

U/C サーバー業務システム間
転送プログラムインターフェース仕様書

2005 年 1 月 1 日
(株) 広告 EDI センター

注意事項

- Ⅰ 本資料を無断で他に転載しないようお願いします。
- Ⅰ 本資料は、予告無しに変更する場合があります。
- Ⅰ 本資料の内容に不備がある場合は、ご連絡ください。

改訂履歴

日付	改訂内容
1999/08	第1版（第3.1版）
2000/06/21	第2版
2000/07/03	修正：事務局名称
2000/09/06	修正：3.1、3.2、3.3 フォーマット表内
2001/04/01	修正： 表紙：発行元を「広告取引 EDI センター推進機構」に変更
2002/10/1	修正： 表紙：発行元を「(株) 広告 EDI センター」に変更
2005/01/01	修正：語句修正

目次

1.	はじめに	1-1
2.	U/C サーバーと転送プログラムのインターフェース使用について	2-1
2.1.	U/C サーバーインターフェース	2-1
2.2.	U/C サーバーインターフェース (業務アプリケーション)	2-2
2.2.1.	各ファイルの説明	2-3
2.2.2.	各ファイルの U/C サーバー側保存ディレクトリー	2-4
2.2.3.	各ファイルのファイル名	2-4
2.3.	処理フロー	2-5
2.3.1.	送信処理フロー	2-5
2.3.2.	受信処理フロー	2-6
3.	ファイルフォーマット	3-1
3.1.	業務取引ファイル	3-1
3.2.	ステータスファイル	3-2
3.3.	リターンファイル	3-5
3.4.	漢字コード	3-5
4.	参考資料 (転送プログラム)	4-1
4.1.	業務集信用転送プログラム	4-1
4.2.	業務配信用転送プログラム	4-1
4.3.	業務集信用転送プログラムサンプルソース	4-2

1. はじめに

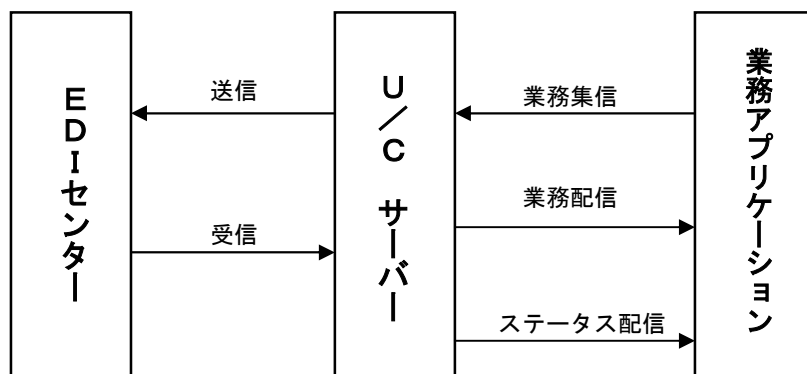
本資料は、U/C サーバーと業務アプリケーションのデータの送受信に関する仕様を記述します。

