

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、誤りが無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.1.00

ネットワーク・ライブラリ

WebServer

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 Webサーバ機能 ...	28
第5章 API関数 ...	34
第6章 RX-NET (WebServer) 依存部 ...	41
索引 ...	77

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
Webサーバ機能
API関数
RX-NET (WebServer) 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 TM , V850/SA1 TM , V850/SB1 TM , V850/SB2 TM , V850/SF1 TM , V850/SV1 TM 用インサートキット・エミュレータ)	U11595J	U11595E	
IE-V850E-MC (V850E/IA1 TM , V850E/IA2 TM 用インサートキット・エミュレータ), IE-V850E-MC-A (V850E/MA1 TM , V850E/MA2 TM 用インサートキット・エミュレータ)	U14487J	U14487E	
IE-703003-MC-EM1 (V853用インサートキット・エミュレータ・オプション・ボード)	U11596J	U11596E	
IE-703017-MC-EM1 (V850/SA1用インサートキット・エミュレータ・オプション・ボード)	U12898J	U12898E	
IE-703037-MC-EM1 (V850/SB1, V850/SB2用インサートキット・エミュレータ・オプション・ボード)	U14151J	U14151E	
IE-703040-MC-EM1 (V850/SV1用インサートキット・エミュレータ・オプション・ボード)	U14337J	U14337E	
IE-703079-MC-EM1 (V850/SF1用インサートキット・エミュレータ・オプション・ボード)	U15447J	U15447E	
IE-703102-MC (V850E/MS1 TM 用インサートキット・エミュレータ)	U13875J	U13875E	
IE-703102-MC-EM1, IE-703102-MC-EM1-A (V850E/MS1用インサートキット・エミュレータ・オプション・ボード)	U13876J	U13876E	
IE-703107-MC-EM1 (V850E/MA1用インサートキット・エミュレータ・オプション・ボード)	U14481J	U14481E	
IE-703116-MC-EM1 (V850E/IA1用インサートキット・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	U15024J	U15024E
	C言語編	U15025J	U15025E
	プロジェクト・マネージャ編	U15026J	U15026E
	アセンブリ言語編	U15027J	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	U13430J	U13430E
	インストレーション編	U13410J	U13410E
	テクニカル編	U13431J	U13431E
RX850 Pro Ver.3.13 リアルタイムOS	基礎編	U13773J	U13773E
	インストレーション編	U13774J	U13774E
	テクニカル編	U13772J	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 概要枠	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	22
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(WebServer) 依存部の記述	23
3.8 処理プログラムの記述	25
3.9 オブジェクト・ファイルの生成	26
3.10 アーカイブ・オブジェクトの生成	26
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	27
第 4 章 Web サーバ機能	28
4.1 概要	28
4.2 処理の流れ	28
4.2.1 リクエスト・メッセージ, レスポンス・メッセージ	29
4.3 Web サーバ機能 API 関数	31
4.3.1 RX-NET(WebServer) の初期化	31
4.3.2 RX-NET(WebServer) の強制終了	32
4.4 Web サーバ機能応答関数	33
第 5 章 API 関数	34
5.1 概要	34
5.2 API 関数の呼び出し	34
5.3 データ・マクロ	35
5.3.1 戻り値	35
5.4 API 関数解説	36
5.4.1 外部インタフェース仕様	38

第 6 章	RX-NET(WebServer) 依存部	41
6.1	概要	41
6.2	基本情報	44
6.2.1	サーバ情報	44
6.2.2	ネットワーク情報	44
6.2.3	ディレクトリ情報	45
6.2.4	受信バッファ情報	45
6.2.5	リクエスト・メッセージ情報	46
6.2.6	タイムアウト時間情報	46
6.2.7	リクエスト・タスク情報	47
6.2.8	エラー・メッセージ情報	48
6.3	データ・マクロ	49
6.3.1	タイプ・コード	49
6.3.2	リクエスト・メタ変数	49
6.4	応答関数解説	51
6.4.1	外部インタフェース仕様	53
索引	77

目次

図 1-1	RX-NET(WebServer) の位置付け	13
図 1-2	RX-NET(WebServer) 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	リクエスト要求の対応手順	28
図 4-2	リクエスト・メッセージの形式	29
図 4-3	レスポンス・メッセージの形式	29
図 6-1	各種応答関数の呼び出しタイミング・イメージ (起動～リクエスト対応)	42
図 6-2	各種応答関数の呼び出しタイミング・イメージ (終了)	43

表目次

表 2-1	RX-NET(WebServer) の提供形式	16
表 3-1	応答関数	24
表 4-1	応答関数	33
表 5-1	戻り値	35
表 5-2	API 関数	38
表 6-1	タイプ・コード	49
表 6-2	リクエスト・メタ変数	49
表 6-3	応答関数	53

第1章 概説

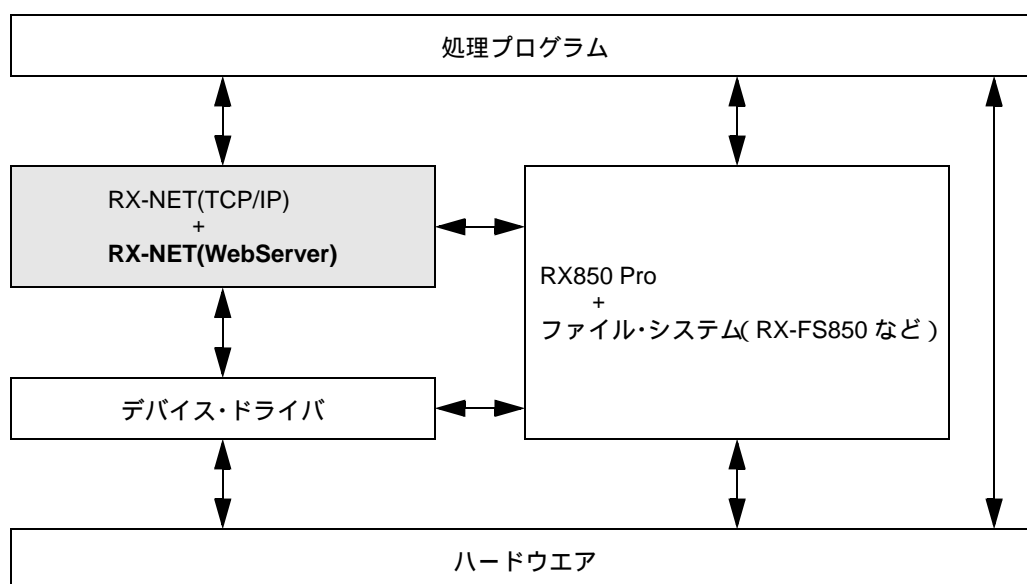
1.1 概要枠

RX-NET(WebServer) 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, HTTP(Hyper Text Transfer Protocol) を利用したサーバ, クライアント間のハイパー・テキスト転送を行うための Web サーバ機能を提供しています。

したがって, ユーザは, RX-NET(WebServer) が提供する Web サーバ機能を利用することにより, HTTP を利用したサーバ, クライアント間のハイパー・テキスト転送を実現することが可能となります。

図 1-1 に, RX-NET(WebServer) の位置付けを示します。

図 1-1 RX-NET(WebServer) の位置付け



1.2 特徴

以下に、RX-NET(WebServer) の特徴を示します。

- RFC に準拠

RX-NET(WebServer) では、RFC に準拠した設計が行われています。

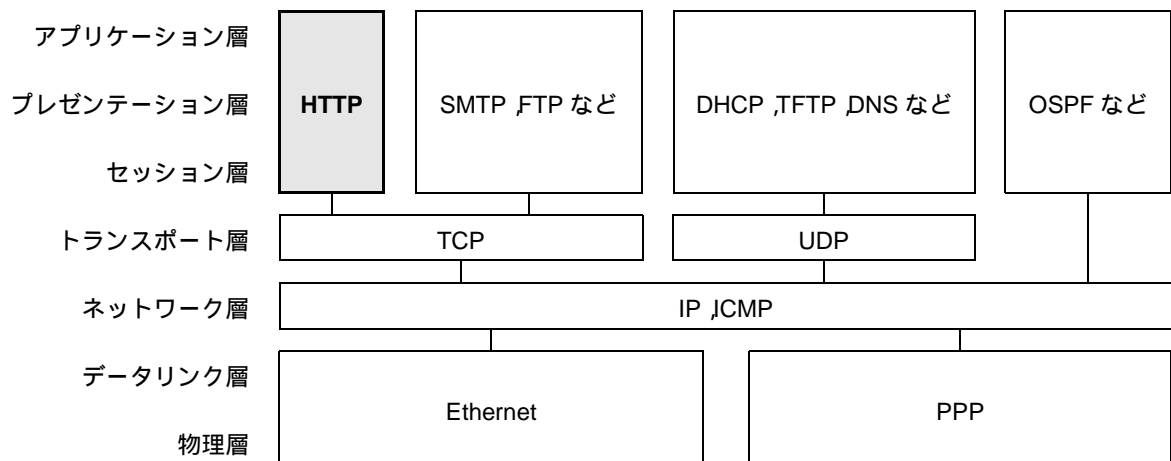
なお、RFC とは、インターネットに関する研究開発機関 IETF (Internet Engineering Task Force) が取りまとめた公開技術文書であり、電子メール、ファイル転送などのプロトコル仕様（情報交換を行う際に必要な手順、および、規約）の他にも、各種サービス、ガイドラインといった多岐に渡った情報（ネットワーク技術の実装と運用に主眼を置いた情報）が記載されています。

- Web サーバ機能をサポート

RX-NET(WebServer) では、ネットワーク上で利用される Web サーバ機能を提供しています。したがって、RX-NET(WebServer) を組み込んだ機器は、サーバ、クライアント間のハイパー・テキスト転送が可能となります。

図 1-2 に、RX-NET(WebServer) の階層的位置付けを示します。

図 1-2 RX-NET(WebServer) の階層的位置付け



- 高い移植性

RX-NET(WebServer) 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(WebServer) 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(WebServer) 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(WebServer) 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(WebServer) 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(WebServer) 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(WebServer) 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(WebServer) のテキスト領域：約 10 K バイト
 - RX-NET(WebServer) のデータ領域：約 2 K バイト

1.4 開発環境

以下に、RX-NET(WebServer) 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/PC09821 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 Softwareoyobi, Inc. 製

第2章 インストール

本章では、RX-NET(WebServer)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(WebServer)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(WebServer)の提供形式一覧を示します。

表 2-1 RX-NET(WebServer)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(WebServer)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(WebServer)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(WebServer)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(WebServer)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(WebServer)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(WebServer)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(WebServer)の提供媒体のディレクトリRX-NET_WEB_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(WebServer)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(WebServer)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(WebServer)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/rxnetweb.tar を、米国 Green Hills Software, Inc. 製 CCV850E の場合は圧縮ファイル ghs/rxnetweb.tar を展開します。

【 CA850 対応版の場合 】

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

【 CCV850E 対応版の場合 】

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

- 6) ファイル群の確認
ls コマンドを実行し、RX-NET(WebServer)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

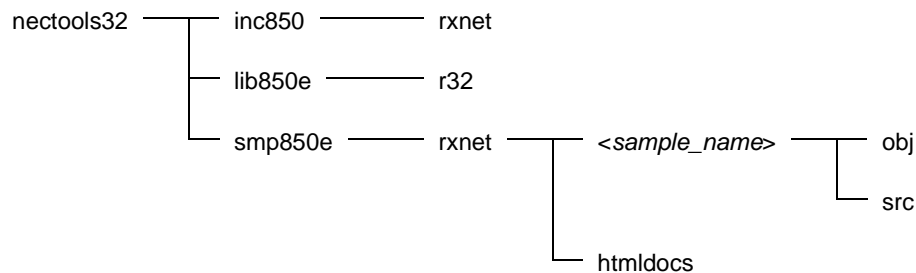
RX-NET(WebServer) の提供媒体に格納されているファイル群のディレクトリ構成は、ユーザが使用する C コンパイラ・パッケージの種類 (NEC 製 CA850, 米国 Green Hills Software, Inc. 製 CCV850E) により異なります。

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(WebServer) 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(WebServer) 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_webs.h : RX-NET(WebServer) 用標準ヘッダ・ファイル
- 2) nectools32\inc850\rxnet
RX-NET(WebServer) 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
HTTP 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。
libwebs.a : HTTP 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。
- 6) nectools32\smp850e\rxnet\htmldocs
サンプル・プログラムで用いられるハイパー・テキスト形式ファイルが格納されているディレクトリです。

