

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日  
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

# ユーザース・マニュアル

## RX-NET Ver.1.10

### ネットワーク・ライブラリ

#### FTP

---

#### 対象デバイス

V850シリーズ™

#### 対象リアルタイムOS

RX850 Pro Ver.3.15

#### 対象TCP/IP ライブラリ

RX-NET(TCP/IP) Ver.1.20

(メモ)

# 目次要約

第1章 概説 ...	13
第2章 インストレーション ...	16
第3章 システム構築 ...	20
第4章 FTPサーバ機能 ...	27
第5章 API関数 ...	31
第6章 RX-NET (FTP) 依存部 ...	38
索引 ...	73

V850 シリーズ, V853, V850/SA1, V850/SB1, V850/SB2, V850/SF1, V850/SV1, V850E/MS1, V850E/MA1, V850E/MA2, V850E/IA1, V850E/IA2は, 日本電気株式会社の商標です。

PC/ATは米国IBM社の商標です。

MS-DOS, Windows, Windows NTは, 米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。

Green Hills Software, MULTIは米国Green Hills Software, Inc.の商標です。

その他, 記載の会社名 / 製品名は, 各社の商標, または, 登録商標です。

**本製品は外国為替および外国貿易管理法の規定により規制貨物等（または役務）に該当しますので、日本国外に輸出する場合には、同法に基づき日本国政府の輸出許可が必要です。**

- **本資料の内容は予告なく変更することがありますので、最新のものであることをご確認の上ご使用ください。**
- 文書による当社の承諾なしに本資料の転載複製を禁じます。
- 本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して、当社は当社もしくは第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合、当社はその責を負うものではありませんのでご了承ください。
- 本資料に記載された回路、ソフトウェア、及びこれらに付随する情報は、半導体製品の動作例、応用例を説明するためのものです。従って、これら回路・ソフトウェア・情報をお客様の機器に使用される場合には、お客様の責任において機器設計をしてください。これらの使用に起因するお客様もしくは第三者の損害に対して、当社は一切その責を負いません。

M7A 98.8

巻末にアンケート・コーナーを設けております。このドキュメントに対するご意見をお気軽にお寄せください。



# はじめに

**対象者** このマニュアルは、V850シリーズの応用システムを設計、開発するユーザを対象とします。

**目的** このマニュアルは、次の構成に示すRX-NETの機能をユーザに理解していただくことを目的としています。

**構成** このマニュアルは、大きく分けて次の内容で構成しています。

概 説  
インストレーション  
システム構築  
FTPサーバ機能  
API関数  
RX-NET (FTP) 依存部

**読み方** このマニュアルの読者には、電気、論理回路、マイクロコンピュータ、C言語、アセンブリ言語に関する一般知識が必要です。

V850シリーズのハードウェア機能、命令機能を知りたいとき  
各製品のユーザズ・マニュアルを参照してください。

**凡 例**

注 : 本文中につけた注の説明  
注意 : 気をつけて読んでいただきたい内容  
備考 : 本文の補足説明  
数の表記 : 2進数 ...XXXX または B'XXXX  
10進数...XXXX  
16進数...0XXXX または H'XXXX

2のべき数を示す接頭語 (アドレス空間、メモリ容量) :

K (キロ) :  $2^{10} = 1024$   
M (メガ) :  $2^{20} = 1024^2$   
G (ギガ) :  $2^{30} = 1024^3$

**関連資料** このマニュアルを使用する場合は、次の資料もあわせてご覧ください。

関連資料は暫定版の場合がありますが、この資料では「暫定」の表示をしておりません。

あらかじめご了承ください。

**開発ツールに関する資料（ユーザズ・マニュアル）**

資料名	資料番号		
	和文	英文	
IE-703002-MC ( V853™, V850/SA1™, V850/SB1™, V850/SB2™, V850/SF1™, V850/SV1™用 インサーキット・エミュレータ )	U11595J	U11595E	
IE-V850E-MC ( V850E/IA1™, V850E/IA2™用インサーキット・エミュレータ ) , IE-V850E-MC-A ( V850E/MA1™, V850E/MA2™用インサーキット・エミュレータ )	U14487J	U14487E	
IE-703003-MC-EM1 ( V853用インサーキット・エミュレータ・オプション・ボード )	U11596J	U11596E	
IE-703017-MC-EM1 ( V850/SA1用インサーキット・エミュレータ・オプション・ボード )	U12898J	U12898E	
IE-703037-MC-EM1 ( V850/SB1, V850/SB2用インサーキット・エミュレータ・オプション・ボード )	U14151J	U14151E	
IE-703040-MC-EM1 ( V850/SV1用インサーキット・エミュレータ・オプション・ボード )	U14337J	U14337E	
IE-703079-MC-EM1 ( V850/SF1用インサーキット・エミュレータ・オプション・ボード )	U15447J	U15447E	
IE-703102-MC ( V850E/MS1™用インサーキット・エミュレータ )	U13875J	U13875E	
IE-703102-MC-EM1, IE-703102-MC-EM1-A( V850E/MS1用インサーキット・エミュレータ・オプション・ ボード )	U13876J	U13876E	
IE-703107-MC-EM1 ( V850E/MA1用インサーキット・エミュレータ・オプション・ボード )	U14481J	U14481E	
IE-703116-MC-EM1 ( V850E/IA1用インサーキット・エミュレータ・オプション・ボード )	U14700J	U14700E	
CA850 Ver.2.40以上 C コンパイラ・パッケージ	操作編 C 言語編 プロジェクト・マネージャ編 アセンブリ言語編	U15024J U15025J U15026J U15027J	U15024E U15025E U15026E U15027E
ID850 Ver.2.40 統合ディバग्ガ	操作編 Windows®ベース	U15181J	作成予定
SM850 Ver.2.40以上 システム・シミュレータ	操作編 Windowsベース	U15182J	作成予定
SM850 Ver.2.00以上 システム・シミュレータ	外部部品ユーザ・オープン・インタフェース仕様編	U14873J	U14873E
RX850 Ver.3.13以上 リアルタイムOS	基礎編 インストレーション編 テクニカル編	U13430J U13410J U13431J	U13430E U13410E U13431E
RX850 Pro Ver.3.13 リアルタイムOS	基礎編 インストレーション編 テクニカル編	U13773J U13774J U13772J	U13773E U13774E U13772E
RX-NET ネットワーク・ライブラリ (TCP/IP)		U15083J	-
RX-NET ネットワーク・ライブラリ (PPP)		U15303J	-
RX-NET ネットワーク・ライブラリ (DNS)		U15304J	-
RX-NET ネットワーク・ライブラリ (DHCP)		U15382J	-
RX-NET ネットワーク・ライブラリ (SMTP)		U15505J	-
RX-NET ネットワーク・ライブラリ (POP)		U15539J	-
RX-NET Ver.1.00 ネットワーク・ライブラリ (telnet)		このマニュアル	-
RD850 Ver.3.01 タスク・ディバग्ガ		U13737J	U13737E
RD850 Pro Ver.3.01 タスク・ディバग्ガ		U13916J	U13916E
AZ850 Ver.3.0 システム・パフォーマンス・アナライザ		U14410J	U14410E
PG-FP4 フラッシュ・メモリ・プログラマ		U15260J	U15260E

# 目次

第 1 章 概説 .....	13
1.1 概要 .....	13
1.2 特徴 .....	14
1.3 実行環境 .....	15
1.4 開発環境 .....	15
第 2 章 インストール .....	16
2.1 概要 .....	16
2.2 インストール手順 .....	16
2.2.1 Windows ベース .....	16
2.2.2 UNIX ベース .....	17
2.3 ディレクトリ構成 .....	18
2.3.1 CA850 対応版 .....	18
2.3.2 CCV850E 対応版 .....	19
第 3 章 システム構築 .....	20
3.1 概要 .....	20
3.2 CF 定義ファイルの記述 .....	21
3.3 情報ファイルの生成 .....	21
3.4 RX850 Pro 依存部の記述 .....	21
3.5 RX-FS850 依存部の記述 .....	23
3.6 RX-NET(TCP/IP) 依存部の記述 .....	23
3.7 RX-NET(FTP) 依存部の記述 .....	23
3.8 処理プログラムの記述 .....	24
3.9 オブジェクト・ファイルの生成 .....	25
3.10 アーカイブ・オブジェクトの生成 .....	25
3.11 リンク・ディレクティブ・ファイルの記述 .....	26
3.12 ロード・モジュールの生成 .....	26
第 4 章 FTP サーバ機能 .....	27
4.1 概要 .....	27
4.2 処理の流れ .....	27
4.3 FTP サーバ機能 API 関数 .....	28
4.3.1 RX-NET(FTP) の初期化 .....	28
4.3.2 RX-NET(FTP) の強制終了 .....	29
4.4 FTP サーバ機能応答関数 .....	30
第 5 章 API 関数 .....	31
5.1 概要 .....	31
5.2 API 関数の呼び出し .....	31
5.3 データ・マクロ .....	32
5.3.1 戻り値 .....	32
5.4 API 関数解説 .....	33
5.4.1 外部インタフェース仕様 .....	35

第 6 章 RX-NET(FTP) 依存部 .....	38
6.1 概要 .....	38
6.2 基本情報 .....	38
6.2.1 受信バッファ情報 .....	38
6.2.2 タイムアウト時間情報 .....	39
6.2.3 メッセージ情報 .....	39
6.3 データ・マクロ .....	40
6.3.1 戻り値 .....	40
6.3.2 データ表現タイプ .....	40
6.3.3 垂直フォーマット制御種別 .....	40
6.3.4 転送モード .....	41
6.4 応答関数解説 .....	42
6.4.1 外部インタフェース仕様 .....	44
索引 .....	73

# 目次

図 1-1	RX-NET(FTP) の位置付け .....	13
図 1-2	RX-NET(FTP) の階層的な位置付け .....	14
図 2-1	ディレクトリ構成 (CA850 対応版) .....	18
図 2-2	ディレクトリ構成 (CCV850E 対応版) .....	19
図 3-1	システム構築手順 .....	20
図 3-2	RX850 Pro 依存部の処理の流れ .....	22
図 4-1	FTP コマンドの対応手順 .....	27

# 表目次

表 2-1	RX-NET(FTP) の提供形式 .....	16
表 3-1	応答関数 .....	23
表 4-1	応答関数 .....	30
表 5-1	戻り値 .....	32
表 5-2	API 関数 .....	35
表 6-1	戻り値 .....	40
表 6-2	データ表現タイプ .....	40
表 6-3	垂直フォーマット制御種別 .....	40
表 6-4	転送モード .....	41
表 6-5	応答関数 .....	44

# 第1章 概説

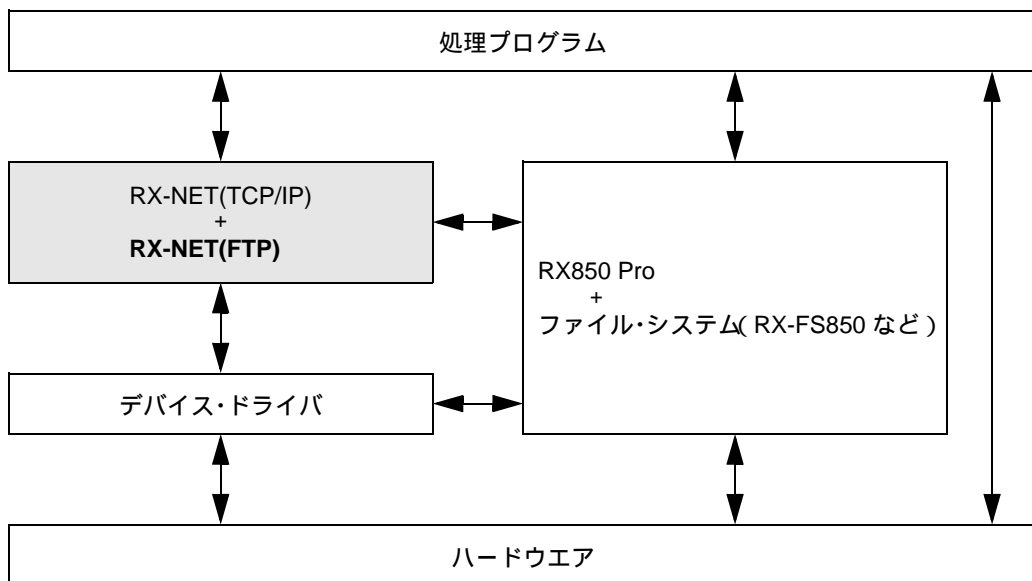
## 1.1 概要

RX-NET(FTP) は、組み込み型制御用リアルタイム・オペレーティング・システム RX850 Pro (μITRON3.0 仕様準拠, NEC 製) 上で動作する TCP/IP ライブラリ RX-NET に対し, FTP(File Transfer Protocol) を利用したサーバ, クライアント間のファイル転送を行うための FTP サーバ機能を提供しています。

したがって, ユーザは, RX-NET(FTP) が提供する FTP サーバ機能を利用することにより, FTP を利用したサーバ, クライアント間のファイル転送を実現することが可能となります。

図 1-1 に, RX-NET(FTP) の位置付けを示します。

図 1-1 RX-NET(FTP) の位置付け



## 1.2 特徴

以下に、RX-NET(FTP) の特徴を示します。

- RFC に準拠

RX-NET(FTP) では、RFC に準拠した設計が行われています。

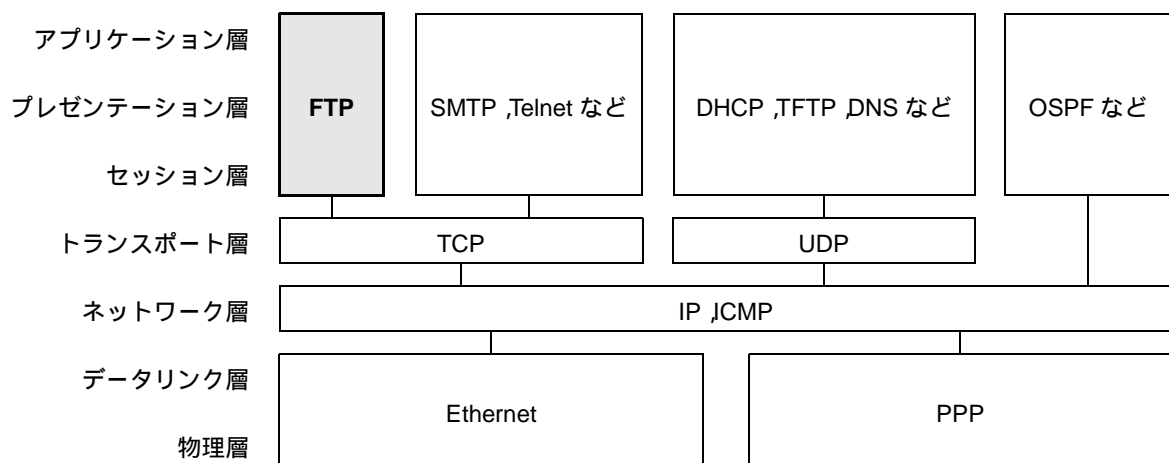
なお、RFC とは、インターネットに関する研究開発機関 IETF (Internet Engineering Task Force) が取りまとめた公開技術文書であり、電子メール、ファイル転送などのプロトコル仕様（情報交換を行う際に必要な手順、および、規約）の他にも、各種サービス、ガイドラインといった多岐に渡った情報（ネットワーク技術の実装と運用に主眼を置いた情報）が記載されています。

- FTP サーバ機能をサポート

RX-NET(FTP) では、ネットワーク上で利用される FTP サーバ機能を提供しています。したがって、RX-NET(FTP) を組み込んだ機器は、サーバ、クライアント間のファイル転送が可能となります。

図 1-2 に、RX-NET(FTP) の階層的位置付けを示します。

図 1-2 RX-NET(FTP) の階層的位置付け



- 高い移植性

RX-NET(FTP) が処理を実行するうえで必要となるユーザの実行環境 / アプリケーション・システムに依存した処理については、RX-NET(FTP) 依存部（ユーザ・OWN・コーディング部）として切り出し、サンプル・ソース・ファイルを提供しています。

これにより、実行環境への移植性を向上させるとともに、カスタマイズ化を容易なものとしています。

- マルチタスク処理を意識した設計

RX-NET(FTP) が提供する API 関数では、マルチタスク処理を考慮した設計が行われています。このため、ユーザが処理プログラムを記述する際、API 関数の発行に伴うタスク間の排他制御などを意識する必要がありません。