なお、本資料は U/C サーバー version 2.2.0 ~ 4.1.0 に対応します。

2. U/C サーバーと転送プログラムのインターフェース使用について

2.1. U/C サーバーインターフェース

EDI センター、U/C サーバー、業務アプリケーションサーバー間の関係を記述します。



I 送信

U/C サーバーが EDI センターへメッセージ（業務取引データ）を送ることを送信と呼びます。

U/C サーバーの下記機能を実行したとき、U/C サーバーはメッセージを送信します。

- ・ スケジュール送信
- ・ 緊急送信

I 受信

U/C サーバーが EDI センターからメッセージ（業務取引データ）を受けることを受信と呼びます。

U/C サーバーの下記機能を実行したとき、U/C サーバーはメッセージを受信します。

- ・ スケジュール受信
- ・ 選択受信
- ・ 緊急受信
- ・ 業務再取込み

I 業務集信

業務アプリケーションから U/C サーバーに業務取引ファイルを転送することを業務集信と呼びます。

U/C サーバーが送信したとき、業務集信を実行します。

I 業務配信

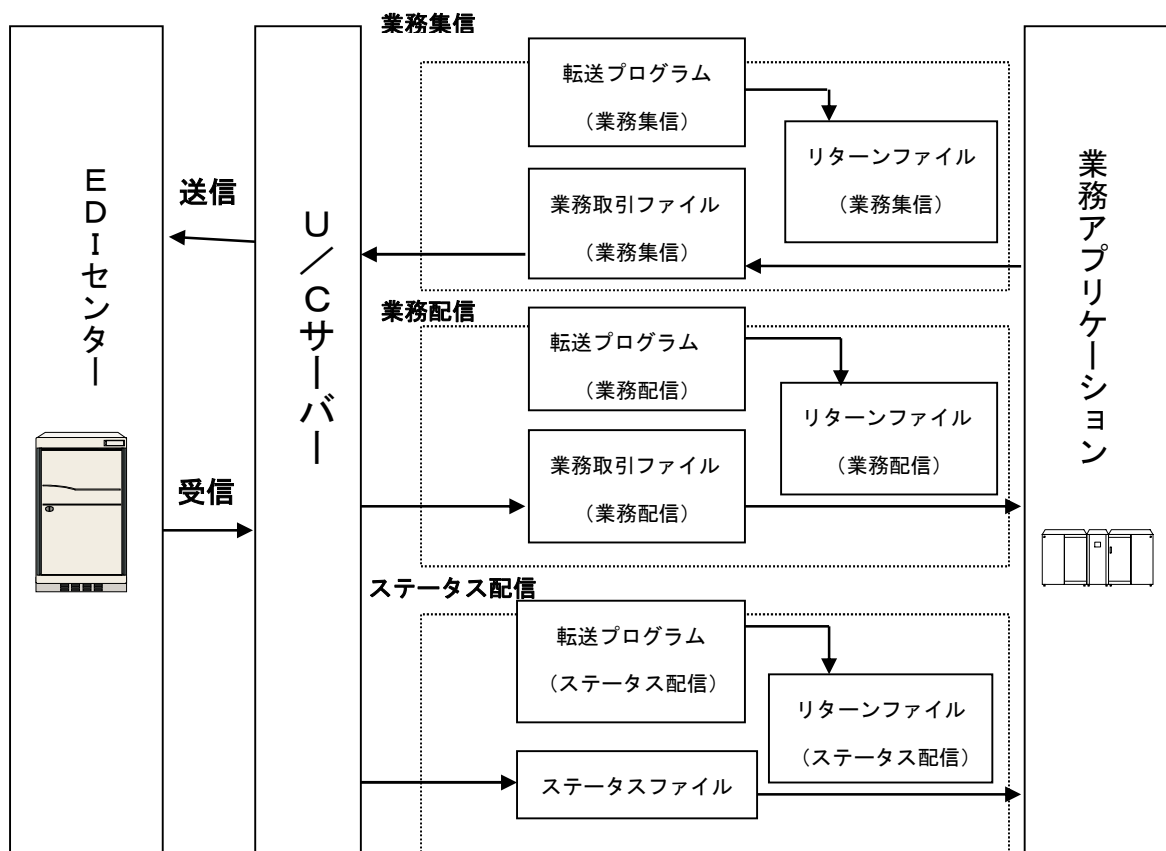
U/C サーバーから業務アプリケーションに業務取引ファイルを転送することを業務配信と呼びます。

U/C サーバーが受信したとき、業務配信を実行します。

I ステータス配信（オプション）

U/C サーバーから業務アプリケーションにステータスファイルを転送することをステータス配信と呼びます。U/C サーバーが業務集信または業務配信したとき、ステータス配信を実行します。

2.2. U/C サーバーインターフェース（業務アプリケーション）



2.2.1. 各ファイルの説明

U/C サーバーと業務アプリケーションのインターフェースを実現する為のファイルは下記のとおりです。

I 転送ファイル

U/C サーバーと業務アプリケーションの間で、業務取引ファイルおよびステータスファイルを転送します。U/C サーバーが転送プログラムを起動するとき、スケジュール設定で指定した引数を転送プログラムに渡します。

転送プログラムの種類は下記のとおりです。

- ・ 転送プログラム（業務集信）
- ・ 転送プログラム（業務配信）
- ・ 転送プログラム（ステータス配信）

I 業務取引ファイル

業務取引データ（新聞：申込、割付、請求予定等/TV：引合、局案、割付等）を保守したファイルです。

業務取引ファイルの種類は下記のとおりです。

- ・ 業務取引ファイル（業務集信）
- ・ 業務取引ファイル（業務配信）

I ステータスファイル（オプション）

U/C サーバー - EDI センター間のメッセージの送達状況を保存したファイルです。1 件（ヘッダー毎）に対して 1 ファイル作成されます。

ステータスファイルの種類は下記のとおりです。

- ・ ステータスファイル（業務集信）
- ・ ステータスファイル（業務配信）

I リターンファイル（オプション）

転送プログラムの実行結果を保存したファイルです。

リターンファイルの種類は下記のとおりです。

- ・ リターンファイル（業務集信）
- ・ リターンファイル（業務配信）
- ・ リターンファイル（ステータス配信）

2.2.2. 各ファイルの U/C サーバー側保存ディレクトリー

I 転送プログラム

ディレクトリーの指定はありません。業務集信、業務配信、およびステータス配信毎に指定できます。

I 業務取引ファイル（業務集信）

C:\AEDI\ユーザー名\COLLECT\ (U/C サーバーでユーザー名を設定する)

I 業務取引ファイル（業務配信）

C:\AEDI\ユーザー名\DELIVER\ (U/C サーバーでユーザー名を設定する)

I ステータスファイル

C:\AEDI\ユーザー名\DELIVER\ (U/C サーバーでユーザー名を設定する)

