connection.
332 Need account for login.
```

(6) TEV_FTD_ACCT ユーザアカウントの識別要求イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UB      *p_name;      ユーザアカウント文字列が格納された領域の先頭アドレス
    INT     len;         ユーザアカウント文字列の長さ
} T_FTD_CBNAME;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントから認証の為のユーザアカウント情報を受信したことを通知します。

svpidにはアカウント情報を受信したサービスポイントのIDが渡されます。
p_nameには受信したユーザアカウント文字列の先頭アドレスが渡されます。
lenには受信したユーザアカウント文字列の長さが渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。
FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_230 (230)	ユーザのログインを許可する
FTD_RCD_202 (202)	アカウントは不要、ログインを許可する
FTD_RCD_421 (421)	サービスは無効である

アカウントが正しいと判断し、ログインを許可する場合は、イベント処理結果としてFTD_RCD_230を指定してください。

ログインを許可しない場合は、イベント処理結果としてFTD_RCD_421を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのACCTコマンドのレスポンスには以下の種類があります。

```
230 User logged in, proceed.
202 Command not implemented, superfluous at this site.
501 Syntax error in parameters or arguments.
503 Bad sequence of commands.
421 Service not available, closing control connection.
```

(7) TEV_FTD_CWD

カレントディレクトリの変更要求イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UB      *p_name;      ディレクトリ名文字列が格納された領域の先頭アドレス
    INT     len;         ディレクトリ名文字列の長さ
} T_FTD_CBNAME;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントからカレントディレクトリの変更要求を受けたことを通知します。

svpidには要求を受けたサービスポイントのIDが渡されます。

p_nameには変更するディレクトリ名文字列の先頭アドレスが渡されます。

lenには変更するディレクトリ名文字列の長さが渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。

FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_257 (257)	カレントディレクトリを変更した
FTD_RCD_421 (421)	サービスは無効である
FTD_RCD_550 (550)	ディレクトリが存在しない、またはアクセス権がない

カレントディレクトリを変更できた場合は、イベント処理結果としてFTD_RCD_257を指定してください。

エラーが発生した場合は、該当するイベント処理結果を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのCWDコマンドのレスポンスには以下の種類があります。