## 1.3 実行環境

以下に、RX-NET(FTP) が処理を実行するうえで必要となるハードウェアを示します。

- プロセッサ

以下に、RX-NET(FTP) が処理を実行するうえで必要となるプロセッサを示します。

V850 シリーズ V850E/xxx

- 周辺コントローラ

RX-NET(FTP) では、処理を実行するうえで、特定の周辺コントローラは必要ありません。

- メモリ容量

以下に、RX-NET(FTP) が処理を実行するうえで必要となるメモリ容量を示します。

RX-NET(FTP) のテキスト領域 : 約 15 K バイト

RX-NET(FTP) のデータ領域 : 約 4K バイト

## 1.4 開発環境

以下に、RX-NET(FTP) を使用した処理プログラムを開発するうえで必要となるハードウェア、および、ソフトウェアを示します。

- ハードウェア

- ホスト・マシン

PC-98NX/PC-9821 シリーズ : Windows 2000, 98, Me, XP, NT 4.0

IBM-PC/AT 互換機 : Windows 2000, 98, Me, XP, NT 4.0

SPARC station : Solaris Rel.2.5.x

- ソフトウェア

- リアルタイム OS

RX850 Pro Ver.3.15 以上 : NEC 製

- ネットワーク・ライブラリ

RX-NET(TCP/IP) Ver.1.20 以上 : NEC 製

- C コンパイラ・パッケージ

CA850 Ver.2.41 以上 : NEC 製

CCV850E Ver.1.8.9 Rel.4.0.2 以上 : 米国 Green Hills Software, Inc. 製

## 第2章 インストール

本章では、RX-NET(FTP)の提供媒体に格納されているファイル群をユーザの開発環境(ホスト・マシン)上にインストールする際の手順について解説しています。

### 2.1 概要

RX-NET(FTP)の提供媒体は、ホスト・マシンの種類(Windowsベース、UNIXベース)に併せて計2種類が用意されています。

表2-1に、RX-NET(FTP)の提供形式一覧を示します。

表2-1 RX-NET(FTP)の提供形式

ホスト・マシン	提供形式	提供媒体
Windowsベース • PC-98NX/PC-9821シリーズ • IBM-PC/AT互換機	CA850対応版オブジェクト・ファイル形式 CCV850E対応版オブジェクト・ファイル形式	CD-ROM
UNIXベース • SPARC station	CA850対応版オブジェクト・ファイル形式 CCV850E対応版オブジェクト・ファイル形式	CD-ROM

**注意** ホスト・マシンの種類別に用意された提供媒体には、2種類(CA850対応版オブジェクト・ファイル形式、CCV850E対応版オブジェクト・ファイル形式)のRX-NET(FTP)が格納されています。したがって、提供媒体からホスト・マシン上にファイル群をインストールする際には、ユーザが使用するCコンパイラ・パッケージに対応したRX-NET(FTP)をインストールする必要があります。

### 2.2 インストール手順

RX-NET(FTP)の提供媒体に格納されているファイル群のインストール手順は、ホスト・マシンの種類(Windowsベース、UNIXベース)により異なります。

そこで、以降に、ホスト・マシンがWindowsベースの場合、UNIXベースの場合のインストール手順をそれぞれに示します。

**注意** RX-NET(FTP)のインストールは、RX-NET(TCP/IP)のインストール完了後に行ってください。

#### 2.2.1 Windowsベース

以下に、RX-NET(FTP)の提供媒体に格納されているファイル群をホスト・マシン(Windowsベース：PC-98NX/PC-9821シリーズ、IBM-PC/AT互換機)上にインストールする際の手順を示します。

- 1) Windowsの起動  
ホスト・マシン、および、周辺機器などの電源を投入し、Windowsを起動します。
- 2) 提供媒体のセット  
RX-NET(FTP)の提供媒体をホスト・マシンの該当デバイス装置(CD-ROMドライブ)にセットすることにより、セットアップ・プログラムが自動実行します。  
以降、モニタ画面に表示されるメッセージに従ってインストール作業を実行します。

**注意** セットアップ・プログラムが自動実行しない場合には、RX-NET(FTP)の提供媒体のディレクトリRX-NET\_FT  
TP\_V850E\_NEC\DISK1に格納されているSETUP.EXEを起動します。

- 3) ファイル群の確認  
Windowsの標準アプリケーションExplorerなどを用いて、RX-NET(FTP)の提供媒体に格納されていたファイル群がホスト・マシン上にインストールされたことを確認します。

なお、各ディレクトリについての詳細は、「2.3 ディレクトリ構成」を参照してください。

## 2.2.2 UNIX ベース

以下に、RX-NET(FTP)の提供媒体に格納されているファイル群をホスト・マシン (UNIX ベース: SPARC station) 上にインストールする際の手順を示します。

ただし、入力例中の“%”はシェル・プロンプトを、“ ”はスペース・キーの入力を、“<Enter>”はエンター・キーの入力を表しています。

- 1) ホスト・マシンへのログイン  
ホスト・マシンにログインします。

```
%
```

- 2) ディレクトリの移動  
cd コマンドを実行し、インストール用ディレクトリに移動します。  
なお、下記入力例では、インストール用ディレクトリとして /usr/local を指定しています。

注意 インストール用ディレクトリのパーミッション (read, write, execute) はインストール作業に対して許可状態である必要があります。そこで、インストール用ディレクトリのパーミッションが不許可状態であった場合には、chmod コマンドを実行し、パーミッションを不許可状態から許可状態に変更します。

```
% cd /usr/local <Enter>
```

- 3) 提供媒体のセット  
RX-NET(FTP)の提供媒体をホスト・マシンの該当デバイス装置 (CD-ROM ドライブ) にセットします。
- 4) デバイスのマウント  
mount コマンドを実行し、該当デバイス装置に対応したデバイスをマウントします。  
なお、下記入力例では、該当デバイス装置のデバイス名 (スペシャル・ファイル名) として /dev/rst8 を、マウント・ディレクトリとして /cdrom を指定しています。  
注意 ホスト・マシンによっては、“デバイスのマウント”が自動的に行われるものがあります。このような場合、mount コマンドを実行する必要はありません。

```
% mount /dev/rst8 /cdrom <Enter>
```

- 5) ファイル群のインストール  
tar コマンドを実行し、マウント・ディレクトリ /cdrom 下の圧縮ファイルをインストール用ディレクトリに展開します。  
ただし、提供媒体には、以下に示した2種類の圧縮ファイルが格納されています。

- CA850 対応版
- CCV850E 対応版

そこで、ユーザが使用する C コンパイラ・パッケージが NEC 製 CA850 の場合は圧縮ファイル nec/rxnetftp.tar を、米国 Green Hills Software, Inc. 製 CCV850E の場合は圧縮ファイル ghs/rxnetftp.tar を展開します。

【 CA850 対応版の場合 】

```
% tar -xvof /cdrom/RXNET/nec/rxnetftp.tar <Enter>
```

【 CCV850E 対応版の場合 】

```
% tar -xvof /cdrom/RXNET/ghs/rxnetftp.tar <Enter>
```

- 6) ファイル群の確認  
ls コマンドを実行し、RX-NET(FTP)の提供媒体に格納されていたファイル群がホスト・マシン上にインストールされたことを確認します。  
なお、各ディレクトリについての詳細は、「2.3 ディレクトリ構成」を参照してください。

```
% ls -CFR /usr/local/nertools32 <Enter>
```

## 2.3 ディレクトリ構成

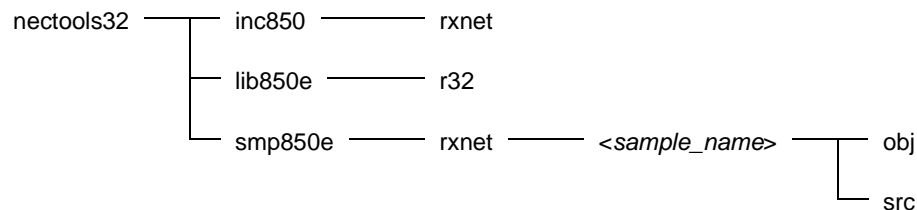
RX-NET(FTP) の提供媒体に格納されているファイル群のディレクトリ構成は、ユーザが使用する C コンパイラ・パッケージの種類 (NEC 製 CA850, 米国 Green Hills Software, Inc. 製 CCV850E) により異なります。

そこで、以降に、C コンパイラ・パッケージが CA850 の場合、CCV850E の場合のディレクトリ構成をそれぞれに示します。

### 2.3.1 CA850 対応版

図 2-1 に、RX-NET(FTP) の提供媒体 (CA850 対応版) に格納されているファイル群をホスト・マシン上にインストールした際に生成されるディレクトリ構成を示します。

図 2-1 ディレクトリ構成 (CA850 対応版)



以下に、各ディレクトリの概要を示します。

- 1) nectools32\inc850  
RX-NET(FTP) の標準ヘッダ・ファイルが格納されているディレクトリです。  
rxnet\_ftps.h : RX-NET(FTP) 用標準ヘッダ・ファイル
- 2) nectools32\inc850\rxnet  
RX-NET(FTP) のヘッダ・ファイルが格納されているディレクトリです。
- 3) nectools32\lib850e\r32  
FTP ライブラリ (32 レジスタ・モード) が格納されているディレクトリです。  
libftps.a : FTP ライブラリ
- 4) nectools32\smp850e\rxnet\*<sample\_name>*\obj  
ロード・モジュールを生成するためのメイク・ファイル Makefile が格納されているディレクトリです。  
なお、本ディレクトリにおいて、make コマンドを実行することにより、ロード・モジュール sample.out が本ディレクトリに生成されます。  
Makefile : ロード・モジュール用メイク・ファイル
- 5) nectools32\smp850e\rxnet\*<sample\_name>*\src  
サンプル・プログラムのソース・ファイル、および、ヘッダ・ファイルが格納されているディレクトリです。

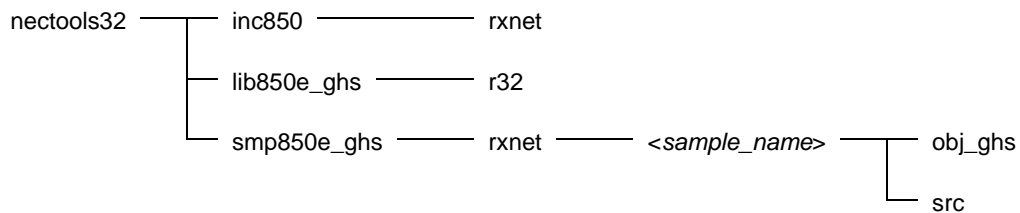
注意 *<sample\_name>* についての詳細は、下記に示したテキスト・ファイルを参照してください。

*<sample\_name>* : nectools32\smp850e\rxnet\README.FTPS

### 2.3.2 CCV850E 対応版

図 2-2 に、RX-NET(FTP) の提供媒体 (CCV850E 対応版) に格納されているファイル群をホスト・マシン上にインストールした際に生成されるディレクトリ構成を示します。

図 2-2 ディレクトリ構成 (CCV850E 対応版)



以下に、各ディレクトリの概要を示します。

- 1) nectools32\inc850  
RX-NET(FTP) の標準ヘッダ・ファイルが格納されているディレクトリです。  
rxnet\_ftps.h : RX-NET(FTP) 用標準ヘッダ・ファイル
- 2) nectools32\inc850\rxnet  
RX-NET(FTP) のヘッダ・ファイルが格納されているディレクトリです。
- 3) nectools32\lib850e\_ghs\r32  
FTP ライブラリ (32 レジスタ・モード) が格納されているディレクトリです。  
libftps.a : FTP ライブラリ
- 4) nectools32\smp850e\_ghs\rxnet\*sample\_name*\obj\_ghs  
ロード・モジュールを生成するためのビルド・ファイル sample.bld が格納されているディレクトリです。  
なお、本ディレクトリの sample.bld を用いることにより、ロード・モジュール sample.out が本ディレクトリに生成されます。  
sample.bld : ロード・モジュール用ビルド・ファイル
- 5) nectools32\smp850e\_ghs\rxnet\*sample\_name*\src  
サンプル・プログラムのソース・ファイル、および、ヘッダ・ファイルが格納されているディレクトリです。

注意 <i>sample\_name> についての詳細は、下記に示したテキスト・ファイルを参照してください。

<i>sample\_name> : nectools32\smp850e\_ghs\rxnet\README.FTPS

# 第3章 システム構築

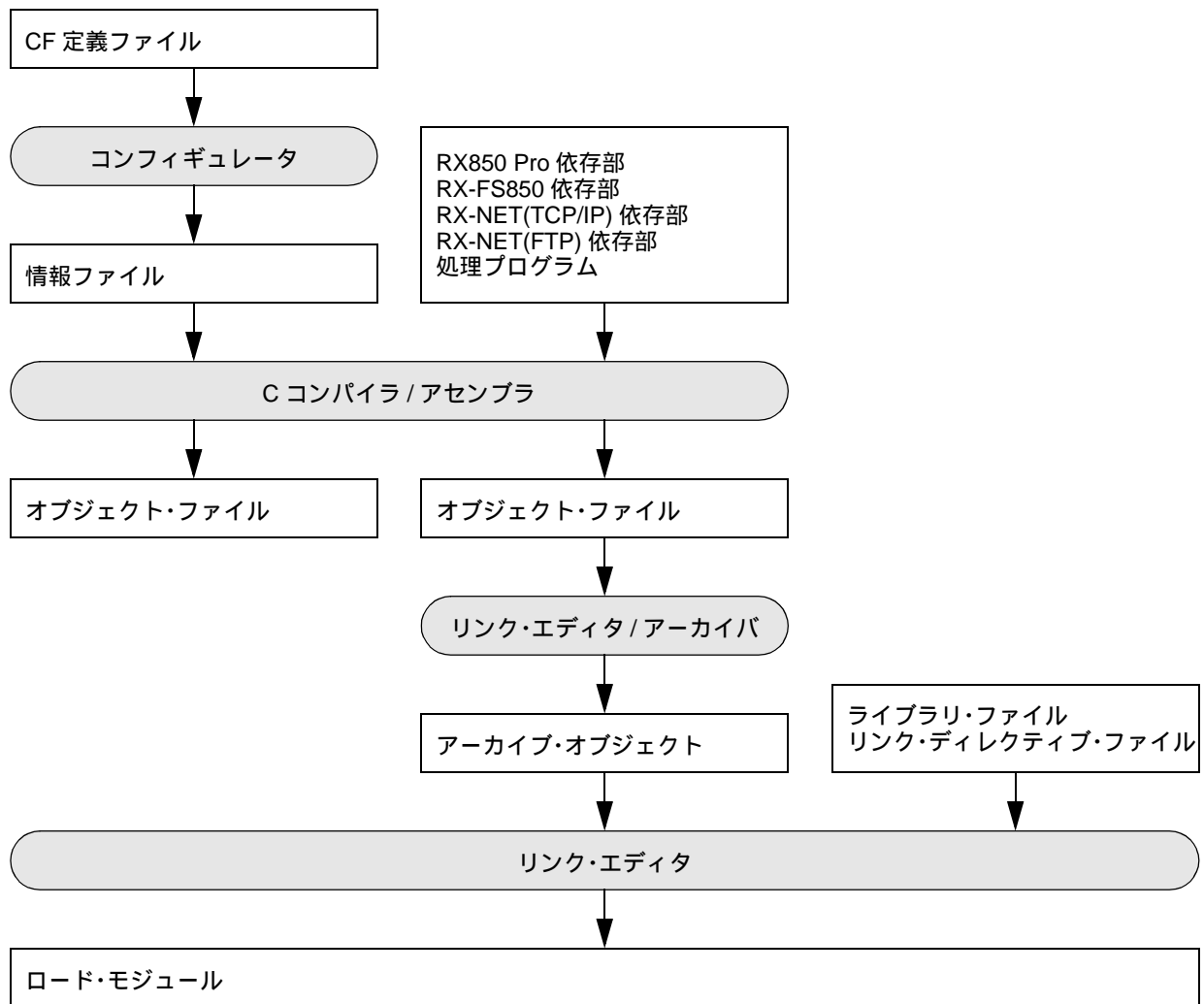
本章では、RX-NET(FTP)を使用したネットワーク・アプリケーション(ロード・モジュール)の構築手順を解説しています。

## 3.1 概要

システム構築とは、RX-NET(FTP)の提供媒体からユーザの開発環境(ホスト・マシン)上にインストールしたファイル群を用いてネットワーク・アプリケーション(ロード・モジュール)を生成することです。

表 3-1 に、RX-NET(FTP)のシステム構築手順を示します。

図 3-1 システム構築手順



## 3.2 CF 定義ファイルの記述

組み込み型制御用リアルタイム・オペレーティング・システム RX850 Pro(μITRON3.0 仕様準拠, NEC 製)の管理下で動作する処理プログラムを作成する場合, RX850 Pro に提供するコンフィギュレーション情報(リアルタイム OS 情報, SIT 情報, SCT 情報)を保持した CF 定義ファイルが必要となります。

なお, RX-NET(FTP)では1種類のシステム・コールを利用して各種機能を実現しています。

- SCT 情報

- 時間管理機能情報

RX-NET(FTP)用に“get\_tim”を定義。

注意 1 CF 定義ファイルを記述する際の注意事項, および, コンフィギュレーション情報についての詳細は, 「**RX850 Pro ユーザーズ・マニュアル インストレーション編**」を参照してください。

