

---

**Send e-mail directly from your application. Works with all 32-bit Windows development environments and includes Paradox, Access, and Visual Basic examples.**

---



[www.prestwood.com](http://www.prestwood.com)

7525 Auburn Blvd., #8

Citrus Heights, CA 95610

Office: 1-916-726-5675

Fax: 1-916-726-5675

Email: [mike@prestwood.com](mailto:mike@prestwood.com)

## Section 1 - Introduction

psSendMail™ is a Windows™ DLL that provides the ability to send e-mail messages. The purpose of this DLL is to make e-mail capability possible for those development environments that are capable of communicating with a DLL but cannot send e-mail directly.

### 1.1 Two Interfaces

There are two interfaces to the DLL. The first interface, or Standard Interface, does not limit the length of the strings passed into each of the functions. This allows the calling program to add as many addresses as desired in the To, CC, and BCC fields. It also allows for unlimited length in the subject and the body of the e-mail. The downside, however, to this interface is that the calling program must make a call to each step in the process and must utilize PChars (Pointer to array of character, null terminated). The second interface, or Quick Send Interface, is a wrapper function that internal makes a call to each of the appropriate functions from the first interface. The downside to this interface is that you cannot supply CC or BCC addresses and the To, Subject, and Message Body are each limited to 255 characters, but they are passed as standard strings.

### 1.2 Demo Examples Included

This version includes the following examples:

- ⊕ Corel Paradox
- ⊕ Microsoft Access
- ⊕ Microsoft Visual Basic

If you create a demo in another development environment (language), please send it to [mike@prestwood.com](mailto:mike@prestwood.com) for inclusion in the next version.



## Section 2 - QuickSend Interface

The QuickSend interface consists of a single function call:

```
Function QuickSend(Host, From, SendTo, Subject, MsgText: ShortString): Integer;
```

### 2.1 QuickSend Parameters

All of the following parameters are limited to 255 characters each.

- ⊕ Host - Server Name or IP address of the SMTP Server to connect to (i.e. smtp.mailserver.com or 10.0.0.1)
- ⊕ From - Email address of person (or machine) who is sending the message
- ⊕ SendTo - Email address of the intended recipient. Multiple addresses should be separated with a comma.
- ⊕ Subject - The subject of the email message.
- ⊕ MsgText - Main body of the email message.

### 2.2 QuickSend Return Value

Return values for QuickSend:

- ⊕ 0 = No Error
- ⊕ -1 = MsgText is blank – empty email message.
- ⊕ -2 = An error occurred while sending the message.
- ⊕ -3 = An error occurred while connecting to the SMTP Server.



## Section 3 - Standard Interface

The standard interface consists of five function calls:

- ⊕ **Connect** - Establishes a physical connection to the SMTP host.
- ⊕ **Quit** - Closes the SMTP connection.
- ⊕ **InitMsg** - Prepares the header information for the message.
- ⊕ **Attach** - Attaches a file to the message. Call this once for each file to attach.
- ⊕ **SendMsg** - Sends the e-mail message

To use the standard interface, use the following sequence:

1. Connect
2. InitMsg
3. Attach (optional)
4. SendMsg
5. Quit

### 3.1 Standard Interface Functions

Following are the interface, parameters, and return value definitions for each function:

#### 3.1.1 Connect

Interface

```
function Connect(Host: PChar): Integer;
```

Parameter

For the *Host* parameter use a server name or IP Address of the SMTP Server to connect to (i.e. smtp.mailserver.com or 10.0.0.1)

Return Value

- ⊕ 0 = Connection established
- ⊕ -1 = Unable to establish connection

#### 3.1.2 Quit

Interface

```
procedure Quit;
```



### 3.1.3 InitMsg

#### Interface

```
function InitMsg(Host, From, SendTo, CC, BCC, Subject: PChar): Integer;
```

#### Parameters

- + Host - Server Name or IP Address of SMTP Server to connect to (i.e. smtp.mailserver.com or 10.0.0.1)
- + From - Email address of the person (or machine) who is sending the email message.
- + SendTo - Email address(es) of the intended recipient(s). Multiple addresses should be separated with a comma.
- + CC - Email address(es) of the intended carbon copy recipient(s). Multiple addresses should be separated with a comma.
- + BCC - Email address(es) of the intended blind carbon copy recipient(s). Multiple addresses should be separated with a comma.
- + Subject - The subject of the email message

