

TDC通信パッケージ 開発者ガイド

第5版

初版	2018年1月17日
2版	2020年4月20日
3版	2020年5月29日
4版	2020年6月26日
5版	2021年3月26日

目次

本書について

第1部 概説	4
第1章 概説	4
本パッケージAPI DLLの位置づけ	4
第2部 DLL	5
第2章 DLL概要	5
DLLファイル	5
DLL関数一覧	5
第3章 DLL関数	6
DLL関数の実行形態	6
DLL関数の同期／非同期	6
処理待ち行列の優先順位	7
DLL関数インターフェース記述形式	7
戻り値と処理結果(rc_code)パラメータについて	8
処理シーケンス(func_seq)パラメータについて	8
DLL関数インターフェース	9
File_Send_Api(ファイル送信)	9
File_Send_Ex_Api(ファイル再送信)	11
File_Recv_Api(ファイル受信)	13
File_Recv_Api2(拡張ファイル受信)	15
File_Recv_Ex_Api(保管取出しファイル受信)	16
File_Recv_Ex_Api2(拡張保管取出しファイル受信)	17
File_List_Api(蓄積状況照会)	18
Audit_List_Api(オーディットトレイル受信)	19
Status_List_Api(状況確認照会)	20
Event_List_Api(処理状況問合せ)	21
第4章 DLL関数呼び出しサンプル	22
C	22
VB.NET	22
VB.NETでの注意事項	22
第3部 コマンド機能	23
第5章 コマンド概要	23
EXEファイル	23
コマンド機能一覧	23
第6章 コマンド操作	24
コマンド機能実行形態	24
コマンド機能インターフェース	25
EDISEND.EXE(ファイル送信)	25
EDIRECV.EXE(ファイル受信)	27
ログ出力	28
処理状況コード一覧	29
付録C. 受信ファイル情報のファイルレイアウト	30
付録D. DLL関数の戻り値一覧	31
付録E. TDC通信パッケージWindows版 コード変換表(詳細)	37

本書について

本書は、TDC通信パッケージ(本パッケージ)のAPIを利用するソフトウェア技術者(開発者)の支援を目的としています。

本パッケージが提供するAPI及びコマンド機能について記述しています。

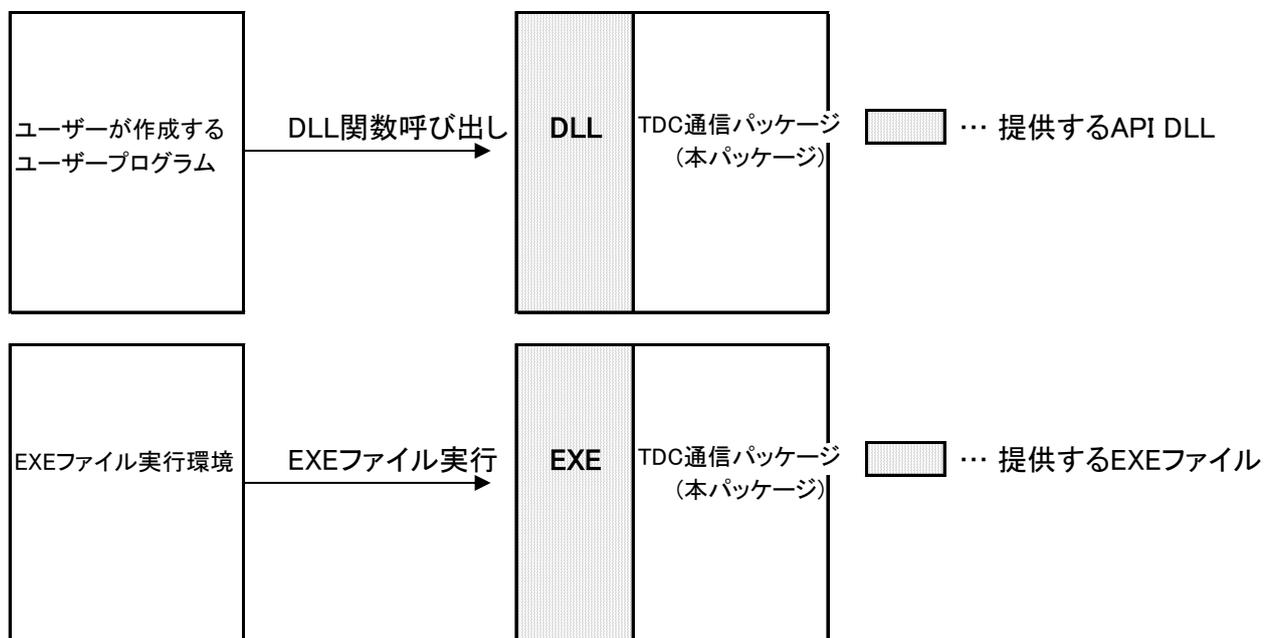
本書の対象読者は、本パッケージAPIを利用し、ユーザー独自のプログラム開発を行う開発者の方を対象としています。

第1部 概説

第1章 概説

本パッケージでは、ユーザープログラムから本パッケージの機能をご利用頂けるAPIをDLL及びコマンド機能(EXEファイル実行)で提供します。

本パッケージ API DLL, EXEの位置づけ



DLL, EXEは、本パッケージをインストールすることにより、提供されます。

第2部 DLL

第2章 DLL概要

DLLファイル

本パッケージのAPI DLLファイル名は、EDI_USER.DLL です。
このDLLファイルは、本パッケージインストール時に、Windowsのシステムディレクトリへコピーされます。

DLL関数一覧

EDI_USER.DLLには、以下のDLL関数があります。

ファイル送信関数

ユーザーアプリケーション・ファイルを指定宛先へ送信するために使用します。

File_Send_Api

ユーザーアプリケーション・ファイルを指定宛先へ送信します。

File_Send_Ex_Api

送信エラーとなったユーザーアプリケーション・ファイルの再送信処理を行いません。

ファイル受信関数

自分宛に送られているファイルを受信するために使用します。

File_Recv_Api

自分宛に送られているファイルを受信します。(ファイル受信)

File_Recv_Api2

自分宛に送られているファイルを受信します。(拡張ファイル受信)

File_Recv_Ex_Api

一度受信したファイルを再度受信します。(保管取り出しファイル受信)

File_Recv_Ex_Api2

一度受信したファイルを再度受信します。(拡張保管取り出しファイル受信)

照会関数

ファイルの蓄積状況・ファイル転送状況など、本パッケージの利用状況を取得します。

File_List_Api

自分宛に送られているファイルの一覧を取得します。

Audit_List_Api

オーディットトレイルを取得します。

Status_List_Api

IE/EXを使用したメッセージ変換処理の状況確認メッセージを取得します。

DLL管理関数

本パッケージのDLL処理状態を確認します。

Event_List_Api

DLLの処理状態を取得します。

第3章 DLL関数

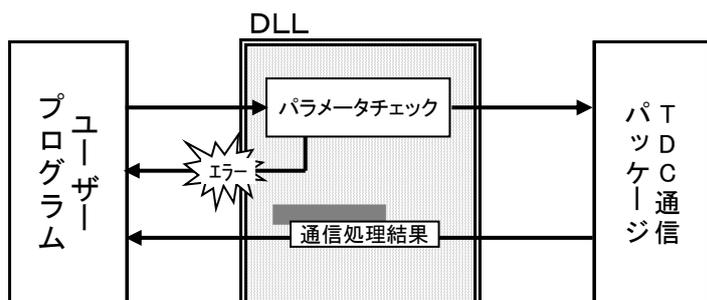
DLL関数の実行形態

本パッケージは、複数の要求を同時に受け付けることが可能です。
よって複数のユーザーアプリケーションからDLL関数を同時に呼出すことができます。
但し、実際に通信が行えるセッション数に制限がありますので、そのセッション数を超えたサービス要求は、本パッケージの処理待ち行列に入ります。

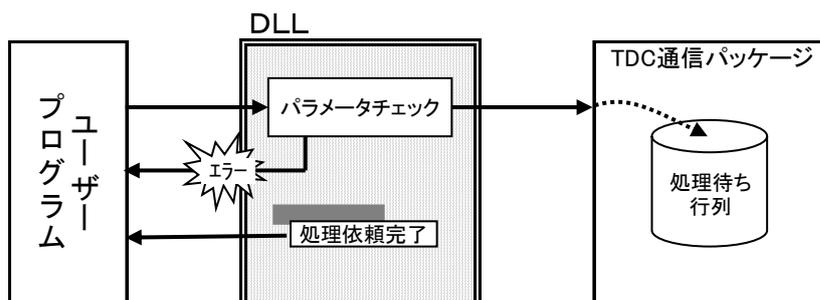
DLL関数の同期／非同期

DLL管理関数を除く各DLL関数は、次の実行形態の中から選択できます。

- ・DLLからの通信処理結果を待つ(同期実行)
- ・DLLからの通信処理結果を待たない(非同期実行)
- ・DLLからの通信処理結果を待つ(同期実行)
ユーザープログラムが要求したサービスの通信処理が完了するまでDLLから戻りません。



- ・DLLからの通信処理結果を待たない(非同期実行)
ユーザープログラムから渡されたパラメータのシンタクスチェックを行い、本パッケージへ処理依頼を通知したら、即座にDLLから戻ります。



処理待ち行列の優先順位

DLL管理関数を除く各DLL関数は、本パッケージが処理する優先順位を指定することができます。下記に実行優先順を示します。

・実行優先順位

優先順位	指定パラメータ
1	同期／優先
2	非同期／優先
3	同期／通常
4	非同期／通常

同一優先順位の場合、先入れ先出しとなります。

DLL関数インターフェース記述形式

DLL関数インターフェースでは、以下の形式で説明します。

機能名

機能名は、各ページの一番上に記され、その下に簡単な説明があります。

書式

パラメータの入出力方向、値の必須／任意を記しています。

入出力方向

入力 (I)
出力 (O)
入出力 (I/O)

値の必須／任意

必須 値は指定は必須です。指定桁数に文字列が満たない場合は、残り桁数に空白をセットするか、文字列の最後にNULL(0x00)をセットして下さい。
任意 値の指定は任意(省略可能)です。省略時は全ての桁数に空白をセットして下さい。

パラメーター

各パラメーターにはC言語データ・タイプ、桁数、表記方法、および簡単な説明が付記されています。
・桁数は各機能が有効とみなす桁数です。よって指定桁数を超えた値は無視されます。
・各パラメーターについて表記方法および簡単な説明があります。該当するものについては、制約事項も付記されています。

戻り値

関数の戻り値を記しています。

解説

必要に応じてその機能についての追加説明を行っています。

例

各機能について、C言語で作成された例を示します。

戻り値と処理結果(rc_code)パラメータについて

各関数の戻り値は、以下の通りです。

戻り値	意味
rc ≠ 0	エラーが発生しました ※
rc = 0	正常に完了しました。

※ 各関数における戻り値の詳細は付録Dを参照してください。

詳細なエラー内容については、処理結果(rc_code)パラメータに、エラーコード(6桁の文字)が返されます。エラーコードとその内容については、本パッケージのエラーコード一覧機能で確認できます。

戻り値rc = 0 の場合でも、処理結果(rc_code)が返される場合があります。従って、必ず処理結果(rc_code)の値をもとにエラーハンドリングを行って下さい。

処理シーケンス(func_seq)パラメータについて

処理シーケンス func_seq を呼び出し元で設定する場合、各DLL関数を呼び出す都度、ユニークとなる値を設定してください。

DLL関数インターフェース

File_Send_Api (ファイル送信)

