

【セッションNo. 3】

知って得する！ 現役ヘルプデスクが答えるDelphiテクニカルエッセンス7.0

株式会社ミガロ.
RAD事業部 技術支援課
吉原 泰介

【アジェンダ】

Q1. OpenOfficeOrgを利用するには？

Q2. メール送信機能を作成するには？

おまけ資料: **メール関連補足**

■ Q1. OpenOfficeOrgを利用するには？

【質問】

DelphiからOpenOfficeをExcelのように利用することはできますか？

【回答】

OpenOfficeのCalcもOLEで操作することができるので、Excelと同様にセル編集やファイル出力といった実装をすることができます。
またExcelにはないPDF出力などの便利な機能も利用することができます。

■ Q1. OpenOfficeOrgを利用するには？

OpenOffice(正式名称: OpenOffice.org)とは
オープンソースのオフィスソフトの一つ。

Microsoft社の「Microsoft Office」と高い互換性があり、
LinuxやWindowsなど様々なOSで動作する。

オープンソースソフトのため、誰でも自由に無償で利用することが
でき、また、興味があれば開発に参加することもできる。

(IT用語辞典引用)

ja: OpenOffice.org 日本語プロジェクト
<http://ja.openoffice.org/>

多くの企業・自治体・教育機関が導入!
みんなで創る無料総合オフィスソフトウェア

OpenOffice.org
ダウンロード
バージョン3.2.1 2010年6月

表計算 文書作成
プレゼンテーション 図形描画
データベース 数式

導入事例
OpenOffice.orgの様々な導入事例を取り上げております。高いコスト削減効果を生み出した過去の事例や現在進行中のプロジェクト等参考になる情報が満載です。

プロジェクトに参加
自分でできることからコツコツと！プログラマーだけでなく、マーケティングや翻訳など喜及活動にご協力頂ける有為を募集しております。

ja.openoffice.org
jaopenoffice.org
もうひとつ、006番の対応でSCAなどのリーガルの処理が速くなるというがある。これもモチベーション低下の要因の一つ。半年かかるのもある。
3 days ago
論理的にクリアな文章を書くというのは、理系大卒だと半論書くからもう少しマシなんですが、

■ Q1. OpenOfficeOrgを利用するには？

OpenOfficeで利用できる主な機能

Writer ワードプロソフト

Math 数式作成ソフト

Calc 表計算ソフト

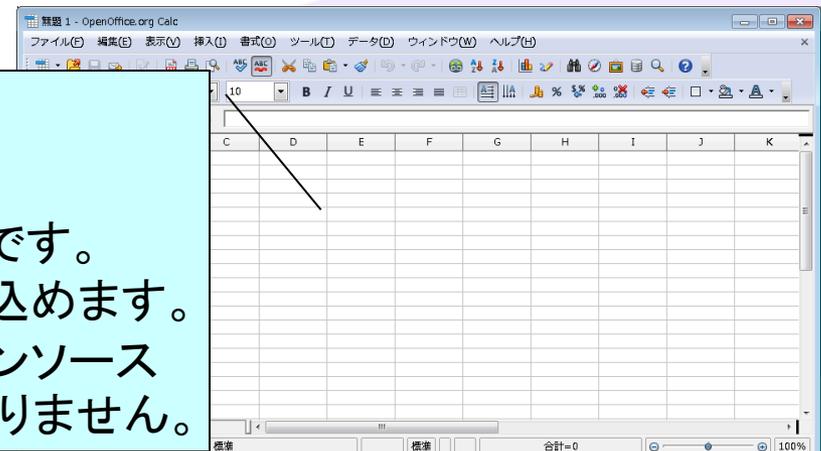
Draw 図形描画ソフト

Impress プレゼンテーションソフト

Base データベースソフト

今回は表計算のCalcを利用

見た目も操作もほぼMicrosoftのExcelと同じです。
ファイル拡張子は.odsです。xlsやxlsxも読み込めます。
無償で使うことができますが、もちろんオープンソース
なので正式な製品サポートがあるわけではありません。



■ Q1. OpenOfficeOrgを利用するには？

以降のDelphiからのCalc操作のサンプルは
OpenOffice.org 3.2.1をインストールした環境での動作です。

The screenshot displays the OpenOffice Calc application window titled '無題 1 - OpenOffice.org Calc'. The spreadsheet shows a table with columns A through E and rows 1 through 6. Cell A1 is selected. The text 'Delphiテクニカルエッ' is visible in cell C2. The interface includes a menu bar (File, Edit, View, Insert, Format, Tools, Data, Window, Help) and a toolbar. To the left of the Calc window is a sidebar with a '基本操作' (Basic Operations) section containing buttons for '起動' (Start) and '終了' (End), and an '出力操作' (Output Operations) section with buttons for '保存' (Save), 'Excel保存' (Save as Excel), 'PDF保存' (Save as PDF), and '印刷' (Print). Overlaid on the Calc window is the Adobe Reader window titled 'test.pdf - Adobe Reader', showing a PDF document with the text 'Delphiテクニカルエッセンス'.

サンプルは説明のために
通常1ボタンで行う処理を分割していますが、
実際のシステムではこれらをCalc(Excel)を
表示せずに出力することが多いです。

■ Q1. OpenOfficeOrgを利用するには？

今回のサンプルでは処理をボタンで分割しているのでサービス、ファイル関係の変数はグローバルで定義します。

Calc操作を行うための共通変数・関数(ソース)

```
private
  vOpenOffice : Variant; //サービスマネージャ
  vStarDesktop: Variant; //サービス
  vDocument   : Variant; //ドキュメント
  vSheet      : Variant; //シート
  function dummyArray: Variant; //空プロパティ配列設定用
```

...

```
//空のプロパティ配列を返却
function TForm1.dummyArray: Variant;
begin
  Result:= VarArrayCreate([0, -1], varVariant);
end;
```

■ Q1. OpenOfficeOrgを利用するには？

CalcのAPIでのPropertyValue

- Calcの操作上で使用するAPIのパラメータにはPropertyValueという構造体がよく使われます。このパラメータは、ドキュメントのオープンと作成に対するさまざまなオプションを OpenOffice.org に提供します。

PropertyValue 構造体には、個々のオプションごとにオプション名(Name)と設定値(Value)をセットで格納します。

ファイル保存のAPIを利用する例

```
//パラメータ用プロパティ配列生成
vaProperties := VarArrayCreate([0, 0], varVariant);
//プロパティ用オプション生成
vProperty:= vOpenOffice.Bridge_GetStruct(' com. sun. star. beans. PropertyValue');
//プロパティ用オプション値設定
vProperty.Name := 'FilterName';
vProperty.Value := 'MS Excel 97';
//オプション内容をプロパティ配列に設定
vaProperties[0] := vProperty;
//ファイルを保存
vDocument.StoreToURL('file:/// ' + sFileName, vaProperties);
```