```
257 "PATHNAME" is current directry.
501 Syntax error in parameters or arguments.
421 Service not available, closing control connection.
530 Not logged in.
550 Requested action not taken.
```

(8) TEV_FTD_CDUP 親ディレクトリへの変更要求イベント

パラメータブロックの構造 (parblk)

なし

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントから親ディレクトリへの変更要求を受けたことを通知します。

svpidには要求を受けたサービスポイントのIDが渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。

FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_257 (257)	カレントディレクトリを変更した
FTD_RCD_421 (421)	サービスは無効である
FTD_RCD_550 (550)	ディレクトリが存在しない、またはアクセス権がない

カレントディレクトリを変更できた場合は、イベント処理結果としてFTD_RCD_257を指定してください。

エラーが発生した場合は、該当するイベント処理結果を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのCDUPコマンドのレスポンスには以下の種類があります。

257 "PATHNAME" is current directry.
501 Syntax error in parameters or arguments.
421 Service not available, closing control connection.
530 Not logged in.
550 Requested action not taken.

(9) TEV_FTD_RMD

ディレクトリの削除要求イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UB      *p_name;      ディレクトリ名文字列が格納された領域の先頭アドレス
    INT     len;         ディレクトリ名文字列の長さ
} T_FTD_CBNAME;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントからディレクトリの削除要求を受けたことを通知します。

svpidには要求を受けたサービスポイントのIDが渡されます。

p_nameには削除するディレクトリ名文字列の先頭アドレスが渡されます。

lenには削除するディレクトリ名文字列の長さが渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。

FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_250 (250)	ファイル操作は正常終了した
FTD_RCD_421 (421)	サービスは無効である
FTD_RCD_550 (550)	ファイルが存在しない、またはアクセス権がない

ディレクトリを削除できた場合は、イベント処理結果としてFTD_RCD_250を指定してください。

エラーが発生した場合は、該当するイベント処理結果を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのRMDコマンドのレスポンスには以下の種類があります。

250 Requested file action okay, completed.
501 Syntax error in parameters or arguments.
421 Service not available, closing control connection.
530 Not logged in.
550 Requested action not taken.

(10) TEV_FTD_MKD

ディレクトリの作成要求イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UB      *p_name;      ディレクトリ名文字列が格納された領域の先頭アドレス
    INT     len;         ディレクトリ名文字列の長さ
} T_FTD_CBNAME;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントからディレクトリの作成要求を受けたことを通知します。

svpidには要求を受けたサービスポイントのIDが渡されます。

p_nameには作成するディレクトリ名文字列の先頭アドレスが渡されます。

lenには作成するディレクトリ名文字列の長さが渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。

FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_257 (257)	ディレクトリを作成した
FTD_RCD_421 (421)	サービスは無効である
FTD_RCD_550 (550)	ディレクトリが作成できない、またはアクセス権がない

ディレクトリを作成できた場合は、イベント処理結果としてFTD_RCD_257を指定してください。

エラーが発生した場合は、該当するイベント処理結果を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのMKDコマンドのレスポンスには以下の種類があります。

- 257 "PATHNAME" is created.
- 501 Syntax error in parameters or arguments.
- 421 Service not available, closing control connection.
- 530 Not logged in.
- 550 Requested action not taken.

(11) TEV_FTD_PWD

カレントディレクトリの送信要求イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UB      *p_name;      カレントディレクトリ名文字列を格納するアドレス
    INT     *p_len;      カレントディレクトリ名文字列の長さを格納するアドレス
} T_FTD_CBPATH;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントからカレントディレクトリ名の送信要求を受けたことを通知します。

p_nameが示す領域に送信するカレントディレクトリ名文字列を、p_lenが示す領域に送信するカレントディレクトリ名文字列の長さを格納してください。

カレントディレクトリ名文字列は最大512文字まで格納することができます。512文字以上格納した場合の動作は保証されません。

svpidには要求を受けたサービスポイントのIDが渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。

FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_257 (257)	ファイル操作は正常終了した
FTD_RCD_421 (421)	サービスは無効である
FTD_RCD_550 (550)	アクセス権がない

カレントディレクトリ名を送信可能な場合は、イベント処理結果としてFTD_RCD_257を指定してください。

エラーが発生した場合は、該当するイベント処理結果を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのPWDコマンドのレスポンスには以下の種類があります。

257 "PATHNAME" is current directry.
501 Syntax error in parameters or arguments.
421 Service not available, closing control connection.
550 Requested action not taken.

(12) TEV_FTD_LIST

ファイル一覧の送信要求イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UB      *p_name;      ディレクトリパス名文字列が格納された領域の先頭アドレス
    INT     len;          ディレクトリパス名文字列の長さ
} T_FTD_CBNAME;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントからファイル一覧の送信要求を受けたことを通知します。

svpidには要求を受けたサービスポイントのIDが渡されます。

p_nameには要求されたディレクトリパス名文字列の先頭アドレスが渡されます。

lenには要求されたディレクトリパス名文字列の長さが渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。

FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_150 (150)	ファイル操作は正常終了した
FTD_RCD_421 (421)	サービスは無効である
FTD_RCD_450 (450)	要求されたファイル操作を行うことができない (システムビジー)
FTD_RCD_451 (451)	要求された操作は中断した (ローカルエラー)

ファイル一覧を送信可能な場合は、イベント処理結果としてFTD_RCD_150を指定し、続けてFTD_put_datにてファイル一覧を送信してください。(注: コールバックルーチン内ではFTD_put_datは発行できません。)

ファイル一覧の例:

```
task1
task2
task3
task4
tcp
test.jpg
test2.jpg
test3.jpg
```

エラーが発生した場合は、該当するイベント処理結果を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのLISTコマンドのレスポンスには以下の種類があります。

```
125 Data connection already open; transfer starting.
150 File status okay; about to open data connection.
450 Requested file action not taken.
501 Syntax error in parameters or arguments.
421 Service not available, closing control connection.
530 Not logged in.
```

ファイル一覧送信完了時のレスポンスには以下の種類があります。

```
226 Closing data connection.
425 Can't open data connection.
426 Connection closed; transfer aborted.
451 Requested action aborted. Local error in processing.
```

(13) TEV_FTD_NLST

詳細なファイル一覧の送信要求イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UB      *p_name;      ディレクトリパス名文字列が格納された領域の先頭アドレス
    INT     len;         ディレクトリパス名文字列の長さ
} T_FTD_CBNAME;
```

リターン値/エラーコード

0 正常終了 (未使用)

解説

クライアントから詳細なファイル一覧の送信要求を受けたことを通知します。

svpidには要求を受けたサービスポイントのIDが渡されます。

p_nameには要求されたディレクトリパス名文字列の先頭アドレスが渡されます。

lenには要求されたディレクトリパス名文字列の長さが渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。

FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

- FTD_RCD_150 (150) ファイル操作は正常終了した
- FTD_RCD_421 (421) サービスは無効である
- FTD_RCD_450 (450) 要求されたファイル操作を行うことができない (システムビジー)
- FTD_RCD_451 (451) 要求された操作は中断した (ローカルエラー)

ファイル一覧を送信可能な場合は、イベント処理結果としてFTD_RCD_150を指定し、続けてFTD_put_datにて詳細なファイル一覧を送信してください。(注: コールバックルーチン内ではFTD_put_datは発行できません。)

詳細なファイル一覧の例:

```
05-10-02 12:57PM          3693399 task1
05-10-02 12:58PM          3693399 task2
05-10-02 01:00PM          3693399 task3
05-10-02 01:01PM          3693399 task4
10-16-01 02:54PM          <DIR>      tcp
12-25-01 11:20AM          26060 test.jpg
12-25-01 11:31AM          26060 test2.jpg
12-25-01 11:20AM          26060 test3.jpg
```

エラーが発生した場合は、該当するイベント処理結果を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのNLSTコマンドのレスポンスには以下の種類があります。

- 125 Data connection already open; transfer starting.
- 150 File status okay; about to open data connection.
- 450 Requested file action not taken.
- 501 Syntax error in parameters or arguments.
- 421 Service not available, closing control connection.
- 530 Not logged in.

ファイル一覧送信完了時のレスポンスには以下の種類があります。

- 226 Closing data connection.
- 425 Can't open data connection.
- 426 Connection closed; transfer aborted.
- 451 Requested action aborted. Local error in processing.

(14) TEV_FTD_DELE

ファイルの削除要求イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UB      *p_name;      ファイル名文字列が格納された領域の先頭アドレス
    INT     len;         ファイル名文字列の長さ
} T_FTD_CBNAME;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントからファイルの削除要求を受けたことを通知します。

svpidには要求を受けたサービスポイントのIDが渡されます。

p_nameには要求されたファイル名文字列の先頭アドレスが渡されます。

lenには要求されたファイル名文字列の長さが渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。

FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_250 (250)	ファイル操作は正常終了した
FTD_RCD_421 (421)	サービスは無効である
FTD_RCD_450 (450)	要求されたファイル操作を行うことができない (システムビジー)
FTD_RCD_451 (451)	要求された操作は中断した (ローカルエラー)
FTD_RCD_550 (550)	ファイルが存在しない、またはアクセス権がない

ファイルを削除できた場合は、イベント処理結果としてFTD_RCD_250を指定してください。

エラーが発生した場合は、該当するイベント処理結果を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのDELEコマンドのレスポンスには以下の種類があります。

```
250 Requested file action okay, completed.
450 Requested file action not taken.
550 Requested action not taken.
501 Syntax error in parameters or arguments.
421 Service not available, closing control connection.
530 Not logged in.
```

(15) TEV_FTD_RNFR

ファイル名の変更要求イベント (変更元ファイル名)

パラメータブロックの構造 (parblk)

```
typedef struct{
    UB      *p_name;      変更前のファイル名文字列が格納された領域の先頭アドレス
    INT     len;         変更前のファイル名文字列の長さ
} T_FTD_CBNAME;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントからファイル名の変更要求 (変更元ファイル名) を受けたことを通知します。
通常、このイベントの次にTEV_FTD_RNTO (ファイル名の変更要求イベント (変更後ファイル名)) イベントが発生します。

svpidには要求を受けたサービスポイントのIDが渡されます。
p_nameには変更元のファイル名文字列の先頭アドレスが渡されます。
lenには変更元のファイル名文字列の長さが渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。
FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_350 (350)	ファイル操作は正常終了した
FTD_RCD_421 (421)	サービスは無効である
FTD_RCD_450 (450)	要求されたファイル操作を行うことができない (システムビジー)
FTD_RCD_550 (550)	ファイルが存在しない、またはアクセス権がない

ファイル名を変更可能な場合は、イベント処理結果としてFTD_RCD_350を指定してください。
エラーが発生した場合は、該当するイベント処理結果を指定してください。
上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。
FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのRNFRコマンドのレスポンスには以下の種類があります。

450 Requested file action not taken.
550 Requested action not taken.
501 Syntax error in parameters or arguments.
421 Service not available, closing control connection.
530 Not logged in.
350 Requested file action pending further information.

(16) TEV_FTD_RNTO

ファイル名の変更要求イベント (変更後ファイル名)

パラメータブロックの構造 (parblk)