I リターンファイル

ディレクトリーの指定はありません。各転送プログラム毎に指定できます。

2.2.3. 各ファイルのファイル名

I 転送プログラム

ファイル名の指定はありません。業務集信、業務配信、およびステータス配信毎に指定できます。

I 業務取引ファイル（業務集信）

ファイル名の指定はありません。

I 業務取引ファイル（業務配信）

XXXXXXXXXX.dat システムでユニークな名称になります。

I ステータスファイル（業務集信）

XXXXXXXXXX.sta

業務取引ファイル（業務集信）毎に作成します。

I ステータスファイル（業務配信）

XXXXXXXXXXXXXXXXXXXX.sta

業務取引ファイル（業務配信）毎に作成します。

I リターンファイル

ファイル名の指定はありません。各転送プログラム毎に指定できます。

【サンプル】

業務集信：C:\AEDI\ユーザー名\Collect.rtn

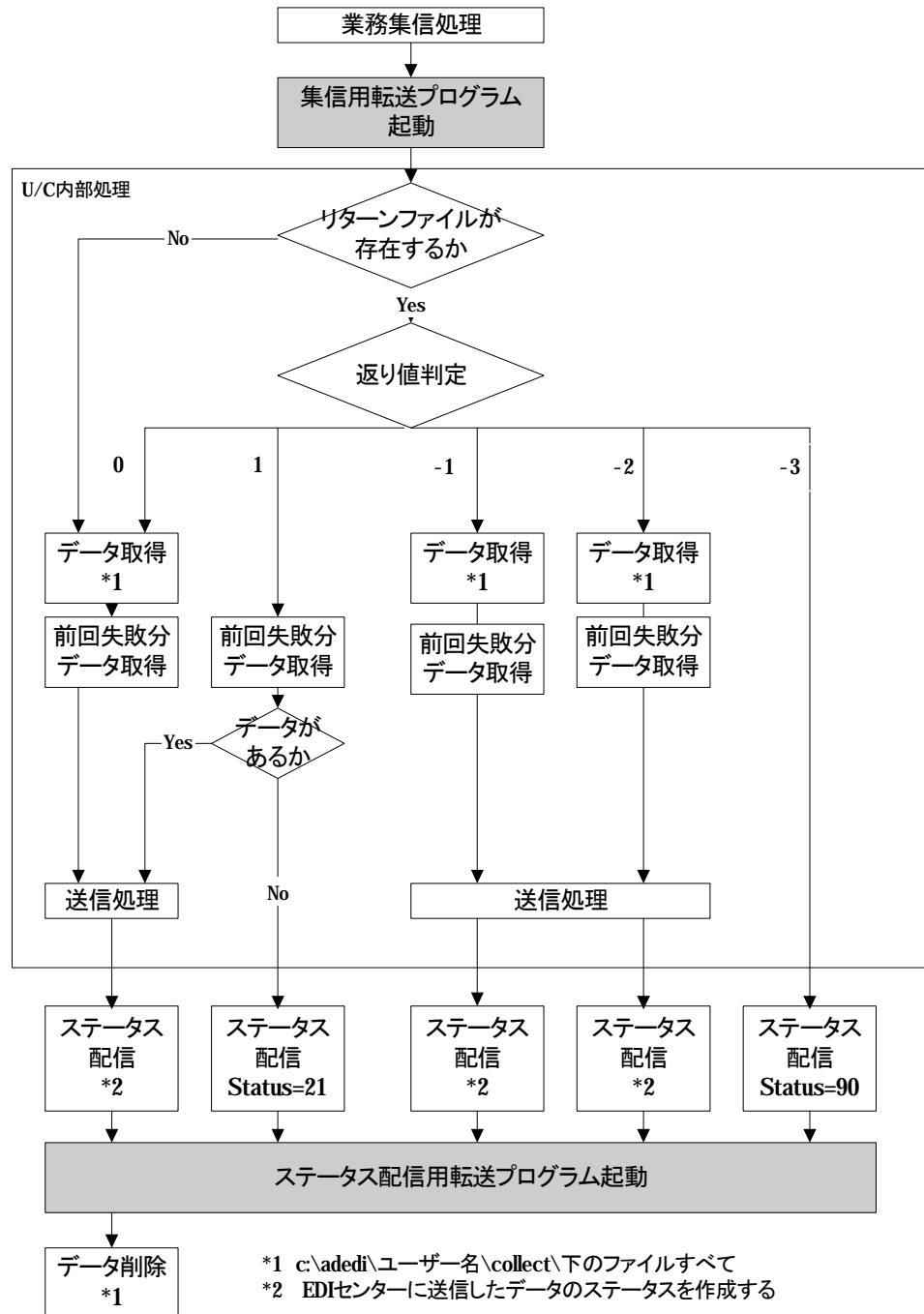
