辻野 健 / 前坂 誠二

株式会社ミガロ.

システム事業部 システム2課

Indyを利用したメール送信機能開発

**ワークフローや注文システムで、確認メールを自動送信したい。
処理ロジックの後に、本稿のメール送信プログラムを追加設定するだけで実現可能だ。**

●はじめに
●Indy とは
●メール送信プログラムの作成（基本編）
●メール送信プログラムの作成（応用編）
●補足
●最後に

略歴　辻野 健
1988 年 06 月 10 日生
2011 年近畿大学 理工学部卒
2011 年 04 月株式会社ミガロ. 入社
2011 年 04 月システム事業部配属

現在の仕事内容
Delphi/400 を利用したシステム開発や保守作業を担当。日々開発スキルの向上を目指し、難易度の高いプログラムにチャレンジしている。

略歴　前坂 誠二
1989 年 03 月 21 日生
2011 年関西大学文学部卒
2011 年 04 月株式会社ミガロ. 入社
2011 年 04 月システム事業部配属

現在の仕事内容
Delphi/400 を利用したシステム開発や保守作業を担当。Delphi、Delphi/400 の開発経験を積みながら、日々スキルを磨いている。

1.はじめに

今日、ワークフローなどのシステムにおいて、一連の処理が完了したことをメールで自動送信するアプリケーションが多くなっている。

しかし、メール送信機能を開発するとなると、技術的に難しいイメージを持たれている方も多いのではないだろうか。

本稿では、Indy を使用し、容易にメール送信機能を実装する方法を紹介する。

2.Indyとは

「Indy」とは、オープンソースのネットワーク関連のコンポーネントのことであり、Delphi に標準で付属している。

TIdSMTP コンポーネントを用いたメールの送信や、TIdPOP3 コンポーネントを用いたメールの受信、TIdFTP コンポーネントを用いた FTP サーバーとの通信などさまざまな機能を実装することができる。

今回は、それらの機能の中で、TIdSMTP コンポーネントを用いてメール送信の実装方法を紹介する。まず初めに基本編として、シンプルな形式のメール送信方法から入り、次に応用編として、ファイルの添付などのメール送信方法を説明する。

なお、本稿で作成しているプログラムは、Delphi/400 XE を用い、Indy バージョンは Indy10_5040 を使用している。

3.メール送信プログラムの作成（基本編）

本章では、画面で入力した件名と本文を、指定したメールアドレスに送信するだけのシンプルなプログラムの作成方法について説明する。

今回は、SMTP サーバーには代表的なメールサービスである Gmail を使用して、プログラムを作成していく。

メール送信用画面の作成

VCL フォームアプリケーションより画面を新規作成し、図 1 のような画面を使用してメール送信プログラムを作成する。【図 1】

使用しているコンポーネントとプロパティの設定については、図 2 を参照してほしい。【図 2】

図 1 の画面の動作としては、送信ボタンを押下すると、To に指定したメールアドレスに、画面で入力した件名と本文のメールを送信する単純なものである。

送信ボタン押下時の処理のおおまかな流れは次の通りである。

① SMTP サーバーへ接続 →
②送信内容の設定と送信 →
③ SMTP サーバーの接続解除

接続設定

メール送信用画面作成後、SMTP サーバーへの接続を行うために TIdSMTP コンポーネントを Form1 に配置する。

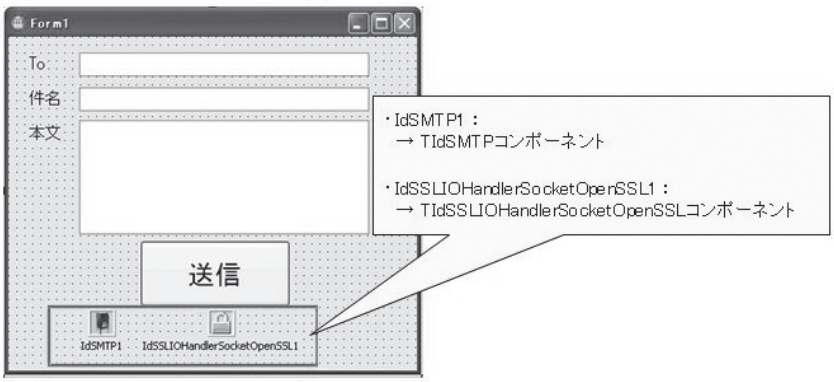
また、今回使用する Gmail のように、SSL を使用しているメールサービスの