PropertyValueを作成してAPIへ渡す

■ Q1. OpenOfficeOrgを利用するには？

CalcのAPIでのPropertyValue(続き)

- ただPropertyValue構造体がパラメータの場合でも、オプションを特に設定する必要がないのであれば毎回Var配列を定義するのも面倒なので、先に実装したdummyArrayで代用します。

ファイル保存のAPIを利用する例

```
//ファイルを保存
```

```
vDocument.StoreToURL('file:/// ' + sFileName, dummyArray);
```

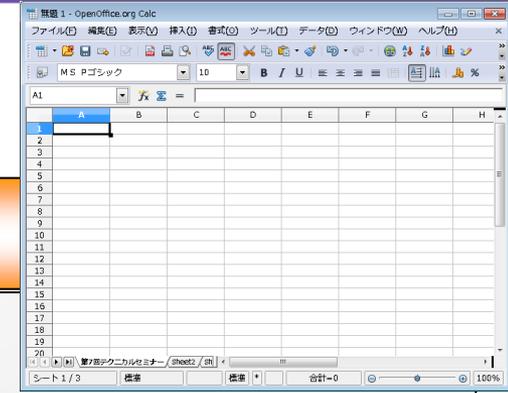
オプション設定が不要な場合、
dummyArrayで代用する

Q1. OpenOfficeOrgを利用するには？

Calcを起動する(ソース)

```
procedure TForm1.btnCalcStartClick(Sender: TObject);
begin
    //サービスマネージャ生成
    vOpenOffice := CreateOleObject(' com. sun. star. ServiceManager' );
    //サービス生成
    vStarDesktop := vOpenOffice.CreateInstance(' com. sun. star. frame. Desktop' );
    //新規ドキュメントを生成する場合
    vDocument := vStarDesktop.LoadComponentFromURL(' private:factory/scalc'
                                                    , '_blank'
                                                    , 0
                                                    , dummyArray);

    //既存ドキュメントを読み込む場合
    // vDocument := StarDesktop.LoadComponentFromURL(' file:///C:/Temp/Test. ods'
    //
    //
    //
    //アクティブなSheetを設定(1番目のSheet)
    vSheet := vDocument.Sheets.GetByIndex(0);
    //Sheet名変更
    vSheet.Name := '第7回テクニカルセミナー';
end;
```



新規ドキュメントは
'private:factory/scalc'を指定

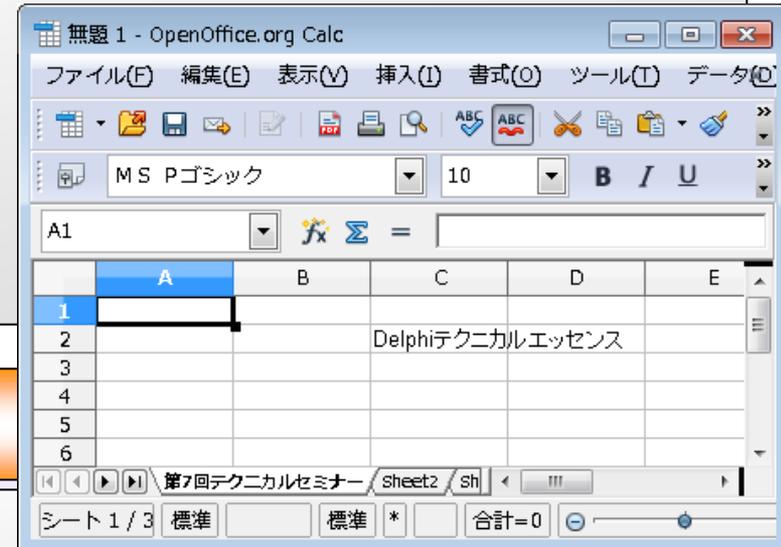
_blank:
すでにファイルが開かれていても新しいウィンドウを使って指定のファイルを開く

既存ドキュメントを開く場合はこちら

■ Q1. OpenOfficeOrgを利用するには？

Calcを終了する(ソース)

```
procedure TForm1.btnCalcEndClick(Sender: TObject);
begin
    //サービス終了
    vStarDesktop.terminate;
    //サービス破棄
    vStarDesktop := unassigned;
    //サービスマネージャ破棄
    vOpenOffice := unassigned;
end;
```



Calcでセル値を編集する(ソース)

```
procedure TForm1.btnCalcCellSetClick(Sender: TObject);
begin
    //セルに値を代入
    vSheet.getCellRangeByName('C2').String := 'Delphiテクニカルエッセンス';
end;
```

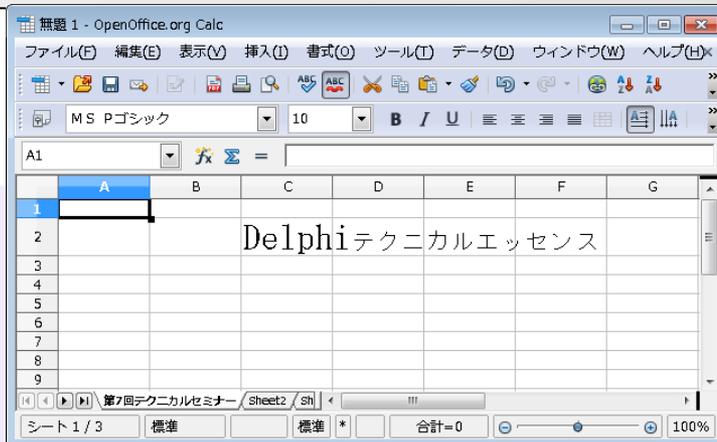
セルを指定してアクセス

■ Q1. OpenOfficeOrgを利用するには？

Calcでフォントを変更する(ソース)

```
procedure TForm1.btnCalcFontClick(Sender: TObject);
begin
  //半角英数記号フォント・サイズ設定
  vSheet.getCellRangeByName('C2').CharFontName      := 'MS 明朝';
  vSheet.getCellRangeByName('C2').CharHeight        := 24;
  //日本語文字（マルチバイト文字や半角カナ文字）フォント・サイズ設定
  vSheet.getCellRangeByName('C2').CharFontNameAsian := 'MS ゴシック';
  vSheet.getCellRangeByName('C2').CharHeightAsian   := 16;
end;
```

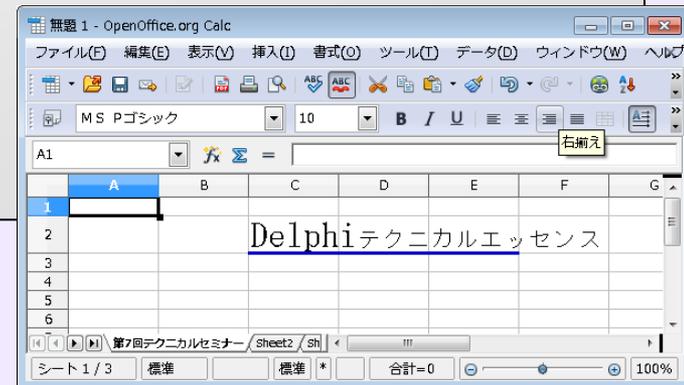
CharXXXでは日本語文字には適用されないので
CharXXXAsianで指定



