



メール / Web Service でFAXを送信 & 受信
InterFAXサービス・デベロッパーガイド

Rev. 06

The logo for doit, consisting of the word "doit" in a bold, blue, lowercase sans-serif font. The dot above the "i" is a small red circle.

株式会社ドゥイット



InterFAXサービスとは？

メールソフトやメールシステム または Web Service (SOAP) でFAXの送受信ができるインターネットFAXサービスです。

メール / Web ServiceでFAX送信： **FAX** **送信** サービス

メール / Web ServiceでFAX受信： **FAX** **受信** サービス

InterFAXサービスの詳細については、カタログおよび各送信・受信サービスの資料をご参照ください。

システムのFAX送受信ソリューションとして

業務アプリケーションからの自動FAX送受信システム利用

◎ InterFAXは多くの業務アプリケーション内で利用されています

例1) アプリケーション内で帳票をPDFやHTML、テキストで生成し、メールやSOAPでFAX送信

例2) 宿泊予約等のWebアプリケーションでお客様がフォームで予約後、
その連絡をCGIからメールで旅館やホテルへFAX送信

例3) OCRとの組み合わせで受信FAXの自動データ処理(人件費計算や受注処理等)

★ 一日、数千件規模のFAX送受信にも対応

◎ InterFAXを利用することで大幅に開発効率を向上させ、導入から保守コストを軽減

★ FAXカードやFAXサーバをアプリケーションでコントロールする為には、専用APIでの開発による多大な開発時間、導入コスト、保守経費が必要 → InterFAXを利用すればメールやWeb APIだけでシステム構築が可能

システム開発 & 運用のポイント(送信サービス-1)

■ メール生成時の注意事項

- ・ テキストメールの本文は、ISO-2022-JPでエンコード
例) Content-Type: text/plain; charset="ISO-2022-JP"
Content-Transfer-Encoding: 7bit
- ・ HTMLメールの本文は、ISO-2022-JP、Shift_JIS、UTF-8をサポート
[マルチパートメールの場合]
作成されたメールのContent-typeが「multipart/alternative」の場合、HTML本文とテキスト本文と一緒に作成、送付されます。(多くのメールソフトがこの形式となります。)この形式では、InterFAXは、HTML本文のみFAXします。
[HTML本文でテキストファイルが添付として作成された場合]
このメール形式の場合、InterFAXでは HTMLを本文、テキストは添付として扱います。ついては、本文をFAXする場合は、オプション設定の「/b」(メール本文と添付ファイルの両方を送信する)が必要です。
- ・ 添付ファイルは、base64でのエンコード
例) Content-Transfer-Encoding: base64

■ 送信履歴の活用

InterFAXサービスでは、アカウント管理Webと呼ばれるWeb管理システムを全てのユーザー様にご提供しております。アカウント管理Web内で、FAXの送信サービス、受信サービス共に履歴をご確認いただくことが可能です。履歴はほぼリアルタイムで更新されます。

送信履歴ページでは、送信FAXイメージをダウンロードして確認することや、再送信することができます。また、緊急時には送信をキャンセルすることもできます。

送信履歴には、検索機能もあり、日付、件名、送信先FAX番号、ステータス等で検索し、必要な履歴情報を表示、またはCSVファイルやXMLファイルで取得することができます。

システム開発 & 運用のポイント(送信サービス-2)

■ ユーザ認証について

InterFAXでは、メールの送信元アドレスでユーザ認証をおこないます。また、なりすまし対策として、パスワードまたは電子署名による追加認証を推奨しています。

送信元アドレスは、SMTP通信での「MAIL FROM:」コマンドの引数として与えられるメールアドレスです。一般的には、メールヘッダ内の「From:」アドレスですが、「From:」アドレスはヘッダ情報の値であり、実際の送信者以外のアドレスを記述することができるため認証には使用いたしません。