注意 2 RX-NET(FTP)では, CF 定義ファイルのサンプル・ソース・ファイルを提供しています。

【CA850 対応版の場合】

```
nectools32\smp850e\rxnet\<sample_name>\src
sys.cf          : CF 定義ファイル
```

【CCV850E 対応版の場合】

```
nectools32\smp850e_ghs\rxnet\<sample_name>\src
sys.cf          : CF 定義ファイル
```

## 3.3 情報ファイルの生成

「3.2 CF 定義ファイルの記述」で作成された CF 定義ファイルに対して RX850 Pro が提供するユーティリティ・ツール(コンフィギュレータ cf850pro)を実行し, 情報ファイル(システム情報テーブル, システム・コール・テーブル, システム情報ヘッダ・ファイル)を生成します。

以下に, シェル・プロンプトのコマンド・ラインから cf850pro を実行する際の入力例(CF 定義ファイル sys.cf を読み込んだのち, システム情報テーブル sit.s, システム・コール・テーブル svc.s, システム情報ヘッダ・ファイル sys.h を出力)を示します。

ただし, 入力例中の“C>”はシェル・プロンプトを, “ ” はスペース・キーの入力を, “<Enter>”はエンター・キーの入力を表しています。

```
C> cf850pro -i sit.s -c svc.s -d sys.h sys.cf <Enter>
```

注意 1 コンフィギュレータ cf850pro の起動オプション, および, 実行方法についての詳細は, 「**RX850 Pro ユーザーズ・マニュアル インストレーション編**」を参照してください。

注意 2 RX-NET(FTP)では, 情報ファイルを生成するためのサンプル・コマンド・ファイルを提供しています。

【CA850 対応版の場合】

```
nectools32\smp850e\rxnet\<sample_name>\obj
Makefile       : ロード・モジュール用メイク・ファイル
```

【CCV850E 対応版の場合】

```
nectools32\smp850e_ghs\rxnet\<sample_name>\obj_ghs
sample.bld     : ロード・モジュール用ビルド・ファイル
```

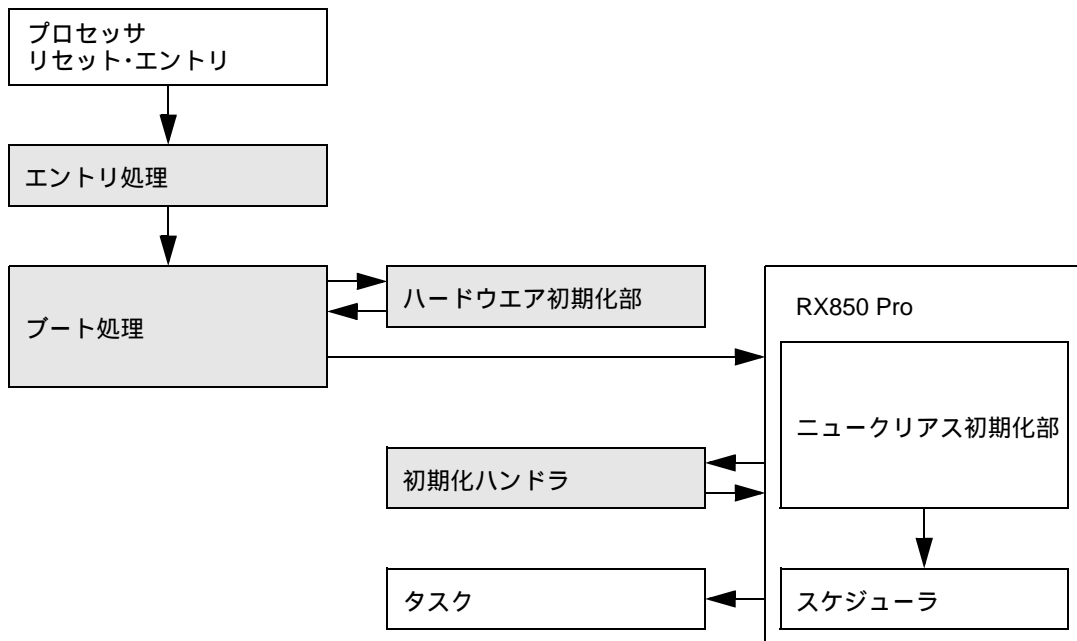
## 3.4 RX850 Pro 依存部の記述

RX-NET(FTP)では, RX850 Pro が提供する機能を利用して各種機能を実現しています。また, ユーザが記述した処理プログラムは, RX850 Pro の管理下でその処理を実行することになります。

したがって, RX850 Pro を正常に動作させるうえで必要となる RX850 Pro 依存部(ユーザ・OWN・コーディング部)の記述が必要となります。

図 3-2 に, RX850 Pro 依存部の処理の流れを示します。

図 3-2 RX850 Pro 依存部の処理の流れ



以下に、RX850 Pro 依存部の一覧を示します。

- エントリ処理

割り込みが発生した際にプロセッサが強制的に制御を移すハンドラ・アドレスに対して該当処理(ブート処理、RX850 Pro が提供する割り込み処理管理機能、直接起動割り込みハンドラ)への分岐処理を割り付けるために用意された処理ルーチンです。

- ブート処理

RX850 Pro が処理を実行するうえで必要となる最低限の初期化処理を行うために用意された処理ルーチンであり、エントリ処理(プロセッサのリセット・エントリに割り付けられた分岐処理)から呼び出されます。

- ハードウェア初期化部

RX850 Pro が処理を実行するうえで必要となるハードウェアの初期化処理を行うために用意された処理ルーチンであり、ブート処理から呼び出されます。

なお、RX850 Pro では、一定周期で発生するタイマ割り込みを利用して時間管理を行っています。そこで、RX850 Pro が時間管理に利用するタイマ割り込みを発生するハードウェア(リアルタイム・パルス・ユニット、または、タイマ・コントローラ)に対しては、CF 定義ファイル作成時にシステム情報で定義した基本クロック周期でタイマ割り込みが発生するような設定を行う必要があります。

- 初期化ハンドラ

ユーザの実行環境/アプリケーション・システムに依存した初期化処理を行うために用意された処理ルーチンであり、ニュークリアス初期化部から呼び出されます。

なお、RX850 Pro では、初期化ハンドラを“タスク”として位置付けています。

注意 1 RX850 Pro 依存部を記述する際の注意事項についての詳細は、「**RX850 Pro ユーザーズ・マニュアル インストラクション編**」を参照してください。

注意 2 RX-NET(FTP)では、初期化ハンドラのサンプル・ソース・ファイルを提供しています。

【CA850 対応版の場合】

```
nectools32\smp850e\rxnet\\src
varfunc.c      : 初期化ハンドラ
```

【CCV850E 対応版の場合】

```
nectools32\smp850e_ghs\rxnet\\src
varfunc.c      : 初期化ハンドラ
```



### 3.5 RX-FS850 依存部の記述

RX-NET(FTP) が提供している応答関数のサンプル・ソース・ファイルでは、RX-FS850 が提供する機能を利用してブロック型デバイスへの入出力処理を実現しています。

したがって、ブロック型デバイスへの入出力処理に RX-FS850 を利用する場合、RX-FS850 依存部 (ユーザ・OWN・コーディング部) の記述が必要となります。

注意 RX-FS850 依存部を記述する際の注意事項についての詳細は、「RX-FS850 ユーザーズ・マニュアル」を参照してください。

### 3.6 RX-NET(TCP/IP) 依存部の記述

RX-NET(FTP) では、RX-NET(TCP/IP) が提供する機能を利用して各種ネットワーク通信機能を実現しています。

したがって、RX-NET(TCP/IP) を正常に動作させるうえで必要となる RX-NET(TCP/IP) 依存部 (ユーザ・OWN・コーディング部) の記述が必要となります。

注意 RX-NET(TCP/IP) 依存部を記述する際の注意事項についての詳細は、「RX-NET(TCP/IP) ユーザーズ・マニュアル」を参照してください。

### 3.7 RX-NET(FTP) 依存部の記述

RX-NET(FTP) では、RX-NET(FTP) が提供する機能を実現する際に必要となる基本情報、および、応答関数を RX-NET(FTP) 依存部 (ユーザ・OWN・コーディング部) として切り出しています。

したがって、RX-NET(FTP) が提供する機能を利用した処理プログラムを作成する場合、RX-NET(FTP) を正常に動作させるうえで必要となる RX-NET(FTP) 依存部 (ユーザ・OWN・コーディング部) の記述が必要となります。

以下に、RX-NET(FTP) 依存部の一覧を示します。

#### • 基本情報

RX-NET(FTP) では、RX-NET(FTP) が提供する機能を実現する際に必要となる基本情報 (受信バッファ情報、タイムアウト時間情報、メッセージ情報など) をユーザ・OWN・コーディング部として切り出しています。

以下に、RX-NET(FTP) のユーザ・OWN・コーディング部として切り出される基本情報の一覧を示します。

- 受信バッファ情報
- タイムアウト時間情報
- メッセージ情報

#### • 応答関数

RX-NET(FTP) では、FTP クライアントから送信された FTP コマンドを受け付けた際、および、各種イベントが発生した際に呼び出す応答関数をユーザ・OWN・コーディング部として切り出しています。

表 3-1 に、ユーザ・OWN・コーディング部として切り出される応答関数の一覧を示します。

表 3-1 応答関数

応答関数名	機能概要
ftps_login_user	USER, PASS コマンドを受け付けた際に呼び出される応答関数
ftps_cwd_user	CWD コマンドを受け付けた際に呼び出される応答関数
ftps_cdup_user	CDUP, XCUP コマンドを受け付けた際に呼び出される応答関数
ftps_logout_user	QUIT コマンドを受け付けた際に呼び出される応答関数
ftps_type_user	TYPE コマンドを受け付けた際に呼び出される応答関数
ftps_mode_user	MODE コマンドを受け付けた際に呼び出される応答関数
ftps_retr_user	RETR コマンドを受け付けた際に呼び出される応答関数
ftps_stor_user	STOR コマンドを受け付けた際に呼び出される応答関数
ftps_stou_user	STOU コマンドを受け付けた際に呼び出される応答関数

応答関数名	機能概要
ftps_appe_user	APPE コマンドを受け付けた際に呼び出される応答関数
ftps_allo_user	ALLO コマンドを受け付けた際に呼び出される応答関数
ftps_rnto_user	RNTO コマンドを受け付けた際に呼び出される応答関数
ftps_abor_user	ABOR コマンドを受け付けた際に呼び出される応答関数
ftps_dele_user	DELE コマンドを受け付けた際に呼び出される応答関数
ftps_rmd_user	RMD, XRMD コマンドを受け付けた際に呼び出される応答関数
ftps_mkd_user	MKD, XMKD コマンドを受け付けた際に呼び出される応答関数
ftps_pwd_user	PWD, XPWD コマンドを受け付けた際に呼び出される応答関数
ftps_list_user	LIST コマンドを受け付けた際に呼び出される応答関数
ftps_nlst_user	NLST コマンドを受け付けた際に呼び出される応答関数
ftps_syst_user	SYST コマンドを受け付けた際に呼び出される応答関数
ftps_recv_user	転送データ受信イベントが発生した際に呼び出される応答関数
ftps_send_user	転送データ送信イベントが発生した際に呼び出される応答関数
ftps_opensession_user	セッション開始イベントが発生した際に呼び出される応答関数
ftps_closesession_user	セッション切断イベントが発生した際に呼び出される応答関数
ftps_timeout_user	タイムアウト・イベントが発生した際に呼び出される応答関数

注意 1 RX-NET(FTP) 依存部についての詳細は、「第6章 RX-NET(FTP) 依存部」を参照してください。

注意 2 RX-NET(FTP) では、RX-NET(FTP) のサンプル・ソース・ファイルを提供しています。

【CA850 対応版の場合】

```
nectools32\smp850e\rxnet\

```

【CCV850E 対応版の場合】

```
nectools32\smp850e_ghs\rxnet\

```

### 3.8 処理プログラムの記述

ネットワーク・アプリケーションとして実現すべき処理 (処理プログラム) を記述します。

なお、処理プログラムは、用途別に以下のように分類 / 区別されています。

- タスク

RX850 Pro の管理下で実行可能な処理プログラムの最小単位です。

- 直接起動割り込みハンドラ

割り込みが発生した際に、RX850 Pro を介在させることなく起動される割り込み処理専用ルーチンです。

なお、RX850 Pro では、直接起動割り込みハンドラを“タスク”とは独立したもの (非タスク) として位置付けています。このため、割り込みが発生した際には、システム内で最高優先度を持つタスクが実行中であっても、その処理は中断され、直接起動割り込みハンドラに制御が移ります。

- 間接起動割り込みハンドラ

割り込みが発生した際に RX850 Pro による割り込み前処理 (レジスタの退避、スタックの切り替えなど) を行われたのちに起動される割り込み処理専用ルーチンです。

なお、RX850 Pro では、間接起動割り込みハンドラを“タスク”とは独立したもの(非タスク)として位置付けています。このため、割り込みが発生した際には、システム内で最高優先度を持つタスクが実行中であっても、その処理は中断され、間接起動割り込みハンドラに制御が移ります。

- 周期起動ハンドラ
 

一定の時間が経過した際に起動される周期処理専用ルーチンです。  
 なお、RX850 Pro では、周期起動ハンドラを“タスク”とは独立したもの(非タスク)として位置付けています。このため、一定の時間が経過した際には、システム内で最高優先度を持つタスクが実行中であっても、その処理は中断され、周期起動ハンドラに制御が移ります。
- 拡張 SVC ハンドラ
 

ユーザが記述した関数を拡張システム・コールとして RX850 Pro に登録した処理ルーチンです。  
 なお、RX850 Pro では、拡張 SVC ハンドラを“拡張 SVC ハンドラを呼び出した処理プログラム(タスク、非タスク)の延長線”として位置付けています。
- 拡張 SVC ハンドラ用インタフェース・ルーチン
 

処理プログラム(タスク、非タスク)から4個以上の引き継ぎデータを持った拡張 SVC ハンドラを呼び出す際に必要となるインタフェース・ルーチンです。

注意 1 処理プログラムを記述する際の注意事項についての詳細は、「RX850 Pro ユーザーズ・マニュアル 基礎編」を参照してください。

注意 2 RX-NET(FTP) では、処理プログラムのサンプル・ソース・ファイルを提供しています。

【CA850 対応版の場合】

```
nectools32\smp850e\rxnet\<sample_name>\src
task.c          : タスク
```

【CCV850E 対応版の場合】

```
nectools32\smp850e_ghs\rxnet\<sample_name>\src
task.c          : タスク
```

## 3.9 オブジェクト・ファイルの生成

「3.2 CF 定義ファイルの記述」～「3.8 処理プログラムの記述」で作成された C 言語ソース・ファイル/アセンブリ言語ソース・ファイルに対して C コンパイラ/アセンブラを実行し、リロケータブルなオブジェクト・ファイルを生成します。

- 注意 1 C コンパイラ/アセンブラの起動オプション、および、実行方法についての詳細は、使用する C コンパイラ・パッケージのユーザーズ・マニュアルを参照してください。
- 注意 2 RX-NET(FTP) では、オブジェクト・ファイルを生成するためのサンプル・コマンド・ファイルを提供しています。

【CA850 対応版の場合】

```
nectools32\smp850e\rxnet\<sample_name>\obj
Makefile       : ロード・モジュール用メイク・ファイル
```

【CCV850E 対応版の場合】

```
nectools32\smp850e_ghs\rxnet\<sample_name>\obj_ghs
sample.bld     : ロード・モジュール用ビルド・ファイル
```

## 3.10 アーカイブ・オブジェクトの生成

「3.9 オブジェクト・ファイルの生成」において作成したリロケータブルなオブジェクト・ファイルのうち、1個のオブジェクトとしてまとめることが可能なものについてはリンク・エディタ/アーカイバを実行し、アーカイブ・オブジェクトを生成します。

- 注意 1 リンク・エディタ/アーカイバの起動オプション、および、実行方法についての詳細は、使用する C コンパイラ・パッケージのユーザーズ・マニュアルを参照してください。
- 注意 2 RX-NET(FTP) では、アーカイブ・オブジェクトを生成するためのサンプル・コマンド・ファイルを提供しています。

## 【CA850 対応版の場合】

```
nctools32\smp850e\rxnet\<sample_name>\obj
Makefile      : ロード・モジュール用メイク・ファイル
```

## 【CCV850E 対応版の場合】

```
nctools32\smp850e_ghs\rxnet\<sample_name>\obj_ghs
sample.bld    : ロード・モジュール用ビルド・ファイル
```

### 3.11 リンク・ディレクティブ・ファイルの記述

リンク・エディタが行うアドレス割り付けをユーザが固定化するためのファイル(リンク・ディレクティブ・ファイル)を記述します。

注意 1 リンク・エディタ/アーカイバの起動オプション,および,実行方法についての詳細は,使用するCコンパイラ・パッケージのユーザズ・マニュアルを参照してください。