【図1】

【図2】

使用コンポーネント	Nameプロパティ	Captionプロパティ
TLabel	Label1	To
TLabel	Label2	件名
TLabel	Label3	本文
TButton	Button1	送信
TEdit	Edit1	−
TEdit	Edit2	−
Tmemo	Memo1	−

【図3】



【図4】

・TIdSMTPコンポーネントのプロパティ設定

プロパティ名	設定値
Host	使用メールサービスのサーバー名 Gmailの場合は、'smtp.gmail.com'を設定
Port	ポート番号 今回は、SSLを使用するため465を設定
Username	使用メールサービスの メール送信を行いたいユーザーアカウント名
Password	上記のユーザーアカウント名で設定したパスワード
IOHandler	TIdSSLIOHandlerSocketOpenSSLの Nameプロパティ値
UseTLS	SSL未使用時は、utNoTLSSupport(デフォルト値) 今回は、utUseExplicitTLSを設定

SSL使用時のみ追加で設定

場合は、SSLの設定を行わなければメール送信が実行できない。そのため、TIdSSLIOHandlerSocketOpenSSLコンポーネントもForm1に配置する必要がある。【図3】

SSL（Secure Socket Layer）とは、インターネット上で送受信を行うデータを暗号化する技術のことである。データの送受信を暗号化することにより、第3者によるデータの盗聴や改ざんなどを防ぐことができるため、安全にデータ通信が行える。

・TIdSMTPコンポーネント

SMTPサーバーへの接続は図4のプロパティを設定し、Connectメソッドを実行するだけで接続できる。SMTPサーバーから接続解除を行う場合には、Disconnectメソッドを実行するだけである。【図4】

また、今回はメール送信にSSLを使用するため、IOHandlerプロパティに図3で配置したIdSSLIOHandlerSocketOpenSSL1を指定し、UseTLSプロパティにutUseExplicitTLSを設定する。

・TIdSSLIOHandlerSocketOpenSSLコンポーネント

SSLの設定についても、SMTPサーバーへの接続設定と同様に、送信ボタン押下時（TButtonのOnClickイベント）に行う。

SMTPサーバーへの接続の前には、図5のプロパティを設定する。【図5】

今回は、送信ボタン押下時（TButtonのOnClickイベント）の最初にSMTPサーバーの設定と接続処理を記述し、最後に接続解除処理を記述する。【ソース1】

メール送信内容の設定

SMTPサーバーへの接続が完了すると、次はメールの送信内容について設定する必要がある。送信内容を設定するために、TIdMessageコンポーネントをForm1に配置する。【図6】

・TIdMessageコンポーネント

送信情報の設定は、送信ボタン押下時（TButtonのOnClickイベント）のSMTPサーバーへの接続処理後に行う。

まず、Clearメソッドで初期化を行った後、図7のプロパティを設定する。今回、送信先のメールアドレス、件名、本文は画面の入力値をそれぞれセットする。【図7】

また、もし複数の宛先に送信したい場合は、複数の送信先のメールアドレスを「,（カンマ）」で区切ることにより実現可能である。

後は、送信情報を設定したIdMessage1を引数として、IdSMTP1のSendメソッドを呼び出すだけでメール送信を行うことができる。【ソース2】

ソース2でコンパイルと実行を行い、送信情報の設定を行った画面が図8である。そして、図8の画面から送信ボタンを押下し、メール送信を行った結果が図9である。【図8】【図9】

・文字コードとエンコーディング

図9を見ると、件名は画面で入力した内容が表示されているが、本文が「?????」という形で文字化けしているのがわかるだろうか。

このような本文の文字化けを防ぐためには、文字コードとエンコーディングの設定を追加で行う必要がある。図7のプロパティ設定にCharSetプロパティ、ContentTransferEncodingプロパティの設定を新たに追加する。【図10】【ソース3】