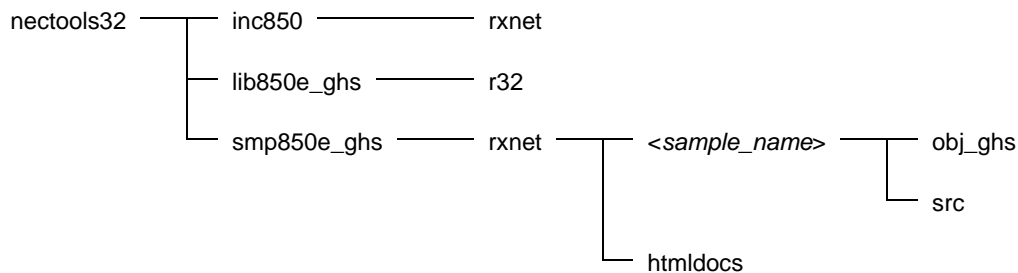
注意 *<sample_name>* についての詳細は、下記に示したテキスト・ファイルを参照してください。

<sample_name> : nectools32\smp850e\rxnet\README.WEBS

2.3.2 CCV850E 対応版

図 2-2 に、RX-NET(WebServer) 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(WebServer) 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_webs.h : RX-NET(WebServer) 用標準ヘッダ・ファイル
- 2) nectools32\inc850\rxnet
RX-NET(WebServer) 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
HTTP 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。
libwebs.a : HTTP 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。
- 6) nectools32\smp850e_ghs\rxnet\htmldocs
サンプル・プログラムで用いられるハイパー・テキスト形式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_ghs\rxnet\README.WEBS

第3章 システム構築

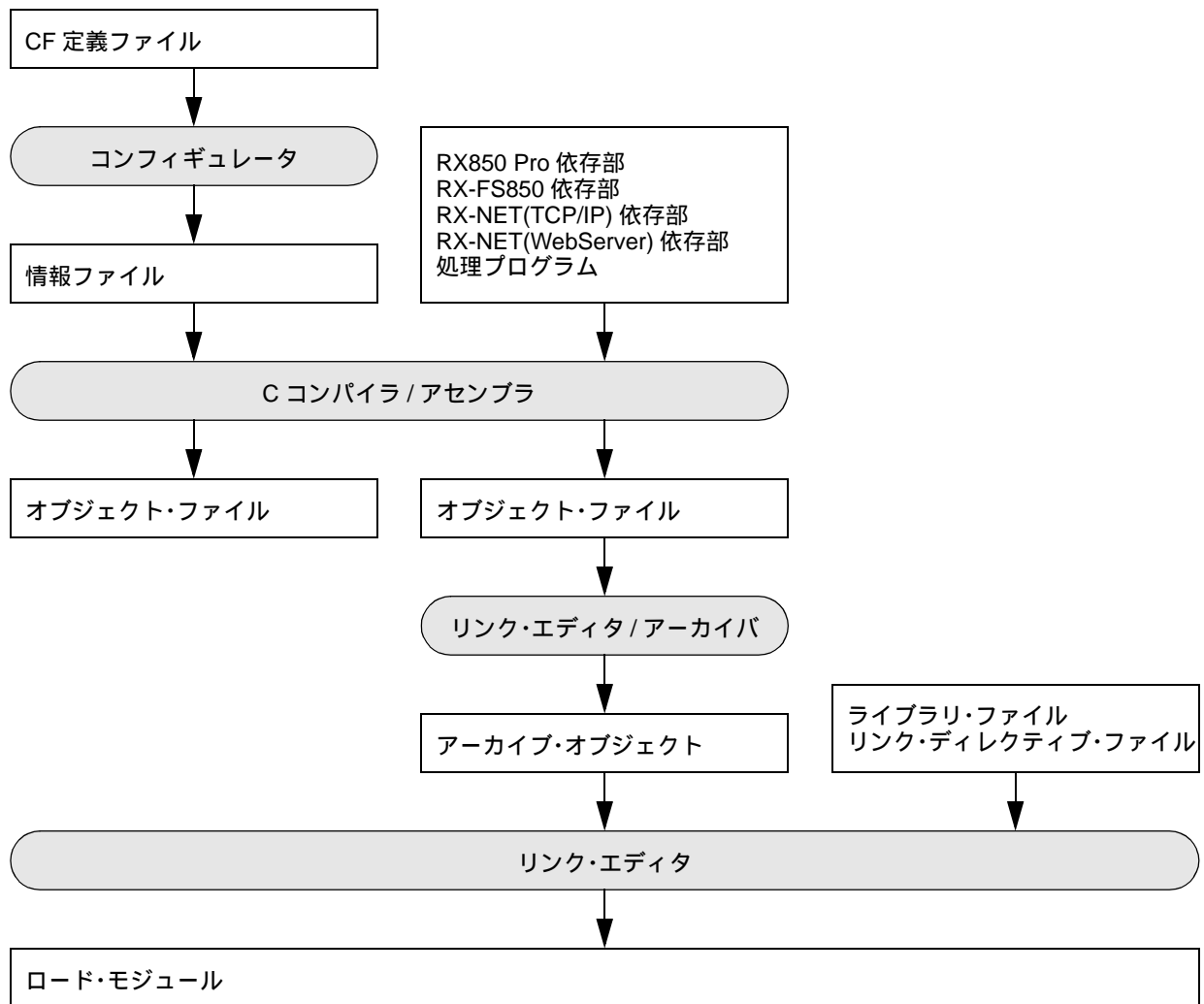
本章では、RX-NET(WebServer)を使用したネットワーク・アプリケーション(ロード・モジュール)の構築手順を解説しています。

3.1 概要

システム構築とは、RX-NET(WebServer)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(WebServer)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(WebServer) では COUNT+1 個のタスク, 10 種類のシステム・コールを, RX-NET(WebServer) 依存部では 1 個のメモリ・プール利用して各種機能を実現しています。

• SIT 情報

- システム情報

RX-NET(WebServer) 用に “タスクの自動割り付け ID 番号” として COUNT + 1 個分を確保。

RX-NET(WebServer) 依存部用に “メモリ・プールの自動割り付け ID 番号” として 1 個分を確保。

注意 COUNT は, API 関数 webs_start を発行する際に指定する最大セッション数 cnt と同値になります。

- システム最大値情報

RX-NET(WebServer) 用に “タスクの最大生成数” として COUNT + 1 個分を確保。

RX-NET(WebServer) 依存部用に “メモリ・プールの最大生成数” として 1 個分を確保。

注意 COUNT は, API 関数 webs_start を発行する際に指定する最大セッション数 cnt と同値になります。

- メモリ・プール情報

RX-NET(WebServer) 用に “システム・メモリ SPOL1 のサイズ” として STK_SIZE × COUNT バイト分を確保。

RX-NET(WebServer) 依存部用に “システム・メモリ UPOL0 のサイズ” として MPL_SIZE バイト分を確保。

注意 1 STK_SIZE は, 基本情報として定義するリクエスト・タスク情報(スタック・サイズ webs_stksize)と同値になります。

注意 2 MPL_SIZE は, 応答関数 webs_callback_prolog においてシステム・コール cre_mpl を発行する際に指定するプール・サイズ mplsz と同値になります。

• SCT 情報

- タスク管理機能情報

RX-NET(WebServer) 用に “cre_tsk, sta_tsk, exd_tsk, chg_pri, rot_rdq” を定義。

- タスク付属同期機能情報

RX-NET(WebServer) 用に “tslp_tsk” を定義。

- メモリ・プール管理機能情報

RX-NET(WebServer) 依存部用に “cre_mpl, del_mpl, get_blk, rel_blk” 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(WebServer) 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\\src
sys.cf          : CF 定義ファイル
```

【CCV850E 対応版の場合】

```
nectools32\smp850e_ghs\rxnet\\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 インストラクション編」を参照してください。

注意 2 RX-NET(WebServer) 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\\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\\obj_ghs
 sample.bld : ロード・モジュール用ビルド・ファイル

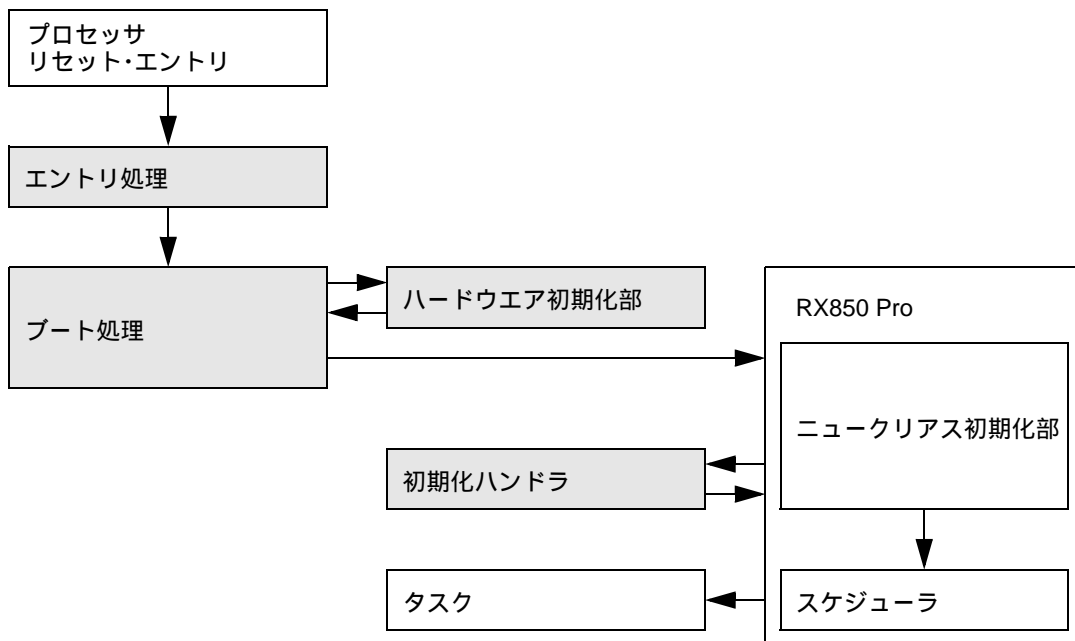
3.4 RX850 Pro 依存部の記述

RX-NET(WebServer) 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、ニユ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(WebServer) 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(WebServer) 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(WebServer) 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 1 RX-NET(WebServer) では、RX-NET(TCP/IP) の初期化時に確保されたヒープ領域から下記計算式 (単位: バイト) で算出されたサイズの領域を RX-NET(WebServer) 用ヒープ領域として使用しています。

このため、ヒープ領域のサイズを定義する際には、RX-NET(WebServer) 用ヒープ領域のサイズを加算する必要があります。

$$\text{WEBS_HEAP} = \{ (4720 + \text{BUFFER}) \times \text{COUNT} \} + 0x8$$

BUFFER : RX-NET(WebServer) 依存部の基本情報としてマクロ定義するバッファ・サイズ WEBS_SIZE と同値。

COUNT : API 関数 webs_start を発行する際に指定する最大セッション数 cnt と同値。

注意 2 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(WebServer) 依存部の記述

RX-NET(WebServer) では、RX-NET(WebServer) 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(WebServer) 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(WebServer) 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(WebServer) を正常に動作させるうえで必要となる RX-NET(WebServer) 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(WebServer) 依存部の一覧を示します。

• 基本情報

RX-NET(WebServer) では、RX-NET(WebServer) が提供する機能を実現する際に必要となる基本情報 (サーバ情報、ネットワーク情報、ディレクトリ情報など) 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(WebServer) 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。

- サーバ情報
- ネットワーク情報
- ディレクトリ情報
- 受信バッファ情報
- リクエスト・メッセージ情報
- タイムアウト時間情報
- リクエスト・タスク情報
- エラー・メッセージ情報

• 応答関数

RX-NET(WebServer) では、RX-NET(WebServer) の初期化処理 / 強制終了処理を実行した際、または、Web クライアントからのリクエストを受け付け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
webs_callback_prolog	webs_start を発行した際に呼び出される応答関数
webs_callback_epilog	webs_abort を発行した際に呼び出される応答関数
webs_task_start	Web クライアントからのリクエストを受け付けた際に呼び出される応答関数
webs_task_end	Web クライアントからのリクエストに対応した処理が完了した際に呼び出される応答関数
webs_fopen	ファイルをオープンする際に呼び出される応答関数
webs_fclose	ファイルのクローズする際に呼び出される応答関数
webs_fread	データを読み込む際に呼び出される応答関数
webs_fwrite	データを書き込む際に呼び出される応答関数
webs_tempfile	一時ファイルを生成 / オープンする際に呼び出される応答関数
webs_tempfile_del	一時ファイルをクローズ / 削除する際に呼び出される応答関数
webs_getsize	ファイル (ハイパー・テキスト形式ファイル、JPEG ファイルなど) のサイズを獲得する際に呼び出される応答関数
webs_rewind	ファイル・ポインタの位置を変更する際に呼び出される応答関数
webs_check_filetype	ディレクトリ、または、ファイルのタイプを獲得する際に呼び出される応答関数
webs_get_cgifunc	Web クライアントからのリクエスト URI が CGI ファイルの際に呼び出される応答関数
webs_get_authname	webs_check_filetype からの戻り値が WEBS_FILETYPE_DIR_AUTH の際に呼び出される応答関数
webs_auth	ユーザ認証を必要とするディレクトリに対するアクセス要求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
webs_get_directoryindex	WebクライアントからのリクエストURIがディレクトリの際に呼び出される応答関数
webs_log_user	Webクライアントからのアクセス・ログを記録する際に呼び出される応答関数
webs_cgifunc	Webクライアントからの呼び出し要求を受け付けた際に呼び出されるCGI関数