業務配信：C:\AEDI\ユーザー名\DeliverD.rtn

ステータス配信：C:\AEDI\ユーザー名\DeliverS.rtn

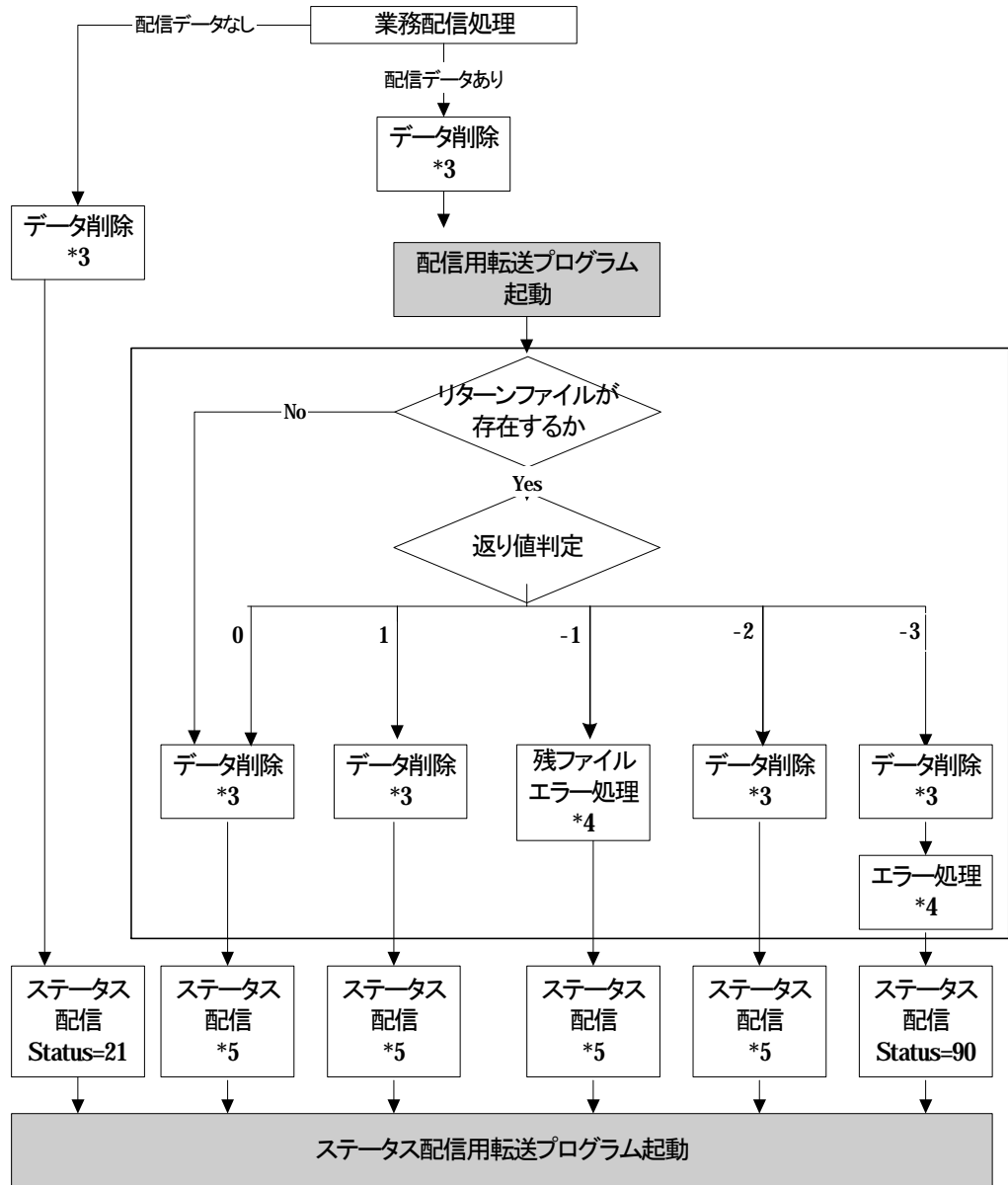
2.3. 処理フロー

U/C サーバーは下記処理フローに従って送信、受信機能中の各処理が動作し、その処理中で転送プログラムを起動します。

2.3.1. 送信処理フロー



2.3.2. 受信処理フロー



*3 c:\adedi\ユーザー名\deliver\下のファイルすべて

*4 対象データのステータスを「業務システム配信エラー」とし、次回スケジュールでリトライの対象とする

*5 EDIセンターから受信したデータのステータスを作成する

3. ファイルフォーマット

3.1. 業務取引ファイル

フォーマット：

フォーマット	byte 数
標準ヘッダー	254byte
スペースパディング	情報区分固有 byte 数 - 254byte
ボディ (業務データ部)	情報区分固有 byte 数 ※デリミタなし