注意 2 RX-NET(FTP)では,リンク・ディレクティブ・ファイルのサンプル・ソース・ファイルを提供しています。

## 【CA850 対応版の場合】

```
nctools32\smp850e\rxnet\<sample_name>\src
sample.dir    : リンク・ディレクティブ・ファイル
```

## 【CCV850E 対応版の場合】

```
nctools32\smp850e_ghs\rxnet\<sample_name>\obj_ghs
sample.lx     : リンク・ディレクティブ・ファイル
```

### 3.12 ロード・モジュールの生成

「3.9 オブジェクト・ファイルの生成」～「3.10 アーカイブ・オブジェクトの生成」で作成したオブジェクト・ファイル,アーカイブ・オブジェクト,「3.11 リンク・ディレクティブ・ファイルの記述」で作成したリンク・ディレクティブ・ファイルの他に,Cコンパイラ・パッケージ,RX850 Pro,RX-NET(TCP/IP),RX-NET(FTP)などが提供しているライブラリ・ファイルに対してリンク・エディタを実行し,ロード・モジュールを生成します。

注意 1 リンク・エディタの起動オプション,および,実行方法についての詳細は,使用するCコンパイラ・パッケージのユーザズ・マニュアルを参照してください。

注意 2 RX-NET(FTP)では,ロード・モジュールを生成するためのサンプル・コマンド・ファイルを提供しています。

## 【CA850 対応版の場合】

```
nctools32\smp850e\rxnet\<sample_name>\obj
Makefile      : ロード・モジュール用メイク・ファイル
```

## 【CCV850E 対応版の場合】

```
nctools32\smp850e_ghs\rxnet\<sample_name>\obj_ghs
sample.bld    : ロード・モジュール用ビルド・ファイル
```

# 第4章 FTPサーバ機能

本章では、RX-NET(FTP)が提供しているFTPサーバ機能について解説しています。

## 4.1 概要

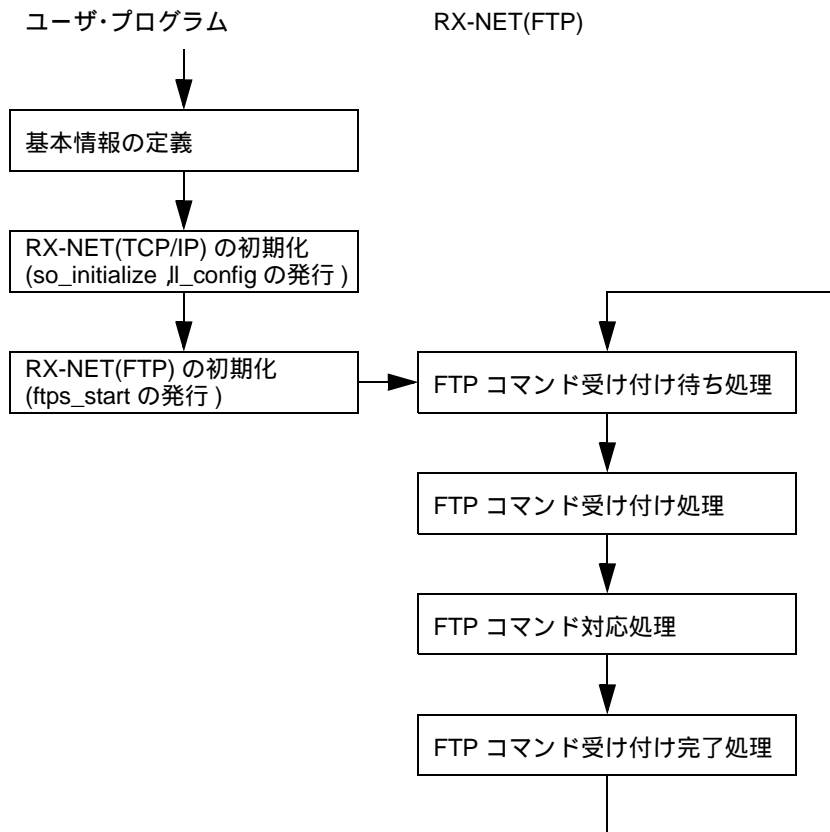
RX-NET(FTP)では、RX-NET(FTP)の初期化処理、強制終了処理の他に、FTPクライアントから送信されたFTPコマンドを受け付けた際、および、各種イベントが発生した際に呼び出す応答関数を“FTPサーバ機能”として提供しています。なお、RX-NET(FTP)では、サーバ、クライアント間の最大同時接続数として、100セッションのリモート・アクセスを可能としています。

## 4.2 処理の流れ

RX-NET(FTP)では、FTPクライアントから送信されたFTPコマンドに対応する際に必要となる各種処理の実行手順を規定しています。

図4-1に、RX-NET(FTP)が提供しているFTPサーバ機能を利用したFTPコマンドの対応手順を示します。

図4-1 FTPコマンドの対応手順



## 4.3 FTP サーバ機能 API 関数

### 4.3.1 RX-NET(FTP) の初期化

RX-NET(FTP) の初期化は、以下に示した API 関数を処理プログラム ( タスク ) から発行することにより実現されます。

- `ftps_start`

パラメータ `cnt` , `port` で指定された情報をもとに、RX-NET(FTP) が提供する機能を実現するうえで必要となる各種初期化処理を実行したのち、FTP クライアントから送信される各種 FTP コマンド ( USER , CWD , CDUP など ) の受け付け待ち状態となります。

なお、RX-NET(FTP) では、RX-NET(FTP) の初期化処理として、RX-NET(FTP) が提供している機能を実現する際に必要となる各種資源の生成、および、メモリ領域の確保を行っています。

以下に、本 API 関数の記述例を示します。

なお、記述例中の `so_initialize` , `ll_config` は RX-NET(TCP/IP) が提供している API 関数、`ext_tsk` は RX850 Pro が提供しているシステム・コールです。

```
#include <stdrx85p.h> /* RX850 Pro 用標準ヘッダ・ファイルの定義 */
#include <rxnet.h> /* RX-NET(TCP/IP) 用標準ヘッダ・ファイルの定義 */
#include <fnsconfig.h> /* 静的設定情報ヘッダ・ファイルの定義 */
#include <rxnet_ftps.h> /* RX-NET(FTP) 用標準ヘッダ・ファイルの定義 */

/*****
基本情報の定義
*****/
#define FTPS_SIZE 4096
int ftps_rcvbufsize = FTPS_SIZE ;
char ftps_rcvbuf [ FTPS_SIZE ] ;
#define FTPS_TIMEOUT 300000
unsigned int ftps_idle_timeout = FTPS_TIMEOUT ;
const char
r125 [] = "Data connection already open on socket %d ; transfer starting." ,
r150 [] = "Opening data connection for (%s) (%s)." ,
r200 [] = "Command okay." ,
.....
.....
.....
r551 [] = "Requested action aborted : page type unknown." ,
r552 [] = "Requested file action aborted." ,
r553 [] = "Requested action not taken. File name not allowed." ;

void
func_task ( INT stacd ) {
    char devname [] = "s91s" ; /* 変数の宣言, 初期化 */
    u32 ipaddress = ine_taddr ( "192.168.0.1" ) ; /* 変数の宣言, 初期化 */
    int ipmask = inet_addr ( "255.255.255.0" ) ; /* 変数の宣言, 初期化 */
    int cnt = 100 ; /* 変数の宣言, 初期化 */
    unsigned short port = 21 ; /* 変数の宣言, 初期化 */

    so_initialize ( ) ; /* RX-NET(TCP/IP) の初期化 */
/* ネットワーク・インタフェースの起動 */
    ll_config ( devname , ipaddress , ipmask , 0x0 , ( u32 ) 0x0 ) ;
    ftps_start ( cnt , port ) ; /* RX-NET(FTP) の初期化 */
    exd_tsk ( ) ; /* タスクの終了処理 */
}
```

- 注意 1 本 API 関数の発行は、基本情報 (受信バッファ情報、タイムアウト時間情報、メッセージ情報など) の定義処理、および、RX-NET(TCP/IP) が提供する API 関数 `so_initialize`、`ll_config` の処理完了後に行う必要があります。  
ただし、RX-NET(DHCP) が提供する機能を併せて使用する場合には、API 関数 `ll_config` の発行が不要となります。
- 注意 2 本 API 関数の発行により遷移した “受け付け待ち状態” の解除は、本 API 関数を発行したタスク以外の処理プログラム (タスク、非タスク) から API 関数 `ftps_abort` を発行することにより実現されます。
- 注意 3 制御用 TCP コネクション・ポート番号 `port` には、通常、21 番を設定します。  
なお、`port` に 0x0 を指定した場合、RX-NET(FTP) は “制御用 TCP コネクション・ポート番号” のデフォルト値 21 が指定されたものとして処理を行います。
- 注意 4 基本情報 (受信バッファ情報、タイムアウト時間情報、メッセージ情報など) についての詳細は、「第 6 章 RX-NET(FTP) 依存部」を参照してください。

### 4.3.2 RX-NET(FTP) の強制終了

RX-NET(FTP) の強制終了は、以下に示した API 関数を処理プログラム (タスク、非タスク) から発行することにより実現されます。

- `ftps_abort`

RX-NET(FTP) の終了処理を強制的に実行します。

なお、RX-NET(FTP) では、RX-NET(FTP) の終了処理として、RX-NET(FTP) が提供している機能を実現する際に必要となる各種資源の削除、メモリ領域の解放を行っています。

これにより、すべての処理プログラムから RX-NET(FTP) が提供する機能を利用することができなくなります。

また、本 API 関数の発行により受け付け待ち状態を解除されたタスクには、受け付け待ち状態へと遷移するきっかけとなった API 関数 `ftps_start` の戻り値として `EFTP_ABORT` が返されます。

以下に、本 API 関数の記述例を示します。

なお、記述例中の `ext_tsk` は、RX850 Pro が提供しているシステム・コールです。

```
#include <stdrx85p.h> /* RX850 Pro 用標準ヘッダ・ファイルの定義 */
#include <rxnet.h> /* RX-NET(TCP/IP) 用標準ヘッダ・ファイルの定義 */
#include <fnsconfig.h> /* 静的設定情報ヘッダ・ファイルの定義 */
#include <rxnet_ftps.h> /* RX-NET(FTP) 用標準ヘッダ・ファイルの定義 */

void
func_task ( INT stacd ) {
    ftps_abort ( ); /* RX-NET(FTP) の強制終了 */
    exd_tsk ( ); /* タスクの終了処理 */
}
```

注意 1 本 API 関数の発行は、API 関数 `ftps_start` を発行したタスク以外の処理プログラム (タスク、非タスク) からに限られています。

注意 2 本 API 関数では、終了処理要求のキューイングが行われません。このため、すでに本 API 関数が発行され、RX-NET(FTP) の終了処理が実行されていた場合には、何も処理は行わず、エラーとしても扱いません。

## 4.4 FTP サーバ機能応答関数

RX-NET(FTP) では、FTP クライアントから送信された FTP コマンドを受け付けた際、および、各種イベントが発生した際、該当する応答関数の呼び出し処理を行っています。

表 4-1 に、応答関数と RX-NET(FTP) が処理対象としている FTP コマンド、および、イベントの対応を示します。

表 4-1 応答関数

応答関数名	機能概要
ftps_login_user	USER, PASS コマンドを受け付けた際に呼び出される応答関数
ftps_cwd_user	CWD コマンドを受け付けた際に呼び出される応答関数
ftps_cdup_user	CDUP, XCUP コマンドを受け付けた際に呼び出される応答関数
ftps_logout_user	QUIT コマンドを受け付けた際に呼び出される応答関数
RX-NET(FTP) 本体処理部	PORT コマンドを受け付けた際に呼び出される応答関数
RX-NET(FTP) 本体処理部	PASV コマンドを受け付けた際に呼び出される応答関数
ftps_type_user	TYPE コマンドを受け付けた際に呼び出される応答関数
RX-NET(FTP) 本体処理部	STRU コマンドを受け付けた際に呼び出される応答関数
ftps_mode_user	MODE コマンドを受け付けた際に呼び出される応答関数
ftps_retr_user	RETR コマンドを受け付けた際に呼び出される応答関数
ftps_stor_user	STOR コマンドを受け付けた際に呼び出される応答関数
ftps_stou_user	STOU コマンドを受け付けた際に呼び出される応答関数
ftps_appe_user	APPE コマンドを受け付けた際に呼び出される応答関数
ftps_allo_user	ALLO コマンドを受け付けた際に呼び出される応答関数
RX-NET(FTP) 本体処理部	RNFR コマンドを受け付けた際に呼び出される応答関数
ftps_rnto_user	RNTO コマンドを受け付けた際に呼び出される応答関数
ftps_abor_user	ABOR コマンドを受け付けた際に呼び出される応答関数
ftps_dele_user	DELE コマンドを受け付けた際に呼び出される応答関数
ftps_rmd_user	RMD, XRMD コマンドを受け付けた際に呼び出される応答関数
ftps_mkd_user	MKD, XMKD コマンドを受け付けた際に呼び出される応答関数
ftps_pwd_user	PWD, XPWD コマンドを受け付けた際に呼び出される応答関数
ftps_list_user	LIST コマンドを受け付けた際に呼び出される応答関数
ftps_nlst_user	NLST コマンドを受け付けた際に呼び出される応答関数
ftps_syst_user	SYST コマンドを受け付けた際に呼び出される応答関数
RX-NET(FTP) 本体処理部	STAT コマンドを受け付けた際に呼び出される応答関数
RX-NET(FTP) 本体処理部	HELP コマンドを受け付けた際に呼び出される応答関数
ftps_recv_user	転送データ受信イベントが発生した際に呼び出される応答関数
ftps_send_user	転送データ送信イベントが発生した際に呼び出される応答関数
ftps_opensession_user	セッション開始イベントが発生した際に呼び出される応答関数
ftps_closesession_user	セッション切断イベントが発生した際に呼び出される応答関数
ftps_timeout_user	タイムアウト・イベントが発生した際に呼び出される応答関数

注意 応答関数についての詳細は、「第6章 RX-NET(FTP) 依存部」を参照してください。



# 第 5 章 API 関数

本章では、RX-NET(FTP) が提供しているアプリケーション・プログラム・インタフェース関数 (API 関数) について解説しています。

## 5.1 概要

RX-NET(FTP) が提供している API 関数は、ユーザが記述した処理プログラムから RX-NET(FTP) が直接管理している資源を間接的に操作するために用意されたサービス・ルーチンです。

以下に、RX-NET(FTP) が提供している API 関数 (2 種類) を示します。

ftps\_start                      ftps\_abort

## 5.2 API 関数の呼び出し

API 関数を C 言語、および、アセンブリ言語で記述された処理プログラムから発行する場合の呼び出し方法を以下に示します。

- C 言語

API 関数を C 言語で記述された処理プログラムから発行する場合、通常の C 言語関数と同様の方法で呼び出しを行うことにより、API 関数のパラメータは RX-NET(FTP) に引き数として渡され、該当処理が実行されます。

- アセンブリ言語

API 関数をアセンブリ言語で記述された処理プログラムから発行する場合、ユーザが開発環境として使用する C コンパイラ・パッケージの関数呼び出し規約に従ったパラメータ、および、戻り番地の設定を行ったのち、jarl 命令による呼び出しを行うことにより、API 関数のパラメータは RX-NET(FTP) に引き数として渡され、該当処理が実行されます。

**注意** RX-NET(FTP) が提供する API 関数を処理プログラムから発行する場合、以下に示したヘッダ・ファイルの定義 (インクルード処理) を行う必要があります。

rxnet.h                      : RX-NET(TCP/IP) 用標準ヘッダ・ファイル  
fnsconfig.h                 : 静的設定情報ヘッダ・ファイル  
rxnet\_ftps.h                : RX-NET(FTP) 用標準ヘッダ・ファイル

なお、rxnet.h、fnsconfig.h は、RX-NET(TCP/IP) が提供しています。

## 5.3 データ・マクロ

RX-NET(FTP)が提供するAPI関数を発行する際に使用する各種データ・マクロ(戻り値など)について以下に示します。

### 5.3.1 戻り値

表 5-1 に、API 関数からの戻り値一覧を示します。

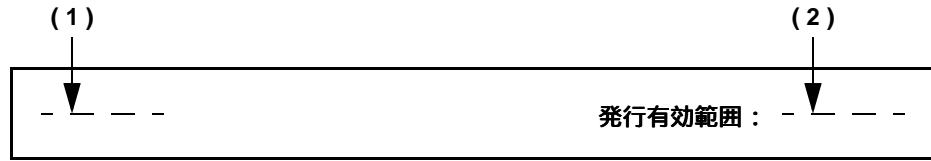
なお、戻り値のマクロ定義は、標準ヘッダ・ファイル `necools32\inc850\rxnet_ftps.h` から呼び出されるヘッダ・ファイル `necools32\inc850\rxnet\ftperrno.h` で行われています。