InterFAX送信サービスで「From:」アドレスを事前登録してもInterFAX側で認証不可となる場合、お客様が送付されるメールヘッダの「Return-path:」アドレスをご確認ください。この「Return-path:」(送信メールがエラーとなった場合、エラーメールを返すアドレス)が存在する場合は、Return-path:が送信元アドレスといえますので、このアドレスをInterFAX側に「FAX送信サービス利用可能者」としてご登録ください。事前にお客様の他のメールアドレスにメール送信し、そのヘッダ内容を確認することを推奨します。よって、Return-path:が存在すればReturn-path: なければFrom: が、認証対象アドレスとご判断ください。

尚、万が一、InterFAXへ送信されたメールがエラーとなった場合は、エラーメールが送信元へ返されます。エラーメールは、Return-path:へ送信されますが、Return-path:が無い場合は、From:へ返されます。従いまして、これらのアドレスは、実在するアドレスの必要があり、定期的にメールボックスをご確認ください。

■ メールの再送信機能

送信されたメールがエラーとなった場合、InterFAX側では処理はおこなっておりません。メールの再送信でのご対応が必要となります。従いまして、お客様のシステムに、メール再送信機能をお持ちいただくことをお勧めしております。

メール再送信機能の付加が難しい場合は、メール送信時にCC:やBCC:で、お客様管理のメールアドレスを指定していただきますと、InterFAX宛と同時に、容易にお客様側にもメールをバックアップでき、送信確認や再送信することが可能です。

例: TO: 03-4477-2288@fax.tc
 CC: faxbackup@abccompany.co.jp

上記の方法により、メール送信後でも次の対応が可能となります。

- 1) メールの再送信
- 2) お客様側システムのメール送信記録とInterFAX側送信履歴との比較

システム開発 & 運用のポイント(送信サービス-3)

■ 通知メールについて

FAX送信結果を通知メールで受信し、その内容をシステムで確認される場合、あくまで「メール」として受信、内容をお読みください。通知メールのフォーマットは、メール本文がUTF-8/base64でエンコードされます。(Content-Type: text/plain; charset="utf-8")

従いまして、メールを正しく読むためには、base64でデコードする必要があります。メールソースを直接読まれても、メール本文がbase64でエンコードされているため送信結果等が正しく読めない可能性があります。

但し、CSVフォーマットを選択された場合は、FAX送信時のメール件名がASCII文字のみの場合は、通知メールの本文は、US-ASCIIとなります。(Content-Type: text/plain; charset="us-ascii")。

■ 送信メールの「件名」の活用

送信履歴には、送信日時(開始・終了)、件名、送信元アドレスおよびステータスが表示されます。例えば、発注書を大量に送信し、エラーが数件発生した場合で、再送信等何らかのアクションが必要な場合、件名にユニークな記号や番号(例えば、発注書番号等)を記述しておく、容易にトランザクションを特定することができます。同一FAX番号へ複数回送信した場合は、FAX番号でのトランザクション特定は難しくなります。

■ セキュリティ機能の活用

InterFAXでは、S/MIMEに対応する電子署名と暗号化機能を送受信サービス共に提供。

- ・ 電子署名: メールアドレスと電子証明書を照合し、一致しなければメール自体を受け付けない仕組みです。不正利用を防止。
- ・ 暗号化: InterFAXの証明書を取得していただき、その証明書でメールを暗号化して送受信することで情報漏洩を防止。

* メールソフトや開発ツールの仕様により、送信メールアドレス単位での電子証明書(デジタルID)が必要となる場合があります。

尚、システムから送信で、S/MIMEのインプリメントが難しい場合、ゲートウェイ型の電子署名、暗号化サーバの利用も効果的です。

加えて、InterFAXでは、[SMTPS\(SMTP over SSL/TLS\)](#)にも対応しています。TLS(port465)またはSTARTTLS(port25)により通信を暗号化してメール送信ができます。* お客様ご利用のメールサーバがTLSまたはSTARTTLSに対応している必要があります。

システム開発 & 運用のポイント(受信サービス-1)

■ 受信履歴の活用

InterFAXでは、アカウント管理Webと呼ばれるWeb管理システムを全てのユーザ様に提供しております。アカウント管理Web内で、FAX受信履歴をご確認いただくことが可能です。履歴はほぼリアルタイムで更新されます。