■ Q1. OpenOfficeOrgを利用するには？

Calcで罫線を引く(ソース)

```
procedure TForm1.btnCalcBorderClick(Sender: TObject);
var
  vBorder : Variant; //罫線処理用
begin
  //罫線のオブジェクトを生成
  vBorder:= vOpenOffice.Bridge_GetStruct(' com. sun. star. table. BorderLine' );
  //罫線の色の設定
  vBorder.Color:= RGB(200, 0, 0); //青
  //罫線の太さを設定
  vBorder.OuterLineWidth:= 100;
  //罫線をセット (C2からE2の下線)
  vSheet.getCellRangeByName(' C2:E2' ).BottomBorder:= vBorder;
end;
```



■ Q1. OpenOfficeOrgを利用するには？

Calcでファイルを保存する(ソース)

```
procedure TForm1.btnCalcSaveClick(Sender: TObject);
var
  sFileName      : String;    //保存ファイル名編集用
begin
  //保存ダイアログ実行
  if SaveDialog1.Execute then
  begin
    //ダイアログの保存パスを/形式に置換
    sFileName := StringReplace(SaveDialog1.FileName, '¥', '/', [rfReplaceAll]);
    //ファイルを保存
    vDocument.StoreToURL('file:/// ' + sFileName, dummyArray);
    //変更保存等ダイアログを表示しない
    vDocument.Modified := False;
  end;
end;
```

保存パスはURL形式で
設定する。

× ...¥...¥Test.ods

○.../.../ Test.ods

保存パスを指定して
保存APIを実行

Calcを終了する際に
変更保存ダイアログが
表示させない設定

■ Q1. OpenOfficeOrgを利用するには？

Calcでxls保存する(ソース)

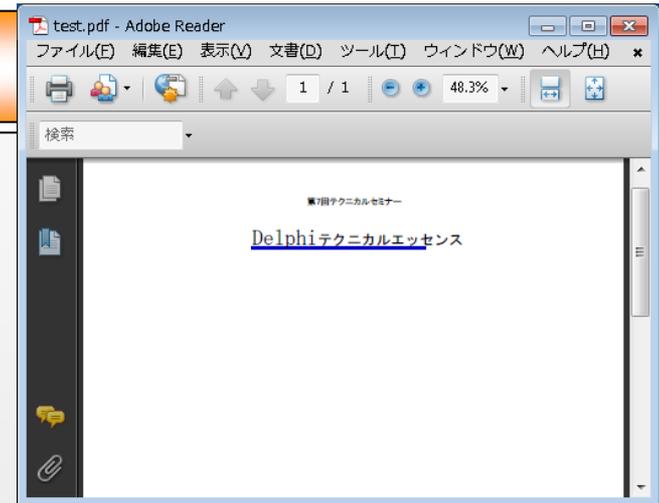
```
procedure TForm1.btnCalcXlsClick(Sender: TObject);
var
  vaProperties : variant; //保存パラメータ用<配列>
  vProperty   : variant; //保存オプション用<配列値>
  sFileName   : String;  //保存ファイル名編集用
begin
  //保存ダイアログ実行
  if SaveDialog1.Execute then
  begin
    //ダイアログの保存パスを/形式に置換
    sFileName := StringReplace(SaveDialog1.FileName, '¥', '/', [rfReplaceAll]);
    //パラメータ用プロパティ配列生成
    vaProperties := VarArrayCreate([0, 0], varVariant);
    //プロパティ用オプション生成
    vProperty := vOpenOffice.Bridge_GetStruct('com.sun.star.beans.PropertyValue');
    //プロパティ用オプション値設定
    vProperty.Name := 'FilterName';
    vProperty.Value := 'MS Excel 97';
    //オプション内容をプロパティ配列に設定
    vaProperties[0] := vProperty;
    //ファイルを保存
    vDocument.StoreToURL('file:/// ' + sFileName, vaProperties);
    //変更保存等ダイアログを表示しない
    vDocument.Modified := False;
  end;
end;
```

基本はodsと同じ
ですが、xlsで保存する
場合は別形式なので
フィルタープロパティで
指定する必要があります。
Xlsx形式保存はCalcでは不可

■ Q1. OpenOfficeOrgを利用するには？

CalcでPDF保存する(ソース)

```
procedure TForm1.btnCalcPDFClick(Sender: TObject);
var
  vaProperties : variant; //保存パラメータ用<配列>
  vProperty   : variant; //保存オプション用<配列値>
  sFileName   : String;  //保存ファイル名編集用
begin
  //保存ダイアログ実行
  if SaveDialog1.Execute then
  begin
    //ダイアログの保存パスを/形式に置換
    sFileName := StringReplace(SaveDialog1.FileName, '¥', '/', [rfReplaceAll]);
    //パラメータ用プロパティ配列生成
    vaProperties := VarArrayCreate([0, 0], varVariant);
    //プロパティ用オプション生成
    vProperty := vOpenOffice.Bridge_GetStruct('com.sun.star.beans.PropertyValue');
    //プロパティ用オプション値設定
    vProperty.Name := 'FilterName';
    vProperty.Value := 'calc_pdf_Export';
    //オプション内容をプロパティ配列に設定
    vaProperties[0] := vProperty;
    //ファイルを保存
    vDocument.StoreToURL('file:/// ' + sFileName, vaProperties);
  end;
end;
```



基本はodsと同じ
ですが、pdfで保存する
場合は別形式なので
フィルタープロパティで
指定する必要があります

■ Q1. OpenOfficeOrgを利用するには？

Calcで印刷する(ソース)

```
procedure TForm1.btnCalcPrintClick(Sender: TObject);
var
  vaProperties : Variant; //プリンタ指定パラメータ用<配列>
  vProperty    : Variant; //プリンタ指定オプション用<配列値>
begin
  //パラメータ用プロパティ配列生成
  vaProperties := VarArrayCreate([0, 0], varVariant);
  //プロパティ用オプション生成
  vProperty := vOpenOffice.Bridge_GetStruct('com.sun.star.beans.PropertyValue');
  //プロパティ用オプション値設定
  vProperty.Name := 'Name';
  vProperty.Value := 'FinePrint'; //設定したいプリンタ名
  //オプション内容をプロパティ配列に設定
  vaProperties[0] := vProperty;
  //プリンター指定
  vDocument.setPrinter(vaProperties);

  //印刷
  vDocument.print(dummyArray);
end;
```