表 5-1 戻り値

マクロ	数値	意味
EFTP_NOMEM	0x1400	メモリ領域が確保できません
EFTP_INVALID	0x1401	パラメータの指定が不正です
EFTP_SOCKET	0x1402	ソケットの生成処理が失敗しました
EFTP_BIND	0x1403	バインド処理 (IP アドレス ポート番号の割り付け処理) が失敗しました
EFTP_DOWN	0x1404	RX-NET(TCP/IP) の初期化処理が完了していません
EFTP_ABORT	0x1405	ftp_abort の発行により ftp_start の処理が強制終了しました
EFTP_RUNNING	0x1406	既に ftp_start が発行されています

## 5.4 API 関数解説

次項から RX-NET(FTP) が提供している API 関数について、以下の記述フォーマットに従って解説します。



(3) —▶ **概要**

-----  
 -----  
 -----

(4) —▶ **C 言語形式**

-----  
 -----  
 -----

(5) —▶ **パラメータ**

I/O	パラメータ	説 明

(6) —▶ **機能**

-----  
 -----  
 -----  
 -----  
 -----

(7) —▶ **戻り値**

-----  
 -----

**( 1 ) 名称**

API 関数の名称を示しています。

**( 2 ) 発行有効範囲**

API 関数の発行が可能な処理プログラムの種別を示しています。

タスク : タスクからのみ発行可能  
 非タスク : 非タスクからのみ発行可能  
 タスク / 非タスク : タスク, 非タスクのどちらかも発行可能

**( 3 ) 概要**

API 関数の機能概要を示しています。

**( 4 ) C 言語形式**

API 関数を C 言語で記述された処理プログラムから発行する際の記述形式を示しています。

**( 5 ) パラメータ**

API 関数のパラメータを以下の形式で示しています。

I/O	パラメータ	説 明
A	B	C

**A ) パラメータの種類**

- I ... RX-NET(FTP) への入力パラメータ
- O ... RX-NET(FTP) からの出力パラメータ

**B ) パラメータのデータ・タイプ****C ) パラメータの説明****( 6 ) 機能**

API 関数の機能詳細を示しています。

**( 7 ) 戻り値**

API 関数からの戻り値をデータ・マクロ, および, 数値で示しています。

### 5.4.1 外部インタフェース仕様

表 5-2 に、RX-NET(FTP) が提供している API 関数の一覧を示します。

表 5-2 API 関数

API 関数名	機能概要
ftps_start	RX-NET(FTP) の初期化
ftps_abort	RX-NET(FTP) の強制終了

次頁以降に、各種 API 関数の外部インタフェース仕様詳細を示します。

**ftps\_start**

発行有効範囲：タスク

**概要**

RX-NET(FTP) の初期化

**C 言語形式**

```
int      ftps_start ( int cnt , unsigned short port );
```

**パラメータ**

I/O	パラメータ	説明
I	int <i>cnt</i> ;	最大セッション数 (1 ~ 100)
I	unsigned short <i>port</i> ;	制御用 TCP コネクション・ポート番号

**機能**

*cnt* , *port* で指定された情報をもとに、RX-NET(FTP) が提供する機能を実現するうえで必要となる各種初期化処理を実行したのち、FTP クライアントから送信される各種 FTP コマンド (USER, CWD, CDUP など) の受け付け待ち状態となります。

なお、RX-NET(FTP) では、RX-NET(FTP) の初期化処理として、RX-NET(FTP) が提供している機能を実現する際に必要となる各種資源の生成、および、メモリ領域の確保を行っています。

- 注意 1 本 API 関数の発行は、基本情報 (受信バッファ情報、タイムアウト時間情報、メッセージ情報など) の定義処理、および、RX-NET(TCP/IP) が提供する API 関数 `so_initialize_ll_config` の処理完了後に行う必要があります。ただし、RX-NET(DHCP) が提供する機能を併せて使用する場合には、API 関数 `ll_config` の発行が不要となります。
- 注意 2 本 API 関数の発行により遷移した“受け付け待ち状態”の解除は、本 API 関数を発行したタスク以外の処理プログラム (タスク、非タスク) から API 関数 `ftps_abort` を発行することにより実現されます。
- 注意 3 制御用 TCP コネクション・ポート番号 *port* には、通常、21 番を設定します。  
なお、*port* に 0x0 を指定した場合、RX-NET(FTP) は“制御用 TCP コネクション・ポート番号”のデフォルト値 21 が指定されたものとして処理を行います。
- 注意 4 基本情報 (受信バッファ情報、タイムアウト時間情報、メッセージ情報など) についての詳細は、「第 6 章 RX-NET(FTP) 依存部」を参照してください。

**戻り値**

EFTP_NOMEM	0x1400	データ管理ブロック用メモリ領域が確保できません
EFTP_INVALID	0x1401	パラメータの指定が不正です
EFTP_SOCKET	0x1402	ソケットの生成処理が失敗しました
EFTP_BIND	0x1403	バインド処理 (IP アドレス、ポート番号の割り付け処理) が失敗しました
EFTP_DOWN	0x1404	RX-NET(TCP/IP) の初期化処理が完了していません
EFTP_ABORT	0x1405	<code>ftps_abort</code> の発行により本 API 関数の処理が強制終了しました
EFTP_RUNNING	0x1406	既に本 API 関数が発行されています

**ftps\_abort**

発行有効範囲：タスク / 非タスク

**概要**

RX-NET(FTP) の強制終了

**C 言語形式**

```
void ftps_abort ( void );
```

**パラメータ**

なし

**機能**

RX-NET(FTP) の終了処理を強制的に実行します。

なお、RX-NET(FTP) では、RX-NET(FTP) の終了処理として、RX-NET(FTP) が提供している機能を実現する際に必要となる各種資源の削除、メモリ領域の解放を行っています。

これにより、すべての処理プログラムから RX-NET(FTP) が提供する機能を利用することができなくなります。

また、本 API 関数の発行により受け付け待ち状態を解除されたタスクには、受け付け待ち状態へと遷移するきっかけとなった API 関数 ftps\_start の戻り値として EFTP\_ABORT が返されます。

注意 1 本 API 関数の発行は、API 関数 ftps\_start を発行したタスク以外の処理プログラム ( タスク、非タスク ) からに限られています。

注意 2 本 API 関数では、終了処理要求のキューイングが行われません。このため、すでに本 API 関数が発行され、RX-NET(FTP) の終了処理が実行されていた場合には、何も処理は行わず、エラーとしても扱いません。

**戻り値**

なし

# 第 6 章 RX-NET(FTP) 依存部

本章では、RX-NET(FTP) が提供している RX-NET(FTP) 依存部について解説しています。

## 6.1 概要

RX-NET(FTP) では、RX-NET(FTP) が提供する機能を実現する際に必要となる基本情報、および、応答関数を RX-NET(FTP) 依存部 (ユーザ・OWN・コーディング部) として切り出し、サンプル・ソース・ファイルを提供しています。

そこで、システム構築時には、これら基本情報、および、応答関数をユーザの実行環境 / アプリケーション・システムにあわせてカスタマイズ化する必要があります。

以下に、RX-NET(FTP) が RX-NET(FTP) 依存部として提供している応答関数 (25 種類) を示します。

ftps_login_user	ftps_cwd_user	ftps_cdup_user	ftps_logout_user
ftps_type_user	ftps_mode_user	ftps_retr_user	ftps_stor_user
ftps_stou_user	ftps_appe_user	ftps_allo_user	ftps_rnto_user
ftps_abor_user	ftps_dele_user	ftps_rmd_user	ftps_mkd_user
ftps_pwd_user	ftps_list_user	ftps_nlst_user	ftps_syst_user
ftps_rcv_user	ftps_send_user	ftps_opensession_user	ftps_closesession_user
ftps_timeout_user			

なお、応答関数の呼び出しは、FTP クライアントから送信された FTP コマンドを受け付けた際、および、各種イベントが発生した際、RX-NET(FTP) により行われます。

したがって、応答関数への入力パラメータに対する設定処理は RX-NET(FTP) が行っています。

## 6.2 基本情報

RX-NET(FTP) が提供する機能を実現する際に必要となる基本情報 (受信バッファ情報、タイムアウト時間情報、メッセージ情報など) について以下に示します。

### 6.2.1 受信バッファ情報

RX-NET(FTP) では、FTP クライアントから送信された転送データを一時的に格納する際に用いる受信バッファのサイズを RX-NET(FTP) 依存部として切り出しています。

このため、ユーザは、規定されたマクロ名、および、変数名を用いて受信バッファ情報を定義する必要があります。

なお、受信バッファのサイズ (単位: バイト) として指定可能な値は、0x1 ~ 0x1000 に限られます。

以下に、受信バッファ情報の記述例を示します。

```
/* バッファ・サイズのマクロ定義 */
#define          FTPS_SIZE          0x1000

/* バッファ・サイズの変数宣言, 初期化 */
int             ftps_rcvbufsize = FTPS_SIZE ;

/* バッファ領域の確保 */
char           ftps_rcvbuf [ FTPS_SIZE ] ;
```

注意 RX-NET(FTP) では、本受信バッファ情報のサンプル・ソース・ファイルを提供しています。

【 CA850 対応版の場合 】

nec tools32\smp850e\rxnet\\src\ftpsconfig.c

【 CCV850E 対応版の場合 】

nec tools32\smp850e\_ghs\rxnet\\src\ftpsconfig.c



## 6.2.2 タイムアウト時間情報

RX-NET(FTP) では、RX-NET(FTP) がタイムアウト・イベントを発生するまでの時限 ( タイムアウト時間 ) を RX-NET(FTP) 依存部として切り出しています。

このため、ユーザは、規定されたマクロ名、および、変数名を用いてタイムアウト時間情報を定義する必要があります。なお、タイムアウト時間 ( 単位：ミリ秒 ) として指定可能な値は、0x1 以上に限られます。

以下に、タイムアウト時間情報の記述例を示します。

```
/* タイムアウト時間のマクロ定義 */
#define          FTPS_TIMEOUT          300000

/* タイムアウト時間の変数宣言，初期化 */
int             ftps_idle_timeout = FTPS_TIMEOUT ;
```

注意 RX-NET(FTP) では、本タイムアウト時間情報のサンプル・ソース・ファイルを提供しています。

【 CA850 対応版の場合 】

necools32\smp850e\rxnet\\src\ftpsconfig.c

【 CCV850E 対応版の場合 】

necools32\smp850e\_ghs\rxnet\\src\ftpsconfig.c

## 6.2.3 メッセージ情報

RX-NET(FTP) では、FTP クライアントに送信する “ 応答コードに続くメッセージ (FTP クライアントから送信された FTP コマンドの受け付け処理結果) ” を RX-NET(FTP) 依存部として切り出しています。

このため、ユーザは、規定された変数名を用いてメッセージ情報を定義する必要があります。

以下に、メッセージ情報の記述例を示します。

```
/* メッセージ情報の変数宣言，初期化 */
const   char

        /* 応答コード 125 用メッセージの定義 */
        r_125 [] = "Data connection already open on socket %d ; transfer starting." ,

        /* 応答コード 150 用メッセージの定義 */
        r_150 [] = "Opening data connection for (%s) (%s)." ,

        /* 応答コード 200 用メッセージの定義 */
        r_200 [] = "Command okay." ,

        .....
        .....
        .....

        /* 応答コード 551 用メッセージの定義 */
        r_551 [] = "Requested action aborted : page type unknown." ,

        /* 応答コード 552 用メッセージの定義 */
        r_552 [] = "Requested file action aborted." ,

        /* 応答コード 553 用メッセージの定義 */
        r_553 [] = "Requested action not taken. File name not allowed." ;
```

注意 RX-NET(FTP) では、本メッセージ情報のサンプル・ソース・ファイルを提供しています。

【 CA850 対応版の場合 】

necools32\smp850e\rxnet\\src\ftpsconfig.c

【 CCV850E 対応版の場合 】

necools32\smp850e\_ghs\rxnet\\src\ftpsconfig.c

## 6.3 データ・マクロ

RX-NET(FTP) が提供する応答関数をカスタマイズ化するために使用する各種データ・マクロ ( データ表現タイプ, 垂直フォーマット制御種別, 転送モードなど ) について以下に示します。

### 6.3.1 戻り値

表 6-1 に, 応答関数 `ftps_login_user` からの戻り値一覧を示します。

なお, 戻り値のマクロ定義は, 標準ヘッダ・ファイル `nctools32\inc850\rxnet_ftps.h` から呼び出されるヘッダ・ファイル `nctools32\inc850\rxnet\spicom.h` で行われています。

表 6-1 戻り値

マクロ	数値	意味
USER_OK	0x0	ユーザ認証処理が正常終了しました
GUEST_OK	0x1	ゲスト認証処理が正常終了しました
NEED_PASSWORD	0x2	FTP クライアント側からのパスワードの送信が必要です
NEED_SIGN	0x3	FTP クライアント側からのメール・アドレスの送信が必要です
BAD_USER	0x4	ユーザ認証処理, または, ゲスト認証処理が失敗しました

### 6.3.2 データ表現タイプ

表 6-2 に, RX-NET(FTP) が FTP クライアントから送信された TYPE コマンドを受け付けた際に呼び出される応答関数 `ftps_type_user` で使用するデータ表現タイプ一覧を示します。

なお, データ表現タイプのマクロ定義は, 標準ヘッダ・ファイル `nctools32\inc850\rxnet_ftps.h` から呼び出されるヘッダ・ファイル `nctools32\inc850\rxnet\ftp.h` で行われています。

表 6-2 データ表現タイプ

マクロ	数値	意味
TYPE_ASCII	0x1	ASCII 表現タイプ
TYPE_BYTE	0x2	バイナリ表現タイプ
TYPE_IMAGE	0x3	イメージ表現タイプ

### 6.3.3 垂直フォーマット制御種別

表 6-3 に, RX-NET(FTP) が FTP クライアントから送信された TYPE コマンドを受け付けた際に呼び出される応答関数 `ftps_type_user` で使用する垂直フォーマット制御種別一覧を示します。

なお, 垂直フォーマット制御種別のマクロ定義は, 標準ヘッダ・ファイル `nctools32\inc850\rxnet_ftps.h` から呼び出されるヘッダ・ファイル `nctools32\inc850\rxnet\ftp.h` で行われています。

表 6-3 垂直フォーマット制御種別

マクロ	数値	意味
FORM_NONPRINTTYPE_ASCII	0x1	ノンプリント
FORM_TELNET	0x2	Telnet フォーマット制御
FORM_ASA	0x3	キャリッジ制御 (ASA)

### 6.3.4 転送モード

表 6-4 に、RX-NET(FTP) が FTP クライアントから送信された MODE コマンドを受け付けた際に呼び出される応答関数 `fps_mode_user` で使用する転送モード一覧を示します。

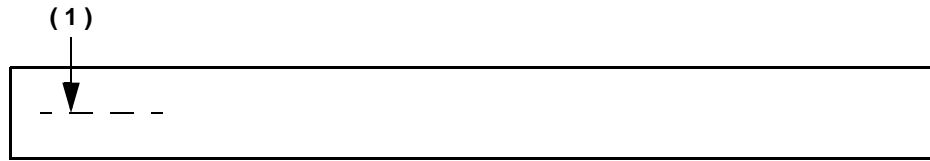
なお、転送モードのマクロ定義は、標準ヘッダ・ファイル `nctools32\inc850\rxnet_ftps.h` から呼び出されるヘッダ・ファイル `nctools32\inc850\rxnet\ftp.h` で行われています。

表 6-4 転送モード

マクロ	数値	意味
MODE_STREAM	0x1	ストリーム・モード
MODE_BLOCK	0x2	ブロック・モード
MODE_COMPRESSED	0x3	圧縮モード