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

注意 2 RX-NET(WebServer) では、RX-NET(WebServer) 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

websconfig.c : 基本情報

callback.c : 応答関数

【CCV850E 対応版の場合】

nectools32\smp850e_ghs\rxnet\\src

websconfig.c : 基本情報

callback.c : 応答関数

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、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) を行わせる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(WebServer) 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、リロケート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(WebServer) 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ト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(WebServer) 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】

```
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
```

3.11 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ーズ・マニュアルを参照してください。
- 注意 2 RX-NET(WebServer) 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(WebServer) 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ーズ・マニュアルを参照してください。

注意 2 RX-NET(WebServer) 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 Web サーバ機能

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

4.1 概要

RX-NET(WebServer) では、RX-NET(WebServer) の初期化処理、強制終了処理の他に、Web クライアントから送信されたリクエスト要求を受け付けた際に呼び出す応答関数を “ Web サーバ機能 ” として提供しています。

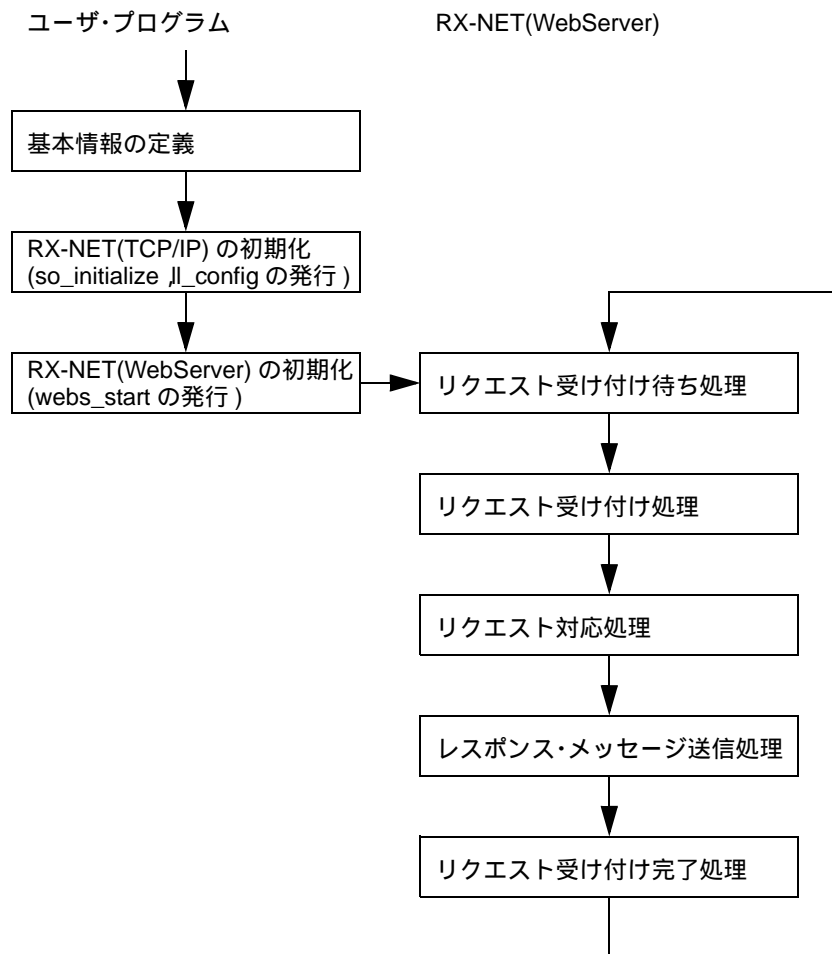
なお、RX-NET(WebServer) では、サーバ、クライアント間の最大同時接続数として、20 セッションのリモート・アクセスを可能としています。

4.2 処理の流れ

RX-NET(WebServer) では、Web クライアントから送信されたリクエスト要求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(WebServer) が提供している Web サーバ機能を利用したリクエスト要求の対応手順を示します。

図 4-1 リクエスト要求の対応手順



4.2.1 リクエスト・メッセージ，レスポンス・メッセージ

RX-NET(WebServer)では，Webクライアントから送信されたリクエスト・メッセージを受信した際，該当する処理を実行した後，レスポンス・メッセージをWebクライアントに対して送信しています。

図4-2，図4-3に，リクエスト・メッセージ，レスポンス・メッセージの形式を示します。

図4-2 リクエスト・メッセージの形式

HTTP メソッド	リクエスト URI	HTTP バージョン
リクエスト・メッセージ (ヘッダ部)		
リクエスト・メッセージ (ボディ部)		

図4-3 レスポンス・メッセージの形式

HTTP バージョン	ステータス・コード	リズン・フレーズ
レスポンス・メッセージ (ヘッダ部)		
レスポンス・メッセージ (ボディ部)		

• HTTP メソッド

RX-NET(WebServer)が処理対象としているHTTPメソッドを以下に示します。

- GET

WebクライアントからWebサーバに対して，レスポンス・メッセージ(ボディ部を含む)の送信要求を行います。なお，レスポンス・メッセージには，WebサーバがリクエストURIから識別した情報が格納されます。

- HEAD

WebクライアントからWebサーバに対して，レスポンス・メッセージ(ボディ部を含まない)の送信要求を行います。

なお，レスポンス・メッセージには，WebサーバがリクエストURIから識別した情報が格納されます。

- POST

WebクライアントからWebサーバに対して，データの送信を行います。

なお，WebクライアントからWebサーバに対して送信されるデータは，リクエスト・メッセージ(ボディ部)に格納されています。

• リクエスト・メッセージ (ヘッダ部)

RX-NET(WebServer)が処理対象としているリクエスト・メッセージ(ヘッダ部)を以下に示します。

- Connection

WebクライアントがKeep-Alive機能(サーバ，クライアント間の持続的ネットワーク接続)を使用することが可能であるか否に関する情報が格納されています。

- Transfer-Encoding

サーバ，クライアント間の転送エンコーディング形式が格納されます。

なお、RX-NET(WebServer) では、転送エンコーディング形式として“チャンク形式”のみを処理対象としています。

- Authorization
認証処理が必要となった際に利用されるユーザ名、および、パスワードが格納されます。
 - Host
Web クライアントがリクエスト要求発行対象としてのマシン名が格納されます。
なお、RX-NET(WebServer) では、本ヘッダ部に設定されている値と基本情報 webs_hostname_list で定義された値を比較し、RX-NET(WebServer) に対するリクエスト要求であるか否かを判断しています。
注意 基本情報 webs_hostname_list についての詳細は、「**6.2.2 ネットワーク情報**」を参照してください。
 - Content-Length
リクエスト・メッセージ (ボディ部) のサイズが格納されます。
- レスポンス・メッセージ (ヘッダ部)
RX-NET(WebServer) が Web クライアントに対して送信するレスポンス・メッセージ (ヘッダ部) を以下に示します。
 - Connection
RX-NET(WebServer) が Keep-Alive 機能 (サーバ、クライアント間の持続的ネットワーク接続) の使用が可能である旨の情報が格納されています。
なお、RX-NET(WebServer) では、Web クライアントからのリクエスト要求 (Connection ヘッダ部) で Keep-Alive 機能の使用が可能である旨の情報が格納されていた場合に限り、本ヘッダ部の送信処理を実行しています。
 - Location
リダイレクト先 URL が格納されます。
 - Server
RX-NET(WebServer) のサーバ情報が格納されます。
なお、RX-NET(WebServer) では、サーバ情報として基本情報 webs_serverinfo で定義された値を本ヘッダ部に設定しています。
注意 基本情報 webs_serverinfo についての詳細は、「**6.2.1 サーバ情報**」を参照してください。
 - WWW-Authenticate
認証方法、および、認証箇所名が格納されます。
なお、RX-NET(WebServer) では、認証方法として“Basic”を、認証箇所名として応答関数 webs_get_authname を実行することにより得られた値を本ヘッダ部に設定しています。
 - Content-Length
レスポンス・メッセージ (ボディ部) のサイズが格納されます。
なお、RX-NET(WebServer) では、レスポンス・メッセージ (ボディ部) のサイズとして応答関数 webs_getsize を実行することにより得られた値を本ヘッダ部に設定しています。
 - Content-Type
レスポンス・メッセージ (ボディ部) のメディア・タイプが格納されます。
なお、RX-NET(WebServer) では、レスポンス・メッセージ (ボディ部) のメディア・タイプとして応答関数 webs_fopen を実行することにより得られた値を本ヘッダ部に設定しています。
 - Allow
RX-NET(WebServer) が処理対象としている HTTP メソッド (GET, HEAD, POST) が格納されます。

4.3 Web サーバ機能 API 関数

4.3.1 RX-NET(WebServer) の初期化

RX-NET(WebServer) 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。

- webs_start

パラメータ *cnt*, *port* で指定された情報をもとに、RX-NET(WebServer) 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、Web クライアントからのリクエスト受け付け待ち状態となります。

なお、RX-NET(WebServer) では、RX-NET(WebServer) の初期化処理として、RX-NET(WebServer) 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、メモリ領域の確保、および、応答関数 *webs_callback_prolog* の呼び出しを行っています。

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 <fnconfig.h> /* 静的設定情報ヘッダ・ファイルの定義 */
#include <rxnet_webs.h> /* RX-NET(WebServer) 用標準ヘッダ・ファイルの定義 */

/*****
基本情報の定義
*****/
#define WEBS_SERVER_INFO "RX-NET(WebServer)850v1.00"
const char webs_serverinfo [ ] = WEBS_SERVER_INFO ;
#define WEBS_HOSTNAME "www"
#define WEBS_FQDN "www.sample1.nec.co.jp"
#define WEBS_IPADDR "192.168.0.1"
char webs_hostname_list [ 0x4 ] [ 0x80 ] = {
    { WEBS_HOSTNAME },
    { WEBS_FQDN },
    { WEBS_IPADDR },
    { 0x0 }
};
#define WEBS_ERROR_DIR "/ErrMsg/"
const char webs_path_errorfile [ ] = WEBS_ERROR_DIR ;
#define WEBS_SIZE 0x1000
int webs_bufsize = WEBS_SIZE ;
#define WEBS_MAX_CONTENTLENGTH 0x1000
int webs_max_contentlength = WEBS_MAX_CONTENTLENGTH ;
#define WEBS_TIMEOUT 0x1000
int webs_idle_timeout = WEBS_TIMEOUT ;
#define WEBS_TASK_IPRI 0x5
unsigned int webs_task_ipri = WEBS_TASK_IPRI ;
#define WEBS_TASK_EPRI 0x1
unsigned int webs_task_epri = WEBS_TASK_EPRI ;
#define WEBS_STKSZ 0x500
unsigned int webs_stksize = WEBS_STKSZ ;

void
func_task ( INT stacd ) {
    char devname [ ] = "s91s"; /* 変数の宣言，初期化 */
    u32 ipaddress = inet_addr ( WEBS_IPADDR ); /* 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 = 20 ; /* 変数の宣言，初期化 */
    unsigned short port = 80 ; /* 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 );
webs_start ( cnt , port );                       /* RX-NET(WebServer) 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（サーバ情報、ネットワーク情報、ディレクトリ情報など）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 `webs_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、80 番を設定します。なお、パラメータ `port` に `0x0` を指定した場合、RX-NET(WebServer) は“制御用 TCP コネクション・ポート番号”のデフォルト値 80 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、および、応答関数 `webs_callback_prolog` についての詳細は、「第 6 章 RX-NET(WebServer) 依存部」を参照してください。

4.3.2 RX-NET(WebServer) の強制終了

RX-NET(WebServer) 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。

- `webs_abort`

RX-NET(WebServer) の終了処理を強制的に実行します。

なお、RX-NET(WebServer) では、RX-NET(WebServer) の終了処理として、RX-NET(WebServer) 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、および、応答関数 `webs_callback_epilog` の呼び出しを行っています。

これにより、すべての処理プログラムから RX-NET(WebServer) 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 `webs_start` の戻り値として `EWEB_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_webs.h>                         /* RX-NET(WebServer) 用標準ヘッダ・ファイルの定義 */

void
func_task ( INT stacd ) {
    webs_abort ( );                             /* RX-NET(WebServer) 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 `webs_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(WebServer) 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。
- 注意 3 応答関数 `webs_callback_epilog` についての詳細は、「第 6 章 RX-NET(WebServer) 依存部」を参照してください。