```
typedef struct{
    UB      *p_name;      変更後のファイル名文字列が格納された領域の先頭アドレス
    INT     len;         変更後のファイル名文字列の長さ
} T_FTD_CBNAME;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントからファイル名の変更要求 (変更元ファイル名) を受けたことを通知します。

svpidには要求を受けたサービスポイントのIDが渡されます。

p_nameには変更後のファイル名文字列の先頭アドレスが渡されます。

lenには変更後のファイル名文字列の長さが渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。

FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_250 (250)	ファイル操作は正常終了した
FTD_RCD_421 (421)	サービスは無効である
FTD_RCD_532 (532)	ファイル操作にはアカウント情報が必要
FTD_RCD_553 (553)	ファイル名が不正

ファイル名を変更できた場合は、イベント処理結果としてFTD_RCD_250を指定してください。

エラーが発生した場合は、該当するイベント処理結果を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのRNTOコマンドのレスポンスには以下の種類があります。

250 Requested file action okay, completed.
532 Need account for storing files.
553 Requested action not taken.
501 Syntax error in parameters or arguments.
503 Bad sequence of commands.
421 Service not available, closing control connection.
530 Not logged in.

(17) TEV_FTD_RETR

ファイルの送信要求イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UB      *p_name;      ファイル名文字列が格納された領域の先頭アドレス
    INT     len;          ファイル名文字列の長さ
    UW     offset;       ファイル先頭からのバイトオフセット
} T_FTD_CBFILE;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントからファイルの送信要求を受けたことを通知します。

svpidには要求を受けたサービスポイントのIDが渡されます。

p_nameには要求されたファイル名文字列の先頭アドレスが渡されます。

lenには要求されたファイル名文字列の長さが渡されます。

offsetにはデータ送信を開始するファイルの先頭からのバイトオフセットが渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。

FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_150 (150)	ファイル操作は正常終了した
FTD_RCD_421 (421)	サービスは無効である
FTD_RCD_450 (450)	要求されたファイル操作を行うことができない (システムビジー)
FTD_RCD_451 (451)	要求された操作は中断した (ローカルエラー)
FTD_RCD_550 (550)	ファイルが存在しない、またはアクセス権がない
FTD_RCD_553 (553)	ファイル名が不正

ファイルデータの送信準備ができた場合は、イベント処理結果としてFTD_RCD_150を指定し、続けてFTD_put_datにてファイルのデータを送信してください。(注: コールバックルーチン内ではFTD_put_datは発行できません。)

エラーが発生した場合は、該当するイベント処理結果を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのRETRコマンドのレスポンスには以下の種類があります。

```
125 Data connection already open; transfer starting.
150 File status okay; about to open data connection.
450 Requested file action not taken.
550 Requested action not taken.
501 Syntax error in parameters or arguments.
421 Service not available, closing control connection.
530 Not logged in.
```

ファイル一覧送信完了時のレスポンスには以下の種類があります。

```
226 Closing data connection.
425 Can't open data connection.
426 Connection closed; transfer aborted.
451 Requested action aborted. Local error in processing.
```

(18) TEV_FTD_STOR

新規ファイルへの書き込み要求イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UB      *p_name;      ファイル名文字列が格納された領域の先頭アドレス
    INT     len;          ファイル名文字列の長さ
    UW     offset;       ファイル先頭からのバイトオフセット
} T_FTD_CBFILE;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントから新規ファイルへの書き込み要求を受けたことを通知します。

svpidには要求を受けたサービスポイントのIDが渡されます。

p_nameには要求されたファイル名文字列の先頭アドレスが渡されます。

lenには要求されたファイル名文字列の長さが渡されます。

offsetにはデータ受信を開始するファイルの先頭からのバイトオフセットが渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。

FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_150 (150)	ファイル操作は正常終了した
FTD_RCD_421 (421)	サービスは無効である
FTD_RCD_450 (450)	要求されたファイル操作を行うことができない (システムビジー)
FTD_RCD_451 (451)	要求された操作は中断した (ローカルエラー)
FTD_RCD_452 (452)	要求されたファイル操作は行われなかった (十分な記憶容量がない)
FTD_RCD_532 (532)	ファイル書き込みにはアカウント情報が必要
FTD_RCD_550 (550)	ファイルが存在しない、またはアクセス権がない
FTD_RCD_552 (552)	要求されたファイル操作は中断した (確保された記憶容量を超えた)
FTD_RCD_553 (553)	ファイル名が不正