センターへ指定ファイルの通常送信、及び上書き再送信を行います。

【書式】

```

INT rc = File_Send_Api( rc_code      , /* ( 0 ) 処理結果           (必須) */
                       func_seq     , /* (I/O) 処理シーケンス     (必須) */
                       async_code    , /* ( I ) 同期/非同期         (必須) */
                       apl_cnv       , /* ( I ) ヘッダ作成         (必須) */
                       file_code     , /* ( I ) メッセージ識別子     (必須) */
                       atesaki_name , /* ( I ) 宛先                 (必須) */
                       send_mode     , /* ( I ) 上書き送信         (必須) */
                       file_name     , /* ( I ) 送信ファイル名       (必須) */
                       file_ext      , /* ( I ) 送信ファイル拡張子   (任意) */
                       file_type     , /* ( I ) ファイル種別         (任意) */
                       compact       , /* ( I ) 圧縮                 (任意) */
                       a_code        , /* ( I ) 暗号化               (任意) */
                       code_cnv      , /* ( I ) コード変換           (任意) */
                       apl_len       , /* ( I ) APL有効長           (任意) */
                       user_id       , /* ( I ) ユーザーID          (任意) */
                       yobi          , /* ( I ) 予備                 (任意) */
                       );
    
```

【パラメーター】

パラメーター名	タイプ	桁数	表記方式	説明
rc_code	char*	6	"XXXXXX"	DLLでの処理結果をセットします。 前3桁は理由コード、後3桁は内部ロジックコードがセットされます。
func_seq	char*	8	"XXXXXXXX"	呼出し元が、発行した各処理を管理するのに使用します。 非同期モードで処理要求を発行した場合、該当する処理の状況確認を行う際に使用します。指定する場合は、8桁フル入力して下さい。 空白を指定した場合は、システムが割り振り、その値を返します。
async_code	char*	1	"1" 非同期&通常 "2" 同期&通常 "3" 非同期&優先 "4" 同期&優先	同期/非同期モードおよび優先/通常モードの確認に使用します。
apl_cnv	char*	1	"Y" ヘッダ作成する "N" ヘッダ作成しない	TNSヘッダを作成するか否かを指定します。
file_code	char*	8	"XXXXXXXX"	アプリケーションで規定したファイルを識別するコードを指定します。 ファイルコードがセットされていない場合、エラーとして処理を中断します。 指定された別名でアドレス帳マスタを検索し、ニックネームもしくは直接宛先がセットされます。 ※アドレス帳マスタに該当が無い場合は、引数を大文字にして宛先として使用します。 ただし、引数が4~8桁で、「@」「.」を含まない場合は 12345678 ⇒ 1234. 12345678 と加工して宛先とします。 先頭1バイト目*指定の場合、送信元ファイルのHレコードより検索します。 (TNS標準形式もしくはTNS拡張形式で、「ヘッダー作成しない」場合のみ) 宛先がセットされていない場合、エラーとして処理を中断します。
send_mode	char*	1	"Y" 上書き送信 "N" 通常送信	ファイル送信するモードを指定します。 "Y" 上書き送信時は、func_seqに過去にファイル送信した際のシーケンスを指定します。 このエリアにセットされたファイル名でファイル検索を行います。 相対パス指定の場合、本DLL関数がCALLされたカレントパス配下から検索します。 ファイルが存在しない場合、エラーとして処理を中断します。 ファイル拡張子として使用します。 指定が無い場合、file_codeに基づき転送制御マスターから求めます。
file_name	char*	120	絶対パス付きのファイル名	ファイル種別を確認するのに使用します。 指定が無い場合、file_codeに基づき転送制御マスターから求めます。
file_ext	char*	3	"XXX"	ファイル種別を確認するのに使用します。 指定が無い場合、file_codeに基づき転送制御マスターから求めます。
file_type	char*	1	"1" TNS標準形式 "2" TNS拡張形式 "3" TDC標準形式 "4" 自由形式	データ圧縮の確認に使用します。 指定が無い場合、file_codeに基づき転送制御マスターから求めます。 file_typeが "1","4" の場合、圧縮が指定されていても圧縮は行いません。
compact	char*	1	"Y" 圧縮有り "N" 圧縮無し	データ暗号化の確認に使用します。 指定が無い場合、file_codeに基づき転送制御マスターから求めます。 file_typeが "1","4" の場合、暗号化が指定されていても暗号化は行いません。
a_code	char*	1	"Y" 暗号化有り "N" 暗号化無し	コード変換の確認に使用します。 指定が無い場合、file_codeに基づき転送制御マスターから求めます。 file_typeが "4" の場合、コード変換が指定されていてもコード変換は行いません。
code_cnv	char*	1	"Y" コード変換有り "N" コード変換無し	APL有効長の確認に使用します。 指定する場合は、5桁フル入力して下さい。 指定が無い場合、file_codeに基づき転送制御マスターから求めます。
apl_len	char*	5	"XXXXX" (10進数)	利用するユーザーを指定します。 指定が無い場合、代表のユーザーIDを使用します。
user_id	char*	16	"XXXXXXXXXXXXXXXXXX"	システム予約。必ず空白を設定する必要があります。
yobi	char*	20	"XXXXXXXXXXXXXXXXXX"	

【戻り値】

0 正常終了
0 以外 エラー ※

※戻り値の詳細は付録Dを参照してください。

尚、DLL関数の戻りコードが正常(0)、異常(0以外)を返したとしても、必ず、処理結果コードの値をもとにエラーハンドリングを行って下さい。

【解説】

任意のインプットパラメータに値がセットされていない場合、file_codeパラメータに定義された内容で転送制御マスターから求めますが、求めた転送制御マスターに各項目データの値が定義されていない場合、転送制御マスター定義エラーとして処理を中断します。

各インプットパラメータに値がセットされていた場合、セットされている値を優先し転送制御マスターの確認は行いません。

通常送信 (send_mode = "N")で、**処理シーケンスを指定した場合、指定した処理シーケンスが過去に使用されていた場合、エラーとして処理を中断します**
非同期モードで処理要求を発行した場合、Event_List_Apiをコールし処理が完了したのを確認して下さい。

マルチファイル送信時は、マシンのメモリサイズによって、送信できるマルチファイル件数が異なります。

マルチファイルの上限値を超えた場合は、「202-202」エラーが発生します。

File_Send_Ex_Api (ファイル再送信)

送信エラーとなった指定ファイルの再送信をします。

【書式】

```

INT rc = File_Send_Ex_Api( rc_code      , /* ( 0 ) 処理結果                (必須) */
                          func_seq     , /* ( 1 ) 処理シーケンス          (必須) */
                          async_code   , /* ( 1 ) 同期/非同期              (必須) */
                          apl_cnv      , /* ( 1 ) ヘッダ作成                (必須) */
                          file_code     , /* ( 1 ) メッセージ識別子          (必須) */
                          atesaki_name , /* ( 1 ) 宛先                      (必須) */
                          send_mode    , /* ( 1 ) 上書き送信                (必須) */
                          file_name     , /* ( 1 ) 送信ファイル名            (必須) */
                          file_ext     , /* ( 1 ) 送信ファイル拡張子        (任意) */
                          file_type    , /* ( 1 ) ファイル種別              (任意) */
                          compact      , /* ( 1 ) 圧縮                      (任意) */
                          a_code       , /* ( 1 ) 暗号化                    (任意) */
                          code_cnv     , /* ( 1 ) コード変換                (任意) */
                          apl_len      , /* ( 1 ) APL有効長                (任意) */
                          user_id      , /* ( 1 ) ユーザーID                (任意) */
                          yobi         , /* ( 1 ) 予備                      (任意) */
                          );
    
```

【パラメーター】

パラメーター名	タイプ	桁数	表記方式	説明
rc_code	char*	6	"XXXXXX"	DLLでの処理結果をセットします。 前3桁は理由コード、後3桁は内部ロジックコードがセットされます。
func_seq	char*	8	"XXXXXXXX"	送信失敗時に使用していた処理シーケンスをセットします。 未入力もしくは空文字の場合はエラーとして処理を中断します。
async_code	char*	1	"1" 非同期&通常 "2" 同期&通常 "3" 非同期&優先 "4" 同期&優先	同期/非同期モードおよび優先/通常モードの確認に使用します。
apl_cnv	char*	1	"Y" ヘッダ作成する "N" ヘッダ作成しない	TNSヘッダを作成するか否か指定します。
file_code	char*	8	"XXXXXXXX"	アプリケーションで規定したファイルを識別するコードを指定します。 ファイルコードがセットされていない場合、エラーとして処理を中断します。
atesaki_name	char*	20	"KKKKKKKKUUUUUUUU"	指定された別名でアドレス帳マスタを検索し、ニックネームもしくは直接宛先がセットされます。 ※アドレス帳マスタに該当が無い場合は、引数を大文字にして宛先として使用します。 ただし、引数が4~8桁で、「@」「.」を含まない場合は 12345678 ⇒ 1234. 12345678 と加工して宛先とします。 先頭1バイト目'*'指定の場合、送信元ファイルのHLコードより検索します。 (TNS標準形式もしくはTNS拡張形式で、「ヘッダー作成しない」場合のみ) 宛先がセットされていない場合、エラーとして処理を中断します。
send_mode	char*	1	"Y" 上書き送信 "N" 通常送信	指定された値に関わらず、上書き送信モードとなります。
file_name	char*	120	絶対パス付きのファイル名	このエリアにセットされたファイル名でファイル検索を行います。 相対パス指定の場合、本DLL関数がCALLされたカレントパス配下から検索します。 ファイルが存在しない場合、エラーとして処理を中断します。
file_ext	char*	3	"XXX"	ファイル拡張子として使用します。 指定が無い場合、file_codeに基づき転送制御マスターから求めます。
file_type	char*	1	"1" TNS標準形式 "2" TNS拡張形式 "3" TDC標準形式 "4" 自由形式	ファイル種別を確認するのに使用します。 指定が無い場合、file_codeに基づき転送制御マスターから求めます。
compact	char*	1	"Y" 圧縮有り "N" 圧縮無し	データ圧縮の確認に使用します。 指定が無い場合、file_codeに基づき転送制御マスターから求めます。 file_typeが"1","4"の場合、圧縮が指定されていても圧縮は行いません。
a_code	char*	1	"Y" 暗号化有り "N" 暗号化無し	データ暗号化の確認に使用します。 指定が無い場合、file_codeに基づき転送制御マスターから求めます。 file_typeが"1","4"の場合、暗号化が指定されていても暗号化は行いません。
code_cnv	char*	1	"Y" コード変換有り "N" コード変換無し	コード変換の確認に使用します。 指定が無い場合、file_codeに基づき転送制御マスターから求めます。 file_typeが"4"の場合、コード変換が指定されていてもコード変換は行いません。
apl_len	char*	5	"XXXXX" (10進数)	APL有効長の確認に使用します。 指定する場合は、5桁フル入力して下さい。 指定が無い場合、file_codeに基づき転送制御マスターから求めます。
user_id	char*	16	"XXXXXXXXXXXXXXXX"	利用するユーザーを指定します。 指定が無い場合、代表のユーザーIDを使用します。
yobi	char*	20	"XXXXXXXXXXXXXXXX"	システム予約。必ず空白を設定する必要があります。

【戻り値】

0 正常終了
0 以外 エラー ※

※戻り値の詳細は付録Dを参照してください。

尚、DLL関数の戻りコードが正常(0)、異常(0以外)を返したとしても、必ず、処理結果コードの値をもとにエラーハンドリングを行って下さい。

【解説】

再処理の対象となるファイル送信処理でセットした各種パラメーターを変更せずに、同一パラメーターにセットして下さい。
処理シーケンスデータに関しては、再処理の対象となったファイル送信処理でセットした値、もしくはDLL関数から返された値を必ずセットして下さい。処理シーケンスパラメーターにセットされた値に基づいて検索処理が行なわれます。

検索処理を行なった結果、該当する転送処理が存在しない場合、エラーとして処理を中断します。

非同期モードで処理要求を発行した場合、Event_List_Apiをコールし処理が完了したのを確認して下さい。

File_Recv_Api(ファイル受信)

センターに蓄積されているファイルを受信します。

【書式】

```
INT rc = File_Recv_Api( rc_code      , /* ( 0 ) 処理結果          (必須) */
                       func_seq     , /* (I/O) 処理シーケンス   (必須) */
                       async_code    , /* ( I ) 同期/非同期       (必須) */
                       file_cat      , /* ( I ) 受信ファイル統合モード (必須) */
                       file_code     , /* ( I ) メッセージ識別子   (必須) */
                       sashidashi_name , /* ( I ) 差し出し          (必須) */
                       recv_path     , /* ( I ) 受信ファイル書込パス (必須) */
                       user_id       , /* ( I ) ユーザーID        (任意) */
                       yobi          , /* ( I ) 予備              (任意) */
                       );
```