4.4 Webサーバ機能応答関数

RX-NET(WebServer) では、RX-NET(WebServer) の初期化処理 / 強制終了処理を実行した際、または、Web クライアントからのリクエストを受け付けた際、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、RX-NET(WebServer) が提供している応答関数一覧を示します。

表 4-1 応答関数

応答関数名	機能概要
webs_callback_prolog	webs_start を発行した際に呼び出される応答関数
webs_callback_epilog	webs_abort を発行した際に呼び出される応答関数
webs_task_start	Web クライアントからのリクエストを受け付けた際に呼び出される応答関数
webs_task_end	Web クライアントからのリクエストに対応した処理が完了した際に呼び出される応答関数
webs_fopen	ファイルをオープンする際に呼び出される応答関数
webs_fclose	ファイルのクローズする際に呼び出される応答関数
webs_fread	データを読み込む際に呼び出される応答関数
webs_fwrite	データを書き込む際に呼び出される応答関数
webs_tempfile	一時ファイルを生成 / オープンする際に呼び出される応答関数
webs_tempfile_del	一時ファイルをクローズ / 削除する際に呼び出される応答関数
webs_getsize	ファイル (ハイパー・テキスト形式ファイル, JPEG ファイルなど) のサイズを獲得する際に呼び出される応答関数
webs_rewind	ファイル・ポインタの位置を変更する際に呼び出される応答関数
webs_check_filetype	ディレクトリ, または, ファイルのタイプを獲得する際に呼び出される応答関数
webs_get_cgifunc	Web クライアントからのリクエスト URI が CGI ファイルの際に呼び出される応答関数
webs_get_authname	webs_check_filetype からの戻り値が WEBS_FILETYPE_DIR_AUTH の際に呼び出される応答関数
webs_auth	ユーザ認証を必要とするディレクトリに対するアクセス要求を受け付けた際に呼び出される応答関数
webs_get_directoryindex	Web クライアントからのリクエスト URI がディレクトリの際に呼び出される応答関数
webs_log_user	Web クライアントからのアクセス・ログを記録する際に呼び出される応答関数
webs_cgifunc	Web クライアントからの呼び出し要求を受け付けた際に呼び出される CGI 関数

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

第 5 章 API 関数

本章では、RX-NET(WebServer) 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(WebServer) 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(WebServer) 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(WebServer) が提供している API 関数 (2 種類) を示します。

webs_start webs_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(WebServer) 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(WebServer) 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(WebServer) 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_webs.h : RX-NET(WebServer) 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(WebServer) 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。

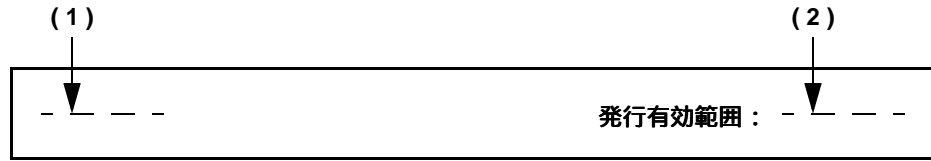
なお、戻り値のマクロ定義は、標準ヘッダ・ファイル `nctools32\inc850\rxnet_webs.h` から呼び出されるヘッダ・ファイル `nctools32\inc850\rxnet\weberno.h` で行われています。

表 5-1 戻り値

マクロ	数値	意味
EWEB_NOMEM	0x1500	メモリ領域が確保できません
EWEB_INVALID	0x1501	パラメータの指定が不正です
EWEB_SOCKET	0x1502	ソケットの生成処理が失敗しました
EWEB_BIND	0x1503	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
EWEB_DOWN	0x1504	RX-NET(TCP/IP) の初期化処理が完了していません
EWEB_NOINIT	0x1505	<code>webs_callback_prolog</code> の処理が失敗しました
EWEB_ABORT	0x1506	<code>webs_abort</code> の発行により <code>webs_start</code> の処理が強制終了しました
EWEB_RUNNING	0x1507	既に <code>webs_start</code> が発行されています
EWEB_TASK	0x1508	タスクの生成処理、または、起動処理が失敗しました

5.4 API 関数解説

次項から RX-NET(WebServer) 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(WebServer) への入力パラメータ
- O ... RX-NET(WebServer) 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(WebServer) 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
webs_start	RX-NET(WebServer) の初期化
webs_abort	RX-NET(WebServer) 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。

webs_start

発行有効範囲：タスク

概要

RX-NET(WebServer) の初期化

C 言語形式

```
int webs_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 ~ 20)
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(WebServer) 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 , Web クライアントからのリクエスト受け付け待ち状態となります。

なお , RX-NET(WebServer) では , RX-NET(WebServer) の初期化処理として , RX-NET(WebServer) 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 , メモリ領域の確保 , および , 応答関数 `webs_callback_prolog` の呼び出しを行っています。

- 注意 1 本 API 関数の発行は , 基本情報 (サーバ情報 , ネットワーク情報 , ディレクトリ情報など) 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 `webs_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 , 80 番を設定します。なお , *port* に 0x0 を指定した場合 , RX-NET(WebServer) は “ 制御用 TCP コネクション・ポート番号 ” のデフォルト値 80 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 , および , 応答関数 `webs_callback_prolog` についての詳細は , 「第 6 章 RX-NET(WebServer) 依存部」を参照してください。

戻り値

EWEB_NOMEM	0x1500	データ管理ブロック用メモリ領域が確保できません
EWEB_INVALID	0x1501	パラメータの指定が不正です
EWEB_SOCKET	0x1502	ソケットの生成処理が失敗しました
EWEB_BIND	0x1503	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
EWEB_DOWN	0x1504	RX-NET(TCP/IP) の初期化処理が完了していません
EWEB_NOINIT	0x1505	<code>webs_callback_prolog</code> の処理が失敗しました
EWEB_ABORT	0x1506	<code>webs_abort</code> の発行により本 API 関数の処理が強制終了しました
EWEB_RUNNING	0x1507	既に本 API 関数が発行されています
EWEB_TASK	0x1508	タスクの生成処理 , または , 起動処理が失敗しました

webs_abort

発行有効範囲：タスク / 非タスク

概要

RX-NET(WebServer) の強制終了

C 言語形式

```
void webs_abort ( void );
```

パラメータ

なし

機能

RX-NET(WebServer) の終了処理を強制的に実行します。

なお、RX-NET(WebServer) では、RX-NET(WebServer) の終了処理として、RX-NET(WebServer) 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、および、応答関数 `webs_callback_epilog` の呼び出しを行っています。

これにより、すべての処理プログラムから RX-NET(WebServer) 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 `webs_start` の戻り値として `EWEB_ABORT` が返されます。

- 注意 1 本 API 関数の発行は、API 関数 `webs_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(WebServer) 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。
- 注意 3 応答関数 `webs_callback_epilog` についての詳細は、「第 6 章 RX-NET(WebServer) 依存部」を参照してください。

戻り値

なし

第 6 章 RX-NET(WebServer) 依存部

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

6.1 概要

RX-NET(WebServer) では、RX-NET(WebServer) 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(WebServer) 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(WebServer) が RX-NET(WebServer) 依存部として提供している応答関数 (19 種類) を示します。

<code>webs_callback_prolog</code>	<code>webs_callback_epilog</code>	<code>webs_task_start</code>	<code>webs_task_end</code>
<code>webs_fopen</code>	<code>webs_fclose</code>	<code>webs_fread</code>	<code>webs_fwrite</code>
<code>webs_tempfile</code>	<code>webs_tempfile_del</code>	<code>webs_getsize</code>	<code>webs_rewind</code>
<code>webs_check_filetype</code>	<code>webs_get_cgifunc</code>	<code>webs_get_authname</code>	<code>webs_auth</code>
<code>webs_get_directoryindex</code>	<code>webs_log_user</code>	<code>webs_cgifunc</code>	

なお、応答関数の呼び出しは、RX-NET(WebServer) の初期化処理 / 強制終了処理を実行した際、または、Web クライアントからのリクエストを受け付けた際、RX-NET(WebServer) 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(WebServer) が行っています。

図 6-1, および, 図 6-2 に, 各種応答関数の呼び出しタイミング・イメージを示します。

図 6-1 各種応答関数の呼び出しタイミング・イメージ (起動~リクエスト対応)

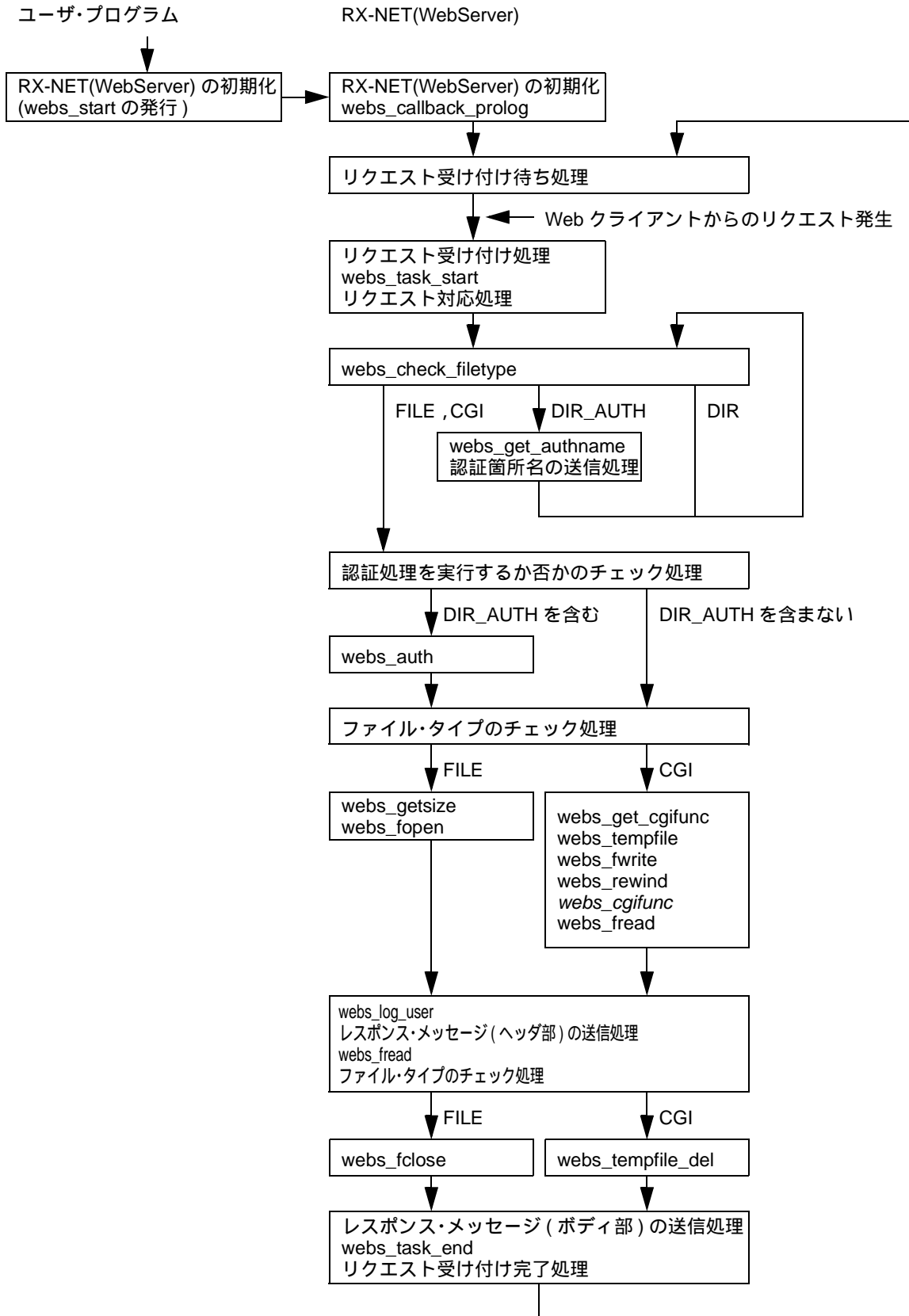
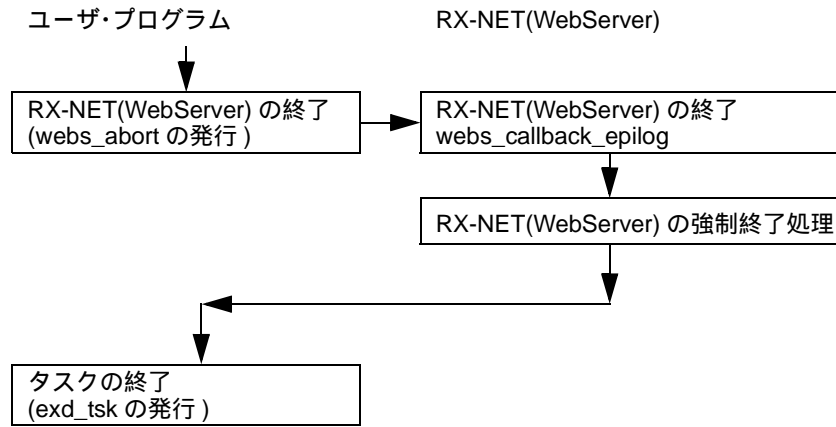


図 6-2 各種応答関数の呼び出しタイミング・イメージ (終了)



6.2 基本情報

RX-NET(WebServer) が提供する機能を実現する際に必要となる基本情報 (サーバ情報, ネットワーク情報, ディレクトリ情報など) について以下に示します。

6.2.1 サーバ情報

RX-NET(WebServer) では, Web クライアントに対して送信するレスポンス・メッセージの Server ヘッダ部を RX-NET(WebServer) 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, 変数名を用いてサーバ情報を定義する必要があります。以下に, サーバ情報の記述例を示します。