InterFAX受信サービスでは、FAX受信後にメールにてFAXイメージファイルを転送しますが、メールがエラーとなる場合があります。万が一、メールがエラーとなりましても、受信履歴にてFAXイメージ(TIFFまたはPDF)をダウンロードしていただくことができます。

従いまして、受信履歴を定期的にご確認いただくことを推奨いたします。

尚、FAXイメージのファイル名は、アカウント管理Webの受信履歴ページからダウンロードした場合と、受信転送メールに添付されてきたファイル名は異なります。

・アカウント管理Webの受信履歴ページからダウンロードしたファイル名: InterFAXで管理されたユニークな番号(受信毎に固有)

・受信転送メールに添付のファイル名: FAXを受信するサーバで仮付与。

※ 複数ある受信サーバでそれぞれ付与するため、ファイル名が同一となる場合もありますので、保存される場合は、貴社管理用のファイル名に変更されることを推奨します。

■ 複数アドレスへのメール転送

InterFAXより送信するメールがエラーとなる場合がございます。エラーの原因は次のことが考えられます。

- 1) メールアドレスの間違い
- 2) お客様のメールボックスが一杯の場合
- 3) お客様のメール環境に何らかの異常がある場合
- 4) お客様のメールサーバ等でのスパム(迷惑メール)対策で、InterFAXからのメールをスパムと判断する場合

InterFAX受信サービスからのメールがエラーとなる場合は、上記の点をご確認ください。

以上のように、メールはエラーとなる可能性が常にごございます。つきましては、バックアップとして他のメールサーバが管理するアドレス(例えば、別のドメインやプロバイダ)にも同時に送信することを推奨します。

■ セキュリティ機能の活用

InterFAXでは、S/MIMEに対応する電子署名と暗号化機能を送受信サービス共に提供。

受信サービスでは、受信メールの暗号化や電子署名付加の機能もご利用いただけます。暗号化では、メール受信者の電子証明書が必要です。電子署名は、メール送信者(InterFAX側)が電子署名を付加したメールを指定アドレスへ送信します。

Web Service (SOAP) でのご利用

InterFAXサービスでは、メールでの送信に加え、**XML Web Service**による送受信をおこなうことができます。

Web Serviceでは、メールと違いリアルタイムトランザクションとなり、送信ステータスや送受信履歴をリアルタイムで取得することができます。

例えば、FAX送信後に定期的にステータスを確認し、エラーの場合は、10分後に再送信をする等、一連の業務を自動化し、効率的なFAX送受信アプリケーションを構築することができます。

また、開発言語を選ばず、**Java、.NET、Perl、PHP、C#**等お客様の開発環境をご利用できます。

Web Serviceでは、**SSLによるセキュア環境**でご利用いただけます。

Web Service プログラマーズ・ガイド

Web Service - 送信

InterFAX Webサービスのご利用で、FAX送信アプリケーションの構築を効率的におこなうことができます。Webサービスへは、.NETやJava、C#、ASP、PHP、Perl等の開発環境で利用できる業界標準SOAPインタフェースでアクセスできます。

■ クイックFAX送信

早く、簡単にFAXを送信するために、**SendCharFax**メソッドでテキストファイル (US-ASCII) を送信する方法と、**Sendfax**メソッドでバイナリファイル (MS-Office、PDF、HTML等) を送信する方法が用意されています。これらのメソッドは、最小限のパラメータで、FAX送信要求ができるのが特徴です。(但し、ドキュメント数は1つ、宛先は一ヶ所に限定されます。) InterFAX Webサービスへのメッセージ要求が成功すると、**SendCharfax** と **Sendfax** メソッドでは、トークンを返します。これは、そのFAX要求の状態 (ステータス) やイメージを取得する際に使用される値です。このトークンを、TransactionID (トランザクションID) と呼び、InterFAXシステムにおけるクライアントが要求したFAX送信毎の固有の識別子 (ID) です。メッセージ要求が失敗した場合は、エラーコードが返されます。