ソース3でコンパイルと実行を行い、図8の送信内容で、再びメール送信を行った結果が図11である。【図11】

図11では、IdMessage1のプロパティに文字コードとエンコーディングの設定を新たに追加したため、本文が文字化けせずに表示されている。このように、メール送信内容の設定では、TIdMessageコンポーネントで文字コードやエンコーディングの設定が重要である。

4.メール送信プログラムの作成（応用編）

本章では、前章の内容を応用したプログラムの作成方法を紹介する。その際、前章で作成したプロジェクトとソースを流用し、説明を行う。

ファイルの添付

メール送信において、送信するメールにファイルを添付する機会が多くある。ファイルの添付は、TIdAttachmentのCreateメソッドを使用するだけで、容易に実現が可能である。

今回は例として、Cドライブに AttachmentFileというフォルダーを作成し、その中にSAMPLE.jpgというファイルを配置する。そして、SAMPLE.jpgが実際に送信されたメールに添付されているかを確認する。

・TIdAttachmentのCreateメソッド

まず、TIdAttachmentを使用するためには、前章のソースのUses節にIdAttachmentFileを追加する。そして、メール送信処理の前にTIdAttachmentのCreateメソッドを呼び出す。

Createメソッドは、第1引数にTIdMessageコンポーネントのMessagePartsプロパティを指定する。第2引数には添付ファイルのパスを設定する。添付ファイルのパスは、絶対パス、相対パスどちらでも使用できる。また、添付ファイル名には日本語名を使用することも可能である。【ソース4】

ソース4のロジックで、実際に送信したメールにファイルが添付されるかを確認してみよう。

まずソース4でコンパイルと実行を行い、送信内容を図8で設定する。送信ボタンを押下した結果が図12である。【図12】

このように、送信メールにファイルを添付するためには、メール送信ロジックに、TIdAttachmentのCreateメソッドを呼び出すロジックを1行追加するだけで、容易に実装できる。

htmlの利用

さらに、Indyを使ったメール送信では、htmlを利用したメール送信も可能である。htmlを利用すると、文字の装飾や表の作成など、送信内容の幅が広がる。

htmlを利用するためには、メッセージのコンテンツタイプをhtml型に設定する必要がある。

【図5】

・TIdSSLIOHandlerSocketOpenSSLコンポーネントのプロパティ設定

プロパティ名	設定値
Host	使用メールサービスのサーバー名 【図4】のHostプロパティ値と同じ値を設定
Port	ポート番号 【図4】のPortプロパティ値と同じ値を設定
Destination	上記の 'Hostプロパティ値:Portプロパティ値' 今回は'smtp.gmail.com:465'を設定

【ソース1】

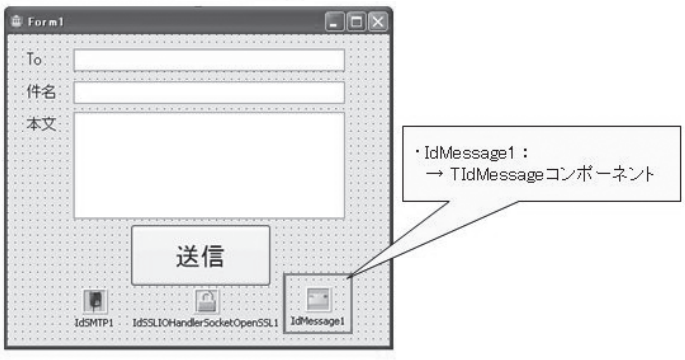
```
{*************************************************************************
 目的 : 送信ボタン押下時処理
 引数 :
 戻値 :
*************************************************************************}
procedure TForm1.Button1Click(Sender: TObject);
begin

  // SMTPサーバー接続設定
  IdSMTP1.Host     := 'smtp.gmail.com';              // サーバー名
  IdSMTP1.Port     := 465;                           // ポート番号
  IdSMTP1.Username := 'MigaroIndySample';            // ログインユーザーID
  IdSMTP1.Password := 'xxxxxxxxxx';                  // ログインパスワード


  IdSMTP1.IOHandler := IdSSLIOHandlerSocketOpenSSL1; // SMTPへSSLを設定
  IdSMTP1.UseTLS    := utUseExplicitTLS;             // 接続方式

  // SSL設定
  IdSSLIOHandlerSocketOpenSSL1.Host := IdSMTP1.Host; // サーバー名
  IdSSLIOHandlerSocketOpenSSL1.Port := IdSMTP1.Port; // ポート番号
  IdSSLIOHandlerSocketOpenSSL1.Destination :=
    IdSMTP1.Host + ':' + IntToStr(IdSMTP1.Port);     // 接続設定

  // SMTPサーバー接続
  IdSMTP1.Connect;

  // SMTPサーバー接続解除
  IdSMTP1.Disconnect;

end;
```