```
/* サーバ情報のマクロ定義 */
#define WEBS_SERVER_INFO "RX-NET(WebServer)850v1.00"

/* サーバ情報の宣言, 初期化 */
const char webs_serverinfo [ ] = WEBS_SERVER_INFO ;
```

注意 RX-NET(WebServer) では, 本サーバ情報のサンプル・ソース・ファイルを提供しています。

【 CA850 対応版の場合 】

nectools32\smp850e\rxnet*<sample_name>*\src\websconfig.c

【 CCV850E 対応版の場合 】

nectools32\smp850e_ghs\rxnet*<sample_name>*\src\websconfig.c

6.2.2 ネットワーク情報

RX-NET(WebServer) では, RX-NET(WebServer) がネットワーク上で動作する際に必要となる情報 (マシン名, FQDN, IP アドレスなど) を RX-NET(WebServer) 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, 変数名を用いてネットワーク情報を定義する必要があります。以下に, ネットワーク情報の記述例を示します。

```
/* マシン名のマクロ定義 */
#define WEBS_HOSTNAME "www"

/* FQDN(Fully Qualified Domain Name) のマクロ定義 */
#define WEBS_FQDN "www.sample1.nec.co.jp"

/* IP アドレスのマクロ定義 */
#define WEBS_IPADDR "192.168.0.1"

/* ネットワーク情報の宣言, 初期化 */
char webs_hostname_list [ 0x4 ] [ 0x80 ] = {
    { WEBS_HOSTNAME },
    { WEBS_FQDN },
    { WEBS_IPADDR },
    { 0x0 }
};
```

注意 RX-NET(WebServer) では, 本ネットワーク情報のサンプル・ソース・ファイルを提供しています。

【 CA850 対応版の場合 】

nectools32\smp850e\rxnet*<sample_name>*\src\websconfig.c

【 CCV850E 対応版の場合 】

nectools32\smp850e_ghs\rxnet*<sample_name>*\src\websconfig.c

6.2.3 ディレクトリ情報

RX-NET(WebServer) では、ステータス・コード 200 以外のレスポンスが発生した場合、予め定義されたディレクトリに格納されているエラー・メッセージ用ハイパー・テキスト形式ファイル(ファイル名: ステータス・コード + .html) の記述内容をレスポンス・メッセージのボディ部に設定したのち、Web クライアントに対して送信しています。

このため、ユーザは、規定されたマクロ名、および、変数名を用いてディレクトリ情報 (エラー・メッセージ用ハイパー・テキスト形式ファイルが格納されているディレクトリ名) を定義する必要があります。

なお、ディレクトリ名として指定可能な文字数は、0x1 ~ 0x1ff に限られます。

以下に、ディレクトリ情報の記述例を示します。

```
/* ディレクトリ名のマクロ定義 */
#define          WEBS_ERROR_DIR          "/ErrMsg/"

/* ディレクトリ情報の宣言, 初期化 */
const   char    webs_path_errorfile [] = WEBS_ERROR_DIR ;
```

注意 RX-NET(WebServer) では、本ディレクトリ情報のサンプル・ソース・ファイルを提供しています。

【 CA850 対応版の場合 】

nectools32\smp850e\rxnet\\src\websconfig.c

【 CCV850E 対応版の場合 】

nectools32\smp850e_ghs\rxnet\\src\websconfig.c

6.2.4 受信バッファ情報

RX-NET(WebServer) では、Web クライアントから送信されたリクエスト・メッセージ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(WebServer) 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、0x400 ~ 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          WEBS_SIZE          0x1000

/* 受信バッファ情報の宣言, 初期化 */
int             webs_bufsize = WEBS_SIZE ;
```

注意 RX-NET(WebServer) 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 】

nectools32\smp850e\rxnet\\src\websconfig.c

【 CCV850E 対応版の場合 】

nectools32\smp850e_ghs\rxnet\\src\websconfig.c

6.2.5 リクエスト・メッセージ情報

RX-NET(WebServer) では、Web クライアントから送信されたリクエスト・メッセージの最大受信可能サイズを RX-NET(WebServer) 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、変数名を用いてリクエスト・メッセージ情報を定義する必要があります。

なお、リクエスト・メッセージの最大受信可能サイズ(単位:バイト)として指定可能な値は、0x1 ~ 0x7ffffff に限られます。以下に、リクエスト・メッセージ情報の記述例を示します。

```
/* 最大受信可能サイズのマクロ定義 */
#define WEBS_MAX_CONTENTLENGTH 0x1000

/* リクエスト・メッセージ情報の宣言, 初期化 */
int webs_max_contentlength = WEBS_MAX_CONTENTLENGTH;
```

注意 RX-NET(WebServer) 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 】

nertools32\smp850e\rxnet\\src\websconfig.c

【 CCV850E 対応版の場合 】

nertools32\smp850e_ghs\rxnet\\src\websconfig.c

6.2.6 タイムアウト時間情報

RX-NET(WebServer) では、RX-NET(WebServer) が Web クライアントに対してレスポンス・メッセージを送信してからタイムアウト・イベント (Web クライアントとのネットワーク接続の遮断処理、および、リクエスト・タスクの終了処理) を実行するまでの時限を RX-NET(WebServer) 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、0x0 ~ 0xffffffff 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 WEBS_TIMEOUT 0x1000

/* タイムアウト時間情報の宣言, 初期化 */
int webs_idle_timeout = WEBS_TIMEOUT;
```

注意 RX-NET(WebServer) 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 】

nertools32\smp850e\rxnet\\src\websconfig.c

【 CCV850E 対応版の場合 】

nertools32\smp850e_ghs\rxnet\\src\websconfig.c

6.2.7 リクエスト・タスク情報

RX-NET(WebServer) では、RX-NET(WebServer) が提供する機能を実現する際に必要となるシステム・タスクに関する情報 (リクエスト・タスクの起動時優先度、終了時優先度、スタック・サイズなど) を RX-NET(WebServer) 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、変数名を用いてリクエスト・タスク情報を定義する必要があります。

なお、リクエスト・タスクの起動時優先度として指定可能な値は “API 関数 `webs_start` を発行するタスクよりも低い優先度 ”、終了時優先度として指定可能な値は “API 関数 `webs_start` を発行するタスクよりも高い優先度 ”、スタック・サイズとして指定可能な値は `0x0 ~ 0x7ffffffc` の 4 バイト境界値に限られます。

以下に、リクエスト・タスク情報の記述例を示します。

```

/* 起動時優先度のマクロ定義 */
#define WEBS_TASK_IPRI 0x5

/* 起動時優先度の宣言, 初期化 */
unsigned int webs_task_ipri = WEBS_TASK_IPRI;

/* 終了時優先度のマクロ定義 */
#define WEBS_TASK_EPRI 0x1

/* 終了時優先度の宣言, 初期化 */
unsigned int webs_task_epri = WEBS_TASK_EPRI;

/* スタック・サイズのマクロ定義 */
#define WEBS_STKSZ 0x500

/* スタック・サイズの宣言, 初期化 */
unsigned int webs_stksize = WEBS_STKSZ;

```

注意 RX-NET(WebServer) では、本リクエスト・タスク情報のサンプル・ソース・ファイルを提供しています。

【 CA850 対応版の場合 】

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

【 CCV850E 対応版の場合 】

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

6.2.8 エラー・メッセージ情報

RX-NET(WebServer) では、ステータス・コード 200 以外のレスポンスが発生した場合、予め定義されたディレクトリに格納されているエラー・メッセージ用ハイパー・テキスト形式ファイルの記述内容をレスポンス・メッセージのボディ部に設定したのち、Web クライアントに対して送信しています。

このため、ユーザは、規定された記述形式でステータス・コードに対応したエラー・メッセージ用ハイパー・テキスト形式ファイル(ファイル名:ステータス・コード + .html)を作成するする必要があります。

なお、作成したエラー・メッセージ用ハイパー・テキスト形式ファイルは、「6.2.3 ディレクトリ情報」で指定されたディレクトリに格納しておきます。

以下に、エラー・メッセージ用ハイパー・テキスト形式ファイルの記述例を示します。

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML>
<HEAD>
  <TITLE>301 Moved Permanently</TITLE>
</HEAD>
<BODY>
  <H1>Moved Permanently</H1>
</BODY>
</HTML>
```

注意 RX-NET(WebServer) では、本ディレクトリ情報のサンプル・ソース・ファイルを提供しています。

【 CA850 対応版の場合 】

nectools32\smp850e\rxnet\htmldocs\htdocs\errmsg\xxx.html

【 CCV850E 対応版の場合 】

nectools32\smp850e_ghs\rxnet\htmldocs\htdocs\errmsg\xxx.html

6.3 データ・マクロ

RX-NET(WebServer) 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 (タイプ・コード , リクエスト・メタ変数など) について以下に示します。

6.3.1 タイプ・コード

表 6-1 に , 応答関数 `webs_check_filetype` の処理結果を RX-NET(WebServer) に通知するために使用するタイプ・コード一覧を示します。

なお , タイプ・コードのマクロ定義は , 標準ヘッダ・ファイル `necools32\inc850\rxnet_webs.h` から呼び出されるヘッダ・ファイル `necools32\inc850\rxnet\webscallback.h` で行われています。

表 6-1 タイプ・コード

マクロ	数値	意味
WEBS_FILETYPE_DIR	0x1	認証処理が不要なディレクトリ
WEBS_FILETYPE_DIR_AUTH	0x2	認証処理が必要なディレクトリ
WEBS_FILETYPE_CGI	0x3	CGI ファイル
WEBS_FILETYPE_FILE	0x4	ハイパー・テキスト形式ファイル , JPEG ファイルなど
WEBS_FILETYPE_NOTEXIST	0x5	対象ディレクトリ , または , 対象ファイルが存在しない
WEBS_FILETYPE_FORBIDDEN	0x6	対象ディレクトリ , または , 対象ファイルに対するアクセス権がない

6.3.2 リクエスト・メタ変数

表 6-2 に , CGI 関数 `webs_cgifunc` の第 2 パラメータ `envf` で指定されたリクエスト・メタ変数格納ファイルに設定されるリクエスト・メタ変数一覧を示します。

表 6-2 リクエスト・メタ変数

リクエスト・メタ変数	意味
AUTH_TYPE	認証方法
CONTENT_TYPE	リクエスト・メッセージ (Content-Type ヘッダ部)
CONTENT_LENGTH	リクエスト・メッセージ (ボディ部) のサイズ
GATEWAY_INTERFACE	サポート対象 CGI のバージョン
PATH_INFO	Web クライアントから送信されたエキストラ・パス情報
QUERY_STRING	リクエスト URI の記号 “?” 以降の情報
REMOTE_ADDR	Web クライアントの IP アドレス
REMOTE_USER	Web クライアントからリクエスト要求を送信したユーザ名
REQUEST_METHOD	リクエストされたメソッド
SCRIPT_NAME	CGI プログラムのパス情報
SERVER_SOFTWARE	Web サーバのサーバ情報
SERVER_NAME	Webサーバのネットワーク情報(マシン名 , FQDN , IPアドレスのいずれか)
SERVER_PROTOCOL	Web サーバがサポートしているネットワーク・プロトコル , および , ネットワーク・プロトコルのバージョン情報
SERVER_PORT	Web サーバのポート番号
HTTP_ACCEPT	リクエスト・メッセージ (Accept ヘッダ部)

リクエスト・メタ変数	意味
HTTP_HOST	リクエスト・メッセージ (Host ヘッダ部)
HTTP_REFERER	リクエスト・メッセージ (Refer ヘッダ部)
HTTP_USER_AGENT	リクエスト・メッセージ (User-Agent 部)