ファイルデータの受信準備ができた場合は、イベント処理結果としてFTD_RCD_150を指定し、続けてFTD_get_datにてファイルのデータを受信してください。(注: コールバックルーチン内ではFTD_get_datは発行できません。)

エラーが発生した場合は、該当するイベント処理結果を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのSTORコマンドのレスポンスには以下の種類があります。

```
125 Data connection already open; transfer starting.
150 File status okay; about to open data connection.
532 Need account for storing files.
450 Requested file action not taken.
452 Requested action not taken.
553 Requested action not taken.
501 Syntax error in parameters or arguments.
421 Service not available, closing control connection.
530 Not logged in.
```


ファイル一覧送信完了時のレスポンスには以下の種類があります。

226 Closing data connection.

425 Can't open data connection.

426 Connection closed: transfer aborted.

451 Requested action aborted. Local error in processing.

552 Requested file action aborted.

(19) TEV_FTD_APPE

既存ファイルへの書き込み要求イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UB      *p_name;      ファイル名文字列が格納された領域の先頭アドレス
    INT     len;          ファイル名文字列の長さ
    UW     offset;       ファイル先頭からのバイトオフセット
} T_FTD_CBFILE;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントから既存ファイルへの書き込み要求を受けたことを通知します。

svpidには要求を受けたサービスポイントのIDが渡されます。

p_nameには要求されたファイル名文字列の先頭アドレスが渡されます。

lenには要求されたファイル名文字列の長さが渡されます。

offsetにはデータ書き込みを開始するファイルの先頭からのバイトオフセットが渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。

FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_150 (150)	ファイル操作は正常終了した
FTD_RCD_421 (421)	サービスは無効のため、接続を切断する
FTD_RCD_450 (450)	要求されたファイル操作は行われなかった (システムビジー)
FTD_RCD_451 (451)	要求された操作は中断した (ローカルエラー)
FTD_RCD_452 (452)	要求されたファイル操作は行われなかった (十分な記憶容量がない)
FTD_RCD_532 (532)	ファイル書き込みにはアカウント情報が必要
FTD_RCD_550 (550)	ファイルが存在しない、またはアクセス権がない
FTD_RCD_552 (552)	要求されたファイル操作は中断した (確保された記憶容量を超えた)
FTD_RCD_553 (553)	ファイル名が不正

ファイルデータの受信準備ができた場合は、イベント処理結果としてFTD_RCD_150を指定し、続けてFTD_get_datにてファイルのデータを受信してください。(注: コールバックルーチン内ではFTD_get_datは発行できません。)

エラーが発生した場合は、該当するイベント処理結果を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのSTORコマンドのレスポンスには以下の種類があります。

```
125 Data connection already open; transfer starting.
150 File status okay; about to open data connection.
532 Need account for storing files.
450 Requested file action not taken.
550 Requested action not taken.
452 Requested action not taken.
553 Requested action not taken.
501 Syntax error in parameters or arguments.
421 Service not available, closing control connection.
530 Not logged in.
```

ファイル一覧送信完了時のレスポンスには以下の種類があります。

226 Closing data connection.

425 Can't open data connection.

426 Connection closed: transfer aborted.

451 Requested action aborted. Local error in processing.

552 Requested file action aborted.

(20) TEV_FTD_STOU

ユニークファイルへの書き込み要求イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UB      *p_name;      ユニークファイル名文字列を格納するアドレス
    INT     *p_len;      ユニークファイル名文字列の長さを格納するアドレス
} T_FTD_CBPATH;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントからユニークファイルへの書き込み要求を受けたことを通知します。

p_nameが示す領域に書き込むユニークファイル名文字列を、p_lenが示す領域にユニークファイル名文字列の長さを格納してください。

ユニークファイル名文字列は最大512文字まで格納することができます。512文字以上格納した場合の動作は保証されません。

svpidには要求を受けたサービスポイントのIDが渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。

FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_150 (150)	ファイル操作は正常終了した
FTD_RCD_421 (421)	サービスは無効である
FTD_RCD_451 (451)	要求された操作は中断した (ローカルエラー)
FTD_RCD_452 (452)	要求されたファイル操作は行われなかった (十分な記憶容量がない)
FTD_RCD_532 (532)	ファイル書き込みにはアカウント情報が必要
FTD_RCD_552 (552)	要求されたファイル操作は中断した (確保された記憶容量を超えた)

ファイルデータの受信準備ができた場合は、イベント処理結果としてFTD_RCD_150を指定し、続けてFTD_get_datにてファイルのデータを受信してください。(注: コールバックルーチン内ではFTD_get_datは発行できません。)