プリンタを
指定

デフォルトプリンタへの印刷
であればこれだけ

■ Q1. OpenOfficeOrgを利用するには？

Calc上での操作のコーディングが分からないときは・・・

- Excelと同じように操作をマクロで記録できるので、マクロの中身を見てコーディングの参考にできます。

①マクロの記録開始

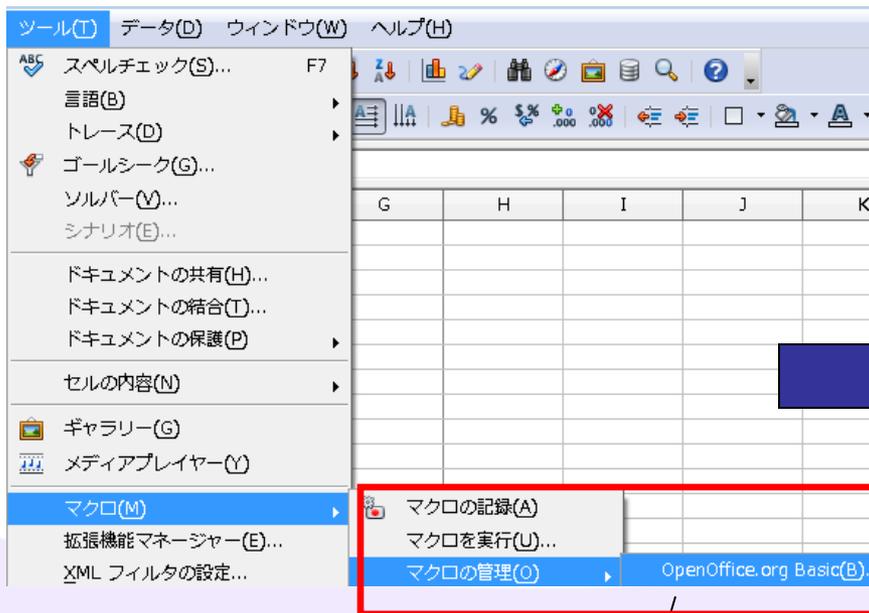
②やりたい操作

③保存

④記録終了

■ Q1. OpenOfficeOrgを利用するには？

Calc上での操作のコーディングが分からないときは・・・



```
REM ***** BASIC *****  
  
sub Main  
rem -----  
rem define variables  
dim document as object  
dim dispatcher as object  
rem -----  
rem get access to the document  
document = ThisComponent.CurrentController.Frame  
dispatcher = createUnoService("com.sun.star.frame.DispatchHelper")  
rem -----  
dim args1(0) as new com.sun.star.beans.PropertyValue  
args1(0).Name = "StringName"  
args1(0).Value = "あいうえお"  
dispatcher.executeDispatch(document, ".uno:EnterString", "", 0, args1())  
rem -----  
dispatcher.executeDispatch(document, ".uno:JumpToNextCell", "", 0, Array())  
rem -----  
end sub
```

⑤ マクロの管理から
保存したマクロを開く

⑥ 操作のマクロが
コーディングの参
考にできます

■Q2.メール送信機能を作成するには？

❗【質問】

Delphi/400のアプリケーションからメールを送信したいのですが、こういった実装が必要でしょうか？

✉【回答】

メールを送信する場合、IndyのTIdSMTPコンポーネントを使用すると比較的簡単にメール送信機能を実装することができます。

※ただしメールはメールサーバや使用するメーカーなど環境によってエンコードなどの調整が必要です。

■Q2.メール送信機能を作成するには？

メール送信サンプルプログラム概要

The screenshot shows a window titled "【Q2】メール送信機能を作成するには?". The interface includes the following fields and controls:

- 送信元アドレス:** tec@migaro.co.jp
- 送信先アドレス:** yoshiwara@migaro.co.jp
- タイトル:** テクニカルセミナー御礼
- 本文:** 本日はテクニカルセミナーにご参加頂き、誠にありがとうございます。
- 添付ファイル:** C:¥Test¥delphi400vD.pdf
- メール送信** button

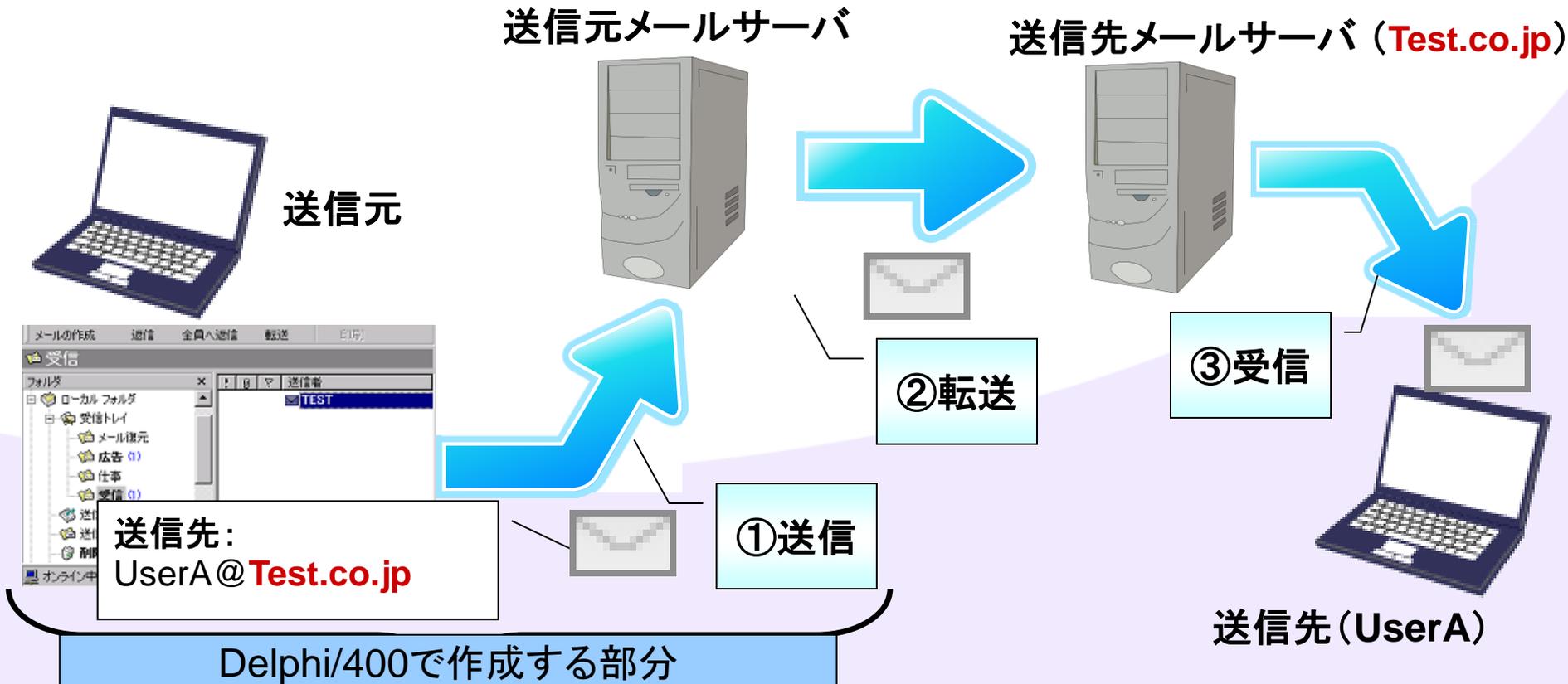
Callouts and annotations:

- 送信元、送信先情報の設定:** Points to the sender and recipient address fields.
- メール内容の設定:** Points to the title and body text fields, listing "タイトル" and "本文" as sub-items.
- 添付ファイルの設定:** Points to the attachment file field.
- メール送信実行:** Points to the "メール送信" button.