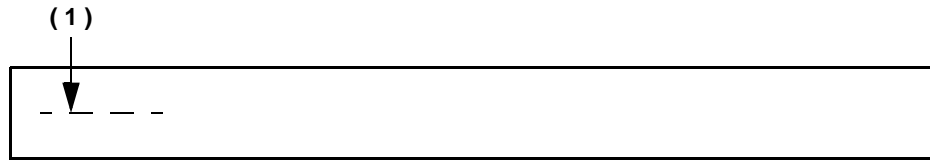
注意 RX-NET(WebServer) では、表 6-2 に示されたリクエスト・メタ変数で対応することができないリクエスト要求を受け付けた際には、以下の操作を実行して作成したリクエスト・メタ変数をリクエスト・メタ変数格納ファイルに設定します。

- ヘッダ名を英大文字に変換
- ハイフン“-”をアンダー・バーに変換
- プレフィックスとして“HTTP_”を付加

したがって、RX-NET(WebServer) が Web クライアントからの CGI プログラムに関するリクエスト要求として Content-Encoding ヘッダを受け付けた際には、HTTP_CONTENT_ENCODING がリクエスト・メタ変数格納ファイルに設定されます。

6.4 応答関数解説

次項から RX-NET(WebServer) 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(WebServer) 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 6-3 に、RX-NET(WebServer) が提供している応答関数の一覧を示します。

表 6-3 応答関数

応答関数名	機能概要
webs_callback_prolog	webs_start を発行した際に呼び出される応答関数
webs_callback_epilog	webs_abort を発行した際に呼び出される応答関数
webs_task_start	Web クライアントからのリクエストを受け付けた際に呼び出される応答関数
webs_task_end	Web クライアントからのリクエストに対応した処理が完了した際に呼び出される応答関数
webs_fopen	ファイルをオープンする際に呼び出される応答関数
webs_fclose	ファイルのクローズする際に呼び出される応答関数
webs_fread	データを読み込む際に呼び出される応答関数
webs_fwrite	データを書き込む際に呼び出される応答関数
webs_tempfile	一時ファイルを生成 / オープンする際に呼び出される応答関数
webs_tempfile_del	一時ファイルをクローズ / 削除する際に呼び出される応答関数
webs_getsize	ファイル (ハイパー・テキスト形式ファイル, JPEG ファイルなど) のサイズを獲得する際に呼び出される応答関数
webs_rewind	ファイル・ポインタの位置を変更する際に呼び出される応答関数
webs_check_filetype	ディレクトリ, または, ファイルのタイプを獲得する際に呼び出される応答関数
webs_get_cgifunc	Web クライアントからのリクエスト URI が CGI ファイルの際に呼び出される応答関数
webs_get_authname	webs_check_filetype からの戻り値が WEBS_FILETYPE_DIR_AUTH の際に呼び出される応答関数
webs_auth	ユーザ認証を必要とするディレクトリに対するアクセス要求を受け付けた際に呼び出される応答関数
webs_get_directoryindex	Web クライアントからのリクエスト URI がディレクトリの際に呼び出される応答関数
webs_log_user	Web クライアントからのアクセス・ログを記録する際に呼び出される応答関数
webs_cgifunc	Web クライアントからの呼び出し要求を受け付けた際に呼び出される CGI 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。

webs_callback_prolog

概要

webs_start を発行した際に呼び出される応答関数

C 言語形式

```
int webs_callback_prolog ( void );
```

パラメータ

なし

機能

RX-NET(WebServer) が提供している API 関数 webs_start を発行し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。

- ユーザ独自の初期化処理 (ファイル・システムの初期化など)

注意 1 本応答関数の呼び出しタイミングは、API 関数 webs_start の処理を実行した直後となります。

注意 2 RX-NET(WebServer) 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 - 本応答関数の処理終了後、API 関数 webs_start を発行したタスクは、Web クライアントからのリクエスト受け付け待ち状態となります。
-1	異常終了

webs_callback_epilog

概要

webs_abort を発行した際に呼び出される応答関数

C 言語形式

```
int webs_callback_epilog ( void );
```

パラメータ

なし

機能

RX-NET(WebServer) が提供している API 関数 webs_abort を発行し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。

- ユーザ独自の終了処理 (ファイル・システムの終了など)

注意 1 本応答関数の呼び出しタイミングは、API 関数 webs_abort の処理を実行する直前となります。

注意 2 RX-NET(WebServer) 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

webs_task_start

概要

Web クライアントからのリクエストを受け付けた際に呼び出される応答関数

C 言語形式

```
int webs_task_start ( void );
```

パラメータ

なし

機能

RX-NET(WebServer) が Web クライアントからのリクエスト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。

- ユーザ独自の初期化処理 (ファイル・システム用タスクの登録など)

注意 1 本応答関数の呼び出しタイミングは、RX-NET(WebServer) が Web クライアントからのリクエスト受け付け処理 (Web クライアントとのネットワーク接続の確立処理、リクエスト・タスクの起動処理など) を実行した直後となります。

なお、リクエスト・タスクは、RX-NET(WebServer) が提供しているシステム・タスクです。

注意 2 RX-NET(WebServer) 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

- 本応答関数の処理終了後、RX-NET(WebServer) は Web クライアントからのリクエスト対応処理を実行します。

-1

異常終了

- 本応答関数の処理終了後、RX-NET(WebServer) は Web クライアントとのネットワーク接続の遮断処理、および、リクエスト・タスクの終了処理を実行します。

webs_task_end

概要

Web クライアントからのリクエストに対応した処理が完了した際に呼び出される応答関数

C 言語形式

```
int webs_task_end ( void );
```

パラメータ

なし

機能

RX-NET(WebServer) が Web クライアントからのリクエストに対応した処理を完了し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。

- ユーザ独自の終了処理 (ファイル・システム用タスクの登録解除など)

注意 1 本応答関数の呼び出しタイミングは、RX-NET(WebServer) が Web クライアントからのリクエスト受け付け完了処理 (Web クライアントとのネットワーク接続の遮断処理、リクエスト・タスクの終了処理など) を実行する直前となります。

なお、リクエスト・タスクは、RX-NET(WebServer) が提供しているシステム・タスクです。

注意 2 RX-NET(WebServer) 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

0x0

正常終了

- 本応答関数の処理終了後、RX-NET(WebServer) は Web クライアントとのネットワーク接続の遮断処理、および、リクエスト・タスクの終了処理を実行します。

webs_fopen

概要

ファイルをオープンする際に呼び出される応答関数

C 言語形式

```
int webs_fopen ( char *filename , char *flags , char *mtype );
```

パラメータ

I/O	パラメータ	説明
I	char *filename ;	オープン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
I	char *flags ;	オープンするファイルの属性を格納した領域へのポインタ r : 読み込み専用
O	char *mtype ;	オープンしたファイルのメディア・タイプを格納する領域へのポインタ text/html : ハイパー・テキスト形式ファイル image/jpeg : JPEG ファイル 上記以外のメディア・タイプについては、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 RFC を参照。

機能

filename で指定されたファイルをオープンする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。

- *filename* で指定されたファイルを *flags* で指定された属性 (読み込み専用 r) でオープン
- オープンしたファイルのメディア・タイプを *mtype* で指定された領域に格納
- オープンしたファイルのファイル記述子を本応答関数の戻り値として設定

注意 1 ファイル名 *filename* には、Web クライアントから受信したリクエスト URI(パス部) が設定されます。
したがって、リクエスト URI が “ http://www.sample1.nec.co.jp/DirName1/DirName2/FileName1 ” の場合、
filename には、“ /DirName1/DirName2/FileName1 ” が設定されます。

注意 2 属性 *flags* には、読み込み専用 r が設定されます。

注意 3 RX-NET(WebServer) では、*mtype* で指定された領域に格納するメディア・タイプの最大サイズを 127 バイトに規定しています。

注意 4 RX-NET(WebServer) 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

NULL 以外

正常終了

- 戻り値として “ オープンしたファイルのファイル記述子 ” が返されます。
- 本応答関数の処理終了後 ,RX-NET(WebServer) はデータの読み込み処理を実行しません。

NULL

異常終了

- 本応答関数の処理終了後 ,RX-NET(WebServer) は Web クライアントからのリクエスト受け付け完了処理 (Web クライアントに対するステータス・コード 500 のレスポンス送信処理 ,Web クライアントからのリクエストに対応した全ファイルのクローズ処理 ,Web クライアントとのネットワーク接続の遮断処理 ,リクエスト・タスクの終了処理など) を実行します。

webs_fclose

概要

ファイルをクローズする際に呼び出される応答関数

C 言語形式

```
int webs_fclose ( int fildes );
```

パラメータ

I/O	パラメータ	説明
I	int <i>fildes</i> ;	クローズするファイルのファイル記述子

機能

fildes で指定されたファイルをクローズする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。

- *fildes* で指定されたファイルのクローズ

注意 1 ファイル記述子 *fildes* には、応答関数 `webs_fopen` の戻り値が設定されます。

注意 2 RX-NET(WebServer) 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
-1	異常終了

webs_fread

概要

データを読み込む際に呼び出される応答関数

C 言語形式

```
int webs_fread ( int fildest , char *ptr , int size );
```

パラメータ

I/O	パラメータ	説明
I	int <i>fildest</i> ;	データを読み込むファイルのファイル記述子
O	char * <i>ptr</i> ;	読み込んだデータを格納する領域へのポインタ
I	int <i>size</i> ;	読み込むデータのサイズ (単位 : バイト)

機能

fildest で指定されたファイルからデータを読み込む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。

- *fildest* で指定されたファイルのファイル・ポインタの位置から *size* で指定されたサイズのデータ、または、EOF までのデータを読み込む
- 読み込んだデータを *ptr* で指定された領域に格納
- 読み込んだデータのサイズを本応答関数の戻り値として設定

注意 1 ファイル記述子 *fildest* には、応答関数 `webs_fopen` の戻り値が設定されます。

注意 2 読み込むデータのサイズ *size* には、RX-NET(WebServer) が提供している送信バッファのサイズが設定されません。

注意 3 RX-NET(WebServer) 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

0x0 , -1 以外	正常終了 <ul style="list-style-type: none"> - 戻り値として “ 読み込んだデータのサイズ (単位 : バイト) ” が返されます。 - 本応答関数の処理終了後、RX-NET(WebServer) は本応答関数の再呼び出し処理を実行します。
0x0	正常終了 <ul style="list-style-type: none"> - 本戻り値は、“ 該当ファイルからのデータの読み込み処理が完了 ” を意味しています。 - 本応答関数の処理終了後、RX-NET(WebServer) は <i>fildest</i> で指定されたファイルのクローズ処理を実行します。
-1	異常終了

- 本応答関数の処理終了後，RX-NET(WebServer) は Web クライアントからのリクエスト受け付け完了処理(Webクライアントからのリクエストに対応した全ファイルのクローズ処理，Web クライアントとのネットワーク接続の遮断処理，リクエスト・タスクの終了処理など) を実行します。

webs_fwrite

概要

データを書き込む際に呼び出される応答関数

C 言語形式

```
int webs_fwrite ( int fildes , char *ptr , int size ) ;
```

パラメータ

I/O	パラメータ	説明
I	int <i>fildes</i> ;	データを書き込む一時ファイルのファイル記述子
I	char * <i>ptr</i> ;	書き込むデータを格納した受信バッファへのポインタ
I	int <i>size</i> ;	書き込むデータのサイズ (単位 : バイト)

機能

fildes で指定された一時ファイルに *ptr* で指定されたデータを書き込む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。

- *fildes* で指定された一時ファイルのファイル・ポインタの位置から *size* で指定されたサイズのデータ、または、EOF までのデータを書き込む

注意 1 ファイル記述子 *fildes* には、応答関数 `webs_tempfile` の戻り値が設定されます。

注意 2 書き込むデータのサイズ *size* には、基本情報として定義した受信バッファのサイズ `WEBS_SIZE` が設定されます。

注意 3 RX-NET(WebServer) 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

- 本応答関数の処理終了後、RX-NET(WebServer) は本応答関数の再呼び出し処理を実行します。
ただし、該当ファイルへのデータの書き込み処理が完了した際には、RX-NET(WebServer) は *fildes* で指定された一時ファイルのファイル・ポインタの位置変更処理 (応答関数 `webs_rewind` の呼び出し) を実行します。

-1

異常終了

- 本応答関数の処理終了後、RX-NET(WebServer) は Web クライアントからのリクエスト受け付け完了処理 (Web クライアントに対するステータス・コード 500 のレスポンス送信処理、Web クライアントからのリクエストに対応した全ファイルのクローズ処理、Web クライアントとのネットワーク接続の遮断処理、リクエスト・タスクの終了処理など) を実行します。

webs_tempfile

概要

一時ファイルを生成 / オープンする際に呼び出される応答関数

C 言語形式

```
int webs_tempfile ( void );
```

パラメータ

なし

機能

一時ファイルを生成 / オープンする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。

- 一時ファイルの生成
- 生成した一時ファイルを読み込み / 書き込み両用 rw でオープン
- オープンした一時ファイルのファイル記述子を本応答関数の戻り値として設定