エラーが発生した場合は、該当するイベント処理結果を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのSTOUコマンドのレスポンスには以下の種類があります。

```
125 Data connection already open; transfer starting.
150 File status okay; about to open data connection.
532 Need account for storing files.
450 Requested file action not taken.
452 Requested action not taken.
553 Requested action not taken.
501 Syntax error in parameters or arguments.
421 Service not available, closing control connection.
530 Not logged in.
```

ファイル一覧送信完了時のレスポンスには以下の種類があります。

```
226 Closing data connection.
425 Can't open data connection.
426 Connection closed; transfer aborted.
451 Requested action aborted. Local error in processing.
552 Requested file action aborted.
```

(21) TEV_FTD_ALLO

ファイル書き込みスペースの確保要求イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UW      size;          確保する容量 (論理バイト数)
} T_FTD_CBSIZE;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントからファイル書き込みスペースの確保要求を受けたことを通知します。

svpidには要求を受けたサービスポイントのIDが渡されます。

sizeには確保する容量が論理バイト数で渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。

FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_200 (200)	ファイル操作は正常終了した
FTD_RCD_202 (202)	コマンドは実装されていない、このサイトでは不要
FTD_RCD_421 (421)	サービスは無効である

ファイル書き込みスペースが確保できた場合は、イベント処理結果としてFTD_RCD_200を指定してください。

エラーが発生した場合は、該当するイベント処理結果を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのALLOコマンドのレスポンスには以下の種類があります。

```
200 Command okay.
202 Command not implemented, superfluous at this site.
501 Syntax error in parameters or arguments.
504 Command not implemented for that parameter.
421 Service not available, closing control connection.
530 Not logged in.
```

(22) TEV_FTD_SITE

サーバ依存コマンドの受信イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UB      *p_cmd;      コマンド文字列が格納された領域の先頭アドレス、および
                        レスポンスメッセージを格納する領域の先頭アドレス
    INT     *p_clen;     コマンド文字列の長さが格納された領域のアドレス、および
                        レスポンスメッセージの長さを格納するアドレス
} T_FTD_CBCMD;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントからサーバ依存のコマンドを受けたことを通知します。

p_cmdが示す領域にクライアントに送信するレスポンスメッセージを、p_clenが示す領域にレスポンスメッセージの長さを格納してください。

レスポンスメッセージは最大512文字まで格納することができます。512文字以上格納した場合の動作は保証されません。レスポンスメッセージの終端には必ず改行コード (CR+LF) を付加してください。

svpidには要求を受けたサービスポイントのIDが渡されます。

p_cmdには受信したコマンド文字列の先頭アドレスが渡されます (レスポンスメッセージを格納する領域と同一です)。

p_clenには受信したコマンド文字列の長さが格納されたアドレスが渡されます (レスポンスメッセージの長さを格納する領域と同一です)。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。

cmdおよびclenに0を指定した場合、デフォルトのレスポンスメッセージを送信します。

レスポンスメッセージの例：

```
XXXX command successful.
```

FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

FTD_RCD_200 (200)	ファイル操作は正常終了した
FTD_RCD_202 (202)	コマンドは実装されていない、このサイトでは不要
FTD_RCD_500 (500)	コマンドが認識できない
FTD_RCD_501 (501)	引数やパラメータに文法エラーがある

処理が正常に完了した場合は、イベント処理結果としてFTD_RCD_200を指定してください。

エラーが発生した場合は、該当するイベント処理結果を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのSITEコマンドのレスポンスには以下の種類があります。

```
200 Command successful.
202 Command not implemented, superfluous at this site.
501 Syntax error in parameters or arguments.
530 Not logged in.
```