サンプルは説明のためにメール内容を入力していますが、システムに組み込む場合は自動送信形式が多いです。

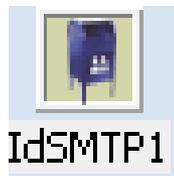
■ Q2.メール送信機能を作成するには？

メール送信の仕組み

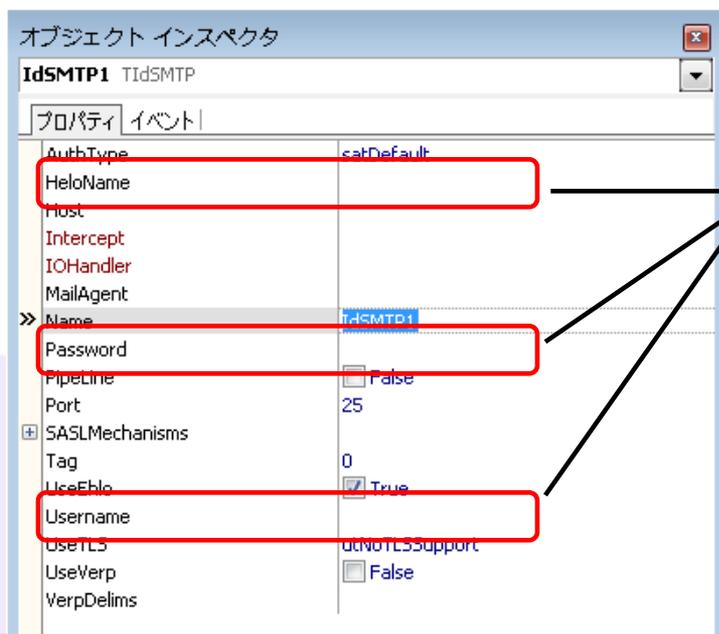


■ Q2.メール送信機能を作成するには？

使用するIndyコンポーネント①



メールサーバ(SMTP)へ接続を行ったり
メールの送信を行うコンポーネント



■ 接続情報 ■

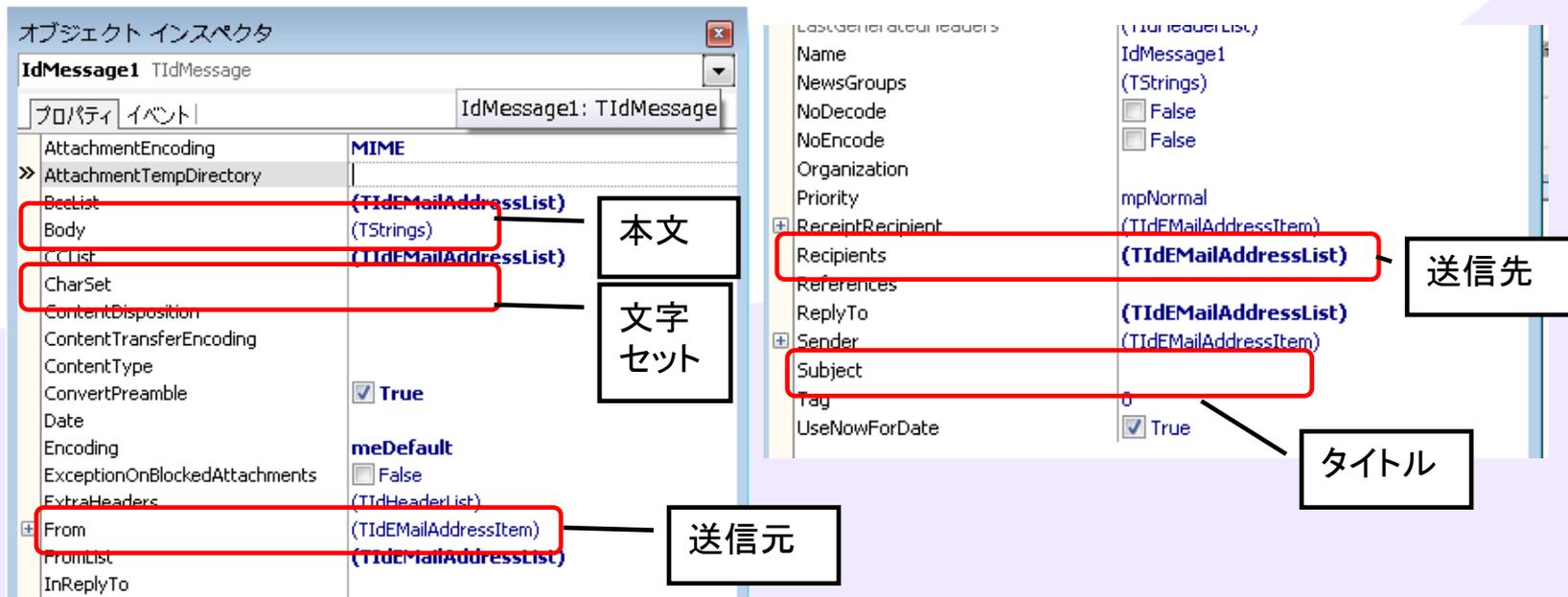
- ・ホスト
- ・ユーザー
- ・パスワード

■ Q2.メール送信機能を作成するには？

使用するIndyコンポーネント②



メールの内容を詳しく制御するための
コンポーネント



オブジェクト インспекタ

IdMessage1 TIdMessage

プロパティ イベント | IdMessage1: TIdMessage

Property	Type	Value	Label
AttachmentEncoding	MIME		
AttachmentTempDirectory			
BccList	(TIdEmailAddressList)		
Body	(TStrings)		本文
CCList	(TIdEmailAddressList)		
CharSet			文字セット
ContentDisposition			
ContentTransferEncoding			
ContentType			
ConvertPreamble	<input checked="" type="checkbox"/> True		
Date			
Encoding	meDefault		
ExceptionOnBlockedAttachments	<input type="checkbox"/> False		
ExtraHeaders	(TIdHeaderList)		
From	(TIdEmailAddressItem)		送信元
FromList	(TIdEmailAddressList)		
InReplyTo			
Name			
NewsGroups			
NoDecode	<input type="checkbox"/> False		
NoEncode	<input type="checkbox"/> False		
Organization			
Priority	mpNormal		
ReceiptRecipient	(TIdEmailAddressItem)		
Recipients	(TIdEmailAddressList)		送信先
References			
ReplyTo	(TIdEmailAddressList)		
Sender	(TIdEmailAddressItem)		
Subject			タイトル
Tag	0		
UseNowForDate	<input checked="" type="checkbox"/> True		