標準ヘッダー：

No	項目名称	属性	説明	必須
1	ヘッダー識別子	an3	標準ヘッダーであることを示す識別子「EDI」固定	○
2	リザーブ	an12	「AEDI」固定	
3	受信者企業コード	an12	受信者の標準企業コード	○
4	送信者企業コード	an12	送信者の標準企業コード	○
5	企業コード変換フラグ	an1	送信および受信企業コードを標準企業コードに変換するか否かを示すフラグ「1」：変換する 「0」：変換しない	○
6	BPID	an8	広告 EDI で定められた BPID	○
7	情報区分	an4	広告 EDI で定められた情報区分	○
8	データ作成日付	n8	データを作成した日付。yyyymmdd 形式	○
9	データ作成時刻	n6	データを作成した時刻。hhmmss 形式	○
10	管理番号	an20	業務システムが管理する管理番号	
11	得意先コード	an12	当該データの得意先コード	
12	得意先名称	an60	当該データの得意先名称	
13	件名	an60	当該データの案件名	
14	掲載期間 (From)	an8	掲載期間 (自)	
15	掲載期間 (To)	an8	掲載期間 (至)	
16	リザーブ	an20	スペース	○

スペースパディング：

No	項目名称	属性	説明	必須
1	スペースパディング	—	(業務データ部の情報区分毎の固有データ長 - 254byte)のスペースをパディングする	○

ボディ (業務データ部)：

新聞、テレビそれぞれの標準メッセージ仕様参照

3.2. ステータスファイル

フォーマット：

フォーマット	byte 数
ボディ	256byte

ボディ詳細仕様：

ステータスファイル（送信）

No	項目名称	属性	説明
1	ヘッダー識別子	an3	標準ヘッダーであることを示す識別子「EDI」固定
2	リザーブ	an12	「AEDI」固定
3	受信者企業コード	an12	受信者の企業コード
4	送信者企業コード	an12	送信者の企業コード
5	企業コード変換フラグ	an1	送信および受信企業コードを標準企業コードに変換するか否かを示すフラグ
6	BPID	an8	広告 EDI で定められた BPID
7	情報区分	an4	広告 EDI で定められた情報区分
8	データ作成日時	n14	データを作成した日付 yyyymmddhhmmss 形式
9	管理番号	an20	業務システムが管理する管理番号
10	得意先コード	an12	当該データの得意先コード
11	得意先名称	an60	当該データの得意先名称
12	件名	an60	当該データの案件名
13	掲載期間 (From)	an8	掲載期間 (自)
14	掲載期間 (To)	an8	掲載期間 (至)
15	リザーブ	an19	スペース
16	送受信区分	an1	スペース
17	ステータスコード	an2	下記説明参照

ステータスファイル（受信）

No	項目名称	属性	説明
1	ヘッダー識別子	an3	標準ヘッダーであることを示す識別子「EDI」固定
2	リザーブ	an12	「AEDI」固定
3	受信者企業コード	an12	受信者の企業コード (社内コード)
4	送信者企業コード	an12	送信者の企業コード (社内コード)
5	リザーブ	an1	「1」固定
6	BPID	an8	広告 EDI で定められた BPID
7	情報区分	an4	広告 EDI で定められた情報区分
8	データ作成日時	n14	送信者 (取引先) がデータを作成した日付 yyyymmddhhmmss 形式
9	リザーブ	an187	スペース
10	送受信区分	n1	「1」固定
11	ステータスコード	an2	下記説明参照

ステータスコード一覧

コード	意味	内容	対象機能
0	正常終了	送信、受信が正常に処理されました。	・送信 ・受信
10	標準ヘッダー必須項目エラー	送信ファイルの標準ヘッダーに未入力の項目があります。 標準ヘッダーの内容を確認してください。	・送信
11	送信者コード変換エラー	送信または受信ファイルの標準ヘッダーの送信者コードが統一企業変換テーブルに登録されていません。標準ヘッダーの内容を確認してください。 また、運用管理・標準企業コード入力画面で内容を確認してください。	・送信
12	受信者コード変換エラー	送信または受信ファイルの標準ヘッダーの受信者コードが標準企業コード変換テーブルに登録されていません。標準ヘッダーの内容を確認してください。 また、運用管理・標準企業コード入力画面で内容を確認してください。	・送信
13	取引データなし	送信ファイル中に取引データが存在しません。送信ファイルの内容を確認してください。	・送信
14	トランスレータ エラー	CII変換プログラムがエラーを返しました。 CII変換プログラムのエラーログを確認してください。	・送信 ・受信
15	標準ヘッダー文字コードエラー	利用可能なANSIコードページに存在しない文字コードが標準ヘッダーに含まれています。業務取引ファイルの内容を確認してください。	・送信
16	送信者コード未登録	送信ファイルの標準ヘッダーの送信者コードが統一企業変換テーブルに登録されていません。標準ヘッダーの内容を確認してください。 また、パートナー設定画面で内容を確認してください。	・送信 ・受信
17	受信者コード未登録	送信ファイルの標準ヘッダーの受信者コードが統一企業変換テーブルに登録されていません。標準ヘッダーの内容を確認してください。 また、パートナー設定画面で内容を確認してください。	・送信 ・受信
21	受信・送信データなし	処理する受信または送信データはありません。	・送信 ・受信
22	標準ヘッダー識別エラー	送信ファイル中に標準ヘッダーが見つからない。 標準ヘッダーの内容を確認してください。	・送信
23	業務取引ファイル レングスエラー	送信ファイルのレコード長が情報区分マスターに定義されているレコード長と合致しません。 送信ファイルのファイル長、または、情報区分マスターの業務取引ファイルレコード長を確認してください。	・送信
24	EDIセンター送信・受信エラー	EDIセンターにデータの送受信中にエラーが発生しました。あるいは、送信したデータにエラーがあります。エラーコードを確認してください。	・送信 ・受信