(23) TEV_FTD_STAT

サーバの状態送信要求イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UB      *p_name;      ディレクトリパス名文字列が格納された領域の先頭アドレス
    INT     len;         ディレクトリパス名文字列の長さ
} T_FTD_CBNAME;
```

リターン値/エラーコード

0 正常終了 (未使用)

解説

クライアントからサーバ状態 (ディレクトリステータスまたはファイルステータス) の送信要求を受けたことを通知します。

svpidには要求を受けたサービスポイントのIDが渡されます。
 p_nameには要求されたディレクトリパス名文字列の先頭アドレスが渡されます。
 lenには要求されたディレクトリパス名文字列の長さが渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。
 FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

- FTD_RCD_213 (213) コマンドは正常終了した
- FTD_RCD_421 (421) サービスは無効である
- FTD_RCD_500 (500) コマンドが認識できない
- FTD_RCD_501 (501) 引数やパラメータに文法エラーがある

ステータスを送信可能な場合は、イベント処理結果としてFTD_RCD_213を指定し、続けてFTD_put_datにてステータスを送信してください。(p_nameにて渡されたディレクトリパス名がディレクトリ名を示す場合はディレクトリステータスを、ファイル名を示す場合はそのファイルステータスを送信してください)
(注：コールバックルーチン内ではFTD_put_datは発行できません。)

ディレクトリステータスおよびファイルステータスの例：

05-10-02	12:57PM		3693399 task1
05-10-02	12:58PM		3693399 task2
05-10-02	01:00PM		3693399 task3
05-10-02	01:01PM		3693399 task4
10-16-01	02:54PM	<DIR>	tcp
12-25-01	11:20AM		26060 test.jpg
12-25-01	11:31AM		26060 test2.jpg
12-25-01	11:20AM		26060 test3.jpg

エラーが発生した場合は、該当するイベント処理結果を指定してください。
 上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのSTATコマンドのレスポンスには以下の種類があります。

- 211 System status, or system help reply.
- 213 Status of XXXXX:
- 450 Requested file action not taken.
- 501 Syntax error in parameters or arguments.
- 421 Service not available, closing control connection.
- 530 Not logged in.

(24) TEV_FTD_HELP

サーバのコマンドヘルプ送信要求イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UB      *p_cmd;      コマンド文字列が格納された領域の先頭アドレス、および
                        ヘルプ文字列を格納する領域の先頭アドレス
    INT     *p_clen;    コマンド文字列の長さが格納された領域のアドレス、および
                        ヘルプ文字列の長さを格納するアドレス
} T_FTD_CBCMD;
```

リターン値/エラーコード

0 正常終了 (未使用)

解説

クライアントからサーバ依存コマンドのヘルプの送信要求を受けたことを通知します。

p_cmdが示す領域に送信するヘルプ文字列を、p_clenが示す領域に送信するヘルプ文字列の長さを格納してください。

ヘルプ文字列は最大512文字まで格納することができます。512文字以上格納した場合の動作は保証されません。ヘルプ文字列の終端には必ず改行コード (CR+LF) を付加してください。

svpidには要求を受けたサービスポイントのIDが渡されます。

p_cmdには受信したサーバ依存コマンド文字列の先頭アドレスが渡されます (ヘルプ文字列を格納する領域と同一です)。コマンドの指定が無い場合(*p_clen=0)、内容は無効です。

p_clenには受信したサーバ依存コマンド文字列の長さが格納されたアドレスが渡されます (ヘルプ文字列の長さを格納する領域と同一です)。コマンドの指定が無い場合は0が渡されます。

本イベントの処理が完了した場合はFTD_evt_endをコールしてください。

FTD_evt_endのrtncd (イベント処理結果) には、次の値を指定することができます。

- FTD_RCD_214 (214) コマンドは正常終了した
- FTD_RCD_421 (421) サービスは無効である
- FTD_RCD_500 (500) コマンドが認識できない
- FTD_RCD_501 (501) 引数やパラメータに文法エラーがある

処理が正常に完了した場合は、イベント処理結果としてFTD_RCD_214を指定してください。

ヘルプ文字列 (コマンド指定無し) の例:

```
CMD1
CMD2
CMD3
CMD4
HELP
```

ヘルプ文字列 (コマンド指定有り) の例:

```
SITE CMD1 <sp> parameter
```

エラーが発生した場合は、該当するイベント処理結果を指定してください。

上記以外のイベント処理結果が指定された場合、内部エラーである応答 (451) をクライアントに返します。

FTD_evt_endの詳細に関しては、「3.2.10 FTPサーバイベントの処理完了を通知する」の項を参照してください。

本FTPサーバのHELPコマンドのレスポンスには以下の種類があります。

211 System status, or system help reply.

214 Syntax: "Help message."

501 Syntax error in parameters or arguments.

421 Service not available, closing control connection.

(25) TEV_FTD_ABOR

コマンドの中断イベント

パラメータブロックの構造 (parblk)

無し

リターン値/エラーコード

0 正常終了 (未使用)

解 説

クライアントから直前コマンドの中断要求を受けたことを通知します。
サービスポイントがデータ転送中の場合、データコネクションは切断されます (コントロールコネクションは切断されません)。
データの受信中に本イベントが発生した場合でも、受信データがある間はFTD_get_datにてデータの受信を繰り返してください。

svpidには要求を受けたサービスポイントのIDが渡されます。
本イベントの処理完了時には、FTD_evt_endをコールする必要はありません。

本FTPサーバのABORコマンドのレスポンスには以下の種類があります。

- 225 Data connection open; no transfer in progress.
- 226 Closing data connection.
- 501 Syntax error in parameters or arguments.
- 421 Service not available, closing control connection.