## 6.4 応答関数解説

次項から RX-NET(FTP) が提供している応答関数について、以下の記述フォーマットに従って解説します。



(2) —▶ 概要

-----  
 -----

(3) —▶ C 言語形式

-----  
 -----

(4) —▶ パラメータ

I/O	パラメータ	説 明

(5) —▶ 機能

-----  
 -----  
 -----  
 -----

(6) —▶ 戻り値

-----  
 -----

**( 1 ) 名称**

応答関数の名称を示しています。

**( 2 ) 概要**

応答関数の機能概要を示しています。

**( 3 ) C 言語形式**

応答関数を RX-NET(FTP) から呼び出す際の記述形式を示しています。

**( 4 ) パラメータ**

応答関数のパラメータを以下の形式で示しています。

I/O	パラメータ	説 明
A	B	C

## A ) パラメータの種類

- I ... 応答関数への入力パラメータ
- O ... 応答関数からの出力パラメータ

## B ) パラメータのデータ・タイプ

## C ) パラメータの説明

**( 5 ) 機能**

応答関数の機能詳細を示しています。

**( 6 ) 戻り値**

応答関数からの戻り値をデータ・マクロ, および, 数値で示しています。

### 6.4.1 外部インタフェース仕様

表 6-5 に、RX-NET(FTP) が提供している応答関数の一覧を示します。

表 6-5 応答関数

応答関数名	機能概要
ftps_login_user	USER, PASS コマンドを受け付けた際に呼び出される応答関数
ftps_cwd_user	CWD コマンドを受け付けた際に呼び出される応答関数
ftps_cdup_user	CDUP, XCUP コマンドを受け付けた際に呼び出される応答関数
ftps_logout_user	QUIT コマンドを受け付けた際に呼び出される応答関数
ftps_type_user	TYPE コマンドを受け付けた際に呼び出される応答関数
ftps_mode_user	MODE コマンドを受け付けた際に呼び出される応答関数
ftps_retr_user	RETR コマンドを受け付けた際に呼び出される応答関数
ftps_stor_user	STOR コマンドを受け付けた際に呼び出される応答関数
ftps_stou_user	STOU コマンドを受け付けた際に呼び出される応答関数
ftps_appe_user	APPE コマンドを受け付けた際に呼び出される応答関数
ftps_allo_user	ALLO コマンドを受け付けた際に呼び出される応答関数
ftps_rnto_user	RNTO コマンドを受け付けた際に呼び出される応答関数
ftps_abor_user	ABOR コマンドを受け付けた際に呼び出される応答関数
ftps_dele_user	DELE コマンドを受け付けた際に呼び出される応答関数
ftps_rmd_user	RMD, XRMD コマンドを受け付けた際に呼び出される応答関数
ftps_mkd_user	MKD, XMKD コマンドを受け付けた際に呼び出される応答関数
ftps_pwd_user	PWD, XPWD コマンドを受け付けた際に呼び出される応答関数
ftps_list_user	LIST コマンドを受け付けた際に呼び出される応答関数
ftps_nlst_user	NLST コマンドを受け付けた際に呼び出される応答関数
ftps_syst_user	SYST コマンドを受け付けた際に呼び出される応答関数
ftps_recv_user	転送データ受信イベントが発生した際に呼び出される応答関数
ftps_send_user	転送データ送信イベントが発生した際に呼び出される応答関数
ftps_opensession_user	セッション開始イベントが発生した際に呼び出される応答関数
ftps_closesession_user	セッション切断イベントが発生した際に呼び出される応答関数
ftps_timeout_user	タイムアウト・イベントが発生した際に呼び出される応答関数

次頁以降に、各種応答関数の外部インタフェース仕様詳細を示します。

## ftps\_login\_user

### 概要

USER , PASS コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_login_user ( int session , char *name , char *passwd );
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
I	char <i>*name</i> ;	ログイン名を格納した領域へのポインタ
I	char <i>*passwd</i> ;	パスワードを格納した領域へのポインタ

### 機能

RX-NET(FTP) が FTP クライアントから送信された USER コマンド , または , PASS コマンドを受け付けた際に呼び出される応答関数です。

なお , USER コマンドにより本応答関数が呼び出された場合には , *passwd* で指定された領域に 0x0 が設定されます。以下に , 本応答関数で実行すべき処理を示します。

- *name* , および , *passwd* で指定されたユーザに対する認証処理
- *name* , および , *passwd* で指定されたユーザに対するログイン処理
- 認証結果を本応答関数の戻り値として設定

注意 RX-NET(FTP) では , 本応答関数のサンプル・ソース・ファイルを提供しています。

#### 【CA850 対応版の場合】

```
necools32\smp850e\rxnet\<sample_name>\src\callback.c
```

#### 【CCV850E 対応版の場合】

```
necools32\smp850e_ghs\rxnet\<sample_name>\src\callback.c
```

### 戻り値

USER_OK	0x0	ユーザ認証処理が正常終了しました
GUEST_OK	0x1	ゲスト認証処理が正常終了しました
NEED_PASSWORD	0x2	FTP クライアント側からのパスワードの送信が必要です
NEED_SIGN	0x3	FTP クライアント側からのメール・アドレスの送信が必要です
BAD_USER	0x4	ユーザ認証処理 , または , ゲスト認証処理が失敗しました

## ftps\_cwd\_user

### 概要

CWD コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_cwd_user ( int session , char **msgbuf , char *dirname ) ;
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
O	char <i>**msgbuf</i> ;	応答コードに続くメッセージを格納するキャラクタ・バッファへのポインタ
I	char <i>*dirname</i> ;	変更後のカレント・ディレクトリの絶対パスを格納した領域へのポインタ

### 機能

RX-NET(FTP) が FTP クライアントから送信された DWD コマンドを受け付けた際に呼び出される応答関数です。以下に、本応答関数で実行すべき処理を示します。

- カレント・ディレクトリを CWD コマンドを受け付けた時点のディレクトリから *dirname* で指定されたディレクトリに変更
- 応答コードに続くメッセージを *msgbuf* で指定された領域に格納
  - 250 の場合 : Requested file action okay, completed.
  - 550 の場合 : Requested action not taken.
- 応答コードを本応答関数の戻り値として設定

注意 RX-NET(FTP) では、本応答関数のサンプル・ソース・ファイルを提供しています。

【CA850 対応版の場合】

nec-tools32\smp850e\rxnet\

【CCV850E 対応版の場合】

nec-tools32\smp850e\_ghs\rxnet\

### 戻り値

250                    正常終了  
550                    異常終了



## ftps\_cdup\_user

### 概要

CDUP, XCUP コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_cdup_user ( int session , char **msgbuf );
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
O	char <i>**msgbuf</i> ;	応答コードに続くメッセージを格納するキャラクタ・バッファへのポインタ

### 機能

RX-NET(FTP) が FTP クライアントから送信された CDUP コマンド, または, XCUP コマンドを受け付けた際に呼び出される応答関数です。

以下に, 本応答関数で実行すべき処理を示します。

- カレント・ディレクトリを CDUP コマンド, または, XCUP コマンドを受け付けた時点のディレクトリから親ディレクトリに変更
- 応答コードに続くメッセージを *msgbuf* で指定された領域に格納
  - 200 の場合 : Requested file action okay, completed.
  - 550 の場合 : Requested action not taken.
- 応答コードを本応答関数の戻り値として設定

注意 RX-NET(FTP) では, 本応答関数のサンプル・ソース・ファイルを提供しています。

#### 【CA850 対応版の場合】

nectools32\smp850e\rxnet\\src\callback.c

#### 【CCV850E 対応版の場合】

nectools32\smp850e\_ghs\rxnet\\src\callback.c

### 戻り値

200                    正常終了  
550                    異常終了

## ftps\_logout\_user

### 概要

QUIT コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_logout_user ( int session , char *name );
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
I	char <i>*name</i> ;	ログイン名を格納した領域へのポインタ

### 機能

RX-NET(FTP) が FTP クライアントから送信された QUIT コマンドを受け付けた際に呼び出される応答関数です。以下に、本応答関数で実行すべき処理を示します。

- *name* で指定されたユーザに対するログアウト処理

注意 RX-NET(FTP) では、本応答関数のサンプル・ソース・ファイルを提供しています。

【CA850 対応版の場合】

```
nectools32\smp850e\rxnet\\src\callback.c
```

【CCV850E 対応版の場合】

```
nectools32\smp850e_ghs\rxnet\\src\callback.c
```

### 戻り値

なし

## ftps\_type\_user

### 概要

TYPE コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_type_user ( int session , char **msgbuf , int type , int option ) ;
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
O	char <i>**msgbuf</i> ;	応答コードに続くメッセージを格納するキャラクタ・バッファへのポインタ
I	int <i>type</i> ;	データ表現フォーマット TYPE_ASCII : ASCII 表現タイプ TYPE_BYTE : バイナリ表現タイプ TYPE_IMAGE : イメージ表現タイプ
I	int <i>option</i> ;	type で指定されたデータ表現フォーマットに対応した値 ( 垂直フォーマット制御種別 , データ・サイズ ) FORMAT_NONPRINT : ノンプリント FORMAT_TELNET : Telnet フォーマット制御 FORMAT_ASA : キャリッジ制御 (ASA)

### 機能

RX-NET(FTP) が FTP クライアントから送信された TYPE コマンドを受け付けた際に呼び出される応答関数です。

なお、*option* で指定された領域には、*type* で指定された領域に TYPE\_ASCII が設定されていた場合には垂直フォーマット制御種別 (FORMAT\_NONPRINT, FORMAT\_TELNET, FORMAT\_ASA) が、TYPE\_BYTE が設定されていた場合にはデータ・サイズが設定されます。

以下に、本応答関数で実行すべき処理を示します。

- 応答コードに続くメッセージを *msgbuf* で指定された領域に格納  
200 の場合 : Requested file action okay, completed.  
504 の場合 : Command not implemented for that parameter.
- 応答コードを本応答関数の戻り値として設定

注意 RX-NET(FTP) では、本応答関数のサンプル・ソース・ファイルを提供しています。

#### 【CA850 対応版の場合】

```
necools32\smp850e\rxnet\\src\callback.c
```

#### 【CCV850E 対応版の場合】

```
necools32\smp850e_ghs\rxnet\\src\callback.c
```

### 戻り値

200

正常終了

504

異常終了

## ftps\_mode\_user

### 概要

MODE コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_mode_user ( int session , char **msgbuf , int mode );
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
O	char <i>**msgbuf</i> ;	応答コードに続くメッセージを格納するキャラクタ・バッファへのポインタ
I	int <i>mode</i> ;	転送モード MODE_STREAM : ストリーム・モード MODE_BLOCK : ブロック・モード MODE_COMPRESSED : 圧縮モード

### 機能

RX-NET(FTP) が FTP クライアントから送信された MODE コマンドを受け付けた際に呼び出される応答関数です。以下に、本応答関数で実行すべき処理を示します。

- 応答コードに続くメッセージを *msgbuf* で指定された領域に格納  
200 の場合 : Requested file action okay, completed.  
504 の場合 : Command not implemented for that parameter.
- 応答コードを本応答関数の戻り値として設定

注意 RX-NET(FTP) では、本応答関数のサンプル・ソース・ファイルを提供しています。

#### 【CA850 対応版の場合】

nectools32\smp850e\rxnet\\src\callback.c

#### 【CCV850E 対応版の場合】

nectools32\smp850e\_ghs\rxnet\\src\callback.c

### 戻り値

200 正常終了  
504 異常終了

## ftps\_retr\_user

### 概要

RETR コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_retr_user ( int session , char **msgbuf , char *filename );
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
O	char <i>**msgbuf</i> ;	応答コードに続くメッセージを格納するキャラクタ・バッファへのポインタ
I	int <i>*filename</i> ;	FTP クライアントに送信するファイルのファイル名を格納した領域へのポインタ

### 機能

RX-NET(FTP) が FTP クライアントから送信された RETR コマンドを受け付けた際に呼び出される応答関数です。なお、RX-NET(FTP) では、本応答関数の処理完了後、応答関数 `ftps_send_user` の呼び出しを行っています。以下に、本応答関数で実行すべき処理を示します。

- *filename* で指定されたファイルのオープン
- *filename* で指定されたファイルの転送データの作成
- 応答コードに続くメッセージを *msgbuf* で指定された領域に格納  
550 の場合 : Requested action not taken.
- 応答コードを本応答関数の戻り値として設定

注意 1 RX-NET(FTP) では、本応答関数からの戻り値が 200 であった場合には、RX-NET(FTP) が用意しているメッセージを“ 応答コードに続くメッセージ ”として FTP クライアントに送信しています。このため、本応答関数からの戻り値が 200 となる場合には、“ 応答コードに続くメッセージを *msgbuf* で指定された領域に格納 ”を実行する必要はありません。

注意 2 RX-NET(FTP) では、本応答関数のサンプル・ソース・ファイルを提供しています。

【CA850 対応版の場合】

nertools32\smp850e\rxnet\<sample\_name>\src\callback.c

【CCV850E 対応版の場合】

nertools32\smp850e\_ghs\rxnet\<sample\_name>\src\callback.c

### 戻り値

200	正常終了
550	異常終了

## ftps\_stor\_user

### 概要

STOR コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_stor_user ( int session , char **msgbuf , char *filename ) ;
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
O	char <i>**msgbuf</i> ;	応答コードに続くメッセージを格納するキャラクタ・バッファへのポインタ
I	int <i>*filename</i> ;	FTPクライアントから受信した転送データを保存(上書き込み)するファイルのファイル名を格納した領域へのポインタ

### 機能

RX-NET(FTP) が FTP クライアントから送信された STOR コマンドを受け付けた際に呼び出される応答関数です。なお、RX-NET(FTP) では、本応答関数の処理完了後、応答関数 `ftps_recv_user` の呼び出しを行っています。以下に、本応答関数で実行すべき処理を示します。

- *filename* で指定されたファイルを“上書き込みモード”でオープン
- 応答コードに続くメッセージを *msgbuf* で指定された領域に格納  
500 の場合 : Syntax error, command unrecognized.
- 応答コードを本応答関数の戻り値として設定

注意 1 RX-NET(FTP) では、本応答関数からの戻り値が 200 であった場合には、RX-NET(FTP) が用意しているメッセージを“応答コードに続くメッセージ”として FTP クライアントに送信しています。このため、本応答関数からの戻り値が 200 となる場合には、“応答コードに続くメッセージを *msgbuf* で指定された領域に格納”を実行する必要はありません。

注意 2 RX-NET(FTP) では、本応答関数のサンプル・ソース・ファイルを提供しています。

【CA850 対応版の場合】

```
nectools32\smp850e\rxnet\<sample_name>\src\callback.c
```

【CCV850E 対応版の場合】

```
nectools32\smp850e_ghs\rxnet\<sample_name>\src\callback.c
```

### 戻り値

200	正常終了
500	異常終了

## ftps\_stou\_user

### 概要

STOU コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_stou_user ( int session , char **msgbuf , char **filename ) ;
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
O	char <i>**msgbuf</i> ;	応答コードに続くメッセージを格納するキャラクタ・バッファへのポインタ
O	int <i>**filename</i> ;	FTPクライアントから受信した転送データを保存(上書き込み)するファイルのファイル名を格納した領域へのポインタ

### 機能

RX-NET(FTP) が FTP クライアントから送信された STOU コマンドを受け付けた際に呼び出される応答関数です。なお、RX-NET(FTP) では、本応答関数の処理完了後、応答関数 `ftps_recv_user` の呼び出しを行っています。以下に、本応答関数で実行すべき処理を示します。

- FTP クライアントから受信した転送データを保存するファイルのファイル名を *filename* で指定された領域に格納
- 応答コードに続くメッセージを *msgbuf* で指定された領域に格納  
500 の場合 : Syntax error, command unrecognized.
- 応答コードを本応答関数の戻り値として設定

注意 1 RX-NET(FTP) では、本応答関数からの戻り値が 200 であった場合には、RX-NET(FTP) が用意しているメッセージを“ 応答コードに続くメッセージ ”として FTP クライアントに送信しています。このため、本応答関数からの戻り値が 200 となる場合には、“ 応答コードに続くメッセージを *msgbuf* で指定された領域に格納 ”を実行する必要はありません。

注意 2 RX-NET(FTP) では、本応答関数のサンプル・ソース・ファイルを提供しています。

【CA850 対応版の場合】

`nectools32\smp850e\rxnet\<sample_name>\src\callback.c`

【CCV850E 対応版の場合】

`nectools32\smp850e_ghs\rxnet\<sample_name>\src\callback.c`

### 戻り値

200	正常終了
500	異常終了



## ftps\_appe\_user

### 概要

APPE コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_appe_user ( int session , char **msgbuf , char *filename ) ;
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
O	char <i>**msgbuf</i> ;	応答コードに続くメッセージを格納するキャラクタ・バッファへのポインタ
I	int <i>*filename</i> ;	FTPクライアントから受信した転送データを保存(追加書き込み)するファイルのファイル名を格納した領域へのポインタ

### 機能

RX-NET(FTP) が FTP クライアントから送信された APPE コマンドを受け付けた際に呼び出される応答関数です。なお、RX-NET(FTP) では、本応答関数の処理完了後、応答関数 `ftps_recv_user` の呼び出しを行っています。以下に、本応答関数で実行すべき処理を示します。

- *filename* で指定されたファイルを“追加書き込みモード”でオープン
- 応答コードに続くメッセージを *msgbuf* で指定された領域に格納  
550 の場合 : Requested action not taken.
- 応答コードを本応答関数の戻り値として設定

注意 1 RX-NET(FTP) では、本応答関数からの戻り値が 200 であった場合には、RX-NET(FTP) が用意しているメッセージを“応答コードに続くメッセージ”として FTP クライアントに送信しています。このため、本応答関数からの戻り値が 200 となる場合には、“応答コードに続くメッセージを *msgbuf* で指定された領域に格納”を実行する必要はありません。