【パラメーター】

パラメーター名	タイプ	桁数	表記方式	説明
rc_code	char*	6	"XXXXXX"	DLLでの処理結果をセットします。 前3桁は理由コード、後3桁は内部ロジックコードがセットされます。
func_seq	char*	8	"XXXXXXXX"	呼出し元が、発行した各処理を管理するのに使用します。 非同期モードで処理要求を発行した場合、該当する処理の状況確認を行う際に使用します。指定する場合は、8桁フル入力して下さい。 空白を指定した場合は、システムが割り振り、その値を返します。
async_code	char*	1	"1" 非同期&通常 "2" 同期&通常 "3" 非同期&優先 "4" 同期&優先	同期/非同期モードおよび優先/通常モードの確認に使用します。
file_cat	char*	1	"Y" 統合する "N" 統合しない	複数ファイルを受信した場合に統合するかどうかの確認に使用します。 処理モードが"Y" or "N" 以外の場合、エラーとして処理を中断します。
file_code	char*	8	"XXXXXXXX"	アプリケーションで規定したファイルを識別するコードを指定します。 ファイルコードがセットされていない場合、エラーとして処理を中断します。 「*」を指定した場合、内部的にEvent_List_Apiをコールします。 (取得した未受信情報ファイルは削除されます)
sashidashi_name	char*	20	"KKKKKKKKUUUUUUUU"	指定された別名でアドレス帳マスタを検索し、ニックネームもしくは直接宛先がセットされます。 ※アドレス帳マスタに該当が無い場合は、引数を大文字にして宛先として使用します。 ただし、引数が4~8桁で、「@」「.」を含まない場合は 12345678 ⇒ 1234. 12345678 と加工して宛先とします。 先頭1バイト目'*'指定の場合、送信元ファイルのHレコードより検索します。 (TNS標準形式もしくはTNS拡張形式で、「ヘッダー作成しない」場合のみ) 宛先がセットされていない場合、エラーとして処理を中断します。
recv_path	char*	84		受信ファイルを書き込むパス 相対パス指定の場合、本システムの導入ドライブのルートディレクトリ(ドライブ:¥)配下のパスとして扱います。受信パスがセットされていない場合、エラーとして処理を中断します。 該当するパスが存在しない場合もエラーとして処理を中断します。
user_id	char*	16	"XXXXXXXXXXXXXXXX"	利用するユーザーを指定します。 指定が無い場合、代表のユーザーIDを使用します。
yobi	char*	20	"XXXXXXXXXXXXXXXX"	システム予約。必ず空白を設定する必要があります。

【戻り値】

0 正常終了
0 以外 エラー ※
※戻り値の詳細は付録Dを参照してください。

尚、DLL関数の戻りコードが正常(0)、異常(0以外)を返したとしても、必ず、処理結果コードの値をもとにエラーハンドリングを行って下さい。

【解説】

任意のインプットパラメータに値がセットされていない場合、file_codeパラメータに定義された内容で転送制御マスターから求めますが、求めた転送制御マスターに各項目データの値が定義されていない場合、転送制御マスター定義エラーとして処理を中断します。
各インプットパラメータに値がセットされていた場合、セットされている値を優先し転送制御マスターの確認は行いません。
ファイルコード・差し出しに基づく自分宛のファイルが存在し場合のみ、受信処理を行います。
処理シーケンスを指定した場合、指定した処理シーケンスが過去に使用されていた場合、エラーとして処理を中断します。
非同期モードで処理要求を発行した場合、Event_List_Apiをコールし処理が完了したのを確認して下さい。
一度に受信できるファイル件数は150件となります。

【補足説明】

受信ファイル名は、本システムが次のように名前づけして指定されたパスに格納します。

・TNS標準形式(ファイルタイプ=1)

統合なしの場合

ファイル名=FFF.YYMMDDSEQ.TTTTT.EXT	
FFF	Hヘッダーのファイルコード
YYMMDD	Hヘッダーの作成日付
SEQ	Hヘッダーの作成日付SEQ
TTTTT	Hヘッダーの発信元取引先コード
EXT	転送制御マスターのファイル拡張子

統合あり & 差し出し≠'*'の場合

ファイル名=MMMMMMMM.YYYYMMDDSEQ.UUUUU.EXT	
MMMMMMMM	DLL関数パラメータのメッセージ識別子
YYYMMDD	システム日付
SEQ	メッセージ識別子毎の当日受信ファイルのカウンタ数(000~999)
UUUUU	DLL関数パラメータの差し出し
EXT	転送制御マスターのファイル拡張子

統合あり & 差し出し='*'の場合

ファイル名=MMMMMMMM.YYYYMMDDSEQ.EXT	
MMMMMMMM	DLL関数パラメータのメッセージ識別子
YYYMMDD	システム日付
SEQ	メッセージ識別子毎の当日受信ファイルのカウンタ数(000~999)
EXT	転送制御マスターのファイル拡張子

・TNS拡張形式(ファイルタイプ=2)

統合なしの場合

ファイル名=FFFGGGG.SSYMMDDNNSEQ.TTTTTUUU.EXT	
FFF	Hヘッダーのファイルコード
GGGGG	Hヘッダーのファイル詳細コード
SS	Hヘッダーの作成日付世紀
YYMMDD	Hヘッダーの作成日付
NN	Hヘッダーの作成日付SEQ拡張
SEQ	Hヘッダーの作成日付SEQ
TTTTT	Hヘッダーの発信元取引先コード
UUU	Hヘッダーの発信元サブアドレス
EXT	転送制御マスターのファイル拡張子

統合あり & 差し出し≠'*'の場合

ファイル名=MMMMMMMM.YYYYMMDDSEQ.UUUUU.EXT	
MMMMMMMM	DLL関数パラメータのメッセージ識別子
YYYMMDD	システム日付
SEQ	メッセージ識別子毎の当日受信ファイルのカウンタ数(000~999)
UUUUU	DLL関数パラメータの差し出し
EXT	転送制御マスターのファイル拡張子

統合あり & 差し出し='*'の場合

ファイル名=MMMMMMMM.YYYYMMDDSEQ.EXT	
MMMMMMMM	DLL関数パラメータのメッセージ識別子
YYYMMDD	システム日付
SEQ	メッセージ識別子毎の当日受信ファイルのカウンタ数(000~999)
EXT	転送制御マスターのファイル拡張子

・TDC標準形式/自由形式(ファイルタイプ=3/4)

差し出し≠'*'の場合

ファイル名=MMMMMMMM.YYYYMMDDSEQ.UUUUU.EXT	
MMMMMMMM	DLL関数パラメータのメッセージ識別子
YYYMMDD	システム日付
SEQ	メッセージ識別子毎の当日受信ファイルのカウンタ数(000~999)
UUUUU	DLL関数パラメータの差し出し
EXT	転送制御マスターのファイル拡張子

差し出し='*'の場合

ファイル名=MMMMMMMM.YYYYMMDDSEQ.EXT	
MMMMMMMM	DLL関数パラメータのメッセージ識別子
YYYMMDD	システム日付
SEQ	メッセージ識別子毎の当日受信ファイルのカウンタ数(000~999)
EXT	転送制御マスターのファイル拡張子

File_Recv_Api2(拡張ファイル受信)

センターに蓄積されているファイルを受信します。

【書式】

```
INT rc = File_Recv_Api2( rc_code      , /* ( 0 ) 処理結果          (必須) */
                        func_seq     , /* ( I/O) 処理シーケンスNo. (必須) */
                        async_code    , /* ( I ) 同期/非同期        (必須) */
                        file_cat      , /* ( I ) 受信ファイル統合モード (必須) */
                        file_code     , /* ( I ) メッセージ識別子    (必須) */
                        sashidashi_name , /* ( I ) 差し出し            (必須) */
                        recv_path     , /* ( I ) 受信ファイル書込みパス (必須) */
                        recv_ctl_file , /* ( I ) 受信ファイル情報書込みファイル名 (必須) */
                        recv_cnt_tbl  , /* ( 0 ) 受信ファイル件数テーブル (必須) */
                        apl_cnv       , /* ( I ) ヘッダ削除          (任意) */
                        user_id       , /* ( I ) ユーザーID          (任意) */
                        yobi          ); /* ( I ) 予備                (任意) */
```

【パラメーター】

パラメーター名	タイプ	桁数	表記方式	説明
rc_code	char*	6	"XXXXXX"	DLLでの処理結果をセットします。 前3桁は理由コード、後3桁は内部ロジックコードがセットされます。
func_seq	char*	8	"XXXXXXXX"	呼び出し元が、発行した各処理を管理するのに使用します。 非同期で処理要求を行なった場合は、該当する処理要求の状況確認を行う際に使用します。指定する場合は、8桁フル入力して下さい。 空白を指定した場合は、システム側で割り振りその値を返します。
async_code	char*	1	"1" 非同期&通常 "2" 同期&通常 "3" 非同期&優先 "4" 同期&優先	同期/非同期モードおよび優先/通常モードの確認に使用します。
file_cat	char*	1	"Y" 統合する "N" 統合しない	複数ファイルを受信した場合に統合するかどうかの確認に使用します。 処理モードが "Y" or "N"以外の場合は、エラーとして処理を中断します。
file_code	char*	8	"XXXXXXXX"	アプリケーションで規定したファイルを識別するコードを指定します。 メッセージ識別子コードがセットされていない場合、エラーとして処理を中断します。
sashidashi_name	char*	20	"KKKKKKKKUUUUUUUU"	指定された別名でアドレス帳マスタを検索し、ニックネームもしくは直接宛先がセットされます。 ※アドレス帳マスタに該当が無い場合は、引数を大文字にして宛先として使用します。 ただし、引数が4~8桁で、「@」「.」を含まない場合は 12345678 ⇒ 1234. 12345678 と加工して宛先とします。 先頭1バイト目'*'指定の場合、送信元ファイルのHレコードより検索します。 (TNS標準形式もしくはTNS拡張形式で、「ヘッダー作成しない」場合のみ) 宛先がセットされていない場合、エラーとして処理を中断します。
recv_path	char*	84		受信ファイルを書き込むパス 相対パス指定の場合、本システムの導入ドライブのルートディレクトリ(ドライブ: ¥) 配下のパスとして扱います。受信パスがセットされていない場合、エラーとして処理を中断します。 該当するパスが存在しない場合もエラーとして処理を中断します。
recv_ctl_file	char*	120		受信した各ファイルのヘッダー情報を書き込むファイル名をセットします。
recv_cnt_tbl	char*	12	"XXXXXXXXXXXX"	相対パス指定の場合、本DLL関数がCALLされたカレントパス配下に作成します。 受信したファイル件数を返します。
apl_cnv	char*	1	"Y" ヘッダ削除有り "N" ヘッダ削除無し	TNSヘッダの削除を行なうか否かを指定します。
user_id	char*	16	"XXXXXXXXXXXXXXXX"	ブランクを指定した場合、転送要求を行なったメッセージ識別子に基づいて定義された転送制御マスターデータの値を使用します。
yobi	char*	20	"XXXXXXXXXXXX"	転送制御マスターデータに基づく 利用するユーザーを指定します。 指定が無い場合、代表のユーザーIDを使用します。 システム予約。必ず空白を設定する必要があります。

【戻り値】

0 正常終了
0 以外 エラー ※
※戻り値の詳細は付録Dを参照してください。

尚、DLL関数の戻りコードが正常(0)、異常(0以外)を返したとしても、必ず、処理結果コードの値をもとにエラーハンドリングを行って下さい。

【解説】

任意のインプットパラメータに値がセットされていない場合、file_codeパラメータに定義された内容で転送制御マスターから求めますが、求めた転送制御マスターに各項目データの値が定義されていない場合、転送制御マスター定義エラーとして処理を中断します。
各インプットパラメータに値がセットされていた場合、セットされている値を優先し転送制御マスターの確認は行いません。
ファイルコード・差し出しに基づく自分宛のファイルが存在し場合のみ、受信処理を行います。
処理シーケンスを指定した場合、指定した処理シーケンスが過去に使用されていた場合、エラーとして処理を中断します。
非同期モードで処理要求を発行した場合、Event_List_Apiをコールし処理が完了したのを確認して下さい。