コード	意味	内容	対象機能
25	業務集配信プログラムエラー	ファイル転送プログラムがエラーを返しました。一部正常データあり。	・送信 ・受信
26	業務集配信プログラムエラー	ファイル転送プログラムがエラーを返しました。データなし。	・送信 ・受信
27	ステータス配信プログラムエラー	業務システムステータス配信のファイル転送プログラムがエラーを返しました。一部正常データあり。	・送信 ・受信
28	ステータス配信プログラムエラー	業務システムステータス配信のファイル転送プログラムがエラーを返しました。データなし。	・送信 ・受信
29	CIIファイルレングスエラー	CIIフォーマットに変換されたファイルのレングスが不正です。(251バイトで割り切れない)業務取引ファイルのデータ内容、トランスレータのマッピングテーブル内容を確認してください。	・送信 ・受信
90	業務集配信プログラムの致命的エラー	ファイル転送プログラムがエラーを返しました。処理続行不可能なエラー。システム担当者に連絡してください。	・送信 ・受信
91	トランスレータの致命的エラー	CII変換プログラムがエラーを返しました。処理続行不可能なエラー。システム担当者に連絡してください。	・送信 ・受信
92	EDIセンター接続 致命的エラー	EDIセンターに接続を試みましたが、異常がありました。システム担当者に連絡してください。	・送信 ・受信
93	ステータス配信プログラム 致命的エラー	業務システムステータス配信のファイル転送プログラムがエラーを返しました。処理続行不可能なエラー。システム担当者に連絡してください。	・送信 ・受信
99	広告EDI U/Cサーバープログラムエラー	広告EDI U/Cサーバーのプログラムエラーです。システム担当者に連絡してください。	・送信 ・受信

3.3. リターンファイル

フォーマット：

フォーマット	byte 数
ボディ	3byte~13byte

ボディ：

No	項目名称	属性	説明	必須
1	返り値	an3	下記戻り値文字を左詰めでセット	○
2	メッセージ	an10	将来拡張用	

返り値一覧

コード	メッセージ	処理	転送対象の業務取引ファイルのステータス
00	業務アプリケーション正常終了 (データ有)	業務集信	正常終了後、送信処理のステータスが反映される
		業務配信	正常終了、送信処理のステータスが反映される
01	業務アプリケーション正常終了 (データ無)	業務集信	データなしのためステータスはなし
		業務配信	すべて業務システム配信正常終了
-1	業務アプリケーションエラー終了 (正常データ有)	業務集信	集信されたデータのみを処理し、送信処理のステータスが反映される
		業務配信	<u>ディレクトリ内に残っている業務取引ファイルは次回スケジュール受信で再度配信対象になります。</u>
-2	業務アプリケーションエラー終了 (データ無)	業務集信	データなしのためステータスはなし
		業務配信	すべて業務システム配信正常終了
-3	業務アプリケーションエラー終了 (FATAL エラー)	業務集信	データが転送されないためステータスはなし
		業務配信	すべて業務システム配信エラー終了 <u>受信処理で対象となった業務取引ファイルは次回スケジュール受信で再度配信対象になります。</u>

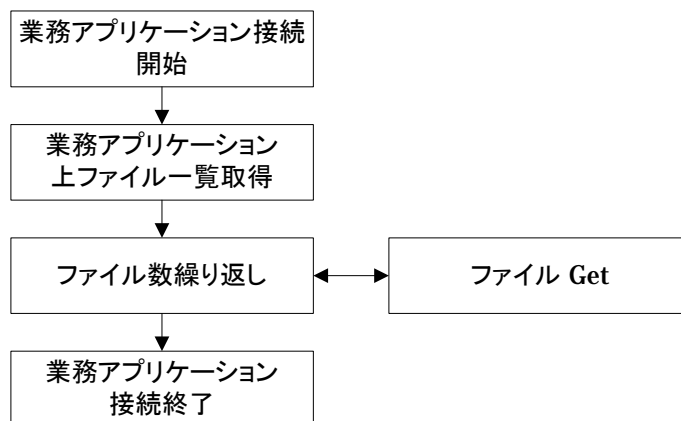
ステータス配信の場合、リターンファイルの値に基づいて U/C サーバーにメッセージを出力します。