注意 2 RX-NET(FTP) では、本応答関数のサンプル・ソース・ファイルを提供しています。

【CA850 対応版の場合】

```
nectools32\smp850e\rxnet\<sample_name>\src\callback.c
```

【CCV850E 対応版の場合】

```
nectools32\smp850e_ghs\rxnet\<sample_name>\src\callback.c
```

### 戻り値

200	正常終了
550	異常終了

## ftps\_allo\_user

### 概要

ALLO コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_allo_user ( int session , char **msgbuf , int n1 , int n2 ) ;
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
O	char <i>**msgbuf</i> ;	応答コードに続くメッセージを格納するキャラクタ・バッファへのポインタ
I	int <i>n1</i> ;	転送データを受信する際に必要となるメモリ領域 ( 単位 : バイト )
I	int <i>n2</i> ;	<i>n1</i> で指定されたメモリ領域の “ ページ数 / レコード数 ” ( 単位 : バイト )

### 機能

RX-NET(FTP) が FTP クライアントから送信された ALLO コマンドを受け付けた際に呼び出される応答関数です。以下に、本応答関数で実行すべき処理を示します。

- *n1* , および *n2* で指定された情報をもとに、FTP クライアントから送信される転送データを格納するためのメモリ領域を確保
- 応答コードに続くメッセージを *msgbuf* で指定された領域に格納
  - 200 の場合 : Command okay.
  - 202 の場合 : Command not implemented, superfluous at this site.
  - 501 の場合 : Syntax error in parameters or arguments.
- 応答コードを本応答関数の戻り値として設定

注意 RX-NET(FTP) では、本応答関数のサンプル・ソース・ファイルを提供しています。

#### 【CA850 対応版の場合】

```
nectools32\smp850e\rxnet\<sample_name>\src\callback.c
```

#### 【CCV850E 対応版の場合】

```
nectools32\smp850e_ghs\rxnet\<sample_name>\src\callback.c
```

### 戻り値

200	正常終了
202	異常終了 (ALLO コマンドに対応した処理が未実装)
501	異常終了 (その他のエラー)

## ftps\_rnto\_user

### 概要

RNTO コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_rnto_user ( int session , char **msgbuf , char *oldname , char *newname ) ;
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
O	char <i>**msgbuf</i> ;	応答コードに続くメッセージを格納するキャラクタ・バッファへのポインタ
I	char <i>*oldname</i> ;	変更前のディレクトリ名, または, ファイル名を格納した領域へのポインタ
I	char <i>*newname</i> ;	変更後のディレクトリ名, または, ファイル名を格納した領域へのポインタ

### 機能

RX-NET(FTP) が FTP クライアントから送信された RNTO コマンドを受け付けた際に呼び出される応答関数です。  
 なお, *oldname* で指定された領域には, RX-NET(FTP) が FTP クライアントから送信された RNFR コマンドを受け付けた際に獲得した変更前のディレクトリ名, または, ファイル名が格納されます。  
 以下に, 本応答関数で実行すべき処理を示します。

- *oldname* で指定されたディレクトリ名, または, ファイル名を *newname* で指定されたディレクトリ名, または, ファイル名に変更
- 応答コードに続くメッセージを *msgbuf* で指定された領域に格納
  - 250 の場合 : Requested file action okay, completed.
  - 532 の場合 : Need account for storing files.
  - 553 の場合 : Requested action not taken.
- 応答コードを本応答関数の戻り値として設定

注意 RX-NET(FTP) では, 本応答関数のサンプル・ソース・ファイルを提供しています。

#### 【CA850 対応版の場合】

```
nctools32\smp850e\rxnet\

```

#### 【CCV850E 対応版の場合】

```
nctools32\smp850e_ghs\rxnet\

```

### 戻り値

250	正常終了
532	異常終了 ( ユーザ認証処理, または, ゲスト認証処理が未実装 )
553	異常終了 ( その他のエラー )

## ftps\_abor\_user

### 概要

ABOR コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_abor_user ( int session );
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号

### 機能

RX-NET(FTP) が FTP クライアントから送信された ABOR コマンドを受け付けた際に呼び出される応答関数です。なお、RX-NET(FTP) では、本応答関数の処理完了後、FTP クライアントとのネットワーク接続を遮断します。以下に、本応答関数で実行すべき処理を示します。

- RX-NET(FTP) に対する強制終了イベント発生の通知
- 0x0 を本応答関数の戻り値として設定

注意 RX-NET(FTP) では、本応答関数のサンプル・ソース・ファイルを提供しています。

#### 【CA850 対応版の場合】

```
nec-tools32\smp850e\rxnet\\src\callback.c
```

#### 【CCV850E 対応版の場合】

```
nec-tools32\smp850e_ghs\rxnet\\src\callback.c
```

### 戻り値

0x0 正常終了

## ftps\_dele\_user

### 概要

DELE コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_dele_user ( int session , char **msgbuf , char *filename ) ;
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
O	char <i>**msgbuf</i> ;	応答コードに続くメッセージを格納するキャラクタ・バッファへのポインタ
I	char <i>*filename</i> ;	削除するファイルのファイル名を格納した領域へのポインタ

### 機能

RX-NET(FTP) が FTP クライアントから送信された DELE コマンドを受け付けた際に呼び出される応答関数です。  
 なお、*oldname* で指定された領域には、RX-NET(FTP) が FTP クライアントから送信された RNFR コマンドを受け付けた際に獲得した変更前のディレクトリ名、または、ファイル名が格納されます。  
 以下に、本応答関数で実行すべき処理を示します。

- *filename* で指定されたファイル名を持つファイルの削除
- 応答コードに続くメッセージを *msgbuf* で指定された領域に格納
  - 250 の場合 : Requested file action okay, completed.
  - 450 の場合 : Requested file action not taken.
  - 550 の場合 : Requested action not taken.
- 応答コードを本応答関数の戻り値として設定

注意 RX-NET(FTP) では、本応答関数のサンプル・ソース・ファイルを提供しています。

#### 【CA850 対応版の場合】

```
nertools32\smp850e\rxnet\\src\callback.c
```

#### 【CCV850E 対応版の場合】

```
nertools32\smp850e_ghs\rxnet\\src\callback.c
```

### 戻り値

250	正常終了
450	異常終了 (ファイル・ビジー)
550	異常終了 (その他のエラー)

## ftps\_rmd\_user

### 概要

RMD, XRMD コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_rmd_user ( int session , char **msgbuf , char *dirname );
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
O	char <i>**msgbuf</i> ;	応答コードに続くメッセージを格納するキャラクタ・バッファへのポインタ
I	char <i>*dirname</i> ;	削除するディレクトリのディレクトリ名を格納した領域へのポインタ

### 機能

RX-NET(FTP) が FTP クライアントから送信された RMD コマンド, または, XRMD コマンドを受け付けた際に呼び出される応答関数です。

以下に, 本応答関数で実行すべき処理を示します。

- *dirname* で指定されたディレクトリ名を持つディレクトリの削除
- 応答コードに続くメッセージを *msgbuf* で指定された領域に格納
  - 250 の場合 : Requested file action okay, completed.
  - 550 の場合 : Requested action not taken.
- 応答コードを本応答関数の戻り値として設定

注意 RX-NET(FTP) では, 本応答関数のサンプル・ソース・ファイルを提供しています。

#### 【CA850 対応版の場合】

```
necools32\smp850e\rxnet\\src\callback.c
```

#### 【CCV850E 対応版の場合】

```
necools32\smp850e_ghs\rxnet\\src\callback.c
```

### 戻り値

250                    正常終了  
550                    異常終了

## ftps\_mkd\_user

### 概要

MKD , XMKD コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_mkd_user ( int session , char **msgbuf , char *dirname ) ;
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
O	char <i>**msgbuf</i> ;	応答コードに続くメッセージを格納するキャラクタ・バッファへのポインタ
I	char <i>*dirname</i> ;	生成するディレクトリのディレクトリ名を格納した領域へのポインタ

### 機能

RX-NET(FTP) が FTP クライアントから送信された MKD コマンド , または , XMKD コマンドを受け付けた際に呼び出される応答関数です。

以下に , 本応答関数で実行すべき処理を示します。

- *dirname* で指定されたディレクトリ名を持つディレクトリの生成
- 応答コードに続くメッセージを *msgbuf* で指定された領域に格納  
257 の場合 : "*dirname*" created.  
550 の場合 : Requested action not taken.
- 応答コードを本応答関数の戻り値として設定

注意 RX-NET(FTP) では , 本応答関数のサンプル・ソース・ファイルを提供しています。

#### 【CA850 対応版の場合】

nectools32\smp850e\rxnet\*<sample\_name>*\src\callback.c

#### 【CCV850E 対応版の場合】

nectools32\smp850e\_ghs\rxnet\*<sample\_name>*\src\callback.c

### 戻り値

257                    正常終了  
550                    異常終了

## ftps\_pwd\_user

### 概要

PWD, XPWD コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_pwd_user ( int session , char **msgbuf );
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
O	char <i>**msgbuf</i> ;	応答コードに続くメッセージを格納するキャラクタ・バッファへのポインタ

### 機能

RX-NET(FTP) が FTP クライアントから送信された PWD コマンド, または, XPWD コマンドを受け付けた際に呼び出される応答関数です。

以下に, 本応答関数で実行すべき処理を示します。

- PWD コマンド, または, XPWD コマンドを受け付けた時点のカレント・ディレクトリを獲得
- 応答コードに続くメッセージを *msgbuf* で指定された領域に格納  
257 の場合 : "*dirname*" created.  
550 の場合 : Requested action not taken.
- 応答コードを本応答関数の戻り値として設定

注意 RX-NET(FTP) では, 本応答関数のサンプル・ソース・ファイルを提供しています。

#### 【CA850 対応版の場合】

```
nctools32\smp850e\rxnet\<sample_name>\src\callback.c
```

#### 【CCV850E 対応版の場合】

```
nctools32\smp850e_ghs\rxnet\<sample_name>\src\callback.c
```

### 戻り値

257	正常終了
550	異常終了



## ftps\_list\_user

### 概要

LIST コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_list_user ( int session , char **msgbuf , char *dirname ) ;
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
O	char <i>**msgbuf</i> ;	応答コードに続くメッセージを格納するキャラクタ・バッファへのポインタ
I	char <i>*dirname</i> ;	ファイル・リスト情報を作成するディレクトリのディレクトリ名を格納した領域へのポインタ

### 機能

RX-NET(FTP) が FTP クライアントから送信された LIST コマンドを受け付けた際に呼び出される応答関数です。なお、RX-NET(FTP) では、本応答関数の処理完了後、応答関数 ftps\_send\_user の呼び出しを行っています。以下に、本応答関数で実行すべき処理を示します。

- *dirname* で指定されたディレクトリのファイル・リスト情報の作成

【ファイル・リスト情報のフォーマット例】

-rw-w--w--	1	root	2393	Jan	28	08:30	filelist.txt
-rw-rw-rw-	1	nobody	254423	Oct	10	1984	scaninfo.log
-rw-rw----	1	uucp	1065	Sep	15	17:15	system.bak
-rw-rw----	1	uucp	512	Sep	25	21:28	command.bat
属性		オーナー	サイズ	作成日時			ファイル名

- 応答コードに続くメッセージを *msgbuf* で指定された領域に格納  
451 の場合 : Requested action aborted : local error in processing.
- 応答コードを本応答関数の戻り値として設定

注意 1 RX-NET(FTP) では、本応答関数からの戻り値が 200 であった場合には、RX-NET(FTP) が用意しているメッセージを “ 応答コードに続くメッセージ ” として FTP クライアントに送信しています。このため、本応答関数からの戻り値が 200 となる場合には、“ 応答コードに続くメッセージを *msgbuf* で指定された領域に格納 ” を実行する必要はありません。

注意 2 RX-NET(FTP) では、本応答関数のサンプル・ソース・ファイルを提供しています。

【CA850 対応版の場合】

```
nectools32\smp850e\rxnet\<sample_name>\src\callback.c
```

【CCV850E 対応版の場合】

```
nectools32\smp850e_ghs\rxnet\<sample_name>\src\callback.c
```

## 戻り値

200	正常終了
451	異常終了

## ftps\_nlst\_user

### 概要

NLST コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_nlst_user ( int session , char **msgbuf , char *dirname ) ;
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
O	char <i>**msgbuf</i> ;	応答コードに続くメッセージを格納するキャラクタ・バッファへのポインタ
I	char <i>*dirname</i> ;	ファイル・リスト情報を作成するディレクトリのディレクトリ名を格納した領域へのポインタ

### 機能

RX-NET(FTP) が FTP クライアントから送信された NLST コマンドを受け付けた際に呼び出される応答関数です。なお、RX-NET(FTP) では、本応答関数の処理完了後、応答関数 ftps\_send\_user の呼び出しを行っています。以下に、本応答関数で実行すべき処理を示します。

- *dirname* で指定されたディレクトリのファイル・リスト情報の作成

【ファイル・リスト情報のフォーマット例】

```
filelist.txt
scaninfo.log
system.bak
command.bat

ファイル名
```

- 応答コードに続くメッセージを *msgbuf* で指定された領域に格納  
451 の場合 : Requested action aborted : local error in processing.
- 応答コードを本応答関数の戻り値として設定

注意 1 RX-NET(FTP) では、本応答関数からの戻り値が 200 であった場合には、RX-NET(FTP) が用意しているメッセージを “ 応答コードに続くメッセージ ” として FTP クライアントに送信しています。このため、本応答関数からの戻り値が 200 となる場合には、“ 応答コードに続くメッセージを *msgbuf* で指定された領域に格納 ” を実行する必要はありません。

注意 2 RX-NET(FTP) では、本応答関数のサンプル・ソース・ファイルを提供しています。

【CA850 対応版の場合】

```
nectools32\smp850e\rxnet\<sample_name>\src\callback.c
```

【CCV850E 対応版の場合】

```
nectools32\smp850e_ghs\rxnet\<sample_name>\src\callback.c
```

## 戻り値

200	正常終了
451	異常終了

## ftps\_syst\_user

### 概要

SYST コマンドを受け付けた際に呼び出される応答関数

### C 言語形式

```
int ftps_syst_user ( int session , char **msgbuf );
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
O	char <i>**msgbuf</i> ;	応答コードに続くメッセージを格納するキャラクタ・バッファへのポインタ

### 機能

RX-NET(FTP) が FTP クライアントから送信された SYST コマンドを受け付けた際に呼び出される応答関数です。以下に、本応答関数で実行すべき処理を示します。

- RFC1700 に従ったシステム・タイプ *systype* の作成
- 応答コードに続くメッセージを *msgbuf* で指定された領域に格納  
215 の場合 : *systype*  
502 の場合 : Command not implemented.
- 応答コードを本応答関数の戻り値として設定

注意 RX-NET(FTP) では、本応答関数のサンプル・ソース・ファイルを提供しています。

#### 【CA850 対応版の場合】

```
necools32\smp850e\rxnet\<sample_name>\src\callback.c
```

#### 【CCV850E 対応版の場合】

```
necools32\smp850e_ghs\rxnet\<sample_name>\src\callback.c
```

### 戻り値

215	正常終了
502	異常終了

## ftps\_recv\_user

### 概要

転送データ受信イベントが発生した際に呼び出される応答関数

### C 言語形式

```
int ftps_recv_user ( int session , char *rcvbuf , int len );
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
I	char <i>*rcvbuf</i> ;	FTP クライアントから受信した転送データを格納した受信バッファへのポインタ
I	int <i>len</i> ;	FTP クライアントから受信した転送データのサイズ (単位: バイト)

### 機能

応答関数 `ftps_stor_user` , `ftps_stou_user` , `ftps_appe_user` の処理が完了した際に呼び出される応答関数です。

なお、RX-NET(FTP) では、基本情報として定義された受信バッファのサイズを単位として、FTP クライアントから送信された転送データの受信処理を行っています。このため、*len* で指定された値が受信バッファのサイズよりも大きい場合には、本応答関数が複数回呼び出されることになります。

以下に、本応答関数で実行すべき処理を示します。

- FTP クライアントから受信した転送データをデータ表現タイプ (ASCII 表現タイプ、バイナリ表現タイプ、イメージ表現タイプ) に従って変換
- 応答関数 `ftps_stor_user` , `ftps_stou_user` , `ftps_appe_user` で指定されたファイルに FTP クライアントから受信した転送データを保存
- 受信処理結果を本応答関数の戻り値として設定

注意 1 受信バッファについての詳細は、「6.2.1 受信バッファ情報」を参照してください。

注意 2 RX-NET(FTP) では、本応答関数のサンプル・ソース・ファイルを提供しています。

【CA850 対応版の場合】

```
nectools32\smp850e\rxnet\<sample_name>\src\callback.c
```

【CCV850E 対応版の場合】

```
nectools32\smp850e_ghs\rxnet\<sample_name>\src\callback.c
```