【補足説明】

受信ファイル名については、ファイル受信(File_Recv_Api)DLL関数の説明を参照して下さい。
受信ファイル情報書込みファイルのファイルレイアウトについては、付録C・ファイルレイアウトの説明を参照して下さい。

File_Recv_Ex_Api(保管取り出しファイル受信)

一旦受信したファイルの再受信を行います。

【書式】

```
INT rc = File_Recv_Ex_Api( rc_code      , /* ( 0 ) 処理結果          (必須) */
                          func_seq     , /* ( I ) 処理シーケンス   (必須) */
                          async_code    , /* ( I ) 同期/非同期       (必須) */
                          recv_path     , /* ( I ) 受信ファイル書込パス (必須) */
                          user_id       , /* ( I ) ユーザーID       (任意) */
                          yobi          ); /* ( I ) 予備              (任意) */
```

【パラメーター】

パラメーター名	タイプ	桁数	表記方式	説明
rc_code	char*	6	"XXXXXX"	DLLでの処理結果をセットします。 前3桁は理由コード、後3桁は内部ロジックコードがセットされます。
func_seq	char*	8	"XXXXXXXX"	過去にファイル受信を行った際に指定したシーケンスを設定します。
async_code	char*	1	"1" 非同期&通常 "2" 同期&通常 "3" 非同期&優先 "4" 同期&優先	同期/非同期モードおよび優先/通常モードの確認に使用します。
recv_path	char*	84		受信ファイルを書き込むパス 相対パス指定の場合、本システムの導入ドライブのルートディレクトリ(ドライブ:¥)配下のパスとして扱います。受信パスがセットされていない場合、エラーとして処理を中断します。 該当するパスが存在しない場合もエラーとして処理を中断します。
user_id	char*	16	"XXXXXXXXXXXXXXXXXX"	利用するユーザーを指定します。 指定が無い場合、代表のユーザーIDを使用します。
yobi	char*	20	"XXXXXXXXXXXXXXXXXX"	システム予約。必ず空白を設定する必要があります。

【戻り値】

0 正常終了

0以外 エラー ※

※戻り値の詳細は付録Dを参照してください。

尚、DLL関数の戻りコードが正常(0)、異常(0以外)を返したとしても、必ず、処理結果コードの値をもとにエラーハンドリングを行って下さい。

【解説】

再処理の対象となるファイル受信処理でセットした各種パラメーターを変更せずに、同一パラメーターにセットして下さい。

処理シーケンスデータに関しては、再処理の対象となったファイル受信処理でセットした値、もしくはDLL関数から返された値を必ずセットして下さい。処理シーケンスパラメーターにセットされた値に基づいて検索処理が行なわれます。

検索処理を行なった結果、該当する転送処理が存在しない場合、エラーとして処理を中断します。

非同期モードで処理要求を発行した場合、Event_List_Apiをコールし処理が完了したのを確認して下さい。

【補足説明】

受信ファイル名については、ファイル受信(File_Recv_Api)DLL関数の説明を参照して下さい。

File_Recv_Ex_Api2 (拡張保管取り出しファイル受信)

拡張ファイル受信で受信した受信ファイルの再受信を行います。

【書式】

```
INT rc = File_Recv_Ex_Api2( rc_code      , /* ( 0 ) 処理結果 (必須) */
                           hokan_key   , /* ( 1 ) 保管取り出しキー (必須) */
                           async_code  , /* ( 1 ) 同期/非同期 (必須) */
                           recv_path   , /* ( 1 ) 受信ファイル書込みパス (必須) */
                           recv_ctl_file, /* ( 1 ) 受信ファイル情報書込みファイル名 (必須) */
                           user_id     , /* ( 1 ) ユーザーID (任意) */
                           yobi        ); /* ( 1 ) 予備 (任意) */
```

【パラメーター】

パラメーター名	タイプ	桁数	表記方式	説明
rc_code	char*	6	"XXXXXX"	DLLでの処理結果をセットします。 前3桁は理由コード、後3桁は内部ロジックコードがセットされます。
hokan_key	char*	12	"XXXXXXXXXXXX"	拡張ファイル受信で返された個別再受信キーをセットします。
async_code	char*	1	"1" 非同期&通常 "2" 同期&通常 "3" 非同期&優先 "4" 同期&優先	同期/非同期モードおよび優先/通常モードの確認に使用します。
recv_path	char*	84		受信ファイルを書き込むパス 相対パス指定の場合、本システムの導入ドライブのルートディレクトリ(ドライブ: ¥) 配下のパスとして扱います。受信パスがセットされていない場合、エラーとして処理を中断します。 該当するパスが存在しない場合もエラーとして処理を中断します。
recv_ctl_file	char*	120		受信した各ファイルのヘッダー情報を書込むファイル名をセットします。 相対パス指定の場合、本DLL関数がCALLされたカレントパス配下に作成します。
user_id	char*	16	"XXXXXXXXXXXX"	利用するユーザーを指定します。 指定が無い場合、代表のユーザーIDを使用します。
yobi	char*	20	"XXXXXXXXXXXX"	システム予約。必ず空白を設定する必要があります。

【戻り値】

0 正常終了
0 以外 エラー ※
※戻り値の詳細は付録Dを参照してください。

尚、DLL関数の戻りコードが正常(0)、異常(0以外)を返したとしても、必ず、処理結果コードの値をもとにエラーハンドリングを行って下さい。

【解説】

再処理の対象となる拡張ファイル受信処理でセットした各種パラメーターを変更せずに、同一パラメーターをセットして下さい。
個別再受信キーデータに関しては、再処理の対象となった拡張ファイル受信処理で返された値をセットして下さい。
個別再受信キーパラメーターにセットされた値に基づいて検索処理が行なわれます。
検索処理を行なった結果、該当する転送処理が存在しない場合、エラーとして処理を中断します。
非同期モードで処理要求を発行した場合、Event_List_Apiをコールし処理が完了したのを確認して下さい。

【補足説明】

受信ファイル名については、ファイル受信 (File_Recv_Api) DLL関数の説明を参照して下さい。
受信ファイル情報書込みファイルのファイルレイアウトについては、付録C・ファイルレイアウトの説明を参照して下さい。
拡張ファイル受信で受信したファイルのみ、拡張保管取出しが行えます。
拡張ファイル受信で受信した際に、受信ファイル統合有り・ヘッダ削除有りで受信処理が行われた受信指示は、拡張保管取出しは行えません。

File_List_Api(蓄積状況照会)

センターに蓄積されている自分宛のファイル一覧の照会を行います。

【書式】

```
INT rc = File_List_Api( rc_code      , /* ( 0 ) 処理結果          (必須) */
                      func_seq     , /* (I/O) 処理シーケンス   (必須) */
                      async_code   , /* ( I ) 同期/非同期       (必須) */
                      file_code    , /* ( I ) メッセージ識別子  (必須) */
                      sashidasii_name , /* ( I ) 差し出し         (必須) */
                      recv_file    , /* ( I ) 受信ファイル     (必須) */
                      user_id      , /* ( I ) ユーザーID       (任意) */
                      yobi         ); /* ( I ) 予備             (任意) */
```

【パラメーター】

パラメーター名	タイプ	桁数	表記方式	説明
rc_code	char*	6	"XXXXXX"	DLLでの処理結果をセットします。 前3桁は理由コード、後3桁は内部ロジックコードがセットされます。
func_seq	char*	8	"XXXXXXXX"	呼出し元が、発行した各処理を管理するのに使用します。 非同期モードで処理要求を発行した場合、該当する処理の状況確認を行う際に使用します。指定する場合は、8桁フル入力して下さい。 空白を指定した場合は、システムが割り振り、その値を返します。
async_code	char*	1	"1" 非同期&通常 "2" 同期&通常 "3" 非同期&優先 "4" 同期&優先	同期/非同期モードおよび優先/通常モードの確認に使用します。
file_code	char*	8	"XXXXXXXX"	アプリケーションで規定したファイルを識別するコードを指定します。 ワイルドカードの指定は可能、指定する場合は先頭1バイト目に"*"をセット。
sashidashi_name	char*	20	"KKKKKKKKUUUUUUUU"	指定された別名でアドレス帳マスタを検索し、ニックネームもしくは直接宛先がセットされます。 ※アドレス帳マスタに該当が無い場合は、引数を大文字にして宛先として使用します。 ただし、引数が4~8桁で、「@」「.」を含まない場合は 12345678 ⇒ 1234. 12345678 と加工して宛先とします。 先頭1バイト目'*'指定の場合、送信元ファイルのHレコードより検索します。 (TNS標準形式もしくはTNS拡張形式で、「ヘッダー作成しない」場合のみ) 宛先がセットされていない場合、エラーとして処理を中断します。
recv_file	char*	120	絶対パス付きのファイル名	受信ファイルを書き込むパス 相対パス指定の場合、本システムの導入ドライブのルートディレクトリ(ドライブ:¥)配下のパスとして扱います。受信パスがセットされていない場合、エラーとして処理を中断します。 該当するパスが存在しない場合もエラーとして処理を中断します。
user_id	char*	16	"XXXXXXXXXXXXXXXX"	利用するユーザーを指定します。
yobi	char*	20	"XXXXXXXXXXXXXXXX"	指定が無い場合、代表のユーザーIDを使用します。 システム予約。必ず空白を設定する必要があります。

【戻り値】

0 正常終了
0以外 エラー ※
※戻り値の詳細は付録Dを参照してください。

尚、DLL関数の戻りコードが正常(0)、異常(0以外)を返したとしても、必ず、処理結果コードの値をもとにエラーハンドリングを行って下さい。

【解説】

非同期モードで処理要求を発行した場合、Event_List_Apiをコールし処理が完了したのを確認して下さい。
処理シーケンスを指定した場合、指定した処理シーケンスが過去に使用されていた場合、エラーとして処理を中断します。

Audit_List_Api(オーデットトレイル受信)

オーデットトレイルの受信処理を行います。

【書式】

```
INT rc = Audit_List_Api( rc_code      , /* ( 0 ) 処理結果 (必須) */
                        func_seq     , /* (I/O) 処理シーケンス (必須) */
                        async_code    , /* ( 1 ) 同期/非同期 (必須) */
                        from_date     , /* ( 1 ) 参照開始日付 (任意) */
                        to_date      , /* ( 1 ) 参照終了日付 (任意) */
                        msg_kubun     , /* ( 1 ) メッセージ区分 (任意) */
                        msg_status    , /* ( 1 ) メッセージ状態 (任意) */
                        recv_file     , /* ( 1 ) 受信ファイル (必須) */
                        user_id       , /* ( 1 ) ユーザーID (任意) */
                        yobi          , /* ( 1 ) 予備 (任意) */
                        );
```

