Title: FRANKING MACHINE CONTROL

Abstract: A system for controlling a franking machine from a remote computer comprises control software for controlling the franking machine, communications apparatus for the franking machine to communicate with the computer, a user application operating on the computer to interface with the control software via the communications apparatus, and a programming library for communicating with the user application, the programming library providing the user application with a series of functional controls of the franking machine. A method for operating a franking machine using the functional controls. A computer readable medium containing the programming library.
Published:
— with international search report
— before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.
FRANKING MACHINE CONTROL

This application claims priority under 35 U.S.C. §119(e) from provisional patent serial number 60/213,922 filed on June 26, 2000, which is hereby incorporated by reference.

5 Field of the Invention

The present invention relates to control systems for franking machines. More particularly, it relates to control systems which interface a franking machine with a computer system such as a personal computer or a computer network so that postal meter accountability and security are enhanced.

Background Art

In many businesses, one or more franking machines are used in a mailroom, or several are used in mailrooms in different locations. If such franking machine or machines operate autonomously, there is virtually no easy way to exercise management control, so that it is possible to determine how postage is being used. While in principle, it is possible to keep a postal log stating the purpose of each piece of mail being franked, this takes up an enormous amount of time, is subject to accidental or intentional errors, and provides only minimal information.

Summary of the Invention

It is an object of the present invention to provide franking machine control from a computer.
It is another object of the invention to allow a user application to communicate with franking machine control software elements in a computer.

It is yet another object of the invention to allow a user or customer application to easily obtain information needed for management of franking machine operations.

The present invention is directed to a system for controlling a franking machine from a remote computer, comprising control software for controlling the franking machine, communications apparatus for the franking machine to communicate with the computer; a user application operating on the computer to interface with the control software via the communications apparatus; and a programming library for communicating with the user application, the programming library providing the user application with a series of functional controls of the franking machine.

The invention is also directed to a method for remotely operating a franking machine under functional controls provided by the programming library.

The functional controls include at least one item selected from the group consisting of connection, configuration, actions, departments and statistics.

The invention is further directed to a computer readable medium containing the programming library. The computer readable medium may be a magnetic disk, CD, tape, or any other medium which can be read by an appropriate reading device.
Brief Description of the Drawings

The foregoing aspects and other features of the present invention are explained in the following description, taken in connection with the accompanying drawings, wherein:

Fig. 1 is an illustration of a computer using the present invention connected to several franking machines.

Fig. 2 illustrates communications between the franking machine and a computer in accordance with Fig. 1.

Fig. 3 illustrates the interface for the program used by the computer of Fig. 1.

Fig. 4 is state diagram of the operation of a franking machine when remote control is implemented.

Fig. 5a, Fig. 5b and Fig. 5c illustrate the manner in which high value limits are processed.

Fig. 6 illustrates a key code arrangement used to prevent unauthorized copying of the software used on the computer of Fig. 1.

Figs. 7a to Fig. 7d are examples of additions to the service menu of a franking machine used in accordance with the invention.

Detailed Description Of The Preferred Embodiment

Referring to Fig. 1, a remote computer 10, such as a PC, is connected to one or more postal franking machines 12A, 12B... 12N by respective communications lines 14A, 14B,... 14N. Communications may be conducted via an RS232 (V.24) serial interface, but the invention is not limited in any
such way. Other communications protocols may be used. Further, a LAN/WAN type arrangement may be used to connect computer 10 to the franking machines 12A, 12B... 12N, with other protocols such as, for example, TCP/IP. Further, computer 10 need not be a PC, but can be any of several types of computers, such as a mainframe, or other computer associated with the operation of a business.

Referring to Fig. 2, each franking machine 12 has associated with it a software package, represented schematically at 16 which includes a remote function package 18 and a communication task or module 19. Remote function package 18 provides franking machine functions via remote control from computer 10 via a software package represented schematically at 20. Software package 20 includes a customer designed user application 22 and a franking machine programming library, also referred to as a control library 24 herein. Communication may be carried on via the RS 232 communication line 14 using, for example, the MLPV6 protocol, developed by Ascom Hasler Mailing Systems of Shelton, Connecticut. Generally, the franking machine will define line speeds of up to 9600 bps. However, an RS-232 line may be used conservatively at line speeds of 20 Kbps and with line lengths of up to 15 meters.

The franking machine control library 24, is a library, which is integrated into the custom designed application 22. This library allows accessing the franking meter functions from a computer 10 in a comfortable and easy way. It offers the same functionality as is available at the franking meter keyboard. Thus, the entire franking meter user interface can be simulated or rebuilt on computer 10 and adapted to a customer’s needs.
As noted above, the main task of the control library 24 is communication with the franking machine. The MLPV6 protocol takes care of the complex and sensitive communication between the computer 10 and franking machine 12. Another important task of the control library 24 is providing an information wrapper. The information wrapping checks the data from the user and the machine and converts it into compatible values (e.g. money values). In addition the control library 24 provides information not explicitly supplied by the franking meter, such as the availability of franking modes, that depend on the base software in the franking machine.

Referring to Fig. 3, user application 22 may include a Windows® interface 26, a statistic evaluation module 28, a database access module 30, and a communication package 32 for communicating with other peripheral devices associated with computer 10 such as, for example, a postage scale, a printer, a modem, a barcode reader, etc. It will be understood that user application 22 will be custom programmed by the user and that many variations on its contact and function are possible. For example, a statistical database may provide information as to customer mailing patterns and behaviors. An audit trail may be provided. Mailroom resources may be more efficiently scheduled, and budget efficiently managed.

Control library 24 may include an input/output check, a source of additional information to a user of library 24 and software for executing an appropriate communication protocol. It may be programmed in a language such as Visual C++.
The control library 24 interfaces to a user application 22 by way of a programming interface 40 in the form of Component Object Model (COM) interface also known as ActiveX. COM is a quasi standard for the Windows environment. It allows accessing the library with almost any Windows programming language. Thus, any programming language can be extended with franking machine functions, that are as easy to use as any built-in command of the programming language.

The control library 24 includes several objects. These objects include Connection, Configuration, Actions, Departments, and Statistics. Each one of these objects has properties, methods and events defined below in an Interface Definition document set forth below as Appendix I. The last three pages of Appendix I specify software and hardware requirements, as well as general instructions for use. The term FM used therein refers to a franking machine.

Appendix II is a source code listing of various portions of the programming or control library 24. The following files are included in Appendix II.

Conversions.bas

ErrorHandler.bas

Start.bas

frmAbout.frm

frmConfig.frm

frmCounters.frm

frmDateTime.frm
frmFranking.frm
frmInterface.frm
frmMailClassAdjust.frm
frmMain.frm
frmTextToDisplay.frm
frmConfig.log
Mssccprj.scc
FMCtrlDemo.vbp
FMCtrlDemo.vbw

Fig. 4 is a state diagram which is implemented by the programming or control library 24, and is self explanatory to one skilled in the art.

Fig. 5a, Fig. 5b and Fig. 5c illustrate the manner in which high value limits are processed, showing interactions between the user application and the franking machine control on the left, and interaction between the franking machine control and the franking machine, on the right. Other aspects are described in Appendix I.

Most errors which occur in the system are passed as COM-Exceptions over the programming interface to the user application. Only state information that does not represent real errors, but may have the meaning that the function could not be carried out, do come via return values of the COM functions.
A C++ exception calls the destructor of each object that it goes through up to the level where it is caught. This mechanism allows a perfect clean up in case of an exception as long as all the objects are created in the constructor and destroyed in the destructor of objects. The communication of layers (MLPV6 layers) may be taken over from the project service computer for digital postal franking machines. Thus, errors are passed on by return variables through all MLPV6 layers up into the business classes. On business class level the exceptions are thrown out, and on the level of the COM-Interface they are caught. Of course there are also exceptions thrown on the COM-Interface level, especially in the case of invalid input from the user application.

This architecture permits taking advantage of exceptions and at the same time having control over creation and destruction of objects, as the exceptions span only one level.

Referring to Fig. 6, in order to avoid unauthorized copying and use of the franking machine control software, a protection mechanism is implemented in the franking machine software. This mechanism allows enabling of franking machine control remotely only after entering a key code. The key code may depend on the serial number of the franking machine. By this, it is meant that each machine has its own unique key code. A specialized personal computer program may be used for the generation of the key codes. This key code generator is given only to authorized representatives of the manufacturer of the system.

In connection with the key code the term "user" refers to the person who enables the franking machine control
feature in the franking machine. It does not matter whether this is the end user, or a technician of a reseller, or any other person.

Referring specifically to Fig. 6, franking machine control is enabled using the following procedure:

1. A user 50 receives a franking machine 10 with software installed that supports franking machine control.

2. The user contacts an authorized representative 52 of the manufacturer with the serial number and software version of the franking machine 10.

3. The representative 52 enters the data received from the user into the key code generator and obtains the key code that is given to the user 50.

4. The user 50 chooses REMOTE CONTROL, in a SERVICE menu of the franking machine 10. At the prompt KEY CODE, the user 50 enters a 16-digit number.

As soon as the correct key code has been entered, the menu REMOTE CONTROL changes to allow switching this feature ON or OFF.

After this procedure the key code does not have to be reentered until new software is installed in the franking machine and/or the memory is cleared (INIA).

Each time new software is installed into the franking machine 10 a new key code is required. This is because the key code depends on the software version. As long as the software does not change, it is not necessary to get a new key code. If the memory is cleared (INIA), the same key code can be reused.
Thus, there is actually no key code management necessary since the key codes can be generated with the information provided by the user (serial number, software version). Key codes do not have to be stored as they can be reproduced at any time.

If key codes are sold to users, it may be necessary to keep some form of list that shows who has already paid for the key code and therefore is entitled to receive a new key code without charge when the software of their meter changes. In this list, the users would be identified by the serial number of their respective franking machine. Keeping such a list is not difficult.

Referring to Figs. 7a to 7d, the user interface of the franking machine is extended to allow the user to enter the key code. The following is a description of the user interface extensions, explaining the actions of the user.