#### Return Value

- + 0 = No Error
- + -1 = No Host Specified
- + -2 = No From e-mail address supplied
- + -3 = No SendTo e-mail address supplied.

### 3.1.4 Attach

#### Interface

```
function Attach(Filename: PChar): Integer;
```

#### Parameters

Use the *Filename* parameter to specify the path and filename of the file to attach to the message

#### Return value

- + 0 = No error
- + -1 = Path and filename are invalid, unable to locate specified file.

### 3.1.5 SendMsg

#### Interface

```
function SendMsg(MsgText: PChar): Integer;
```

#### Parameters

Use the *MsgText* parameter to specify the main body of the email message.

#### Return Value

- + 0 = No Error
- + -1 = MsgText is blank
- + -2 = Error occurred while sending the message
- + -3 = Error occurred while connecting to the SMTP Server.



## Section 4 - Using psSendMail with Access

Refer to the sendmail.mdb file that shipped with psSendMail. It's included in the AccessDemo folder. The example included was created with Access 2000.

### 4.1 Location of psSendMail.dll

The psSendMail DLL follows the same rules as ALL DLLs. For Access applications, you can place DLLs in any of the following folders:

- Same folder as the Access EXE.
- The Windows folder (something like C:\WINNT).
- The Windows system folder (something like C:\WINNT\SYSTEM32).
- Any folder on your path.



## **Section 5 - Using psSendMail with Visual Basic**

Refer to the files that shipped with psSendMail in the VBDemo folder. The example included was created with Visual Basic 5.

### **5.1 Location of psSendMail.dll**

- The psSendMail DLL follows the same rules as ALL DLLs.



## Section 6 - Using psSendMail with Paradox

The Paradox example included in this section was created with Paradox 9. If you have trouble running the included version, simply open the form in design mode and save to convert it to your version of Paradox.

### 6.1 Location of psSendMail.dll

The psSendMail DLL follows the same rules as ALL DLLs. For Paradox applications, you can place DLLs in any of the following folders:

- Same folder as the form, library, or script
- The Windows folder (something like C:\WINNT).
- The Windows system folder (something like C:\WINNT\SYSTEM32).
- Any folder on your path.

### 6.2 QuickSendDemo.fsl

The QuickSendDemo.fsl form contains all the code required to send an email on the send button. The following screen shot shows the form being used to send an email to two recipients.



The screenshot shows a Windows-style dialog box titled "psSendMail QuickSend Interface Demo". It contains several text input fields for email details:

- Host : 10.0.0.1
- From : mike@prestwood.com
- SendTo : dfought@prestwood.com,joy@prestwood.com
- Subject : Demonstrating psSendMail
- Message : Just demonstrating psSendmail to local user group.

At the bottom left is a "Send" button. To its right, the text reads: "All code is contained on this one button."