3.4. 漢字コード

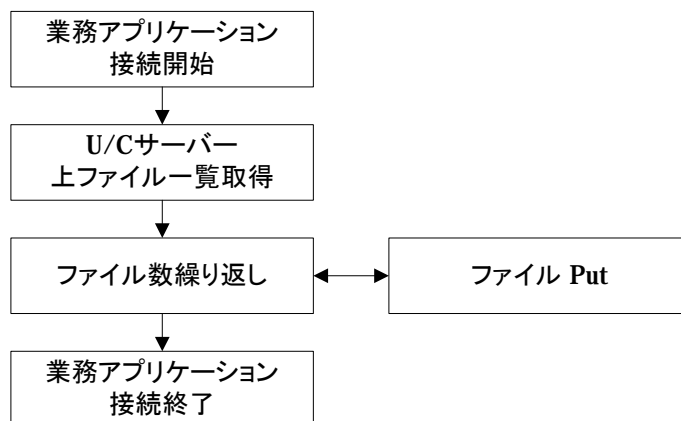
すべてのファイルに含まれる漢字コードは Shift JIS を使用しています。従って業務アプリケーションにて、Shift JIS 以外のコードが必要な場合は転送プログラム内でコード変換の処理が必要となります。

4. 参考資料（転送プログラム）

4.1. 業務集信用転送プログラム



4.2. 業務配信用転送プログラム



4.3. 業務集信用転送プログラムサンプルソース

```
業務 AP→U/Cftp 転送
// ftpgyomu.cpp : アプリケーション用クラスの機能定義を行います。
//

#include "stdafx.h"
#include "afxinet.h"
#include "ftpgyomu.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////

// CFtpgyomuApp

BEGIN_MESSAGE_MAP(CFtpgyomuApp, CWinApp)
   //{{AFX_MSG_MAP(CFtpgyomuApp)
//      ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
        // メモ - ClassWizard はこの位置にマッピング用のマクロを追加または削除します。
        //   この位置に生成されるコードを編集しないでください。
   //}}AFX_MSG_MAP
    // 標準のファイル基本ドキュメント コマンド
END_MESSAGE_MAP()

////////////////////////////////////

// CFtpgyomuApp クラスの構築

/*
CFtpgyomuApp::CFtpgyomuApp()
{
```

```

// TODO: この位置に構築用コードを追加してください。
// ここに InitInstance 中の重要な初期化処理をすべて記述してください。
}
*/

////////////////////////////////////
// The one and only CFtpgyomuApp オブジェクト

CFtpgyomuApp theApp;

// ftpgyomu.h : FTPTYOMU アプリケーションのメイン ヘッダー ファイル
//

#ifdef __AFXWIN_H__
    #error include 'stdafx.h' before including this file for PCH
#endif

// #include "resource.h"          // メイン シンボル

#define GFTP_HOST "gyomu_host"
#define GFTP_APNAME "ftpgyomu"
#define GFTP_PORT 21
#define GFTP_USER "user1"
#define GFTP_PASS "user1"
#define GFTP_REMDIR "/home/user1/ftpgyomu/index"
#define GFTP_REMFDIR "/home/user1/ftpgyomu"
#define GFTP_REMPAT "*"
#define GFTP_LOCDIR "c:\\\\AEDI\\\\USERNAME\\\\COLLECT"

////////////////////////////////////
// CFtpgyomuApp:
// このクラスの動作の定義に関しては ftpgyomu.cpp ファイルを参照してください。
//

```

```

class CFtpgyomuApp : public CWinApp
{
public:
    inline CFtpgyomuApp(){};
    int appret;

// オーバーライド
    // ClassWizard は仮想関数を生成しオーバーライドします。
    //{{AFX_VIRTUAL(CFtpgyomuApp)
    public:
    virtual BOOL InitInstance();
    //}}AFX_VIRTUAL

    virtual int ExitInstance();

// インプリメンテーション

    //{{AFX_MSG(CFtpgyomuApp)
    afx_msg void OnAppAbout();
        // メモ - ClassWizard はこの位置にメンバ関数を追加または削除します。
        //      この位置に生成されるコードを編集しないでください。
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

////////////////////////////////////

////////////////////////////////////

// CFtpgyomuApp クラスの初期化

BOOL CFtpgyomuApp::InitInstance()
{
/*