注意 RX-NET(WebServer) 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

NULL 以外

正常終了

- 戻り値として “ 生成 / オープンした一時ファイルのファイル記述子 ” が返されます。

NULL

異常終了

- 本応答関数の処理終了後、RX-NET(WebServer) は Web クライアントからのリクエスト受け付け完了処理 (Web クライアントに対するステータス・コード 500 のレスポンス送信処理、Web クライアントからのリクエストに対応した全ファイルのクローズ処理、Web クライアントとのネットワーク接続の遮断処理、リクエスト・タスクの終了処理など) を実行します。

webs_tempfile_del

概要

一時ファイルをクローズ / 削除する際に呼び出される応答関数

C 言語形式

```
int webs_tempfile_del ( int filde );
```

パラメータ

I/O	パラメータ	説明
I	int <i>filde</i> ;	クローズ / 削除する一時ファイルのファイル記述子

機能

filde で指定された一時ファイルをクローズ / 削除する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。

- *filde* で指定された一時ファイルのクローズ
- *filde* で指定された一時ファイルの削除

注意 1 ファイル記述子 *filde* には、応答関数 `webs_tempfile` の戻り値が設定されます。

注意 2 RX-NET(WebServer) 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

webs_getsize

概要

ファイル(ハイパー・テキスト形式ファイル, JPEG ファイルなど)のサイズを獲得する際に呼び出される応答関数

C 言語形式

```
int webs_getsize ( char *filename );
```

パラメータ

I/O	パラメータ	説明
I	char *filename ;	サイズを獲得するファイル(ハイパー・テキスト形式ファイル, JPEG ファイルなど)のファイル名を格納した領域へのポインタ

機能

filename で指定されたファイル(ハイパー・テキスト形式ファイル, JPEG ファイルなど)のサイズを獲得する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。

- *filename* で指定されたファイル(ハイパー・テキスト形式ファイル, JPEG ファイルなど)のサイズを獲得
- 獲得したファイル・サイズを本応答関数の戻り値として設定

注意 1 ファイル名 *filename* には, Web クライアントから受信したリクエスト URI(パス部)が設定されます。したがって, リクエスト URI が “http://www.sample1.nec.co.jp/DirName1/DirName2/FileName1” の場合, *filename* には, “/DirName1/DirName2/FileName1” が設定されます。

注意 2 RX-NET(WebServer) 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

-1 以外

正常終了

- 戻り値として “獲得したファイルのサイズ(単位: バイト)” が返されます。

-1

異常終了

- 本応答関数の処理終了後, RX-NET(WebServer) は Web クライアントからのリクエスト受け付け完了処理 (Web クライアントに対するステータス・コード 500 のレスポンス送信処理, Web クライアントからのリクエストに対応した全ファイルのクローズ処理, Web クライアントとのネットワーク接続の遮断処理, リクエスト・タスクの終了処理など) を実行します。

webs_rewind

概要

ファイル・ポインタの位置を変更する際に呼び出される応答関数

C 言語形式

```
int webs_rewind ( int fildes );
```

パラメータ

I/O	パラメータ	説明
I	int <i>fildes</i> ;	ファイル・ポインタの位置を変更する一時ファイルのファイル記述子

機能

fildes で指定された一時ファイルのファイル・ポインタの位置を変更する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。

- *fildes* で指定されたファイルのファイル・ポインタの位置を本応答関数呼び出し以前の位置からファイルの先頭に変更

注意 1 ファイル記述子 *fildes* には、応答関数 `webs_tempfile` の戻り値が設定されます。

注意 2 RX-NET(WebServer) 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

- 本応答関数の処理終了後、RX-NET(WebServer) は Web クライアントからのリクエスト受け付け完了処理 (Web クライアントに対するステータス・コード 500 のレスポンス送信処理、Web クライアントからのリクエストに対応した全ファイルのクローズ処理、Web クライアントとのネットワーク接続の遮断処理、リクエスト・タスクの終了処理など) を実行します。

webs_check_filetype

概要

ディレクトリ, または, ファイルのタイプを獲得する際に呼び出される応答関数

C 言語形式

```
int webs_check_filetype ( char *name );
```

パラメータ

I/O	パラメータ	説明
I	char *name ;	タイプを獲得するディレクトリ名, または, ファイル名を格納した領域へのポインタ

機能

name で指定されたディレクトリ, または, ファイルのタイプを獲得する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* で指定されたディレクトリ, または, ファイルのタイプを獲得
- 獲得したタイプを本応答関数の戻り値として設定

注意 1 RX-NET(WebServer) では, Web クライアントから受信したリクエスト URI(パス部) が複数のディレクトリ, および, ファイルから構成されていた場合, 本応答関数の呼び出し処理を複数回連続して行います。なお, Web クライアントから受信したリクエスト URI(パス部) が “/DirName1/DirName2/FileName1” の際には, 本応答関数の 1 回目の呼び出し時には “/DirName1/” が, 2 回目の呼び出し時には “/DirName1/DirName2/” が, 3 回目の呼び出し時には “/DirName1/DirName2/FileName1” が *name* に設定されます。

注意 2 RX-NET(WebServer) 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

WEBS_FILETYPE_DIR	0x1	正常終了 - 本戻り値は, 該当ディレクトリが “認証処理が不要なディレクトリ” であることを意味しています。 - 本応答関数の処理終了後, RX-NET(WebServer) は本応答関数の再呼び出し処理を実行します。
WEBS_FILETYPE_DIR_AUTH	0x2	正常終了 - 本戻り値は, 該当ディレクトリが “認証処理が必要なディレクトリ” であることを意味しています。 - 本応答関数の処理終了後, RX-NET(WebServer) は本応答関数の再呼び出し処理を実行します。
WEBS_FILETYPE_CGI	0x3	正常終了 - 本戻り値は, 該当ファイルが “CGI ファイル” であることを意味しています。
WEBS_FILETYPE_FILE	0x4	正常終了

WEBS_FILETYPE_NOTEXIST	0x5	<ul style="list-style-type: none">- 本戻り値は、該当ファイルが“ハイパー・テキスト形式ファイル、JPEG ファイルなど”であることを意味しています。 異常終了 <ul style="list-style-type: none">- 対象ディレクトリ、または、対象ファイルが存在しません。- 本応答関数の処理終了後、RX-NET(WebServer) は Web クライアントからのリクエスト受け付け完了処理 (Web クライアントに対するステータス・コード 500 のレスポンス送信処理、Web クライアントからのリクエストに対応した全ファイルのクローズ処理、Web クライアントとのネットワーク接続の遮断処理、リクエスト・タスクの終了処理など) を実行します。
WEBS_FILETYPE_FORBIDDEN	0x6	異常終了 <ul style="list-style-type: none">- 対象ディレクトリ、または、対象ファイルに対するアクセス権がありません。- 本応答関数の処理終了後、RX-NET(WebServer) は Web クライアントからのリクエスト受け付け完了処理 (Web クライアントに対するステータス・コード 500 のレスポンス送信処理、Web クライアントからのリクエストに対応した全ファイルのクローズ処理、Web クライアントとのネットワーク接続の遮断処理、リクエスト・タスクの終了処理など) を実行します。

webs_get_cgifunc

概要

Web クライアントからのリクエスト URI が CGI ファイルの際に呼び出される応答関数

C 言語形式

```
int (*webs_get_cgifunc ( char *filename )) ( int , int , int , int * , char * );
```

パラメータ

I/O	パラメータ	説明
I	char *filename ;	CGI 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

Web クライアントからのリクエスト URI が CGI ファイルの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。

- filename で指定された CGI ファイルに記述されている CGI 関数へのポインタを獲得
- 獲得した CGI 関数へのポインタを本応答関数の戻り値として設定

注意 1 ファイル名 filename には、Web クライアントから受信したリクエスト URI(パス部) が設定されます。
したがって、リクエスト URI が “ http://www.sample1.nec.co.jp/DirName1/DirName2/FileName1 ” の場合、
filename には、“ /DirName1/DirName2/FileName1 ” が設定されます。

注意 2 RX-NET(WebServer) 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

NULL 以外

正常終了

- 戻り値として “ 獲得した CGI 関数へのポインタ ” が返されます。
- 本応答関数の処理終了後、RX-NET(WebServer) は該当 CGI 関数の呼び出し処理を実行します。

NULL

異常終了

- 本応答関数の処理終了後、RX-NET(WebServer) は Web クライアントからのリクエスト受け付け完了処理 (Web クライアントに対するステータス・コード 500 のレスポンス送信処理、Web クライアントからのリクエストに対応した全ファイルのクローズ処理、Web クライアントとのネットワーク接続の遮断処理、リクエスト・タスクの終了処理など) を実行します。

webs_get_authname

概要

webs_check_filetype からの戻り値が WEBS_FILETYPE_DIR_AUTH の際に呼び出される応答関数

C 言語形式

```
int webs_get_authname ( char *dirname , char *authname );
```

パラメータ

I/O	パラメータ	説明
I	char * <i>dirname</i> ;	ユーザ認証を必要と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
O	char * <i>authname</i> ;	ユーザ認証を必要とするディレクトリに対応した認証箇所名を格納する領域へのポインタ

機能

webs_check_filetype からの戻り値が WEBS_FILETYPE_DIR_AUTH の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* で指定されたユーザ認証を必要とするディレクトリに対応した認証箇所名を獲得
- 獲得した認証箇所名を *authname* で指定された領域に格納

注意 1 RX-NET(WebServer) では、*authname* で指定された領域に格納する認証箇所名の最大サイズを 63 バイトに規定しています。

注意 2 RX-NET(WebServer) 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

- 本応答関数の処理終了後、RX-NET(WebServer) は Web クライアントに対する認証箇所名のレスポンス送信処理を実行します。

-1

異常終了

- 本応答関数の処理終了後、RX-NET(WebServer) は Web クライアントからのリクエスト受け付け完了処理 (Web クライアントに対するステータス・コード 500 のレスポンス送信処理、Web クライアントからのリクエストに対応した全ファイルのクローズ処理、Web クライアントとのネットワーク接続の遮断処理、リクエスト・タスクの終了処理など) を実行します。

webs_auth

概要

ユーザ認証を必要とするディレクトリに対するアクセス要求を受け付けた際に呼び出される応答関数

C 言語形式

```
int webs_auth ( char *username , char *password , char *authname );
```

パラメータ

I/O	パラメータ	説明
I	char *username ;	ユーザ名を格納した領域へのポインタ
I	char *password ;	パスワードを格納した領域へのポインタ
I	char *authname ;	ユーザ認証を必要とするディレクトリに対応した認証箇所名を格納した領域へのポインタ

機能

RX-NET(WebServer) が Web クライアントからのリクエスト (ユーザ認証を必要とするディレクトリに対するアクセス要求) 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。

- *username* , *password* , *authname* から得られる情報をもとに、ユーザ認証を必要とするディレクトリに対するアクセス要求の認証処理

注意 1 *authname* で指定された領域には、応答関数 `webs_get_authname` において設定された認証箇所名が設定されます。

注意 2 RX-NET(WebServer) 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

- 本戻り値は、“ 認証成功 ” を意味しています。

-1

異常終了

- 本戻り値は、“ 認証失敗 ” を意味しています。
- 本応答関数の処理終了後、RX-NET(WebServer) は Web クライアントからのリクエスト受け付け完了処理 (Web クライアントに対するステータス・コード 401 のレスポンス送信処理、Web クライアントからのリクエストに対応した全ファイルのクローズ処理、Web クライアントとのネットワーク接続の遮断処理、リクエスト・タスクの終了処理など) を実行します。

webs_get_directoryindex

概要

Web クライアントからのリクエスト URI がディレクトリの際に呼び出される応答関数

C 言語形式

```
int webs_get_directoryindex ( char *dirname , char *filename );
```

パラメータ

I/O	パラメータ	説明
I	char *dirname ;	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
O	char *filename ;	参照ファイルのファイル名を格納する領域へのポインタ

機能

Web クライアントからのリクエスト URI がディレクトリの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。

- リクエスト URI に参照ファイルのファイル名(例:index.htmlなど)を付加したものを *filename* で指定された領域に格納

注意 1 ディレクトリ名 *dirname* には、Web クライアントから受信したリクエスト URI (パス部) が設定されます。したがって、リクエスト URI が “ http://www.sample1.nec.co.jp/DirName1/DirName2/ ” の場合、*dirname* には、“ /DirName1/DirName2/ ” が設定されます。

注意 2 RX-NET(WebServer) では、*filename* で指定された領域に格納するファイル名 (リクエスト URI のパス部を含む) の最大サイズを 511 バイトに規定しています。

注意 3 本応答関数の呼び出しタイミングは、リクエスト URI に対するタイプの獲得処理が完了 (リクエスト URI を構成している各ディレクトリに対する応答関数 *webs_check_filetype* の実行) した直後となります。