IdSMTP1のプロパティ設定

IdSSLIOHandlerSocketOpenSSL1のプロパティ設定

接続メソッドと接続解除メソッドの間に送信処理を記述していく【ソース2】～【ソース5】

【図6】



・IdMessage1：
→ TIdMessageコンポーネント

【図7】

・TIdMessageコンポーネントのプロパティ設定

プロパティ名	設定値
Recipient.EmailAdresses	送信先のメールアドレス
Subject	メールの件名
Body	メールの本文

そのために本稿では、まず先述のファイル添付で作成したロジックのUses節にIdTextを追加する。そして、ソース4の送信ボタン押下時処理のメソッド内において、TIdText型の変数TEXT1、TEXT2を定義しておく。

・ContentTypeプロパティ'multipart/mixed'

次に、IdMessage1のContentTypeプロパティを'multipart/mixed'に設定する。ContentTypeプロパティでは、どのようなコンテンツの種類で記述を行うかが指定できる。

'multipart/mixed'を指定すると、複数のコンテンツで記述を行うことが可能になる。今回の場合、複数のコンテンツとは、通常のメール文とhtml形式のメール文で記述を行うという意味になる。

メール文を設定するには、変数TEXTをTIdTextとして生成し、TEXTのContentTypeプロパティを通常のメール文の場合は'text/plain'、html形式のメール文の場合は'text/html'に設定する。

後は、本文の設定と文字化け防止のための、文字コードとエンコーディングの設定を忘れずに行えば、htmlを利用したメール送信が完成である。【ソース5】

最後に、ソース5でコンパイルと実行を行い、メール送信を行った結果が図13である。【図13】

このように、htmlを利用することで、文章だけでは伝えづらい内容も、視覚的にわかりやすく表現することができる。

5.補足

・SMTPサーバーへの接続と接続解除のタイミング

本稿では、送信処理の直前にSMTPサーバーへの接続を行い、送信処理の直後にSMTPサーバーとの接続解除を行っている。

一見、画面表示時に接続を行い、画面終了時に接続解除を行ったほうが、レスポンスがよくなるのではないかと思われる方もいるかもしれない。

しかし、本稿ではあえて、送信処理の直前と直後のタイミングで接続と接続解除を行っている。

その理由は、画面表示時にSMTPサーバーへの接続処理を行って、画面を終了せずに接続状態のまま一定時間が経過すると、SMTPサーバーとの接続タイムアウトが発生し、自動的に接続が解除されてしまうからである。つまり、送信処理自体が行えなくなるという可能性がある。

・他のメールサービスを使用する際の注意点

本稿では、SMTPサーバーとしてGmailを使用したため、記載のソースでメール送信機能を実現することができた。

ただし、Gmail以外のメールサービスでメール送信プログラムを作成する場合には、次の点に注意が必要となる。

例えば、本稿で記載したソースを参考に、自身が使用したいメールサービスのホスト名、ポート番号を設定し、メール送信処理を実装するとしよう。その際、SMTPサーバーへの接続は正常に行えるが、TIdSMTPのSendメソッドを呼び出しても、指定したメールアドレスに、メールが届かないケースやエラーが発生するケースがある。

そういった場合には、送信元メールアドレスを指定する必要がある。送信元メールアドレスの指定方法については、ソース6を参考にしてほしい。【ソース6】

6.最後に

今回は、Indyを利用したメール送信の実装方法を紹介した。

本稿で説明や記載している送信プログラムを参考にしたり、変更いただければ、実践的なメール自動送信の仕組みも容易に実装することができる。

例えば、注文システムで、注文確認メールを自動送信したい場合には、注文ボタンの処理ロジックの後に、本稿のメール送信プログラムを追加して送信内容を設定するだけで実現できる。