(26) TEV_FTD_DPCN

データコネクションの接続開始イベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    ID      repid;      データコネクションの接続で使用するTCP受付口ID
    ID      cepid;      データコネクションの接続で使用するTCP通信端点ID
} T_FTD_CBDPCN;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

データコネクションの接続を開始したことを通知します。

svpidには開始したサービスポイントのIDが渡されます。

repidにはデータコネクションの接続で使用するTCP/IPマネージャのTCP受付口IDが渡されます。パッシブ接続以外の場合は、0が渡されます。

cepidにはデータコネクションの接続で使用するTCP/IPマネージャのTCP通信端点IDが渡されます。

本イベントの処理完了時には、FTD_evt_endをコールする必要はありません。

(27) TEV_FTD_ERROR エラーイベント

パラメータブロックの構造 (parblk)

```
typedef struct{
    UW      erno;          エラーコード
} T_FTD_CBERR;
```

リターン値/エラーコード

0 正常終了 (未使用)

解 説

FTDサーバにてエラーが発生したことを通知します。

svpidにはエラーが発生したサービスポイントのIDが渡されます。

ernoにはエラーの内容が渡されます。

本イベントの処理完了時には、FTD_evt_endをコールする必要はありません。

エラーの内容には以下の種類があります。

FTD_ENO_CEPNOID (H'00000001) TCP/IPマネージャの通信端点ID空き不足

FTD_ENO_REPNOID (H'00000002) TCP/IPマネージャの受付口ID空き不足

FTD_ENO_MODCPORT (H'00000003) 制御コネクションの動作モード設定に失敗

FTD_ENO_MODDPORT (H'00000004) データコネクションの動作モード設定に失敗

FTD_ENO_MODCPORT、およびFTD_ENO_MODDPORTは、FTD_set_optのFTD_MOD_CPORT (制御コネクションの動作モード設定) またはFTD_MOD_DPORT (データコネクションの動作モードの設定) で指定したパラメータ内容に誤りがあるため、TCP/IPマネージャのTCP通信端点の設定に失敗したこと意味します。この場合、動作モードの設定は行なわれません。FTD_set_optで指定した動作モード設定パラメータ (T_FTD_MODSVP) の内容を確認してください。

5. FTP サーバ レスポンスコード

本FTPサーバがクライアントに返すレスポンスコード、およびレスポンスメッセージの一覧を以下に示します。

表5.1 FTPサーバ レスポンスコード一覧

レスポンスコード	レスポンスメッセージ
125	Data connection already open; transfer starting.
150	File status okay; about to open data connection.
200	Command okay.
202	Command not implemented, superfluous at this site.
211	System status, or system help reply.
212	Directory status.
213	File status.
214	The following commands are recognized.
215	UNKNOWN Type:L8
220	Service ready for new user.
221	Service closing control connection.
225	Data connection open; no transfer in progress.
226	Closing data connection.
227	Entering Passive Mode (h1,h2,h3,h4,p1,p2).
230	User logged in, proceed.
250	Requested file action okay, completed.
257	" PATHNAME " created.
331	User name okay, need password.
332	Need account for login.
350	Requested file action pending further information.
421	Service not available, closing control connection.
425	Can't open data connection.
426	Connection closed; transfer aborted.
450	Requested file action not taken.
451	Requested action aborted. Local error in processing.
452	Requested action not taken.
500	" COMMAND ": Syntax error, command unrecognized.
501	" PARAMETER ": Syntax error in parameters or arguments.
502	" COMMAND ": Command not implemented.
503	" COMMAND ": Bad sequence of commands.
504	" PARAMETER ": Command not implemented for that parameter.
530	Not logged in.
532	Need account for storing files.
550	Requested action not taken.
551	Requested action aborted. Page type unknown.
552	Requested file action aborted.
553	Requested action not taken.

HI.CommunicationEngine
FTPサーバ リファレンスマニュアル
CE7000FTD03J-1

発行年月	2015年 12月 第1版
発行	ルネサスセミコンダクタパッケージ&テストソリューションズ株式会社
編集	ルネサスセミコンダクタパッケージ&テストソリューションズ株式会社

©ルネサスセミコンダクタパッケージ&テストソリューションズ株式会社 2015