```

```

        if (!AfxSocketInit())
        {
            AfxMessageBox(IDP_SOCKETS_INIT_FAILED);
            return FALSE;
        }
    */

    // 標準的な初期化处理
    // もしこれらの機能を使用せず、実行ファイルのサイズを小さく
    // したければ以下の特定の初期化ルーチンの中から不必要なもの
    // を削除してください。

    /*
    #ifdef _AFXDLL
        Enable3dControls(); // 共有 DLL の中で MFC を使用する場合
        にはここを呼び出してください。
    #else
        Enable3dControlsStatic(); // MFC と静的にリンクしている場合にはここを呼び出してくだ
        さい。
    #endif

    LoadStdProfileSettings(); // 標準の INI ファイルのオプションをロードします (MRU を含
    む)
    */

    // アプリケーション用のドキュメント テンプレートを登録します。ドキュメント テンプレー
    // ト
    // はドキュメント、フレーム ウィンドウとビューを結合するために機能します。

    /*
    CSingleDocTemplate* pDocTemplate;
    pDocTemplate = new CSingleDocTemplate(
        IDR_MAINFRAME,
        RUNTIME_CLASS(CFtpgyomuDoc),
        RUNTIME_CLASS(CMainFrame), // メイン SDI フレーム ウィンドウ
        RUNTIME_CLASS(CFtpgyomuView));
    AddDocTemplate(pDocTemplate);
    */

```

```

        // DDE、file open など標準のシェル コマンドのコマンドラインを解析します。
/*
    CCommandLineInfo cmdInfo;
    ParseCommandLine(cmdInfo);
*/
    // コマンドラインでディスパッチ コマンドを指定します。
/*
    if (!ProcessShellCommand(cmdInfo))
        return FALSE;
*/
    appret = 0;

// make ftp client
    CInternetSession cinetses;
    CStringArray      csa;
    int fcnt = 0;
    int jcnt = 0;
    CFtpConnection* cftp = NULL;

    try {

        int ret;

        cftp = cinetses.GetFtpConnection(GFTP_HOST, GFTP_USER, GFTP_PASS);
        if ( (ret = cftp->SetCurrentDirectory(GFTP_REMDIR)) == 0 ){
            throw GetLastError();
        }
        CFtpFileFind cftpff(cftp);

        if ( (ret = cftpff.FindFile(GFTP_REMPAT) ) == 0 ){
            if ( GetLastError() == 18 ){
                throw (DWORD)9999;
            }
            throw GetLastError();
        }else{
            cftpff.FindNextFile();

```

```

        csa.Add(cftpff.GetFileName());
        fcnt++;

#ifdef _DEBUG

        TRACE("First Remote File Name=%s\n", csa[fcnt-1]);

#endif

    }

    while ( ( ret = cftpff.FindNextFile() ) != 0 ) {
        csa.Add(cftpff.GetFileName());
        fcnt++;

#ifdef _DEBUG

        TRACE("Remote File Name=%s\n", csa[fcnt-1]);

#endif

    }

    csa.Add(cftpff.GetFileName());
    fcnt++;
    TRACE("Remote File Name=%s\n", csa[fcnt-1]);

// set local directory
    if ( (ret = SetCurrentDirectory(GFTP_LOCDIR)) == FALSE ){
        throw GetLastError();
    }

// get a file from remote directory and delete index
    for ( jcnt = 0; jcnt < fcnt; jcnt++ ){

// set remote directory

        if ( (ret = cftp->SetCurrentDirectory(GFTP_REMFDIR)) == 0 ){
            throw GetLastError();
        }

// get remote date file

        if ( (ret = cftp->GetFile(csa[jcnt], csa[jcnt], FALSE)) == FALSE ){
            throw GetLastError();
        }

//set remote directory

        if ( (ret = cftp->SetCurrentDirectory(GFTP_REMFDIR)) == 0 ){
            throw GetLastError();
        }

    }

```

```

        throw (DWORD) 0;

    }

    catch (CInternetException cie){
#ifdef _DEBUG
        afxDump << "CInternetException " << cie;
#endif

        if ( jcnt == 0 ){
            appret = -1;
        }else{
            appret = -2;
        }

        if ( cftp != NULL ){
            delete cftp;
            for ( int i = 0; i < fcnt; i++ ){
                csa[i].ReleaseBuffer();
            }
            csa.RemoveAll();
        }
    }

    catch (DWORD err){
        LPVOID lpMsgBuf;

        if ( err != 0 ){
            FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER |
FORMAT_MESSAGE_FROM_SYSTEM NULL, GetLastError(), MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
(LPTSTR) &lpMsgBuf, 0, NULL);

            TRACE("InterNet Error=%d, %d\n", HIWORD(err), LOWORD(err));
            TRACE("ErrMsg = %s\n", lpMsgBuf);
            if ( err == 9999 ){
                appret = 1;
            }else{
                if ( jcnt == 0 ){
                    appret = -1;
                }else{

```

```
                                appret = -2;
                                }
                                }
                                }
                                if ( cftp != NULL ){
                                    delete cftp;
                                    for ( int i = 0; i < fcnt; i++ ){
                                        csa[i].ReleaseBuffer();
                                    }
                                    csa.RemoveAll();
                                }
                                }

                                return FALSE;
                                }
                                //
                                //
                                //
                                int CFtpgyomuApp::ExitInstance(){
                                    return appret;
                                }
                                //////////////////////////////////////
                                // CFtpgyomuApp コマンド
```