■ 高度な機能を利用したFAXの送信

パラメータを指定することで、送信毎に高度な送信機能をご利用できます。

- ★ 複数ドキュメントの送信
- ★ 複数宛先への送信
- ★ 日時指定の送信
- ★ CSIDや通知メール送信先アドレスの指定
- ★ 用紙サイズ、用紙向き (縦・横) や解像度 (標準・ファイン) の設定をおこなえます。
- ★ 参照用として“Subjectフィールド”の指定 (要求メッセージのステータスを入手する際に使用したり、送信履歴で表示できます。
例えば、注文書をFAXする場合等は、注文書番号をSubjectに指定すると、注文書番号でクエリ検索をおこなったり、送信履歴で表示したり、後々の参照が便利になります。)

拡張性のあるFAX送信を行う場合、**SendfaxEx_2**メソッドの使用を推奨します。InterFAX Webサービスへのメッセージ要求が成功した後で、**SendfaxEx_2** メソッドは、FAXのステータスを問い合わせる際に使用するTransactionIDを返します。メッセージ要求が失敗した場合は、エラーコードが返されます。

■ 送信FAXステータスの入手

要求済メッセージのステータスを状況別に検索する場合、**FaxStatus**メソッド、もしくは**FaxQuery**メソッドが用意されています。**FaxStatus**と**FaxQuery**は、一度の要求で、複数トランザクションのステータスを戻すことができます。

■ 送信FAXイメージの取得

送信したFAXのイメージは、**GetFaxImage** メソッドで獲得できます。**GetFaxImage**は、一つのトランザクションIDで要求すると、そのトランザクションのFAXイメージ (TIFF形式) が1個のファイルでダウンロードされます。(G3 TIFF形式で、複数ページでも1ファイルです) そのイメージファイルは、保存したり、Windowsに含まれるビューワで開いたり印刷ができます。

SOAPメソッド・リファレンス

Web Service - 送信

■ FAX送信サービス 要求可能なメソッドの概要

メソッド	概要
SendCharFax	ASCIIテキストをFAX送信する最もシンプルなメソッド。また、HTMLおよびPostScriptの構造化テキストにも対応します。(日本語等2バイト文字は使用できません。)
SendFax	シンプルにファイルをFAX送信するメソッド。オプション設定する場合は、SendfaxEx2を利用。
SendfaxEx_2	豊富な機能が盛り込まれたFAX送信メソッド。FAX送信の詳細な制御を行えます。
Chunk Upload チャンク(分割)アップロード	容量の大きいファイルをInterFAX Webサービスにアップロードする方法です。この方法を用いることで、タイムアウトを起こすことなく、大容量ドキュメントのアップロードを可能にします。個々のファイルサイズが、250KB以上の場合使用を推奨、また、3MB以上の場合必須。
FaxStatus	最小限の情報(11項目)に限定し、素早くFAX送信の処理状況を取得できます。
FaxQuery	"Verb"で指定された送信トランザクションの詳細情報を取得します。
FaxQuery2	高度且つ、柔軟に送信トランザクションの詳細情報を取得するメソッドです。"Verb"と"VerbData"で問い合わせします。
GetFaxImage	指定したトランザクションで、生成されたFAXイメージ(TIFFファイル)を取得します。
ReSendFax	過去に送信したFAXを再送するメソッド。
HideFax	アカウント管理Webの送信履歴よりトランザクションを非表示にします。(一旦非表示にしたトランザクションは、再度表示はできません。)
CancelFax	このメソッドでは、処理中の送信トランザクションをキャンセルできます。但し、送信中のトランザクションはキャンセルできません。

SOAPサンプルコード(送信サービス)