The following screen shot displays the code on the pushbutton event of the Send button:



```
QuickSendDemo::#Button59::pushButton
#Button59  •  pushButton

Uses psSendMail
QuickSend(Host Cptr, strFrom Cptr, SendTo Cptr, Subject Cptr, MsgText Cptr ) clong
endUses

method pushButton(var eventInfo Event)
var
  strHost,
  strFrom,
  strSendTo,
  strSubject,
  strMsgText      string

  intResults      longint  ;;0 no error, -1 no message, -2 error while sending
endvar

;Assign e-mail values.
strHost = fldHost.value
strFrom = fldFrom.value
strSendTo = fldSendTo.value
strSubject = fldSubject.value
strMsgText = fldMessage.value

;Send e-mail.
intResults = QuickSend( strHost, strFrom, strSendTo, strSubject, strMsgText )

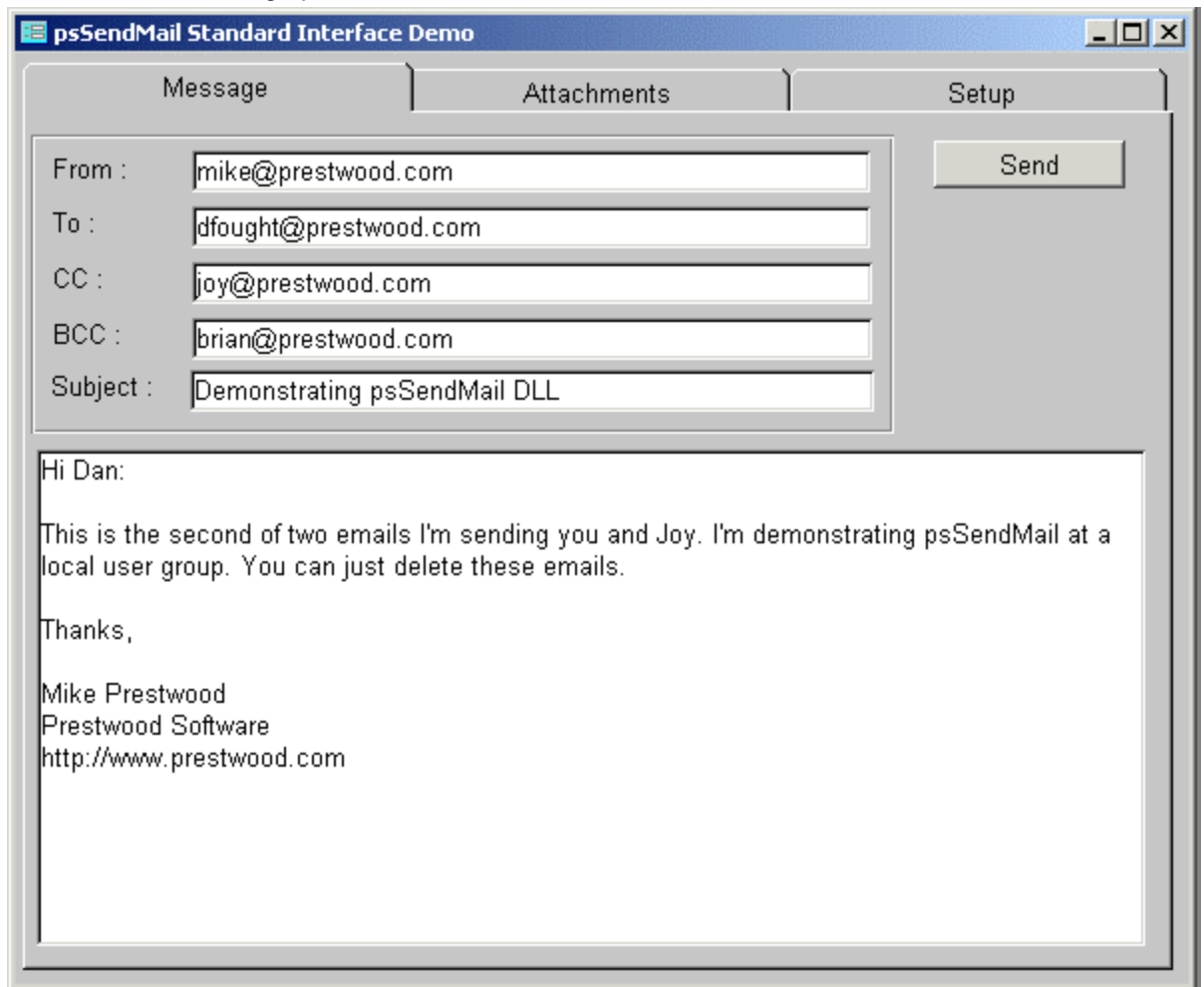
;Check return value.
if not (intResults = 0) then
  msginfo("Mail Error", "Error message returned =" + strval(intResults))
else
  msginfo("", "success")
endif
endMethod
```



## 6.3 StandardDemo.fsl

This Paradox form makes better use of the psSendMail DLL and more closely matches production usage. Refer to the source code for implementation usage.