■ Q2.メール送信機能を作成するには？

テキストのメールを送信する(ソース)

```
procedure TfrmQ1.btmMailSendClick(Sender: TObject);
begin
  try
    IdSMTP1.Host      := 'smtp.XXXXX.XX.XX'; //SMTPサーバ名
    IdSMTP1.Port      := 25;                //SMTPポート番号
    IdSMTP1.Username  := 'ユーザー名';     //SMTPユーザー名
    IdSMTP1.Password  := 'パスワード';     //SMTPパスワード
    IdSMTP1.Connect; //SMTP接続

    IdMessage1.Clear;
    IdMessage1.ContentType := 'text/plain'; // コンテンツタイプ
    IdMessage1.CharSet     := 'ISO-2022-JP'; // 文字セット
    IdMessage1.ContentTransferEncoding := 'BASE64'; // BASE64 エンコーディング
```

メールサーバ(SMTP)に接続
・SMTPホスト名を設定
・ポートはデフォルト25を設定
・ユーザー / パスワードの
認証があれば設定
※環境によって設定は異なります

TIdMessageは毎回クリア
連続して送信すると、
送信情報が残ってしまうため

■ Q2.メール送信機能を作成するには？

テキストのメールを送信する(ソース)

画面で入力したメール内容を
TIdMessageに設定

```
IdMessage1.From.Address      := edtSNDAddress.Text;    // 送信元のメールアドレス
IdMessage1.Recipients.EmailAddresses := edtRCVAddress.Text; // 送信先のメールアドレス
IdMessage1.Subject           := edtTitle.Text;         // タイトル
IdMessage1.Body.Text         := memText.Text;         // 本文
IdSMTP1.Send(IdMessage1);    // メール送信
ShowMessage('メール送信完了しました');
except
  ShowMessage('メール送信中にエラーが発生しました');
end;
IdSMTP1.Disconnect; //SMTP切断

end;
```

メール送信はTIdSMTP.Sendメソッドで
内容設定したTIdMessageを送信

■Q2.メール送信機能を作成するには？

Delphi/400からテキストメールを送信

メール送信機能を作成するには？

送信元アドレス tec@migaro.co.jp ①

送信先アドレス yoshiwara@migaro.co.jp ②

タイトル テクニカルセミナーセッション3 ③

本文 知って得する
Delphiテクニカルエッセンス7.0 ④

添付ファイル

メール



受信

テクニカルセミナーセッション3 受信トレイ | x

☆ From tec@migaro.co.jp <tec@migaro.co.jp> ①

To yoshiwara@migaro.co.jp ②

日付 2010年11月22日 14:35

件名 テクニカルセミナーセッション3 ③

知って得する
Delphiテクニカルエッセンス7.0 ④

■ Q2.メール送信機能を作成するには？

Delphi/400からテキストメールを送信

V2009以降

【1】 メール送信機能を作成するには？

送信元アドレス tec@migaro.co.jp ①

送信先アドレス yoshiwara@migaro.co.jp ②

タイトル ～テクニカルセミナーセッション～ ③

本文 知って得する
～Delphiテクニカルエッセンス7.0～ ④

添付ファイル



③、④で文字化けが発生
なぜならUnicodeで処理しているの
にISO-2022-JPでエンコード
しているから

受信

?テクニカルセミナーセッション? 受信トレイ | x

☆ From tec@migaro.co.jp <tec@migaro.co.jp> ①

To yoshiwara@migaro.co.jp ②

日付 2010年11月22日 15:07

件名 ?テクニカルセミナーセッション? ③

知って得する
?Delphiテクニカルエッセンス7.0? ④

■ Q2.メール送信機能を作成するには？

テキストのメールを送信する(ソース)

V2009以降

```
procedure TfrmQ1.btmMailSendClick(Sender: TObject);
```

```
begin
```

```
  try
```

```
    IdSMTP1.Host      := 'smtp.XXXXX.XX.XX';      //SMTPサーバ名  
    IdSMTP1.Port      := 25;                      //SMTPポート番号  
    IdSMTP1.Username  := 'ユーザー名';          //SMTPユーザー名  
    IdSMTP1.Password  := 'パスワード';          //SMTPパスワード  
    IdSMTP1.Connect;                              //SMTP接続
```

```
    IdMessage1.Clear;
```

```
    IdMessage1.ContentType      := 'text/plain';      // コンテンツタイプ
```

```
    IdMessage1.CharSet          := 'UTF-8';          // 文字セット
```

```
    IdMessage1.ContentTransferEncoding  := 'BASE64';      // BASE64 エンコーディング
```

?テクニカルセミナーセッション? 受信トレイ | x

☆ From **tec@migaro.co.jp** <tec@migaro.co.jp>

To yoshiwara@migaro.co.jp

日付 2010年11月22日15:07

件名 ?テクニカルセミナーセッション?

知って得する
?Delphiテクニカルエッセンス7.0?

④

Delphi/400 V2009以降は
Unicodeが標準なので
CharSetを**UTF-8**で設定
④**本文部分が改善**

■Q2.メール送信機能を作成するには？

テキストのメールを送信する(ソース)

V2009以降

```
procedure TfrmQ1.IdMessage1InitializeISO(var VHeaderEncoding: Char;  
    var VCharSet: string);  
begin  
    VHeaderEncoding := 'B'; // BASE64 エンコーディング  
    VCharSet := 'UTF-8';    // UTF-8;  
end;
```

?テクニカルセミナーセッション? 受信トレイ | x

☆ From **tec@migaro.co.jp** <tec@migaro.co.jp>
To yoshiwara@migaro.co.jp
日付 2010年11月22日 15:07
件名 ?テクニカルセミナーセッション? ③

知って得する
?Delphiテクニカルエッセンス7.0?

TIdMessageのOnInitializeISOイベントで
設定をUTF-8に変更
③タイトル部分が改善

■ Q2.メール送信機能を作成するには？

Delphi/400からテキストメールを送信

V2009以降

【1】メール送信機能を作成するには？

送信元アドレス tec@migaro.co.jp ①

送信先アドレス yoshiwara@migaro.co.jp ②

タイトル ～テクニカルセミナーセッション～ ③

本文 知って得する
～Delphiテクニカルエッセンス7.0～ ④

添付ファイル



受信

～テクニカルセミナーセッション～ 受信トレイ | ×

☆ From tec@migaro.co.jp <tec@migaro.co.jp> ①

To yoshiwara@migaro.co.jp ②

日付 2010年11月22日 15:30

件名 ～テクニカルセミナーセッション～ ③

知って得する
～Delphiテクニカルエッセンス7.0～ ④

■Q2.メール送信機能を作成するには？

メールにファイルを添付する

【1】メール送信機能を作成するには？

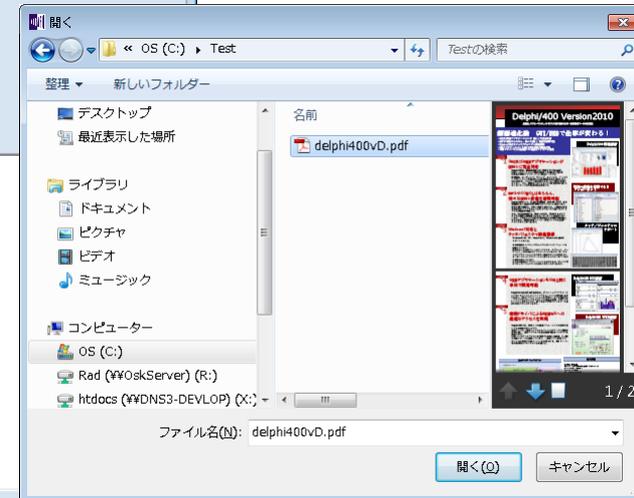
送信元アドレス

送信先アドレス

タイトル

本文
知って得する
~Delphiテクニカルエッセンス7.0~

添付ファイル



■ Q2.メール送信機能を作成するには？

添付ファイルをつけたメールを送信する(ソース)

```
procedure TfrmQ1.btmMailSendClick(Sender: TObject);
begin
  try
    IdSMTP1.Host      := 'smtp.XXXX.XX.XX';      //SMTPサーバ名
    IdSMTP1.Port      := 25;                    //ポート番号
    IdSMTP1.Username  := 'ユーザー名';          //ユーザー名
    IdSMTP1.Password  := 'パスワード';          //パスワード
    IdSMTP1.Connect;                               //接続
    IdMessage1.Clear;
    IdMessage1.ContentType := 'text/plain';      // コンテンツタイプ
    IdMessage1.CharSet     := 'UTF-8';           // 文字セット
    IdMessage1.ContentTransferEncoding := 'BASE64'; // BASE64 エンコーディング
    IdMessage1.From.Address := edtSNDAddress.Text; // 送信元のメールアドレス
    IdMessage1.Recipients.EmailAddresses := edtRCVAddress.Text; // 送信先のメールアドレス
    IdMessage1.Subject     := edtTitle.Text;
    IdMessage1.Body.Text   := memText.Text;
```

“text/plain”だと添付にならない

```
☆ From    tec@migaro.co.jp <tec@migaro.co.jp>
  To      yoshiwara@migaro.co.jp
  日付    2010年11月22日22:01
  件名    ~テクニカルセミナーセッション~

This is a multi-part message in MIME format

--XRpKnn1YJRI3xELDgooldnpbJQ5V8aB=_5
Content-Type: text/plain; charset="UTF-8"
Content-Transfer-Encoding: quoted-printable
Content-Disposition: inline

=E7=9F=A5=E3=81=A3=E3=81=A6=E5=BE=97=E3=81=99=E3=82=BB
=EF=BD=9EDELphi=E3=83=86=E3=82=AF=E3=83=8B=E3=82=AB=E3=
=83=83=E3=82=BB=E3=83=B3=E3=82=B97.0=EF=BD=9E

--XRpKnn1YJRI3xELDgooldnpbJQ5V8aB=_5
Content-Type: application/octet-stream;
```

■ Q2.メール送信機能を作成するには？

添付ファイルをつけたメールを送信する(ソース)

```
if edtFile.Text <> '' then
begin
  IdMessage1.ContentType := 'multipart/mixed'; // コンテンツタイプ
  //添付ファイルの設定 (添付するTIdMessageと添付ファイルパスでCreate)
  with TIdAttachmentFile.Create(IdMessage1.MessageParts, edtFile.Text) do
  begin
    FileName :=ExtractFileName(edtFile.Text);
    ContentType := 'application/octet-stream';
    ContentTransfer := 'base64';
  end;
end;
IdSMTP1.Send(IdMessage1); //メールの送信
ShowMessage('メール送信完了しました');
except
  ShowMessage('メール送信中にエラーが発生しました');
end;
IdSMTP1.Disconnect; //SMTP切断
end;;
```

TIdAttachmentFileで
添付ファイルをCreate
する

■Q2.メール送信機能を作成するには？

主なContentType(メールの形式) ※他にも形式はあります。

text/plain

文字テキストのみのメール。基本中の基本のメール形式。

text/html

文字拡大、強調、色変更、背景設定などhtmlと同様に装飾が可能な形式。

multipart/mixed

複数のメール形式が混在可能な形式。添付ファイルなどでよく使用される。

■ Q2.メール送信機能を作成するには？

もちろん同じ作り方でWEBアプリにも活用可能

http://localhost:90/Smartphone.dll/EXEC/0/...

MIGARO 株式会社ミガロ.

送信元
tec@migaro.co.jp

送信元
yoshiwara@migaro.co.jp

タイトル
テクニカルセミナーセッション3

本文
知って得する
Delphiテクニカルエッセンス7.0

メール送信



Safariを使って
SmartPhoneで動かす
なんてことも！

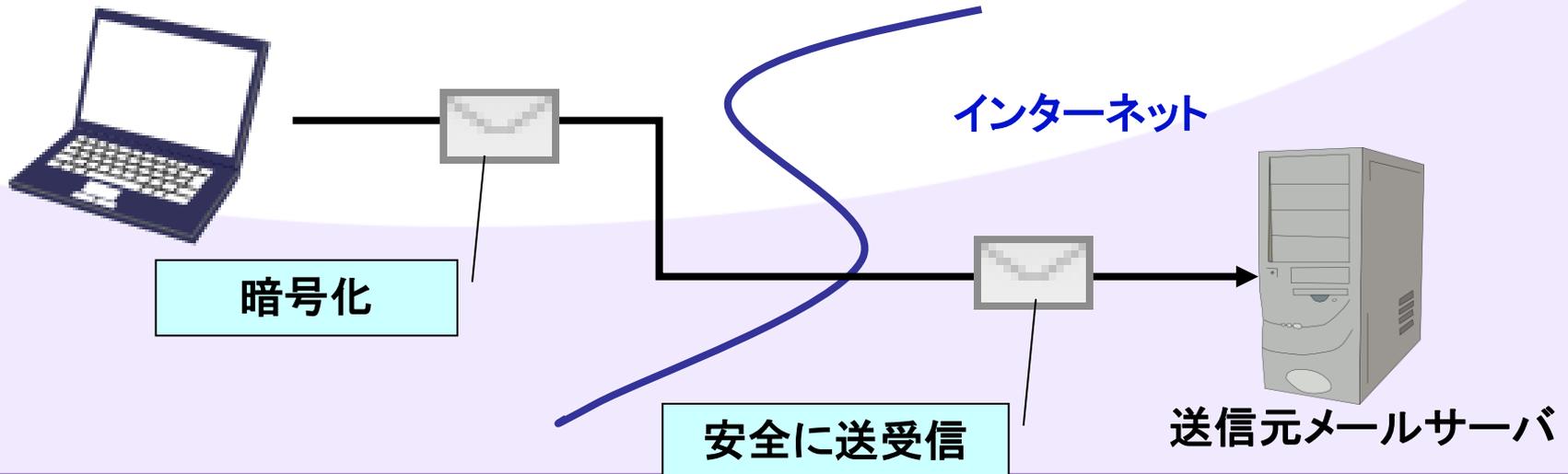
おまけ資料

■おまけ資料(メール関連補足)