Web Service - 送信

Java: SendCharFax - テキストをFAX送信

```

import cc.interfax.www.SendCharFax;
import cc.interfax.www.SendCharFaxResponse;
public class SendCharFaxTest {
    public void run() throws Exception {
        cc.interfax.www.InterFaxSoapStub theBinding;
        try {
            theBinding = (cc.interfax.www.InterFaxSoapStub)
                new cc.interfax.www.InterFaxLocator().getInterFaxSoap();
        }
        catch (javax.xml.rpc.ServiceException jre) {
            if(jre.getLinkedCause()!=null)
                jre.getLinkedCause().printStackTrace();
            throw new RuntimeException("JAX-RPC ServiceException caught: ");
        }
        // Time out after a minute
        theBinding.setTimeout(60000);
        // Send a simple text fax using the InterFax sendCharFax() web service
        System.out.println("Sending Fax using sendCharFax()");
        SendCharFax theParams = new SendCharFax(TestConstants.USERNAME,
            TestConstants.PASSWORD,
            TestConstants.FAX_NUMBER,
            "This is a test fax message.",
            "TXT");
        SendCharFaxResponse theResponse = theBinding.sendCharFax(theParams);
        long theReturnCode = theResponse.getSendCharFaxResult();
        System.out.println("sendCharFax() call returned with code: " + theReturnCode);
    }
    public static void main(String[] anArgs) {
        try {
            new SendCharFaxTest().run();
        } catch(Exception theE) {
            System.out.println("Error encountered while running SendCharFaxTest:");
            theE.printStackTrace();
        }
    }
}

```

Perl: SendFax - バイナリファイルをFAX送信

```

#!/***** Settings begin *****/
my $username = ""; # Enter your Interfax username here
my $password = ""; # Enter your Interfax password here
# Enter the destination fax number here in the format:
# +[country code][area code][fax number] for example: +1212555487
my $faxNumber = "";
my $filename = 'sample.pdf';
my $filetype = 'pdf';
my $NS = 'http://www.interfax.cc';
#!/***** Settings end *****/

# Read file as binary in 1K blocks
my ($file, $data);
open FILE, $filename;
while (read FILE, $data, 1024 != 0) {
    $file .= $data;
}
close FILE;
$file = encode_base64($file);

my $client = SOAP::Lite
    ->uri($NS)
    ->on_action( sub { join '/', $NS, $_[1] })
    ->proxy('https://ws.interfax.net/dfs.asmx?wsdl');

my $result = $client
    ->call(SOAP::Data->name('Sendfax')->attr({xmlns => $NS}) =>
        SOAP::Data->name('Username')->value($username)->type(""),
        SOAP::Data->name('Password')->value($password)->type(""),
        SOAP::Data->name('FaxNumber')->value($faxNumber)->type(""),
        SOAP::Data->name('FileData')->value($file)->type(""),
        SOAP::Data->name('FileType')->value($filetype)->type(""))
    );

if ( $result->fault ) {
    print $result->faultstring . "\n";
} else {
    if ( $result->valueof('//SendfaxResult') > 0 ) {
        print "Success. TransactionID=" . $result->valueof('//SendfaxResult') . "\n";
    } else {
        print "Error, return code=" . $result->valueof('//SendfaxResult') . "\n";
    }
}
}

```

Web Service メソッド機能比較・1

Web Service - 送信

■ FAX送信サービス メソッドの比較

機 能	SendCharFax	Sendfax	SendfaxEx_2
複数の宛先送信			●
テキストファイルの送信 (TXT、HTML等)	●	●	●
バイナリファイルの送信 (DOC、XLS、PDF等)		●	●
8-bit 文字を用いたHTMLファイル (強調文字等)		●	●
複数ドキュメントの送信			●
日時指定送信			●
CSID指定 (デフォルト設定値より優先)			●
参照用 Subject(件名)の入力			●
通知メールデフォルト送信先の指定 (デフォルト設定値より優先)			●
用紙サイズの指定: A4/Lette/Legal/B4 (デフォルト設定値より優先)			●
用紙方向の指定: 縦/横 (デフォルト設定値より優先)			●
送信解像度の指定: 標準/ファイン (デフォルト設定値より優先)			●
送信品質の指定: ハーフトーン/白黒 (デフォルト設定値より優先)			●

Web Service メソッド機能比較・2-1

Web Service - 送信

■ FAX送信サービス ステータス・クエリメソッドの比較 2-1

機 能	FaxStatus	FaxQuery	FaxQuery2
取得するトランザクション数の制御	●	●	●
日時範囲指定での検索			●
宛先FAX番号での検索			●
通知メール送付先アドレスでの検索			●
TransactionIDでの検索	●	●	●
Parent TransactionIDでの検索		●	●
FAXステータス(成功、エラー等)での検索			●
合計トランザクション数の取得	●	●	●
クエリで返されたレコード数の取得	●	●	●
トランザクションの開始日時の取得	●	●	●
日時指定時の日時の取得	●	●	●
トランザクションの終了日時の取得	●	●	●
送信先FAX番号の取得	●	●	●
送信先CSIDの取得	●	●	●

Web Service メソッド機能比較・2-2

Web Service - 送信

■ FAX送信サービス ステータス・クエリメソッドの比較 2-2

機 能	FaxStatus	FaxQuery	FaxQuery2
送信完了ページ数の取得	●	●	●
送信結果の取得	●	●	●
送信時間の取得	●	●	●
Subjectの取得	●	●	●
送信要求ページ数の取得	●	●	●
Parent TransactionIDの取得		●	●
UserIDの取得		●	●
通知メール送付先アドレスの取得		●	●
送信料金の取得 (Unit x ページ単価)		●	●
ページサイズの取得		●	●
用紙方向の取得		●	●
送信解像度 (ファイン/標準) の取得		●	●
送信品質 (ハーフトーン/白黒) の取得		●	●

InterFAX Webサービスのご利用で、FAX受信アプリケーションの構築を効率的におこなうことができます。Webサービスへは、.NETやJava、C#、ASP、PHP、Perl等の開発環境で利用できる業界標準SOAPインタフェースでアクセスできます。

■ 受信FAXリストの取得

受信FAXリストは GetListメソッドをコールして取得できます。

GetList のリクエストはリストタイプ(全て、または未読)と MaxItem (リスト中のアイテムの最大数は100まで)と共に提示されます。

リストは MessageItems からなり、クライアントからのリファレンスがついたアレーのフォーマットになります。

■ 受信FAX画像の取得

FAX画像は GetImageChunk メソッドを使用して取得できます。

GetImageChunk リクエストはMessageID、画像がはいるバッファ、バッファサイズ(受信したいチャンクサイズ)、と画像がどのバイトからスタートするかの情報と共に提示されます。画像の最終バッファが送信された後、メッセージは既読とマークされます。

■ 受信メッセージを既読にマークする

クライアントはメッセージのステータスを「未読」から「既読」へ、またその反対にもできます。受信メッセージは自動的に「未読」にマークされます。クライアントが画像ファイルを完全にダウンロードすることにより「既読」となるか、MarkMessage メソッドを使用して「既読」とすることができます。MarkMessage リクエストは MessageID と Boolean (true:メッセージを「既読」にマーク、false:「未読」にマーク)と共に提示されます。

SOAPメソッド・リファレンス

Web Service – 受信

■ FAX受信サービス 要求可能なメソッドの概要

メソッド	概要
GetList	受信FAXのリストを取得するメソッドです。
GetList2	受信FAXのリストをより柔軟に、詳細に取得するメソッドです。
GetImageChunk	受信FAXのイメージを取得するメソッドです。GetListで取得した情報(MessageIDおよびMessageSize)が必要です。
GetImageChunkEx2	受信FAXのイメージとそのファイル形式を取得するメソッドです。
MarkMesssage	受信FAXに「既読」マークを付加するメソッドです。既読となった受信FAXのトランザクションはGetListで取得できません。
GetInboundLogEmails	受信FAXが転送されたメールアドレスのリストを取得するメソッドです。メールの送信ステータス(成功、失敗)を含みます。
ResendInboundToEmail	受信FAXを指定したメールアドレスへ再転送するメソッドです。

SOAPサンプルコード(受信サービス)