【パラメーター】

パラメーター名	タイプ	桁数	表記方式	説明
rc_code	char*	6	"XXXXXX"	DLLでの処理結果をセットします。 前3桁は理由コード、後3桁は内部ロジックコードがセットされます。
func_seq	char*	8	"XXXXXXXX"	呼出し元が、発行した各処理を管理するのに使用します。 非同期モードで処理要求を発行した場合、該当する処理の状況確認を行う際に使用します。指定する場合は、8桁フル入力して下さい。 空白を指定した場合は、システムが割り振り、その値を返します。
async_code	char*	1	"1" 非同期&通常 "2" 同期&通常 "3" 非同期&優先 "4" 同期&優先	同期/非同期モードおよび優先/通常モードの確認に使用します。
from_date	char*	8	"YYYYMMDD"	メッセージのオーデットを取出す開始日付。 指定がない場合、19500101となります。
to_date	char*	8	"YYYYMMDD"	メッセージのオーデットを取出す終了日付。 指定がない場合、20991231となります。
msg_kubun	char*	1	"1" 送信メッセージ "2" 受信メッセージ "*" 全てのメッセージ	取出すメッセージの種類を指定します。 指定がない場合、すべてのメッセージを対象とします。
msg_status	char*	1	"1" 未受信状態 "2" 受信状態 "3" 受信前に削除された状態 "*" 全ての状態	取出すメッセージの状態を指定します。 指定がない場合、すべての状態を対象とします。
recv_file	char*	120	絶対パス付きのファイル名	受信ファイルを書き込むパス 相対パス指定の場合、本システムの導入ドライブのルートディレクトリ(ドライブ: ¥) 配下のパスとして扱います。受信パスがセットされていない場合、エラーとして処理を中断します。 該当するパスが存在しない場合もエラーとして処理を中断します。
user_id	char*	16	"XXXXXXXXXXXXXXXX"	利用するユーザーを指定します。 指定が無い場合、代表のユーザーIDを使用します。
yobi	char*	20	"XXXXXXXXXXXXXXXX"	システム予約。必ず空白を設定する必要があります。

【戻り値】

0 正常終了
0 以外 エラー ※
※戻り値の詳細は付録Dを参照してください。

尚、DLL関数の戻りコードが正常(0)、異常(0以外)を返したとしても、必ず、処理結果コードの値をもとにエラーハンドリングを行って下さい。

【解説】

非同期モードで処理要求を発行した場合、Event_List_Apiをコールし処理が完了したのを確認して下さい。
処理シーケンスを指定した場合、指定した処理シーケンスが過去に使用されていた場合、エラーとして処理を中断します。

Status_List_Api(状況確認照会)

IE/EXを使用したメッセージ変換処理の状況確認メッセージの受信を行います。

【書式】

```
INT rc = Status_List_Api( rc_code      , /* ( 0 ) 処理結果                (必須) */
                        func_seq     , /* (I/0) 処理シーケンス          (必須) */
                        async_code   , /* ( I ) 同期/非同期              (必須) */
                        recvfile     , /* ( I ) 受信ファイル            (必須) */
                        user_id      , /* ( I ) ユーザーID              (任意) */
                        yobi         ); /* ( I ) 予備                    (任意) */
```

【パラメーター】

パラメーター名	タイプ	桁数	表記方式	説明
rc_code	char*	6	"XXXXXX"	DLLでの処理結果をセットします。 前3桁は理由コード、後3桁は内部ロジックコードがセットされます。
func_seq	char*	8	"XXXXXXXX"	呼出し元が、発行した各処理を管理するのに使用します。 非同期モードで処理要求を発行した場合、該当する処理の状況確認を行う際に使用します。指定する場合は、8桁フル入力して下さい。 空白を指定した場合は、システムが割り振り、その値を返します。
async_code	char*	1	"1" 非同期&通常 "2" 同期&通常 "3" 非同期&優先 "4" 同期&優先	同期/非同期モードおよび優先/通常モードの確認に使用します。
recv_file	char*	120	絶対パス付きのファイル名	受信ファイルを書き込むパス 相対パス指定の場合、本システムの導入ドライブのルートディレクトリ(ドライブ: ¥) 配下のパスとして扱います。受信パスがセットされていない場合、エラーとして処理を中断します。 該当するパスが存在しない場合もエラーとして処理を中断します。
user_id	char*	16	"XXXXXXXXXXXXXXXX"	利用するユーザーを指定します。 指定が無い場合、代表のユーザーIDを使用します。
yobi	char*	20	"XXXXXXXXXXXXXXXX"	システム予約。必ず空白を設定する必要があります。

【戻り値】

0 正常終了
0 以外 エラー ※
※戻り値の詳細は付録Dを参照してください。

尚、DLL関数の戻りコードが正常(0)、異常(0以外)を返したとしても、必ず、処理結果コードの値をもとにエラーハンドリングを行って下さい。

【解説】

非同期モードで処理要求を発行した場合、Event_List_Apiをコールし処理が完了したのを確認して下さい。
処理シーケンスを指定した場合、指定した処理シーケンスが過去に使用されていた場合、エラーとして処理を中断します。

Event_List_Api(処理状況問合せ)

パッケージへ処理要求を通知した各処理状況の状況照会を行います。

【書式】

```
INT rc = Event_List_Api( rc_code    , /* ( 0 ) 処理結果          (必須) */
                       info_code   , /* ( 0 ) 処理状況          (必須) */
                       user_id     , /* ( I ) ユーザー I D      (任意) */
                       func_seq    ); /* ( I ) 処理シーケンス    (必須) */
```

【パラメーター】

パラメーター名	タイプ	桁数	表記方式	説明
rc_code	char*	6	"XXXXXX"	DLLでの処理結果をセットします。 前3桁は理由コード、後3桁は内部ロジックコードがセットされます。
info_code	char*	4	"XXXX"(10進数)	処理状況コードをセットします。
user_id	char*	16	"XXXXXXXXXXXXXXXX"	利用するユーザーを指定します。 指定が無い場合、代表のユーザーIDを使用します。
func_seq	char*	8	"XXXXXXXX"	各機能要求時に指定した処理シーケンスを指定します。

【戻り値】

0 正常終了
0 以外 エラー ※
※戻り値の詳細は付録Dを参照してください。

【解説】

処理完了分の詳細な状況は、処理結果(rc_code)の内容により判定します。
該当処理シーケンスが存在しない場合には、次のように戻り値がセットされます。

```
戻り値       (rc       ) ≠ 0  
処理結果     (rc_code ) = "137000"  
処理状況コード (info_code) = 空白
```

DLL関数呼び出しサンプル

各言語毎のDLL関数呼び出しサンプルです。

C

Tool/C/ H/EDI_USER.H	プロトタイプ宣言ヘッダーファイル
Lib/x64/EDI_USER.lib	x64用インポートライブラリ
Lib/x86/EDI_USER.lib	x86用インポートライブラリ
Sample/SAMPLE.C	サンプルコード

VB.NET

Tool/VB/ EDIGCOM.vb	モジュールファンクションプロシージャ宣言
VBSample/Sample.vb	サンプルコード
VBSample/VBSample.vbproj	VisualStudio用プロジェクトファイル

VB.NETでの注意事項

API起動時に変数内の値が最大桁数未満の場合には、変数へNULL(Chr(0))が設定されている必要がある為、サンプル・プログラムにおいて、送信ファイル名変数へNULL(Chr(0))を設定しております。この変数をお客様の開発されたアプリにおいて使用した場合に下記の事象が発生する可能性がありますので、ご注意下さい。

【事象】

ユーザーアプリ開発時に、NULL(Chr(0))が設定されている送信ファイル名変数を用いて、ファイル操作処理(ファイルの読み込み・書き込み等)を実施すると、ファイル無しエラーが発生する。
※ NULL(Chr(0))まで、ファイルパスの一部であると認識されてしまう為。

【対応方法】

ファイル操作処理実施前に送信ファイル名変数よりNULL(Chr(0))を削除する事で、正常にファイル操作処理を実施することが可能です。

第3部 コマンド機能

第5章 コマンド機能概要

EXEファイル

EDI¥Tool¥Cmd

本パッケージのコマンド機能実行ファイルは EDISEND.EXE、EDIRECV.EXE という名称です。

このEXEファイルは、本パッケージインストール時に、EDI¥Tool¥Cmd 配下に

64ビット版と32ビット版のそれぞれがコピーされます。

コマンド機能一覧

提供するコマンド機能は以下の通りです。

ファイル送信コマンド

ユーザーアプリケーション・ファイルを指定宛先へ送信するために使用します。

EDISEND.EXE

ユーザーアプリケーション・ファイルを指定宛先へ送信します。

ファイル受信コマンド

自分宛に送られているファイルを受信するために使用します。

EDIRECV.EXE

自分宛に送られているファイルを受信します。

第6章 コマンド操作

コマンド機能実行形態

各コマンドは引数とともにEXEファイルを実行することで動作します。
EXEファイルが実行されると、引数をパラメータとして各種DLL関数がコールされ、DLL関数の実行結果に応じて、コマンドの戻り値が設定されます。

コールされるDLL

- EDISEND.EXE ... File_Send_Api
- EDIRECV.EXE ... File_Recv_Api

コマンド機能インターフェース

EDISEND.EXE (ファイル送信)

センターへ指定ファイルの通常送信、及び上書き再送信を行います。

```
EDISEND { file_code } { atesaki_name } { file_name } [-e:{ file_ext }][ -s? ][ -h? ][ -f? ]
        [-m? ][ -a? ][ -c? ][ -r:{ apl_len } ][ -u:{ user_id } ]
```

【パラメーター】

パラメーター名	必須	名称	最大桁数	説明
file_code	○	メッセージ識別子	8	アプリケーションで規定したファイルを識別するコードを指定します。
atesaki_name	○	宛先	16	別名、ニックネーム、テーブル名、顧客コード(8桁)+ユーザーID(8桁) 先頭1バイト目*指定の場合、送信元ファイルのHレコードより検索します
file_name	○	送信ファイルパス	128	送信ファイル名称を指定します。 フルパスで指定してください。
-e:{ file_ext }	-	送信ファイル拡張子	3	ファイル拡張子として使用します。指定が無い場合、file_codeに基づきメッセージマスタから求めます。
-s?	-	同期 / 非同期	-	同期/非同期モードおよび優先/通常モードの確認に使用します。 -s1 非同期&通常 -s2 同期&通常(デフォルト) -s3 非同期&優先 -s4 同期&優先
-h?	-	ヘッダ作成	-	TNSヘッダを作成するか否かを指定します。 -hy ヘッダ作成する -hn ヘッダ作成しない(デフォルト)
-f?	-	ファイル種別	-	ファイル種別を確認するのに使用します。指定が無い場合、file_codeに基づきメッセージマスタから求めます。 -f1 TNS標準形式 -f2 TNS拡張形式 -f3 TDC標準形式 -f4 自由形式
-m?	-	圧縮	-	データ圧縮の確認に使用します。指定が無い場合、file_codeに基づきメッセージマスタから求めます。 -my 圧縮有り -mn 圧縮無し
-a?	-	暗号化	-	データ暗号化の確認に使用します。指定が無い場合、file_codeに基づきメッセージマスタから求めます。 -ay 暗号化有り -an 暗号化無し
-c?	-	文字コード変換	-	コード変換の確認に使用します。指定が無い場合、file_codeに基づきメッセージマスタから求めます。 -cy コード変換有り -cn コード変換無し
-r:{ apl_len }	-	APL有効長	5	APL有効長の確認に使用します。指定する場合は、5桁フル入力して下さい。指定が無い場合、file_codeに基づきメッセージマスタから求めます。
-u:{ user_id }	-	利用者名称	16	利用するユーザーを指定します。指定が無い場合、代表ユーザーIDを使用します。

※パラメータ識別子「-」は「/」で代用可能です。

※「:」は省略可能です。

【戻り値】

- 0:正常 APIで正常(RET=[0] で RC_CODE=[000000])の場合
- 2:APIエラー APIで異常(RET=[0]以外か RC_CODE=[000000]以外)の場合
- 3:その他エラー 上記以外のパラメータエラー等

【使用例】

①必須パラメータのみ指定

EDISEND TNS01 "9999" C:¥WORK¥SendData.txt

- ⇒宛先「9999」へファイル「SendData.txt」をメッセージ識別子「TNS01」で「同期&通常」モードで送信
- その他パラメータはメッセージマスタを検索してセットされる

②任意パラメータを指定

EDISEND TNSEX01 "*" C:¥WORK¥REC1000.txt -e:txt -s4 -hn -f2 -mn -an -cn -r:01000 -u:USERID

- ⇒ファイル「REC1000.txt」をメッセージ識別子「TNSEX01」で送信

- 宛先 「REC1000.txt」のヘッダレコード内より取得
- 拡張子 txt
- 送信モード 同期&優先
- ヘッダ作成 しない
- ファイル種別 TNS拡張形式
- 圧縮 しない
- 暗号化 しない
- 文字コード変換 しない
- アプリ有効長 1000
- 利用者名称 USERID

EDIRECV.EXE (ファイル受信)

センターに蓄積されているファイルを受信します。

EDIRECV { file_code } { sasidasi_name } { recv_path } [-s?] [-c?] [-u:{ user_id }]

【パラメーター】

パラメーター名	必須	名称	最大桁数	説明
file_code	○	メッセージ識別子	8	アプリケーションで規定したファイルを識別するコードを指定します。
sasidasi_name	○	差し出し元	16	別名、ニックネーム、テーブル名、顧客コード(8桁)+ユーザーID(8桁) 先頭1バイト目*指定の場合、差し出し元を特定しません。
recv_path	○	受信ファイル書き込みパス	128	受信ファイルを書き込むパスを指定します。フルパスで指定してください。
-s?	-	同期 / 非同期	-	同期/非同期モードおよび優先/通常モードの確認に使用します。 -s1 非同期&通常 -s2 同期&通常(デフォルト) -s3 非同期&優先 -s4 同期&優先
-c?	-	受信ファイル統合モード	-	複数ファイルを受信した場合に統合するかどうかの確認に使用します。 -cy 統合する -cn 統合しない(デフォルト)
-u:{ user_id }	-	利用者名称	16	利用するユーザーを指定します。指定が無い場合、代表ユーザーIDを使用します。

※パラメータ識別子「-」は「/」で代用可能です。

※「:」は省略可能です。

【戻り値】

- 0: 正常 APIで正常(RET=[0] で RC_CODE=[000000])の場合
- 1: データ無し APIでデータ無し(RC_CODE=[90????])の場合
- 2: APIエラー APIで異常(RET=[0]以外か RC_CODE=[000000],[90????]以外)の場合
- 3: その他エラー 上記以外のパラメータエラー等

【使用例】

①必須パラメータのみ指定

EDIRECV TNS01 "*" C:\WORK

⇒ メッセージ識別子「TNS01」の受信可能なファイルを全て「同期&通常」モードで受信
受信したファイルはファイル統合されずに「C:\WORK」フォルダ配下へ配置される

②任意パラメータを指定

EDIRECV FREE "9999" C:\WORK -s4 -cy -u:USERID

⇒ 差し出し元「9999」からメッセージ識別子「FREE」のファイルを「同期&優先」モードで受信
受信したファイルは「C:\WORK」フォルダ配下へファイル統合された状態で配置される

ログ出力

各コマンドを実行した場合、標準出力及び標準エラーにメッセージを出力します。
ただし、下記パラメータを追記した場合、出力されるログの情報及び出力先を変更することが可能となります。

{コマンド} [-d?] [-l:{ log_file }]

【パラメーター】

パラメーター名	必須	名称	最大桁数	説明
-d?	-	ログ出力レベル	long型の桁数	出力レベルに応じてログファイルに出力される内容が決まります。 -d0 ログ出力なし -d1 エラーのみ出力 -d2 処理状況出力(デフォルト) -d3 デバッグ情報出力 (3以上の値を設定した場合は-d3と同じ動作となります。)
-l:{ log_file }	-	ログファイルパス	511	標準出力、標準エラーに出力されるメッセージをログファイルに出力する場合に指定します。フルパスで指定してください。 指定したパスにファイルが存在しない場合は、ファイルを新規作成します。ファイルが存在する場合は、ファイル末尾に追加書き込みします。

【使用例】

① デバッグ情報をログファイルに出力する場合

EDIRECV TNS01 "*" C:¥WORK -d3 -l:C:¥Work¥tnslog.log
⇒ エラー、処理状況、デバッグ情報の全てをログファイル「tnslog.log」へ出力します。

② ログレベルに応じて出力先を変更する場合

EdiRecv TNS01 "*" C:¥WORK -d2 1>>C:¥Work¥tnslog_info.log 2>>C:¥Work¥tnslog_err.log
⇒ 標準出力内容(処理状況)をログファイル「tnslog_info.log」へ
標準エラー出力内容(エラーログ)をログファイル「tnslog_err.log」へ出力します。

【出力例】

① 標準出力への出力(処理開始、正常終了)

```
2014/10/01 10:01:02,Inf,EDISEND Start! FILE_CD=[TNS01] FILE_NAME=[C:¥Work¥Senddata.txt]
2014/10/01 10:01:02,Inf,EDISEND OK End! RET=[0] RC_CODE=[000000] FUNC_SEQ=[12345678]

2014/10/01 10:01:02,Inf,EDIRECV Start! FILE_CD=[TNS01] RECV_PATH=[C:¥Work]
2014/10/01 10:01:02,Inf,EDIRECV OK End! RET=[0] RC_CODE=[000000] FUNC_SEQ=[12345678]
```

② 標準エラーへの出力(異常終了)

```
2014/10/01 10:01:02,Err,EDIRECV NG End! RET=[0] RC_CODE=[140022] FUNC_SEQ=[12345678]
2014/10/01 10:01:02,Err,EDIRECV NG End! RET=[0] RC_CODE=[140022] FUNC_SEQ=[12345678]
```

処理状況コード一覧

処理状況コード	処理フェーズ	説明
0104	JOB管理	処理受付リストからの取り出し待機中。
0105	JOB管理	転送処理フェーズに移行しました。
0201	ダイヤルアップ	ダイヤル接続中です。(ダイヤル接続が完了していた場合、この処理フェーズは飛ばします)
0301	管理サービス	認証サーバーへのログオン中です。(認証サーバーへのログオンが完了していた場合、この処理フェーズは飛ばします)
0303	管理サービス	認証サーバーへアクセスリクエストを発行中です。
0401	FTPサービス (制御線スレッド)	G/Wとのセッションを確立中です。
0404	FTPサービス (転送線スレッド)	G/Wとの転送処理は完了しました。
0411	FTPサービス(u-DIEX) (制御線スレッド)	G/Wとのセッションを確立中です。(パスワード変更処理)
0421	FTPサービス(u-DIEX) (制御線スレッド)	G/Wとのセッションを確立中です。
0424	FTPサービス(u-DIEX) (制御線スレッド)	G/Wとの転送処理は完了しました。
0901 0909	JOB管理	転送要求処理は完了しました。

付録C 受信ファイル情報書込みファイルのファイルレイアウト

通知する項目／レイアウトは次の通り。

フォーマット区分	ヘッダレコードのファイルタイプ	X(01)
発信元	ヘッダレコードの発信元	X(16)
宛先	ヘッダレコードの宛先	X(16)
メッセージ識別子	受信したメッセージ識別子	X(08)
日付	ヘッダの日付	X(08)
SEQ	ヘッダのSEQ	X(03)
件数	トレーラの件数	X(07)
個別再受信キー	システム側で割り振ったユニークな値	X(12)
ファイル名	受信したファイルの物理ファイル名	X(45)

	ファイルタイプ	発信元	宛先	メッセージ識別子	日付	SEQ	件数	個別再受信キー	ファイル名
TNS標準	1	T1000 (残りブランク11)	O3101 (残りブランク11)	ITNSMO1△	000306△△ ※1	03△ ※2	00023△△	00000010001	AAAA.DAT (残りブランク)
TNS拡張	2	T1000003 ※1 (残りブランク8)	O3101001 ※3 (残りブランク8)	ITNSMO1△	20000306	003 ※2	0000023	00000010001	AAAA.DAT (残りブランク)
TDC標準	3	OTNS△△△△ OTNS001△	OTNSD△△△ OTNS038△	ITNSAAZ1	ブランク	ブランク	ブランク	00000010001	AAAA.DAT (残りブランク)
自由	4	OTNS△△△△ OTNS001△	OTNSD△△△ OTNS038△	ITNSAAZ1	ブランク	ブランク	ブランク	00000010001	AAAA.DAT (残りブランク)

※1 TNS標準は作成世紀を含まない、TNS拡張は作成世紀(2桁)＋作成日付(6桁:YYMMDD)

※2 TNS標準は拡張SEQを含まない、TNS拡張は拡張SEQ(1桁)＋SEQ(2桁)

※3 アドレス(5桁)＋サブアドレス(3桁)

付録D DLL関数の戻り値一覧

関数名	戻り値	戻り値内容	エラーが発生するケース
File_Send_Api(ファイル送信)	0	正常終了	
	336	利用者ユーザーIDエラー	ユーザIDの取得に失敗
	272	パラメータエラー (処理結果)	処理結果エリアにNULLをセット
	274	パラメータエラー (処理シーケンス)	処理シーケンスにNULLをセット または処理シーケンスの先頭<' '
	313	処理シーケンスエラー ※非上書き送信	送信モードエリア=NULLをセット または送信モードの先頭≠'Y' かつ、処理シーケンスを使用済み
	277	パラメータエラー (同期/非同期)	同期/非同期エリアにNULLをセット または同期/非同期コード<1 または同期/非同期コード>4
	275	パラメータエラー (ヘッダ作成)	ヘッダ作成エリアにNULLをセット またはAPLコンバートの先頭≠'Y' またはAPLコンバートの先頭≠'N'
	278	パラメータエラー (メッセージ識別子)	メッセージ識別子にNULLをセット または送信メッセージ識別子の先頭<' '
	279	パラメータエラー (宛先)	宛先にNULLをセット または宛先コード=¥0' または (宛先コードの先頭<'='かつ宛先コード[1]<'=')
	287	パラメータエラー (上書き送信指定)	上書き送信指定エリアにNULLをセット または (送信モードの先頭≠'Y'かつ送信モードの先頭≠'N')
	280	パラメータエラー (送信ファイルパス)	送信ファイルパスにNULLをセット または送信ファイル名の先頭<' '
	281	パラメータエラー (ファイル種別)	ファイル種別エリアに無効な値([1-4]以外)を指定
	282	パラメータエラー (圧縮)	TNS拡張形式もしくはTDC標準形式 (ファイル種別='2'または'3') かつ 圧縮指定エリアに無効な値('Y','N','0','1','2'以外)を指定
	283	パラメータエラー (暗号化)	TNS拡張形式もしくはTDC標準形式 (ファイル種別='2'または'3') かつ 暗号化指定エリアに無効な値('Y','N'以外)を指定
	284	パラメータエラー (コード変換)	TNS標準形式、TNS拡張形式、TDC標準形式のいずれか (ファイル種別='1'または'2'または'3') かつ コード変換指定エリアに無効な値('Y','N'以外)を指定
	279	ヘッダ作成指定エラー	宛先='*' かつ (ファイル種別='1'または'2') またはAPLコンバートの先頭='Y'
	322	アドレス帳マスタ検索エラー	宛先≠'*' かつ アドレス帳マスタ検索に失敗
	296	送信ファイル取得エラー	フルパス付送信ファイルが存在しない かつ 拡張子なしフルパス付送信ファイルが存在しない
	307	転送制御ログ取得エラー	送信モード='Y' かつ転送制御ログファイル名の取得に失敗 送信モード='Y' かつ保管ファイル名の取得に失敗
	311	上書き送信エラー	送信モード='Y' かつ送信済みデータの検索に成功 かつ該当なし
	307	転送制御ログ取得エラー	送信モード='Y' かつ送信済みデータの検索に失敗
	321	キューアクセスエラー	データキュー作成処理に失敗 要求通知後、キューの読み込みに失敗
320	処理要求通知エラー	キュー作成後、プロセス間通知に失敗	

File_Send_Ex_Api	0	正常終了	
	336	利用者ユーザーIDエラー	ユーザIDの取得に失敗
	272	パラメータエラー (処理結果)	処理結果にNULLをセット
	274	パラメータエラー (処理シーケンス)	処理シーケンスにNULLをセット または処理シーケンスの先頭<' '
	277	パラメータエラー (同期/非同期)	同期/非同期エリアにNULLをセット または同期/非同期コード<1 または同期/非同期コード>4
	275	パラメータエラー (ヘッダ作成)	ヘッダ作成エリアにNULLをセット またはAPLコンバートの先頭≠'Y' またはAPLコンバートの先頭≠'N'
	278	パラメータエラー (メッセージ識別子)	メッセージ識別子にNULLをセット または送信メッセージ識別子の先頭<' '
	279	パラメータエラー (宛先)	宛先にNULLをセット または宛先コード='*0' または (宛先コードの先頭<' 'かつ宛先コード[1]<' ')
	280	パラメータエラー (送信ファイルパス)	送信ファイルパスにNULLをセット または送信ファイル名の先頭<' '
	281	パラメータエラー (ファイル種別)	ファイル種別エリアに無効な値([1-4]以外)を指定
	282	パラメータエラー (圧縮)	TNS拡張形式もしくはTDC標準形式 (ファイル種別='2'または'3') かつ 圧縮指定エリアに無効な値('Y','N','0','1','2'以外)を指定
	283	パラメータエラー (暗号化)	TNS拡張形式もしくはTDC標準形式 (ファイル種別='2'または'3') かつ 暗号化指定エリアに無効な値('Y','N'以外)を指定
	284	パラメータエラー (コード変換)	TNS標準形式、TNS拡張形式、TDC標準形式のいずれか (ファイル種別='1'または'2'または'3') かつ コード変換指定エリアに無効な値('Y','N'以外)を指定
	279	ヘッダ作成指定エラー	宛先='*' かつ (ファイル種別='1'または'2') またはAPLコンバートの先頭='Y'
	322	アドレス帳マスタ検索エラー	宛先≠'*' かつ アドレス帳マスタ検索に失敗
	307	転送制御ログ取得エラー	送信モード='Y' かつ転送制御ログファイル名の取得に失敗 送信モード='Y' かつ保管ファイル名の取得に失敗
	311	上書き送信エラー	送信モード='Y' かつ送信済みデータの検索に成功 かつ該当なし
	307	転送制御ログ取得エラー	送信モード='Y' かつ送信済みデータの検索に失敗
321	キューアクセスエラー	データキュー作成処理に失敗 要求通知後、キューの読み込みに失敗	
320	処理要求通知エラー	キュー作成後、プロセス間通知に失敗	

File_Recv_Api	0	正常終了	
	336	利用者ユーザーIDエラー	ユーザIDの取得に失敗
	272	パラメータエラー (処理結果)	処理結果にNULLをセット
	274	パラメータエラー (処理シーケンス)	処理シーケンスにNULLをセット または処理シーケンスの先頭<' '
	313	処理シーケンス重複エラー	指定もしくは生成された処理シーケンスが、既に使用済み
	277	パラメータエラー (同期/非同期)	同期/非同期エリアにNULLをセット または同期/非同期コード<1 または同期/非同期コード>4
	276	パラメータエラー (ファイル結合)	ファイル結合エリアにNULLをセット または 無効な値('Y','N'以外)をセット
	278	パラメータエラー (メッセージ識別子)	メッセージ識別子にNULLをセット または送信メッセージ識別子の先頭<=' '
	279	パラメータエラー (差し出し)	差し出し元指定エリアにNULLをセット または差し出し元='¥0' または (差し出し元の先頭<='かつ差し出し元[1]<=')
	286	パラメータエラー (受信ファイルパス)	受信ファイルパスにNULLをセット または受信ファイルパスの先頭<=' '
	295	受信ファイルパスエラー	指定された受信ファイルパスがディレクトリではない かつ (ファイル統合='N' または メッセージ識別子='*') 指定された受信ファイルパスがディレクトリではない かつ いずれかのディレクトリ配下にあるパスを指定していない 指定された受信ファイルパスのファイル属性取得に失敗
	338	ワークファイルパス取得エラー	メッセージ識別子='*' かつ iniファイルからテンポラリファイルパス取得に失敗 メッセージ識別子='*' かつ 蓄積照会で取得したファイルリストから メッセージ識別子の検索を行う処理でエラーが発生
	304	メッセージマスタ検索エラー	メッセージ識別子≠'*' かつ メッセージマスタ検索に失敗(該当無し以外のエラー)
	288		メッセージ識別子≠'*' かつ メッセージマスタ検索に該当無し
	322	アドレス帳検索エラー	宛先≠'*' かつ アドレス帳マスタ検索に失敗
321	キューアクセスエラー	データキュー作成処理に失敗 要求通知後、キューの読み込みに失敗	
320	処理要求通知エラー	キュー作成後、プロセス間通知に失敗	
File_Recv_Ex_Api	0	正常終了	
	336	利用者ユーザーIDエラー	ユーザIDの取得に失敗
	272	パラメータエラー (処理結果)	処理結果にNULLをセット
	274	パラメータエラー (処理シーケンス)	処理シーケンスにNULLをセット または処理シーケンスの先頭<' '
	277	パラメータエラー (同期/非同期)	同期/非同期エリアにNULLをセット または同期/非同期コード<1 または同期/非同期コード>4
	286	パラメータエラー (受信ファイルパス)	受信ファイルパスにNULLをセット または受信ファイルパスの先頭<=' '
	295	受信ファイルパスエラー	指定された受信ファイルパスで、ディレクトリの 存在チェックを行いエラーもしくはディレクトリが存在しない
	307	転送制御ログ取得エラー	iniファイルから転送制御ログパスの取得に失敗
	311	シーケンスコードエラー	指定された処理シーケンスで受信処理を行っていない
	307	転送制御ログ検索エラー	指定された処理シーケンスでの受信処理検索でエラー
	304	メッセージマスタ検索エラー	メッセージ識別子≠'*' かつ メッセージマスタ検索に失敗(該当無し以外のエラー)
	288		メッセージ識別子≠'*' かつ メッセージマスタ検索に該当無し
	321	キューアクセスエラー	データキュー作成処理に失敗 要求通知後、キューの読み込みに失敗
	320	処理要求通知エラー	キュー作成後、プロセス間通知に失敗

File_Recv_Api2	0	正常終了	
	336	利用者ユーザーIDエラー	ユーザIDの取得に失敗
	337	システムパッケージ種別エラー	iniファイルのEDI CLIMODEが'ON'
	272	パラメータエラー (処理結果)	処理結果にNULLをセット
	274	パラメータエラー (処理シーケンス)	処理シーケンスにNULLをセット または処理シーケンスの先頭<' '
	313	処理シーケンス重複エラー	指定もしくは生成された処理シーケンスが、既に使用済み
	277	パラメータエラー (同期/非同期)	同期/非同期エリアにNULLをセット または同期/非同期コード<1 または同期/非同期コード>4
	276	パラメータエラー (ファイル結合)	ファイル結合エリアにNULLをセット または 無効な値('Y','N'以外)をセット
	278	パラメータエラー (メッセージ識別子)	メッセージ識別子にNULLをセット または送信メッセージ識別子の先頭<' '
	279	パラメータエラー (差し出し)	差し出し元指定エリアにNULLをセット または差し出し元='¥0' または (差し出し元先頭<' 'かつ差し出し元[1]<=' ')
	286	パラメータエラー (受信ファイルパス)	受信ファイルパスにNULLをセット または受信ファイルパス先頭<' '
	295	受信ファイルパスエラー	指定された受信ファイルパスで、ディレクトリの 存在チェックを行いエラーもしくはディレクトリが存在しない
	297	パラメータエラー (受信ファイル情報書き込みパス)	受信ファイル情報書き込みファイルパスが無効 (NULL,' 'ファイルが存在しない)
	298	パラメータエラー (受信ファイル件数)	受信ファイル件数にNULLをセット
	299	パラメータエラー (ヘッダ削除)	ヘッダ削除指定エリアにNULLをセット または 無効な値(' ','Y','N'以外)をセット
	304	メッセージマスタ検索エラー	メッセージ識別子≠* かつ メッセージマスタ検索に失敗(該当無し以外のエラー)
	288		メッセージ識別子≠* かつ メッセージマスタ検索に該当無し
	322	アドレス帳検索エラー	宛先≠* かつ アドレス帳マスタ検索に失敗
	321	キューアクセスエラー	データキュー作成処理に失敗 要求通知後、キューの読み込みに失敗
	320	処理要求通知エラー	キュー作成後、プロセス間通知に失敗
File_Recv_Ex_Api2	0	正常終了	
	336	利用者ユーザーIDエラー	ユーザIDの取得に失敗
	337	システムパッケージ種別エラー	iniファイルのEDI CLIMODEが'ON'
	272	パラメータエラー (処理結果)	処理結果にNULLをセット
	300	パラメータエラー (保管取り出しキー)	保管取り出しキーにNULLをセット
	277	パラメータエラー (同期/非同期)	同期/非同期エリアにNULLをセット または同期/非同期コード<1 または同期/非同期コード>4
	286	パラメータエラー (受信ファイルパス)	受信ファイルパスにNULLをセット または受信ファイルパス先頭<' '
	295	受信ファイルパスエラー	指定された受信ファイルパスで、ディレクトリの 存在チェックを行いエラーもしくはディレクトリが存在しない
	297	パラメータエラー (受信ファイル情報書き込みパス)	受信ファイル情報書き込みファイルパスが無効 (NULL,' 'ファイルが存在しない)
	307	転送制御ログ取得エラー	iniファイルから転送制御ログパスの取得に失敗
	311	シーケンスコードエラー	指定された処理シーケンスで受信処理を行っていない
	307	転送制御ログ検索エラー	指定された処理シーケンスでの受信処理検索でエラー
	302	要求元データ種別エラー	ファイル取り出しの対象が「ファイル統合有り」で受信済み
	304	メッセージマスタ検索エラー	メッセージ識別子≠* かつ メッセージマスタ検索に失敗(該当無し以外のエラー)
	288		メッセージ識別子≠* かつ メッセージマスタ検索に該当無し
	321	キューアクセスエラー	データキュー作成処理に失敗 要求通知後、キューの読み込みに失敗
	320	処理要求通知エラー	キュー作成後、プロセス間通知に失敗

File_List_Api	0	正常終了	
	336	利用者ユーザーIDエラー	ユーザIDの取得に失敗
	272	パラメータエラー (処理結果)	処理結果にNULLをセット
	274	パラメータエラー (処理シーケンス)	処理シーケンスにNULLをセット または処理シーケンスの先頭<' '
	313	処理シーケンス重複エラー	指定もしくは生成された処理シーケンスが、既に使用済み
	277	パラメータエラー (同期/非同期)	同期/非同期エリアにNULLをセット または同期/非同期コード<1 または同期/非同期コード>4
	278	パラメータエラー (メッセージ識別子)	メッセージ識別子にNULLをセット または送信メッセージ識別子の先頭<=' '
	279	パラメータエラー (差し出し)	差し出し元指定エリアにNULLをセット または差し出し元='¥0' または (差し出し元の先頭<=' 'かつ差し出し元[1]<=' ')
	286	パラメータエラー (受信ファイルパス)	受信ファイルパスにNULLをセット または 受信ファイルパスの先頭<=' '
	295	パスエラー	指定された受信ファイル情報書き込みパスの ディレクトリが存在しない
	304	メッセージマスタ検索エラー	メッセージ識別子≠'*' かつ メッセージマスタ検索に失敗(該当無し以外のエラー)
	288		メッセージ識別子≠'*' かつ メッセージマスタ検索に該当無し
	Audit_List_Api	322	アドレス帳検索エラー
321		キューアクセスエラー	データキュー作成処理に失敗 要求通知後、キューの読み込みに失敗
320		処理要求通知エラー	キュー作成後、プロセス間通知に失敗
0		正常終了	
336		利用者ユーザーIDエラー	ユーザIDの取得に失敗
272		パラメータエラー (処理結果)	処理結果にNULLをセット
274		パラメータエラー (処理シーケンス)	処理シーケンスにNULLをセット または処理シーケンスの先頭<' '
313		処理シーケンス重複エラー	指定もしくは生成された処理シーケンスが、既に使用済み
277		パラメータエラー (同期/非同期)	同期/非同期エリアにNULLをセット または同期/非同期コード<1 または同期/非同期コード>4
496		パラメータエラー (開始日付)	開始日付=NULL
497		パラメータエラー (終了日付)	終了日付=NULL
498		開始日付が数字コード以外	開始日付入力あり、かつ 入力値が数値でない
499		開始日付範囲エラー	開始日付入力あり、かつ 入力値が19500101以前
500	終了日付が数字コード以外	終了日付入力あり、かつ 入力値が数値でない	
501	終了日付範囲エラー	終了日付入力あり、かつ 入力値が20991231以後	
502	開始日付が無効	月[1-12],日[1-28,29,30,31](各月に応じた範囲)以外	
503	終了日付が無効	月[1-12],日[1-28,29,30,31](各月に応じた範囲)以外	
504	開始日付範囲エラー(未来)	開始日付が未来日を指定している	
505	検索期間エラー	開始日付 > 終了日付	
286	パラメータエラー (受信ファイルパス)	受信ファイルパスにNULLをセット または 受信ファイルパス先頭<=' '	
295	パスエラー	指定された受信ファイルパスのディレクトリが存在しない	
321	キューアクセスエラー	データキュー作成処理に失敗 要求通知後、キューの読み込みに失敗	
320	処理要求通知エラー	キュー作成後、プロセス間通知に失敗	

Status_List_Api	0	正常終了	
	336	利用者ユーザーIDエラー	ユーザIDの取得に失敗
	272	パラメータエラー (処理結果)	処理結果にNULLをセット
	274	パラメータエラー (処理シーケンス)	処理シーケンスにNULLをセット または処理シーケンスの先頭<' '
	313	処理シーケンス重複エラー	指定もしくは生成された処理シーケンスが、既に使用済み
	277	パラメータエラー (同期/非同期)	同期/非同期エリアにNULLをセット または同期/非同期コード<1 または同期/非同期コード>4
	286	パラメータエラー (受信ファイルパス)	受信ファイルパスにNULLをセット または 受信ファイルパス先頭<= '
	295	パスエラー	指定された受信情報書き込みパスの ディレクトリが存在しない
	321	キューアクセスエラー	データキュー作成処理に失敗 要求通知後、キューの読み込みに失敗
	320	処理要求通知エラー	キュー作成後、プロセス間通知に失敗
Event_List_Api	0	正常終了	
	336	利用者ユーザーIDエラー	ユーザIDの取得に失敗
	272	パラメータエラー (処理結果)	処理結果にNULLをセット
	273	パラメータエラー (処理状況)	処理状況にNULLをセット
	274	パラメータエラー (処理シーケンス)	処理シーケンスにNULLをセット または処理シーケンスの先頭<' '
	7	DLLログ取得エラー	iniファイルからDLLログファイル名の取得に失敗
	1	ファイルオープンエラー	DLLログファイルのオープンに失敗
	2	ファイルリードエラー	DLLログファイルの読み込みに失敗
	311	シーケンスコードエラー	DLLログファイルを検索した結果該当無しの場合
	312	シーケンスコードエラー(完了)	処理状況コード='0909'の時
	7	ファイル/Oエラー	DLLログファイルの更新に失敗した時

付録E TDC通信パッケージWindows版 コード変換表(詳細)

JIS-8 → EBCDIC

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			SP (40)	0 (F0)	@ (7C)	P (D7)	^ (60)	p (D7)	ナシ (80)	ナシ (90)	ナシ (A0)	- (58)	タ (91)	ミ (A5)	ナシ (E0)	ナシ (F0)
1			! (5A)	1 (F1)	A (C1)	Q (D8)	a (C1)	q (D8)	ナシ (81)	ナシ (91)	。 (41)	ア (81)	チ (92)	ム (A6)	ナシ (E1)	ナシ (F1)
2			" (7F)	2 (F2)	B (C2)	R (D9)	b (C2)	r (D9)	ナシ (82)	ナシ (92)	〒 (42)	イ (82)	ツ (93)	メ (A7)	ナシ (E2)	ナシ (F2)
3			# (7B)	3 (F3)	C (C3)	S (E2)	c (C3)	s (E2)	ナシ (83)	ナシ (93)	J (43)	ウ (83)	テ (94)	モ (A8)	ナシ (E3)	ナシ (F3)
4			\$ (E0)	4 (F4)	D (C4)	T (E3)	d (C4)	t (E3)	ナシ (84)	ナシ (94)	、 (44)	エ (84)	ト (95)	ヤ (A9)	ナシ (E4)	ナシ (F4)
5			% (6C)	5 (F5)	E (C5)	U (E4)	e (C5)	u (E4)	ナシ (85)	ナシ (95)	・ (45)	オ (85)	ナ (96)	ユ (AA)	ナシ (E5)	ナシ (F5)
6			& (50)	6 (F6)	F (C6)	V (E5)	f (C6)	v (E5)	ナシ (86)	ナシ (96)	ヲ (46)	カ (86)	ニ (97)	ヨ (AC)	ナシ (E6)	ナシ (F6)
7			^ (7D)	7 (F7)	G (C7)	W (E6)	g (C7)	w (E6)	ナシ (87)	ナシ (97)	ア (47)	キ (87)	ヌ (98)	ラ (AD)	ナシ (E7)	ナシ (F7)
8			((4D)	8 (F8)	H (C8)	X (E7)	h (C8)	x (E7)	ナシ (88)	ナシ (98)	イ (48)	ク (88)	ネ (99)	リ (AE)	ナシ (E8)	ナシ (F8)
9) (5D)	9 (F9)	I (C9)	Y (E8)	i (C9)	y (E8)	ナシ (89)	ナシ (99)	う (49)	ケ (89)	ノ (9A)	ル (AF)	ナシ (E9)	ナシ (F9)
A			* (5C)	: (7A)	J (D1)	Z (E9)	j (D1)	z (E9)	ナシ (8A)	ナシ (9A)	エ (51)	コ (8A)	ハ (9D)	レ (BA)	ナシ (EA)	ナシ (FA)
B			+ (4E)	; (5E)	K (D2)	[(5B)	k (D2)	[(C0)	ナシ (8B)	ナシ (9B)	オ (52)	サ (8C)	ヒ (9E)	ロ (BB)	ナシ (EB)	ナシ (FB)
C			. (6B)	< (4C)	L (D3)	¥ (5B)	l (D3)	l (4F)	ナシ (8C)	ナシ (9C)	ヤ (53)	シ (8D)	フ (9F)	ウ (BC)	ナシ (EC)	ナシ (FC)
D			- (60)	= (7E)	M (D4)] (5D)	m (D4)] (D0)	ナシ (8D)	ナシ (9D)	ユ (54)	ス (8E)	ヘ (A2)	ウ (BD)	ナシ (ED)	ナシ (FD)
E			> (4B)	> (6E)	N (D5)	^ (5E)	n (D5)	~ (A1)	ナシ (8E)	ナシ (9E)	ヨ (55)	セ (8F)	ホ (A3)	^ (BE)	ナシ (EE)	ナシ (FE)
F			/ (61)	? (6F)	O (D6)	_ (6D)	o (D6)	ナシ (7F)	ナシ (8F)	ナシ (9F)	ッ (56)	ソ (90)	マ (A4)	° (BF)	ナシ (EF)	ナシ (FF)

変換しない
 英小文字から英大文字へ変換
 ()内は変換後のコード

EBCDIC → JIS-8

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			ナシ (20)	ナシ (30)	SP (20)	& (26)	- (2D)	ナシ (70)	ナシ (80)	ソ (BF)	ナシ (A0)	ナシ (B0)	[(7B)] (7D)	\$ (24)	0 (30)
1			ナシ (21)	ナシ (31)	。(A1)	エ (AA)	/ (2F)	ナシ (71)	ア (B1)	タ (C0)	~ (7E)	ナシ (B1)	A (41)	J (4A)	ナシ (E1)	1 (31)
2			ナシ (22)	ナシ (32)	〒 (A2)	オ (AB)	ナシ (62)	ナシ (72)	イ (B2)	チ (C1)	へ (CD)	ナシ (B2)	B (42)	K (4B)	S (53)	2 (32)
3			ナシ (23)	ナシ (33)	J (A3)	ヤ (AC)	ナシ (63)	ナシ (73)	ウ (B3)	ツ (C2)	ホ (CE)	ナシ (B3)	C (43)	L (4C)	T (54)	3 (33)
4			ナシ (24)	ナシ (34)	、 (A4)	ユ (AD)	ナシ (64)	ナシ (74)	エ (B4)	テ (C3)	マ (CF)	ナシ (B4)	D (44)	M (4D)	U (55)	4 (34)
5			ナシ (25)	ナシ (35)	・ (A5)	ヨ (AE)	ナシ (65)	ナシ (75)	オ (B5)	ト (C4)	ミ (D0)	ナシ (B5)	E (45)	N (4E)	V (56)	5 (35)
6			ナシ (26)	ナシ (36)	ヲ (A6)	ッ (AF)	ナシ (66)	ナシ (76)	カ (B6)	ナ (C5)	ム (D1)	ナシ (B6)	F (46)	O (4F)	W (57)	6 (36)
7			ナシ (27)	ナシ (37)	ア (A7)	ナシ (57)	ナシ (67)	ナシ (77)	キ (B7)	ニ (C6)	モ (D2)	ナシ (B7)	G (47)	P (50)	X (58)	7 (37)
8			ナシ (28)	ナシ (38)	イ (A8)	- (B0)	ナシ (68)	ナシ (78)	ク (B8)	ヌ (C7)	メ (D3)	ナシ (B8)	H (48)	Q (51)	Y (59)	8 (38)
9			ナシ (29)	ナシ (39)	ウ (A9)	ナシ (59)	ナシ (69)	ナシ (79)	ケ (B9)	ネ (C8)	ヤ (D4)	ナシ (B9)	I (49)	R (52)	Z (5A)	9 (39)
A			ナシ (2A)	ナシ (3A)	ナシ (4A)	! (21)	ナシ (6A)	:	コ (BA)	ノ (C9)	ユ (D5)	レ (DA)	ナシ (CA)	ナシ (DA)	ナシ (EA)	ナシ (FA)
B			ナシ (2B)	ナシ (3B)	、 (2E)	¥ (5C)	.	# (23)	ナシ (8B)	ナシ (9B)	ナシ (AB)	ロ (DB)	ナシ (CB)	ナシ (DB)	ナシ (EB)	ナシ (FB)
C			ナシ (2C)	ナシ (3C)	< (3C)	* (2A)	% (2C)	@ (40)	ナシ (8B)	ナシ (9C)	ヨ (D6)	ワ (DC)	ナシ (CC)	ナシ (DC)	ナシ (EC)	ナシ (FC)
D			ナシ (2D)	ナシ (3D)	((28)) (29)	_ (5F)	' (27)	シ (BC)	ハ (CA)	ヲ (D7)	ン (DD)	ナシ (CD)	ナシ (DD)	ナシ (ED)	ナシ (FD)
E			ナシ (2E)	ナシ (3E)	+ (2B)	; (3B)	> (3E)	≡ (3D)	ス (BD)	ヒ (CB)	リ (D8)	° (DE)	ナシ (CE)	ナシ (DE)	ナシ (EE)	ナシ (FE)
F			ナシ (2F)	ナシ (3F)	(7C)	ナシ (5F)	? (3F)	” (22)	セ (BE)	フ (CC)	ル (D9)	° (DF)	ナシ (CF)	ナシ (DF)	ナシ (EF)	ナシ (FF)

変換しない
 EBCDIC(C) のコード
 EBCDIC(K) のコード
 EBCDIC(C) では「¥」 EBCDIC(K) では「\$」
 ()内は変換後のコード

※ 「`」 JIS (60) EBCDIC (79) は変換しません！

注意事項

の部分に関しては、JIS8文字コード・EBCDIC文字コード共、変換しませんので注意が必要です。
 仮に、JIS8→EBCDIC→JIS8変換を仮定した場合に、下記の問題が発生する可能性が有ります。

最初のJIS8変換データに、変換対象外の0x81データが含まれていたとします。
 0x81データは、変換対象外なので無変換のままEBCDIC変換データとして扱われます。
 次に、EBCDICデータをJIS8データに変換する場合に、JIS8→EBCDIC変換時は対象外だった0x81データは、
 EBCDICデータとして見た場合、半角カナのアと判断され、JIS8データのア(0xB1)に変換され、
 0x81が0xB1に置き換わる問題が発生します。