メール補足 SSLを利用したい場合

SSLとは

Netscape Communications社が開発した、インターネット上で情報を暗号化して送受信するプロトコル。(IT用語辞典引用)

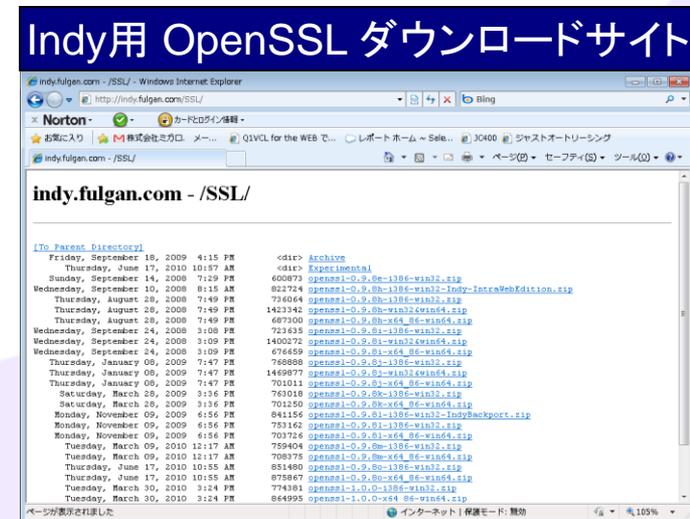
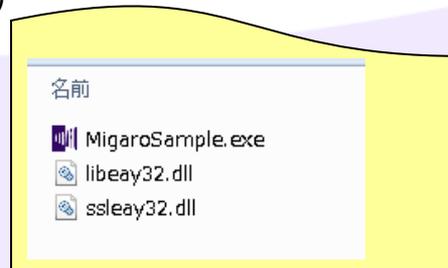


■おまけ資料(メール関連補足)

メール補足 SSLを利用したい場合

POP3/SMTP に SSL を利用する場合には、
“[Indy用 OpenSSL DLL](http://indy.fulgan.com/SSL/)” が別途必要となります。

DLLは以下のURLからダウンロードすることができます。
<http://indy.fulgan.com/SSL/>
アーカイブを解凍して得られる DLL をEXEと同じフォルダ
に配置して下さい。
(Libeay32.dllとssleay32.dll)



■おまけ資料(メール関連補足)

メール補足

SSL認証を行ってメールを送信する(ソース)

usesにIdSSLOpenSSLを追記

```

procedure TfrmQ1.btmMailSendClick(Sender: TObject);
var
  SSL: TIdSSLIOHandlerSocketOpenSSL; //SSLクラス
begin
  try
    try
      IdSMTP1.Host      := 'smtp.XXXX.XX.XX'; //SMTPサーバ名
      IdSMTP1.Port      := 25; //ポート番号
      IdSMTP1.Username  := 'ユーザー名'; //ユーザー名
      IdSMTP1.Password  := 'パスワード'; //パスワード
      SSL := TIdSSLIOHandlerSocketOpenSSL.Create; //SSLクラスを生成
      SSL.Host := IdSMTP1.Host; //ホスト設定
      SSL.Port := IdSMTP1.Port; //ポート設定
      SSL.Destination := SSL.Host + ':' + IntToStr(SSL.Port); //ホスト:ポート
      IdSMTP1.IOHandler := SSL; //IdSMTPのハンドラに設定
      IdSMTP1.UseTLS := utUseExplicitTLS;
      IdSMTP1.Connect; //接続
      IdMessage1.Clear; //メッセージクリア
      IdMessage1.ContentType := 'multipart/mixed'; //コンテンツタイプ
      IdMessage1.CharSet := 'UTF-8'; //文字セット
      IdMessage1.ContentTransferEncoding := 'BASE64'; //BASE64エンコーディング
    end;
  end;
end;
    
```

■おまけ資料(メール関連補足)

メール補足

SSL認証を行ってメールを送信する(ソース)

```
IdMessage1.From.Address := edtSNDAddress.Text; // 送信元のメールアドレス
IdMessage1.Recipients.EMailAddresses := edtRCVAddress.Text; // 送信先のメールアドレス
IdMessage1.Subject := edtTitle.Text; // タイトル
IdMessage1.Body.Text := memText.Text; // 本文
if edtFile.Text <> '' then
begin
  //添付ファイルの設定
  with TIdAttachmentFile.Create(IdMessage1.MessageParts, edtFile.Text) do
  begin
    FileName :=ExtractFileName(edtFile.Text);
    ContentType := 'application/octet-stream';
    ContentTransfer := 'base64';
  end;
end;
IdSMTP1.Send(IdMessage1); //メールの送信
ShowMessage('メール送信完了しました');
except
  ShowMessage('メール送信中にエラーが発生しました');
end;
finally
  SSL.Free; //SSLクラスを破棄
end;
IdSMTP1.Disconnect; //SMTP切断
end;
```

■おまけ資料(メール関連補足)

メール補足

Delphi/400のバージョンとQuickSendメソッド

- TIdSMTPコンポーネントにはQuickSendという簡易メール送信メソッドがあり、次のような簡単なコーディングで送信が可能です。

```
procedure TfrmQ1.btmMailSendClick(Sender: TObject);
begin
  IdSMTP1.QuickSend('ホスト名',
                    'タイトル',
                    '送信先アドレス',
                    '送信元アドレス',
                    '本文');
end;
```



注意点

TIdMessageのように文字セットなどの柔軟な対応が難しい

受信

?テクニカルセミナーセッション? 受信トレイ | x

☆ From tec@migaro.co.jp <tec@migaro.co.jp> **1**

To yoshiwara@migaro.co.jp **2**

日付 2010年11月22日19:55

件名 ?テクニカルセミナーセッション? **3**

?????? **4**
~Delphi??????????????.0~

■おまけ資料(メール関連補足)

メール補足

jconvert.pasを利用したい場合

V2009以降

jconvert.pas(フリーの文字コード変換ライブラリ)
主にメール機能の文字変換などで過去のバージョンでよく使用されていましたが、ダウンロードができるソースは古いためV2009以降の環境で使用すると正しく動作しません。
以下のような点は修正する必要がありますのでご参考ください。

ソースを変更すべき箇所

String → AnsiString
Char → AnsiChar
PChar → PAnsiChar