Web Service – 受信

VB.NET: 受信したFAXイメージの取得

```

Dim objSOAP As MSSOAPLib30.SoapClient30
Dim FaxStatusXML() As Object
Dim objNL As MSXML2.IXMLDOMNodeList

Private Sub Form_Load()

Me.cmbListType.ListIndex = 0
'
' Initialization of SOAP object
'
Set objSOAP = New MSSOAPLib30.SoapClient30
objSOAP.MSSoapInit "https://ws.interfax.net/Inbound.asmx?wsdl"

End Sub

Private Sub cmdGetList_Click()

Dim lngResult As Long
Dim i As Integer, j As Integer
Dim strPages As String, strMessageId As String
Dim blsEmptyArray As Boolean

On Error Resume Next
lngResult = objSOAP.GetList(Me.txtUsername.Text, Me.txtPassword.Text, Me.c
10, FaxStatusXML)
blsEmptyArray = (Err.Number = -2147352565)

If lngResult <> 0 Then
MsgBox ("Bad status=" & lngResult)
ElseIf blsEmptyArray Then 'check if no results - TBD; IsEmpty does not see
MsgBox ("No data returned")
Else
Me.List1.Clear
Me.List1.AddItem "MessageID" & vbTab & "Pages"
For i = 0 To UBound(FaxStatusXML)
Set objNL = FaxStatusXML(i)
For j = 0 To objNL.length - 1
If objNL.Item(j).baseName = "MessageID" Then
strMessageId = objNL.Item(j).Text
ElseIf objNL.Item(j).baseName = "Pages" Then
strPages = objNL.Item(j).Text
End If
Next
Me.List1.AddItem strMessageId & vbTab & vbTab & strPages
Next
Me.Refresh
End If
End Sub

```

VB.NET: 受信FAX履歴の取得

```

Private mobjIF As New Interfax.Inbound()
Private Items() As Interfax.MessageItem
Private b() As Byte

Private Sub btnGetList_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles btnGetList.Click
Dim intResult As Integer

Me.lstMsgs.Items.Clear()
Me.lstMsgs.Items.Add("Getting data from server, please wait...")
Me.ComboBox1.SelectedIndex = 0
Me.Refresh()
intResult = mobjIF.GetList(Me.txtUser.Text, Me.txtPass.Text, Interfax.ListType.NewMessages, Me.txtNumItems.Text, Items)
Select Case intResult
Case 0
ShowList()
Case Else
MsgBox("An error occured. Code=" & intResult)
End Select
End Sub

Private Sub ShowList()
Dim i As Integer
Dim strLine As String

Me.lstMsgs.Items.Clear()
For i = 0 To Items.Length - 1
With Items(i)
strLine = "(" & (i + 1).ToString & ") " & .Pages & " Page(s) Received at " & .ReceiveTime & " Caller ID=" &
CallerID & "]" CSID=[" & .RemoteCSID & "]"
End With
Me.lstMsgs.Items.Add(strLine)
Next
Me.Refresh()
End Sub

Private Sub lstMsgs_DoubleClick(ByVal sender As Object, ByVal e As System.EventArgs) Handles lstMsgs.DoubleClick
Dim i As Integer
Dim intResult As Integer
StatusBar1.Text = "Retrieving message, please wait"
i = Me.lstMsgs.SelectedIndex

b = Nothing 'Do not send data for outbound-only parameter
intResult = mobjIF.GetImageChunk(Me.txtUser.Text, Me.txtPass.Text, Items(i).MessageID, False, Items(i).MessageSize, 0, b)
Select Case intResult
Case 0
ShowImage()
Case Else
MsgBox("An error occured. Code=" & intResult)
End Select
End Sub

Private Sub ShowImage()
Dim strPath As String
strPath = IO.Path.GetTempFileName & "." & Me.ComboBox1.Text
Dim fs As New IO.FileStream(strPath, IO.FileMode.Create)
StatusBar1.Text = FormatNumber(b.Length / 1024, 0, TriState.True, TriState.False, TriState.True) & " KBytes received"
fs.Write(b, 0, b.Length)
fs.Close()
fs = Nothing
System.Diagnostics.Process.Start(strPath)
End Sub
End Class

```