Screen shot of message parameters:



The screenshot shows a window titled "psSendMail Standard Interface Demo" with three tabs: "Message", "Attachments", and "Setup". The "Message" tab is active, displaying a form for composing an email. The form includes fields for "From", "To", "CC", "BCC", and "Subject", each with a text input box. A "Send" button is located to the right of the "From" field. Below the form is a large text area containing the following text:

Hi Dan:

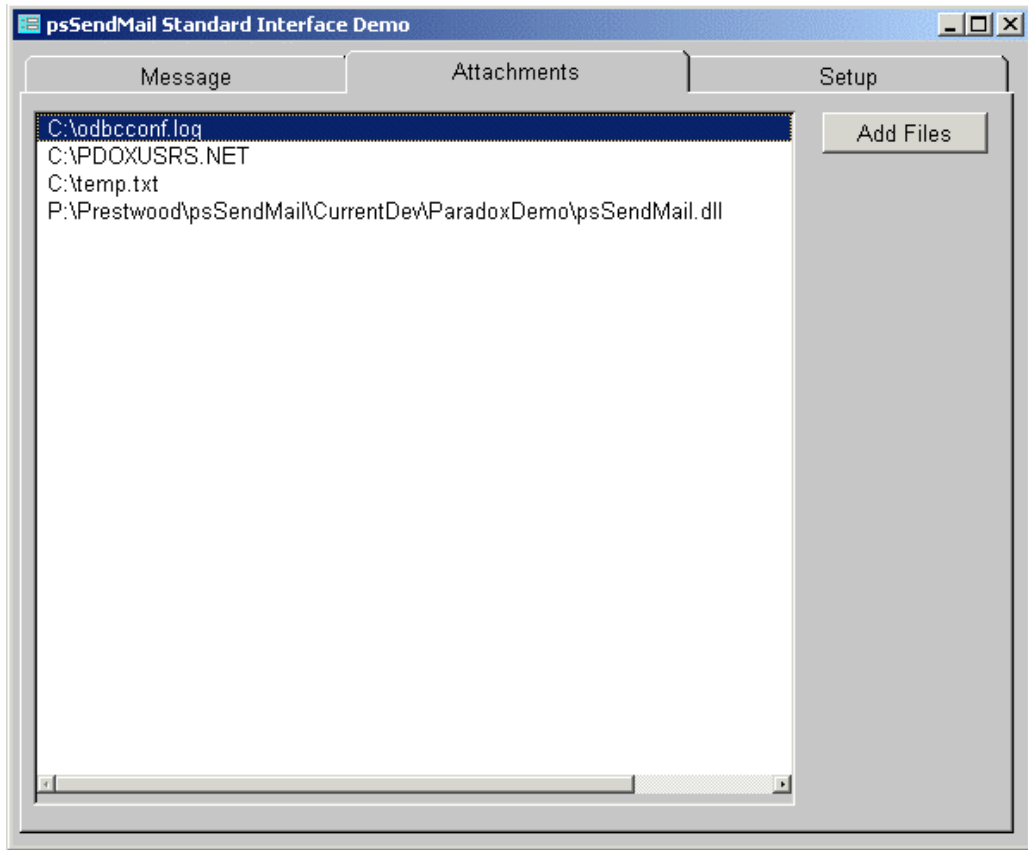
This is the second of two emails I'm sending you and Joy. I'm demonstrating psSendMail at a local user group. You can just delete these emails.

Thanks,

Mike Prestwood  
Prestwood Software  
<http://www.prestwood.com>



Screen shot of attachments tab:



Screen shot of setup tab:

