

**7. SEND/RECEIVE FILES (FTP OR DIAL-UP).....2**  
    7.1 COMMREQUEST.DLL.....2  
    7.2 SENDEMAIL DLL: .....5  
    7.3 HOW THE SENDEMAIL.DLL WORKS.....6

## **7. Send/Receive Files (FTP or Dial-up)**

### ***7.1 CommRequest.dll***

The CommRequest.dll is used to facilitate the transfer of files for the WIC systems. The CommRequest.dll will send and receive files between locations using one of the following communication processes, the File Transfer Protocol or Dialup processing.

To support the CommRequest.dll process a new oracle database table will be needed. The CommRequest table will contain the information needed by the CommRequest.dll to automatically send and receive files.

The information in the CommRequest table will be static in content and will not need a user interface to maintain the information (system table).

### 7.1.1 CommRequest Table

The CommRequest table contains the following information:

ProcessName	Varchar2 (50)	Example – EOD, MONTHEND
SystemName	VarChar2 (50)	Example – BANKING, PNSS, PEDNSS
CommType	VarChar2 (10)	Example – FTP, DIALUP
CommPort	VarChar2 (20)	Example – “Standard Modem”
CommPortSpeed	Number (6)	Example - 9600
CommIdleTime	Number (6)	Example - 600
PhoneNumber	VarChar2 (20)	Example – “1 (888) 123-4567”
ServerName	VarChar2 (25)	
UserID	VarChar2 (25)	
PassWord	VarChar2 (25)	
RecieveFromDirectory	VarChar2 (50)	
SendToDirectory	VarChar2 (50)	
OutBoundFolder	VarChar2 (50)	
OutBoundErrorFolder	VarChar2 (50)	
OutBoundPostFolder	VarChar2 (50)	
InBoundFolder	VarChar2 (50)	
InBoundPostFolder	VarChar2 (50)	
LogFileName	VarChar2 (50)	
SendEmail	VarChar2 (1)	Y/N

The information in the CommRequest table will be used to connect and process the file transfers. It will also control the file transfer process based on the ProcessName and SystemName columns. If the CommRequest.dll is called with only the ProcessName property set then all file transfers for the ProcessName will be processed. If the ProcessName and SystemName property are both set then only the SystemName file transfer will be processed. This will provide some flexibility on what files get transferred when the CommRequest.dll is called and will allow multiple file transfers per process. It will also allow the CommRequest.dll to be called only once for a process if desired. Example – End of Day will need to transfer both the Bank files and well as the CDC files but only one call to the CommRequest process will be needed.

Following is a definition for each column:

- **ProcessName** – Name of the calling process Example – “End of Day”
- **SystemName** – Name of the System within the Process needing a file transfer CommType (“FTP” or “DIALUP”)
- **CommPort** – “Standard Modem”
- **CommPortSpeed** –
- **CommIdleTime** –
- **PhoneNumber** – “DIALUP” phone number
- **ServerName** – Name of the FTP server
- **UserID** – UserID for the FTP server
- **PassWord** – Password for the FTP server
- **RecieveFromDirectory** – Directory on client machine where files will be retrieved
- **SendToDirectory** – Directory on client machine where files will be placed
- **OutBoundFolde** – Local directory where files needing to be sent reside
- **OutBoundErrorFolder** – Local directory where files that had an error reside
- **OutBoundPostFolder** – Local directory where files from the OutBoundFolder are placed after being transferred.
- **InBoundFolder** – Local directory where files being received will reside
- **InBoundPostFolder** – Local directory where files from the InBoundFolder are placed after being processed.
- **LogFileName** – Name of the communication log file.
- **SendEMail** – Y/N value indicating the need to Send and Email notification of the file transfer.

To use the CommRequest.dll make sure that the CommRequest oracle table entry exist for each Process and System that need to do file transfers. From the calling program set the ProcessName property to process all file transfers for the ProcessName and then call the CommRequest.dll. To process only one Systems file transfer set the ProcessName and SystemName properties and then call the CommRequest.dll.

If any errors occur during the CommRequest process a message will be displayed to the user and the return code from the CommRequest.dll will be set to false. If the CommRequest was processed successfully then the return code will be set to true.

## **7.2 SendEmail Dll:**

The SendEmail.dll is used to send predefined E-Mail messages. E-Mail messages are predefined per system. The SendEmail.dll uses the mapiSession and mapiMessage controls provided with Visual Basic to facilitate the sending of the E-Mail messages. An E-Mail client profile must be setup on the computer that the SendEmail.dll runs on for it to work.

There are two database tables that support the sending of the E-Mail message. These tables will contain the information needed to send the E-Mail message and will be predefined and somewhat static in content. The tables are:

- EMAILCONTENT
- EMAILRECIPIENT

### **7.2.1 EMAILCONTENT**

The EMAILCONTENT table contains the information needed to build the content portion of the E-Mail message. The following columns are in the EMAILCONTENT table:

<b>SystemName</b>	<b>Primary key</b>
Subject	Subject line for the E-Mail Message
Priority	Importance Level for the E-Mail Message  0 – Low importance  1 – Normal importance  2 – High importance
MessageContent	E-Mail message text

## 7.2.2 EMAILRECIPIENT

The EMAILRECIPIENT table contains the recipient information needed to send the Email- message. The following columns are in the EMAILRECIPIENT table:

SystemName	Primary key
EmailAddress	Recipient E-Mail address
RecipientName	Display name for the Recipient
RecipientType	Distribution type for Recipient  1 – To recipient  2 – Copy recipient  3 – Blind copy recipient

To use the SendEMail.dll you must do the following.

- Set the SendEMail.dll SystemName property.
- \*Set the SendEMail.dll SuperUser property – System registry entry.
- \*Set the SendEMail.dll ServerName property – System registry entry.
- If you have an attachment you want to send with the E-Mail message these additional properties will need to be set –
  - EMailAttachment\_FileName – Path and file name for the attachment.
  - EMailAttachment\_AttchmntName – Name that will appear on the attachment.
  - EmailAttachment\_Type – Type of attachment –
    - 0 – attachment is a data file
    - 1 – attachment is an embedded OLE object
    - 2 – attachment is a static OLE object

### 7.3 How the SendEMail.dll works.

The SendEmail.dll will take the SystemName property and read the EMAILCONTENT and EMAILRECIPIENT data. It will take the data read from the tables and build the E-Mail message. If the EmailAttachment\_FileName property is not null then the E-Mail attachment will be added to the E-Mail message. The E-Mail message is then sent.

If any errors occurred during the SendEmail process a message will be displayed to the user and the return code from the SendEmail.dll will be set to false. If the E-Mail message was sent successfully then the return code from the SendEmail.dll will be set to true.

\* - To keep from having to read registry entries for a specific application I decided to have the user send the registry entries to me as properties so I can connect to the data base using the supplied properties. This should allow the SendEmail.dll to be used with any application as long as the supporting data base structure exists.