### 戻り値

0x0	正常終了
-1	異常終了

## ftps\_send\_user

### 概要

転送データ送信イベントが発生した際に呼び出される応答関数

### C 言語形式

```
int ftps_send_user ( int session , char **sdbuf , int *len );
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号
O	char <i>*sdbuf</i> ;	FTP クライアントに送信する転送データを格納した送信バッファへのポインタ
O	int <i>*len</i> ;	FTP クライアントに送信する転送データのサイズ ( 単位: バイト ) を格納する領域へのポインタ

### 機能

応答関数 `ftps_retr_user` , `ftps_list_user` , `ftps_nlst_user` の処理が完了した際に呼び出される応答関数です。

なお、RX-NET(FTP) では、送信バッファのサイズ (1500 バイト) を単位として、FTP クライアントに対する転送データの送信処理を行っています。このため、`len` で指定された値が送信バッファのサイズよりも大きい場合には、本応答関数が複数回呼び出されることになります。

以下に、本応答関数で実行すべき処理を示します。

- FTP クライアントに送信する転送データを `sdbuf` で指定された領域に格納
- FTP クライアントに送信する転送データのサイズ ( 単位: バイト ) を `len` で指定された領域に格納
- FTP クライアントに送信する転送データをデータ表現タイプ (ASCII 表現タイプ、バイナリ表現タイプ、イメージ表現タイプ) に従って変換
- 応答関数 `ftps_retr_user` , `ftps_list_user` , `ftps_nlst_user` で作成された転送データを FTP クライアントに送信
- 0x0 を本応答関数の戻り値として設定

注意 RX-NET(FTP) では、本応答関数のサンプル・ソース・ファイルを提供しています。

#### 【CA850 対応版の場合】

```
nec-tools32\smp850e\rxnet\<sample_name>\src\callback.c
```

#### 【CCV850E 対応版の場合】

```
nec-tools32\smp850e_ghs\rxnet\<sample_name>\src\callback.c
```

### 戻り値

0x0

正常終了

## ftps\_opensession\_user

### 概要

セッション開始イベントが発生した際に呼び出される応答関数

### C 言語形式

```
int ftps_opensession_user ( int session );
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号

### 機能

RX-NET(FTP) が FTP クライアントから送信されたセッション開始イベントを受け付けた際に呼び出される応答関数です。

なお、RX-NET(FTP) では、本応答関数の処理完了後、FTP クライアントとのネットワーク接続を確立します。以下に、本応答関数で実行すべき処理を示します。

- セッション用メモリ領域の確保
- 0x0 を本応答関数の戻り値として設定

注意 RX-NET(FTP) では、本応答関数のサンプル・ソース・ファイルを提供しています。

#### 【CA850 対応版の場合】

nectools32\smp850e\rxnet\\src\callback.c

#### 【CCV850E 対応版の場合】

nectools32\smp850e\_ghs\rxnet\\src\callback.c

### 戻り値

0x0                      正常終了



**ftps\_closesession\_user**

**概要**

セッション切断イベントが発生した際に呼び出される応答関数

**C 言語形式**

```
int ftps_closesession_user ( int session );
```

**パラメータ**

I/O	パラメータ	説 明
I	int session ;	セッション番号

**機能**

RX-NET(FTP) が FTP クライアントから送信されたセッション切断イベントを受け付けた際に呼び出される応答関数です。

なお、RX-NET(FTP) では、本応答関数の処理完了後、FTP クライアントとのネットワーク接続を遮断します。以下に、本応答関数で実行すべき処理を示します。

- 応答関数 ftps\_opensession\_user において確保したセッション用メモリ領域の解放
- 0x0 を本応答関数の戻り値として設定

注意 RX-NET(FTP) では、本応答関数のサンプル・ソース・ファイルを提供しています。

【CA850 対応版の場合】  
nectools32\smp850e\rxnet\\src\callback.c

【CCV850E 対応版の場合】  
nectools32\smp850e\_ghs\rxnet\\src\callback.c

**戻り値**

0x0                      正常終了

## ftps\_timeout\_user

### 概要

タイムアウト・イベントが発生した際に呼び出される応答関数

### C 言語形式

```
int ftps_timeout_user ( int session );
```

### パラメータ

I/O	パラメータ	説明
I	int <i>session</i> ;	セッション番号

### 機能

RX-NET(FTP) がタイムアウト・イベントを発生した際に呼び出される応答関数です。  
 なお、RX-NET(FTP) では、本応答関数の処理完了後、FTP クライアントとのネットワーク接続を遮断します。  
 以下に、本応答関数で実行すべき処理を示します。

- RX-NET(FTP)、および、FTP クライアントに対するタイムアウト・イベント発生のお知らせ
- 0x0 を本応答関数の戻り値として設定

注意 1 タイムアウト時間についての詳細は、「**6.2.1 受信バッファ情報**」を参照してください。

注意 2 RX-NET(FTP) では、本応答関数のサンプル・ソース・ファイルを提供しています。

【CA850 対応版の場合】

nertools32\smp850e\rxnet\\src\callback.c

【CCV850E 対応版の場合】

nertools32\smp850e\_ghs\rxnet\\src\callback.c

### 戻り値

0x0

正常終了

# 索引

## A

API 関数 .....	28, 31, 35
ftp_abort .....	29, 31, 35, 37
ftp_start .....	28, 31, 35, 36
外部インタフェース仕様 .....	35
データ・マクロ .....	32
呼び出し方法 .....	31

## C

cf850pro .....	21
CF 定義ファイル .....	21
SCT 情報 .....	21
情報ファイル .....	21

## F

fnsconfig.h .....	31
ftp_abort .....	29, 31, 35, 37
ftp_abor_user .....	24, 30, 38, 44, 58
ftp_allo_user .....	24, 30, 38, 44, 56
ftp_appe_user .....	24, 30, 38, 44, 55
ftp_cdup_user .....	23, 30, 38, 44, 47
ftp_closesession_user .....	24, 30, 38, 44, 71
ftp_cwd_user .....	23, 30, 38, 44, 46
ftp_dele_user .....	24, 30, 38, 44, 59
ftp_list_user .....	24, 30, 38, 44, 63
ftp_login_user .....	23, 30, 38, 44, 45
ftp_logout_user .....	23, 30, 38, 44, 48
ftp_mkd_user .....	24, 30, 38, 44, 61
ftp_mode_user .....	23, 30, 38, 44, 51
ftp_nlst_user .....	24, 30, 38, 44, 65
ftp_opensession_user .....	24, 30, 38, 44, 70
ftp_pwd_user .....	24, 30, 38, 44, 62
ftp_recv_user .....	24, 30, 38, 44, 68
ftp_retr_user .....	23, 30, 38, 44, 52
ftp_rmd_user .....	24, 30, 38, 44, 60
ftp_rnto_user .....	24, 30, 38, 44, 57
ftp_send_user .....	24, 30, 38, 44, 69
ftp_start .....	28, 31, 35, 36
ftp_stor_user .....	23, 30, 38, 44, 53
ftp_stou_user .....	23, 30, 38, 44, 54
ftp_syst_user .....	24, 30, 38, 44, 67
ftp_timeout_user .....	24, 30, 38, 44, 72
ftp_type_user .....	23, 30, 38, 44, 49
FTP サーバ機能 .....	27
FTP ライブラリ .....	18, 19
libftps.a .....	18, 19

## L

libftps.a .....	18, 19
-----------------	--------

## M

Makefile .....	18
----------------	----

## R

README.FTPS .....	18, 19
RFC .....	14
RX850 Pro 依存部 .....	21
エントリ処理 .....	22
初期化ハンドラ .....	22
ハードウェア初期化部 .....	22
ブート処理 .....	22
RX-FS850 依存部 .....	23
RX-NET(FTP) .....	13
API 関数 .....	31
RX-NET(FTP) 依存部 .....	38
位置付け .....	13
インストールレーション .....	16
階層的 position 付け .....	14
開発環境 .....	15
FTP サーバ機能 .....	27
システム構築 .....	20
実行環境 .....	15
特徴 .....	14
rxnet_ftps.h .....	18, 19, 31
RX-NET(FTP) 依存部 .....	23, 38
応答関数 .....	23
基本情報 .....	23, 38
データ・マクロ .....	40
RX-NET(FTP) 用標準ヘッダ・ファイル .....	18, 19, 31
rxnet_ftps.h .....	19, 31, 18
rxnet.h .....	31
RX-NET(TCP/IP) 依存部 .....	23
RX-NET(TCP/IP) 用標準ヘッダ・ファイル .....	31
rxnet.h .....	31

## S

sample.bld .....	19
SCT 情報 .....	21
時間管理機能情報 .....	21

## あ

アーカイブ・オブジェクト ..... 25

## い

インストール ..... 16  
  UNIX ベース ..... 17  
  Windows ベース ..... 16

## え

エントリ処理 ..... 22

## お

応答関数 ..... 23, 30, 38, 44  
  ftps\_abor\_user ..... 24, 30, 38, 44, 58  
  ftps\_allo\_user ..... 24, 30, 38, 44, 56  
  ftps\_appe\_user ..... 24, 30, 38, 44, 55  
  ftps\_cdup\_user ..... 23, 30, 38, 44, 47  
  ftps\_closesession\_user ..... 24, 30, 38, 44, 71  
  ftps\_cwd\_user ..... 23, 30, 38, 44, 46  
  ftps\_dele\_user ..... 24, 30, 38, 44, 59  
  ftps\_list\_user ..... 24, 30, 38, 44, 63  
  ftps\_login\_user ..... 23, 30, 38, 44, 45  
  ftps\_logout\_user ..... 23, 30, 38, 44, 48  
  ftps\_mkd\_user ..... 24, 30, 38, 44, 61  
  ftps\_mode\_user ..... 23, 30, 38, 44, 51  
  ftps\_nlst\_user ..... 24, 30, 38, 44, 65  
  ftps\_opensession\_user ..... 24, 30, 38, 44, 70  
  ftps\_pwd\_user ..... 24, 30, 38, 44, 62  
  ftps\_recv\_user ..... 24, 30, 38, 44, 68  
  ftps\_retr\_user ..... 23, 30, 38, 44, 52  
  ftps\_rmd\_user ..... 24, 30, 38, 44, 60  
  ftps\_rnto\_user ..... 24, 30, 38, 44, 57  
  ftps\_send\_user ..... 24, 30, 38, 44, 69  
  ftps\_stor\_user ..... 23, 30, 38, 44, 53  
  ftps\_stou\_user ..... 23, 30, 38, 44, 54  
  ftps\_syst\_user ..... 24, 30, 38, 44, 67  
  ftps\_timeout\_user ..... 24, 30, 38, 44, 72  
  ftps\_type\_user ..... 23, 30, 38, 44, 49  
  外部インタフェース仕様 ..... 44  
オブジェクト・ファイル ..... 25

## か

開発環境 ..... 15  
  ソフトウェア ..... 15  
  ハードウェア ..... 15  
外部インタフェース仕様 ..... 35, 44  
拡張 SVC ハンドラ ..... 25  
拡張 SVC ハンドラ用インタフェース・ルーチン ..... 25  
間接起動割り込みハンドラ ..... 24

## き

基本情報 ..... 23, 38

受信バッファ情報 ..... 38  
タイムアウト時間情報 ..... 39  
メッセージ情報 ..... 39

## こ

コンフィギュレータ ..... 21  
  cf850pro ..... 21

## し

時間管理機能情報 ..... 21  
システム構築 ..... 20  
  CF 定義ファイル ..... 21  
  RX850 Pro 依存部 ..... 21  
  RX-FS850 依存部 ..... 23  
  RX-NET(FTP) 依存部 ..... 23  
  RX-NET(TCP/IP) 依存部 ..... 23  
  アーカイブ・オブジェクト ..... 25  
  オブジェクト・ファイル ..... 25  
  処理プログラム ..... 24  
  リンク・ディレクティブ・ファイル ..... 26  
  ロード・モジュール ..... 26  
システム・コール・テーブル ..... 21  
システム情報テーブル ..... 21  
システム情報ヘッダ・ファイル ..... 21  
実行環境 ..... 15  
  周辺コントローラ ..... 15  
  プロセッサ ..... 15  
  メモリ容量 ..... 15  
周期起動ハンドラ ..... 25  
受信バッファ情報 ..... 38  
  FTPS\_SIZE ..... 38  
情報ファイル ..... 21  
  システム・コール・テーブル ..... 21  
  システム情報テーブル ..... 21  
  システム情報ヘッダ・ファイル ..... 21  
初期化ハンドラ ..... 22  
処理プログラム ..... 24  
  拡張 SVC ハンドラ ..... 25  
  拡張 SVC ハンドラ用インタフェース・ルーチン ..... 25  
  間接起動割り込みハンドラ ..... 24  
  周期起動ハンドラ ..... 25  
  タスク ..... 24  
  直接起動割り込みハンドラ ..... 24

## す

垂直フォーマット制御種別 ..... 40

## せ

静的設定情報ヘッダ・ファイル ..... 31  
  fnconfig.h ..... 31

## た

タイムアウト時間情報 .....	39
FTPS_TIMEOUT .....	39
タスク .....	24

## ち

直接起動割り込みハンドラ .....	24
--------------------	----

## て

ディレクトリ構成 .....	18
CA850 対応版 .....	18
CCV850E 対応版 .....	19
データ表現タイプ .....	40
データ・マクロ .....	32, 40
垂直フォーマット制御種別 .....	40
データ表現タイプ .....	40
転送モード .....	41
戻り値 .....	32, 40
テキスト・ファイル .....	18, 19
README.FTPS .....	18, 19
転送モード .....	41

## は

ハードウェア初期化部 .....	22
------------------	----

## ひ

ビルド・ファイル .....	19
sample.bid .....	19

## ふ

ブート処理 .....	22
-------------	----

## め

メイク・ファイル .....	18
Makefile .....	18
メッセージ情報 .....	39

## も

戻り値 .....	32, 40
-----------	--------

## り

リンク・ディレクティブ・ファイル .....	26
------------------------	----

## ろ

ロード・モジュール .....	26
-----------------	----

---

## — お問い合わせ先 —

### 【技術的なお問い合わせ先】

NEC半導体テクニカルホットライン  
(電話：午前 9:00～12:00，午後 1:00～5:00)

電話 : 044-435-9494  
FAX : 044-435-9608  
E-mail : [info@lsi.nec.co.jp](mailto:info@lsi.nec.co.jp)

### 【営業関係お問い合わせ先】

システムLSI第一営業事業部  
東京 (03)3798-6106, 6107, 6108, 6155  
大阪 (06)6945-3178, 3200, 3208  
名古屋 (052)222-2375  
仙台 (022)267-8740  
水戸 (029)226-1702  
広島 (082)242-5504  
鳥取 (0857)27-5313  
松山 (089)945-4149

システムLSI第二営業事業部  
東京 (03)3798-6110, 6111, 6112, 6151, 6156  
名古屋 (052)222-2170, 2190  
松本 (0263)35-1662  
前橋 (027)243-6060  
立川 (042)526-5981  
静岡 (054)254-4794  
金沢 (076)232-7303  
福岡 (092)261-2806

### 【資料の請求先】

上記営業関係お問い合わせ先またはNEC特約店へお申しつけください。

### 【NECエレクトロニクス デバイス ホームページ】

NECエレクトロニクスデバイスの情報がインターネットでご覧になれます。

URL(アドレス) <http://www.ic.nec.co.jp/>

## アンケート記入のお願い

お手数ですが、このドキュメントに対するご意見をお寄せください。今後のドキュメント作成の参考にさせていただきます。

[ドキュメント名] RX-NET Ver.1.10 ネットワーク・ライブラリ ユーザーズ・マニュアル FTP  
( U15946JJ2V0UM00 ( 第2版 ) )

[お名前など] ( さしつかえのない範囲で )

御社名 ( 学校名, その他 ) ( )  
ご住所 ( )  
お電話番号 ( )  
お仕事の内容 ( )  
お名前 ( )

1. ご評価 ( 各欄に をご記入ください )

項 目	大変良い	良 い	普 通	悪 い	大変悪い
全体の構成					
説明内容					
用語解説					
調べやすさ					
デザイン, 字の大きさなど					
その他 ( )					
( )					

2. わかりやすい所 ( 第 章, 第 章, 第 章, 第 章, その他 )

理由 [ ]

3. わかりにくい所 ( 第 章, 第 章, 第 章, 第 章, その他 )

理由 [ ]

4. ご意見, ご要望

5. このドキュメントをお届けしたのは

NEC販売員, 特約店販売員, その他 ( )

ご協力ありがとうございました。

下記あてにFAXで送信いただくか, 最寄りの販売員にコピーをお渡ししてください。

日本電気(株) NEC エレクトロニクス  
半導体テクニカルホットライン

FAX : ( 044 ) 435-9608

2000.6