また、今回はGmailを例にメール送信機能の実装を行ったが、もちろん他のメールソフトでも利用することができる。

メールを連携した機能実装を検討される場合には、本稿の技術情報を役立てていただければ幸いである。

M

【ソース2】

```
{*************************************************************************
 目的 : 送信ボタン押下時処理
 引数 :
 戻値 :
*************************************************************************}
procedure TForm1.Button1Click(Sender: TObject);
begin

  // SMTPサーバー接続設定
  IdSMTP1.Host     := 'smtp.gmail.com';                  // サーバー名
  IdSMTP1.Port     := 465;                               // ポート番号
  IdSMTP1.Username := 'MigaroIndySample';                // ログインユーザーID
  IdSMTP1.Password := 'xxxxxxxxxx';                      // ログインパスワード

  IdSMTP1.IOHandler := IdSSLIOHandlerSocketOpenSSL1;     // SMTPへSSLを設定
  IdSMTP1.UseTLS    := utUseExplicitTLS;                 // 接続方式

  // SSL設定
  IdSSLIOHandlerSocketOpenSSL1.Host := IdSMTP1.Host;     // サーバー名
  IdSSLIOHandlerSocketOpenSSL1.Port := IdSMTP1.Port;     // ポート番号
  IdSSLIOHandlerSocketOpenSSL1.Destination :=
    IdSMTP1.Host + ':' + IntToStr(IdSMTP1.Port);         // 接続設定

  // SMTPサーバー接続
  IdSMTP1.Connect;

  IdMessage1.Clear;

  IdMessage1.Recipients.EMailAddresses := Edit1.Text;    // 送信先メールアドレス
  IdMessage1.Subject                   := Edit2.Text;    // 件名
  IdMessage1.Body.Text                 := Memo1.Text;    // 本文

  // メール送信
  IdSMTP1.Send(IdMessage1);

  // SMTPサーバー接続解除
  IdSMTP1.Disconnect;

end;
```

IdMessage1のプロパティ設定

【図8】

【図9】

【図10】

・TIdMessageコンポーネントのプロパティ追加設定

プロパティ名	設定値
CharSet	'UTF－8'
ContentTransferEncoding	'BASE64'

【ソース3】

```
{*******************************************************************************
 目的 : 送信ボタン押下時処理
 引数 :
 戻値 :
*******************************************************************************}
procedure TForm1.Button1Click(Sender: TObject);
begin

  // SMTPサーバー接続設定
  IdSMTP1.Host     := 'smtp.gmail.com';                  // サーバー名
  IdSMTP1.Port     := 465;                               // ポート番号
  IdSMTP1.Username := 'MigaroIndySample';                // ログインユーザーID
  IdSMTP1.Password := 'xxxxxxxxxx';                      // ログインパスワード

  IdSMTP1.IOHandler := IdSSLIOHandlerSocketOpenSSL1;     // SMTPへSSLを設定
  IdSMTP1.UseTLS    := utUseExplicitTLS;                 // 接続方式

  // SSL設定
  IdSSLIOHandlerSocketOpenSSL1.Host := IdSMTP1.Host;     // サーバー名
  IdSSLIOHandlerSocketOpenSSL1.Port := IdSMTP1.Port;     // ポート番号
  IdSSLIOHandlerSocketOpenSSL1.Destination :=
    IdSMTP1.Host + ':' + IntToStr(IdSMTP1.Port);         // 接続設定

  // SMTPサーバー接続
  IdSMTP1.Connect;

  IdMessage1.Clear;

  IdMessage1.CharSet                  := 'UTF-8';        // 文字セット
  IdMessage1.ContentTransferEncoding  := 'BASE64';       // エンコーディング

  IdMessage1.Recipients.EMailAddresses := Edit1.Text;    // 送信先メールアドレス
  IdMessage1.Subject                   := Edit2.Text;    // 件名
  IdMessage1.Body.Text                 := Memo1.Text;    // 本文

  // メール送信
  IdSMTP1.Send(IdMessage1);

  // SMTPサーバー接続解除
  IdSMTP1.Disconnect;

end;
```

IdMessage1のプロパティ設定に新たに追加する

【図11】

【ソース4】

```
{*****************************************************************************
 目的 : 画面表示時処理
 引数 :
 戻値 :
*****************************************************************************}
procedure TForm1.FormShow(Sender: TObject);
begin

  // SMTPサーバー接続設定
  IdSMTP1.Host     := 'smtp.gmail.com';                  // サーバー名
  IdSMTP1.Port     := 465;                               // ポート番号
  IdSMTP1.Username := 'MigaroIndySample';                // ログインユーザーID
  IdSMTP1.Password := 'xxxxxxxxxx';                      // ログインパスワード

  IdSMTP1.IOHandler := IdSSLIOHandlerSocketOpenSSL1;     // SMTPへSSLを設定
  IdSMTP1.UseTLS    := utUseExplicitTLS;                 // 接続方式

  // SSL設定
  IdSSLIOHandlerSocketOpenSSL1.Host := IdSMTP1.Host;     // サーバー名
  IdSSLIOHandlerSocketOpenSSL1.Port := IdSMTP1.Port;     // ポート番号
  IdSSLIOHandlerSocketOpenSSL1.Destination :=
    IdSMTP1.Host + ':' + IntToStr(IdSMTP1.Port);         // 接続設定

  // SMTPサーバー接続
  IdSMTP1.Connect;

  IdMessage1.Clear;

  IdMessage1.CharSet                    := 'UTF-8';      // 文字セット
  IdMessage1.ContentTransferEncoding    := 'BASE64';     // エンコーディング

  IdMessage1.Recipients.EMailAddresses := Edit1.Text;    // 送信先メールアドレス
  IdMessage1.Subject                   := Edit2.Text;    // 件名
  IdMessage1.Body.Text                 := Memo1.Text;    // 本文

  // ファイルを添付
  TIdAttachmentFile.Create(IdMessage1.MessageParts, 'C:¥AttachmentFile¥SAMPLE.jpg');

  // メール送信
  IdSMTP1.Send(IdMessage1);

  // SMTPサーバー接続解除
  IdSMTP1.Disconnect;

end;
```

ファイル添付処理を追加するためには、Uses節にIdAttachmentFileを追加しなければならない

【図12】

【ソース5】

```
{*****************************************************************************
 目的 : 送信ボタン押下時処理
 引数 :
 戻値 :
*****************************************************************************}
procedure TForm1.Button1Click(Sender: TObject);
var
  TEXT1, TEXT2 : TIdText;
begin

  // SMTPサーバー接続設定
  IdSMTP1.Host     := 'smtp.gmail.com';                    // サーバー名
  IdSMTP1.Port     := 465;                                 // ポート番号
  IdSMTP1.Username := 'MigaroIndySample';                  // ログインユーザール
  IdSMTP1.Password := 'xxxxxxxxxx';                        // ログインパスワード

  IdSMTP1.IOHandler := IdSSLIOHandlerSocketOpenSSL1;       // SMTPへSSLを設定
  IdSMTP1.UseTLS    := utUseExplicitTLS;                   // 接続方式

  // SSL設定
  IdSSLIOHandlerSocketOpenSSL1.Host := IdSMTP1.Host;       // サーバー名
  IdSSLIOHandlerSocketOpenSSL1.Port := IdSMTP1.Port;       // ポート番号
  IdSSLIOHandlerSocketOpenSSL1.Destination :=
    IdSMTP1.Host + ':' + IntToStr(IdSMTP1.Port);           // 接続設定

  // SMTPサーバー接続
  IdSMTP1.Connect;

  IdMessage1.Clear;

  IdMessage1.CharSet                    := 'UTF-8';        // 文字セット
  IdMessage1.ContentTransferEncoding    := 'BASE64';       // エンコーディング

  IdMessage1.Recipients.EMailAddresses := Edit1.Text;      // 送信先メールアドレス
  IdMessage1.Subject                   := Edit2.Text;      // 件名

  IdMessage1.ContentType               := 'multipart/mixed';  // コンテンツタイプ

  // メッセージのみ送信時
  TEXT1 := TIdText.Create(IdMessage1.MessageParts, nil);
  TEXT1.ContentType      := 'text/plain';                  // コンテンツタイプ
  TEXT1.CharSet          := 'UTF-8';                       // 文字セット
  TEXT1.ContentTransfer := 'BASE64';                       // エンコーディング
  TEXT1.Body.Text        := Memo1.Text;                    // 本文

  // ファイルを添付
  TIdAttachmentFile.Create(IdMessage1.MessageParts, 'C:¥AttachmentFile¥SAMPLE.jpg');

  // HTML形式での送信
  TEXT2 := TIdText.Create(IdMessage1.MessageParts, nil);
  TEXT2.ContentType      := 'text/html';                   // コンテンツタイプ
  TEXT2.CharSet          := 'UTF-8';                       // 文字セット
  TEXT2.ContentTransfer := 'BASE64';                       // エンコーディング

  // 本文(HTML)
  TEXT2.Body.Text := '<font size="5" color="#ff0000">文字のサイズと色を指定します</font>' +
                     '<br/><br/>' +
                     '<table border=4 width=250 align=center>' +
                     ' <caption>【テーブルの例】</caption>' +
                     ' <tr bgcolor="#cccccc">' +
                     '  <th><br></th>' +
                     '  <th>列-A</th>' +
                     '  <th>列-A</th>' +
                     '  <th>列-B</th>' +
                     ' </tr>' +
                     ' <tr align=center>' +
                     '  <td>行-1</td>' +
                     '  <td>A1</td>' +
                     '  <td>A1</td>' +
                     '  <td rowspan=2>B1-B2</td>' +
                     ' </tr>' +
                     ' <tr align=center>' +
                     '  <td>行-2</td>' +
                     '  <td>A2</td>' +
                     '  <td>A2</td>' +
                     ' </tr>' +
                     ' <tr align=center>' +
                     '  <td>行-3</td>' +
                     '  <td>A3</td>' +
                     '  <td colspan=2>A3-B3</td>' +
                     ' </tr>' +
                     '</table>';

  // メール送信
  IdSMTP1.Send(IdMessage1);

  // SMTPサーバー接続解除
  IdSMTP1.Disconnect;

end;
```

●通常の本文の設定
・ContentTypeプロパティを
'text/plain'に設定している

●html形式の本文の設定
・ContentTypeプロパティを
'text/html'に設定している

htmlで以下の処理を記述している
①色とサイズを指定した文字列の設定
②テーブルの作成

【図13】

【ソース6】

※【ソース3】を参考にしている。

```
{*******************************************************************************
 目的 : 送信ボタン押下時処理
 引数 :
 戻値 :
*******************************************************************************}
procedure TForm1.Button1Click(Sender: TObject);
begin
  // SMTPサーバー接続設定
  IdSMTP1.Host     := 'smtp.mail.yahoo.co.jp';          // サーバー名
  IdSMTP1.Port     := 465;                              // ポート番号
  IdSMTP1.Username := 'MigaroIndySample';               // ログインユーザーID
  IdSMTP1.Password := 'xxxxxxxxxx';                     // ログインパスワード

  IdSMTP1.IOHandler := IdSSLIOHandlerSocketOpenSSL1;    // SMTPへSSLを設定
  IdSMTP1.UseTLS    := utUseExplicitTLS;                // 接続方式

  // SSL設定
  IdSSLIOHandlerSocketOpenSSL1.Host := IdSMTP1.Host;    // サーバー名
  IdSSLIOHandlerSocketOpenSSL1.Port := IdSMTP1.Port;    // ポート番号
  IdSSLIOHandlerSocketOpenSSL1.Destination :=
    IdSMTP1.Host + ':' + IntToStr(IdSMTP1.Port);        // 接続設定

  // SMTPサーバー接続
  IdSMTP1.Connect;

  IdMessage1.Clear;

  IdMessage1.CharSet                  := 'UTF-8';       // 文字セット
  IdMessage1.ContentTransferEncoding  := 'BASE64';      // エンコーディング

  IdMessage1.From.Address             :=
    'MigaroIndySample@yahoo.co.jp';                     // 送信元メールアドレス

  IdMessage1.Recipients.EMailAddresses := Edit1.Text;   // 送信先メールアドレス
  IdMessage1.Subject                   := Edit2.Text;   // 件名
  IdMessage1.Body.Text                 := Memo1.Text;   // 本文

  // メール送信
  IdSMTP1.Send(IdMessage1);

  // SMTPサーバー接続解除
  IdSMTP1.Disconnect;

end;

end.
```

送信元メールアドレスの指定