注意 4 RX-NET(WebServer) 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

0x0

正常終了

-1

異常終了

- 本応答関数の処理終了後、RX-NET(WebServer) は Web クライアントからのリクエスト受け付け完了処理 (Web クライアントに対するステータス・コード 403 のレスポンス送信処理、Web クライアントからのリクエストに対応した全ファイルのクローズ処理、Web クライアントとのネットワーク接続の遮断処理、リクエスト・タスクの終了処理など) を実行します。

webs_log_user

概要

Web クライアントからのアクセス・ログを記録する際に呼び出される応答関数

C 言語形式

```
int webs_log_user ( unsigned int ipaddr , char *reqmsg , int stscd );
```

パラメータ

I/O	パラメータ	説明
I	unsigned int <i>ipaddr</i> ;	Web クライアントの IP アドレス (ネットワーク・バイト・オーダー)
I	char * <i>reqmsg</i> ;	Web クライアントから受信したリクエスト・メッセージ (先頭 1 行目のみ) を格納した領域へのポインタ
I	int <i>stscd</i> ;	Web クライアントに対して送信したステータス・コード

機能

Web クライアントからのアクセス・ログを記録する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。

- *ipaddr* , *reqmsg* , *stscd* から得られる情報をもとに、アクセス・ログの記録

注意 RX-NET(WebServer) 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

webs_cgifunc

概要

Web クライアントからの呼び出し要求を受け付けた際に呼び出される CGI 関数

C 言語形式

```
int webs_cgifunc ( int stdinf , int envf , int headeroutf , int bodyoutf , int *bodysize , char *reason ) ;
```

パラメータ

I/O	パラメータ	説明
I	int <i>stdinf</i> ;	リクエスト・メッセージ(ボディ部)格納用ファイルの記述子
I	int <i>envf</i> ;	リクエスト・メタ変数格納ファイルの記述子
I	int <i>headeroutf</i> ;	レスポンス・メッセージ(ヘッダ部)格納用ファイルの記述子
I	int <i>bodyoutf</i> ;	レスポンス・メッセージ(ボディ部)格納用ファイルの記述子
O	int <i>*bodysize</i> ;	レスポンス・メッセージ(ボディ部)格納用ファイルのサイズを格納する領域へのポインタ
O	char <i>*reason</i> ;	ステータス・コードに対応したステータス・フレーズを格納する領域へのポインタ

機能

Web クライアントからの呼び出し要求を受け付けた際に呼び出される CGI 関数です。

以下に、本 CGI 関数で実行すべき処理を示します。

- *envf* で指定されたファイルに設定されているリクエスト・メタ変数に対応した処理
- Web クライアントに送信するレスポンス・メッセージのヘッダ部を *headeroutf* で指定された領域に格納
- Web クライアントに送信するレスポンス・メッセージのボディ部を *bodyoutf* で指定された領域に格納
- *headeroutf* で指定されたメッセージ・ボディ出力ファイルのサイズを獲得
- 獲得したファイル・サイズを *bodysize* で指定された領域に格納
- ステータス・コードに対応したリズン・フレーズを獲得
- 獲得したリズン・フレーズを *reason* で指定された領域に格納
- ステータス・コードを本 CGI 関数の戻り値として設定

注意 1 Web クライアントからの HTTP メソッドが POST の場合、*stdinf* で指定された領域には、RX-NET(WebServer) が Web クライアントから受信したリクエスト・メッセージのボディ部が格納されます。Web クライアントからの HTTP メソッドが POST 以外の場合、*stdinf* で指定された領域の内容は不定となります。

注意 2 RX-NET(WebServer) では、*reason* で指定された領域に格納するリズン・フレーズの最大サイズを 63 バイトに規定しています。

注意 3 RX-NET(WebServer) では、本 CGI 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\cgi.c
```

【CCV850E 対応版の場合】

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

戻り値

0x0, -1 以外の正数	正常終了 <ul style="list-style-type: none">- 本戻り値は、RX-NET(WebServer) から Web クライアントに対して送信される “ステータス・コード” を意味しています。- 本 CGI 関数の処理終了後、RX-NET(WebServer) は Web クライアントからのリクエスト受け付け完了処理 (Web クライアントに対するステータス・コード、および、レスポンス・メッセージのレスポンス送信処理など) を実行します。
0x0	正常終了 <ul style="list-style-type: none">- 本 CGI 関数の処理終了後、RX-NET(WebServer) は Web クライアントからのリクエスト受け付け完了処理 (Web クライアントに対するステータス・コード 200、および、レスポンス・メッセージのレスポンス送信処理など) を実行します。
-1	異常終了 <ul style="list-style-type: none">- 本 CGI 関数の処理終了後、RX-NET(WebServer) は Web クライアントからのリクエスト受け付け完了処理 (Web クライアントに対するステータス・コード 500 のレスポンス送信処理、Web クライアントからのリクエストに対応した全ファイルのクローズ処理、Web クライアントとのネットワーク接続の遮断処理、リクエスト・タスクの終了処理など) を実行します。

索引

A

Allow	30
API 関数	31, 34, 38
webs_abort	32, 34, 38, 40
webs_start	31, 34, 38, 39
外部インタフェース仕様	38
データ・マクロ	35
呼び出し方法	34
Authorization	30

C

cf850pro	21
CF 定義ファイル	21
SCT 情報	21
SIT 情報	21
情報ファイル	21
Connection	29, 30
Content-Length	30
Content-Type	30

F

fnsconfig.h	34
-------------------	----

G

GET	29
-----------	----

H

HEAD	29
Host	30
HTTP メソッド	29
GET	29
HEAD	29
POST	29
HTTP ライブラリ	18, 19
libwebs.a	18, 19

L

libwebs.a	18, 19
Location	30

M

Makefile	18
----------------	----

P

POST	29
------------	----

R

README.WEBS	18, 19
RFC	14
RX850 Pro 依存部	22
エントリ処理	22
初期化ハンドラ	23
ハードウェア初期化部	23
ブート処理	22
RX-FS850 依存部	23
rxnet.h	34
RX-NET(TCP/IP) 依存部	23
RX-NET(TCP/IP) 用標準ヘッダ・ファイル	34
rxnet.h	34
RX-NET(WebServer)	13
API 関数	34
RX-NET(WebServer) 依存部	41
Web サーバ機能	28
位置付け	13
インストール	16
階層的な位置付け	14
開発環境	15
システム構築	20
実行環境	15
特徴	14
RX-NET(WebServer) 依存部	23, 41
応答関数	24, 53
基本情報	24, 44
データ・マクロ	49
RX-NET(WebServer) 用標準ヘッダ・ファイル	18, 19, 34
rxnet_webs.h	18, 19, 34
rxnet_webs.h	18, 19, 34

S

sample.bld	19
SCT 情報	21
タスク管理機能情報	21
タスク付属同期機能情報	21
メモリ・プール管理機能情報	21
Server	30
SIT 情報	21
システム最大値情報	21
システム情報	21
メモリ・プール情報	21

T

Transfer-Encoding	29
-------------------------	----

W

webs_abort	32, 34, 38, 40
webs_auth	24, 33, 41, 53, 72
webs_callback_epilog	24, 33, 41, 53, 55
webs_callback_prolog	24, 33, 41, 53, 54
webs_cgifunc	25, 33, 41, 53, 75
webs_check_filetype	24, 33, 41, 53, 68
webs_fcclose	24, 33, 41, 53, 60
webs_fopen	24, 33, 41, 53, 58
webs_fread	24, 33, 41, 53, 61
webs_fwrite	24, 33, 41, 53, 63
webs_get_authname	24, 33, 41, 53, 71
webs_get_cgifunc	24, 33, 41, 53, 70
webs_get_directoryindex	25, 33, 41, 53, 73
webs_getsize	24, 33, 41, 53, 66
webs_log_user	25, 33, 41, 53, 74
webs_rewind	24, 33, 41, 53, 67
webs_start	31, 34, 38, 39
webs_task_end	24, 33, 41, 53, 57
webs_task_start	24, 33, 41, 53, 56
webs_tempfile	24, 33, 41, 53, 64
webs_tempfile_del	24, 33, 41, 53, 65
Web サーバ機能	28
WWW-Authenticate	30

あ

アーカイブ・オブジェクト 26

い

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

え

エラー・メッセージ情報 48
エントリ処理 22

お

応答関数 24, 33, 41, 53
 webs_auth 24, 33, 41, 53, 72
 webs_callback_epilog 24, 33, 41, 53, 55
 webs_callback_prolog 24, 33, 41, 53, 54
 webs_cgifunc 25, 33, 41, 53, 75
 webs_check_filetype 68, 24, 33, 41, 53
 webs_fclose 24, 33, 41, 53, 60
 webs_fopen 24, 33, 41, 53, 58
 webs_fread 24, 33, 41, 53, 61
 webs_fwrite 24, 33, 41, 53, 63
 webs_get_authname 24, 33, 41, 53, 71
 webs_get_cgifunc 24, 33, 41, 53, 70
 webs_get_directoryindex 25, 33, 41, 53, 73
 webs_getsize 24, 33, 41, 53, 66
 webs_log_user 25, 33, 41, 53, 74
 webs_rewind 24, 33, 41, 53, 67
 webs_task_end 24, 33, 41, 53, 57
 webs_task_start 24, 33, 41, 53, 56
 webs_tempfile 24, 33, 41, 53, 64
 webs_tempfile_del 24, 33, 41, 53, 65
 外部インタフェース仕様 53
オブジェクト・ファイル 26

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 38, 53
 webs_auth 72
 webs_callback_epilog 55
 webs_callback_prolog 54
 webs_cgifunc 75
 webs_check_filetype 68
 webs_fclose 60
 webs_fopen 58
 webs_fread 61
 webs_fwrite 63
 webs_get_authname 71
 webs_get_cgifunc 70
 webs_get_directoryindex 73
 webs_getsize 66
 webs_log_user 74

webs_rewind 67
webs_task_end 57
webs_task_start 56
webs_tempfile 64
webs_tempfile_del 65

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 25

き

基本情報 24, 44
エラー・メッセージ情報 48
サーバ情報 44
受信バッファ情報 45
タイムアウト時間情報 46
ディレクトリ情報 45
ネットワーク情報 44
リクエスト・タスク情報 47
リクエスト・メッセージ情報 46

こ

コンフィギュレータ 21
 cf850pro 21

く

サーバ情報 44
 WEBS_FQDN 44
 WEBS_IPADDR 44
 WEBS_SERVER_INFO 44

し

システム構築 20
 CF 定義ファイル 21
 RX850 Pro 依存部 22
 RX-FS850 依存部 23
 RX-NET(TCP/IP) 依存部 23
 RX-NET(WebServer) 依存部 23
 アーカイブ・オブジェクト 26
 オブジェクト・ファイル 26
 処理プログラム 25
 リンク・ディレクティブ・ファイル 26
 ロード・モジュール 27
システム・コール・テーブル 21
システム最大値情報 21
システム情報 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 45
 WEBS_SIZE 45

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	23
処理プログラム	25
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	25
周期起動ハンドラ	25
タスク	25
直接起動割り込みハンドラ	25

せ

静的設定情報ヘッダ・ファイル	34
fnconfig.h	34

た

タイプ・コード	49
タイムアウト時間情報	46
WEBS_TIMEOUT	46
タスク	25
タスク管理機能情報	21
タスク付属同期機能情報	21

ち

直接起動割り込みハンドラ	25
--------------	----

て

ディレクトリ構成	18
CA850 対応版	18
CCV850E 対応版	19
ディレクトリ情報	45
WEBS_ERROR_DIR	45
データ・マクロ	35, 49
タイプ・コード	49
戻り値	35
リクエスト・メタ変数	49
テキスト・ファイル	18, 19
README.WEBS	19, 18

ね

ネットワーク情報	44
WEBS_HOSTNAME	44

は

ハードウェア初期化部	23
------------	----

ひ

ビルド・ファイル	19
sample.bld	19

ふ

ブート処理	22
-------	----

へ

ヘッダ部	29, 30
Allow	30
Authorization	30
Connection	29, 30
Content-Length	30
Content-Type	30
Host	30
Location	30
Server	30
Transfer-Encoding	29
WWW-Authenticate	30

め

メイク・ファイル	18
Makefile	18
メモリ・プール管理機能情報	21
メモリ・プール情報	21

も

戻り値	35
-----	----

り

リクエスト・タスク情報	47
WEBS_STKSZ	47
WEBS_TASK_EPRI	47
WEBS_TASK_IPRI	47
リクエスト・メタ変数	49
リクエスト・メッセージ	29
HTTP メソッド	29
ヘッダ部	29
リクエスト・メッセージ情報	46
WEBS_MAX_CONTENTLENGTH	46
リンク・ディレクティブ・ファイル	26

れ

レスポンス・メッセージ	29
ヘッダ部	30

ろ

ロード・モジュール	27
-----------------	----

— 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

【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.1.00 ネットワーク・ライブラリ ユーザーズ・マニュアル WebServer

(U16294JJ1V0UM00 (第1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