REMOTE CONTR. [E]

CONTINUE: [>] The menu command REMOTE CONTROL is added to the SERVICE menu.

KEY CODE:

By pressing the E button 60, the user gets to the prompt (only if the correct key code has not yet been entered). With the C-button (not shown) the user can go back to the menu command.

The key code may be a sixteen digit number, such as: 1234567890123456. The user now enters the 16-digit key code.

REMOTE CONTROL (Fig. 7c and Fig. 7d)
Fig. 7c [>] ON [E] if the entered key code is correct. Remote control can be switched ON and OFF by pressing the > button 62.

If the Key Code is incorrect, the user gets an error message and gets back to the menu (first display; Fig. 7a).

Once the correct key code has been entered, the prompt will not appear again. The user will get directly to the last display Fig. 7d (ON/OFF) when pressing E at the menu command. To avoid abuse by someone trying random codes, the key code can be entered at the keyboard of the franking machine only (not from the computer or any other external device).

The key code generator may be a quite simple and small Windows® program that allows generating key codes by entering the serial number and the software version of the franking machine. It may use a one-way hash algorithm that is already in use in franking machines having certain telemetering system (TMS) functions. Thus, this part of the franking machine software can be reused. The software of other franking machines not so equipped may be enhanced by adding the hash algorithm.

In order to enhance security, it is possible for the input parameters for the hash algorithm to be not only the serial number and the software version, but also a secret key, as described below. The user of the key code generator does not need to know the secret key.

Thus, the hash algorithm that calculates the key code may have three input parameters: serial number and software version of the franking machine and an additional secret key.
This secret key must be known to the key code generator and to the franking meter software that checks the correctness of the entered key code. The secret key is hardcoded in the franking meter software (to be precise: in a parameter file). Since all source code of the franking machine software is stored in a database (MADE), no secret key will be lost and the key code for any meter software ever released, can be reproduced.

The secret key provides additional security against cracking of the key code. Since the hash algorithm is known to the public, it is important to have at least one input parameter that is kept secret.

Another advantage of the introduction of a secret key is its exchangeability. If, for any reason, it becomes necessary to change the key codes, one can introduce a different secret key into the new franking machine software versions. This scenario is practical if the key code generator falls into the wrong hands. The key code generator can determine which secret key to use from the software version, as the secret key is hardcoded into the franking machine software. It is possible for each meter software version to have its own secret key, but this would result in fairly large administrative expenditures. Thus, a simple version of this key code system may have only one secret key for all franking machines.

It should be understood that the foregoing description is only illustrative of the invention. Various alternatives and modifications can be devised by those skilled in the art without departing from the invention. Accordingly, the present invention is intended to embrace all such alternatives, modifications and variances which fall within the scope of the appended claims.
Conversions

Attribute VB_Name = "Conversions"
'Module Conversions contains routines to convert values from one
'type to another.

Option Explicit

'Converts enumerated type FMTYPES to a string with the name

Function FMTypeToString(FMType As FMTYPES) As String
    Select Case FMType
        Case F3XXPLUS
            FMTypeToString = "F3XXPLUS"
        Case Else
            FMTypeToString = "(unknown)"
    End Select
End Function

'Converts enumerated type BASES to a string with the name

Function BaseModelToString(Base As BASES) As String
    Select Case Base
        Case B120
            BaseModelToString = "B120"
        Case B150
            BaseModelToString = "B150"
        Case B220
            BaseModelToString = "B220"
        Case B220P
            BaseModelToString = "B220P"
        Case B250P
            BaseModelToString = "B250P"
        Case B320P
            BaseModelToString = "B320P"
        Case B335
            BaseModelToString = "B335"
        Case B335P
            BaseModelToString = "B335P"
        Case B335R
            BaseModelToString = "B335R"
        Case B337
            BaseModelToString = "B337"
        Case B337P
            BaseModelToString = "B337P"
        Case B340P
            BaseModelToString = "B340P"
        Case BTEST
            BaseModelToString = "BTEST"
        Case Else
Conversions

BaseModelToString = "/(unknown)"
End Select

End Function

'Composes the print image out of the information read from the FM and
'returns it as string (e.g. 99.99 or 9999900 etc.)

Function CreatePrintImage() As String
    Dim Config As New FMCTRLLib.FMConfig
    'Create an new FMC
    Dim strPrintImage As String
    'Define needed variables
    Dim strFixZeros As String
    Dim strPostPoint As String
    Dim strPrePoint As String

    '!!! Error handling must be done in the calling function !!!
    Config.ActiveConnection = fMainForm.Con
    'Define the connection to be used by Config

    strPrintImage = String(Config.DecadeNumber - 1, "9") 'A "9"
    'for each decade wheel do not add the last wheel yet
    If Config.LastDecadeType = LDL09 Then
        strPrintImage = strPrintImage + "9"  'Add a "9" for the
        last decade wheel if last decade can be 0..9
    Else
        strPrintImage = strPrintImage + "5"  'Add a "5" for the
        last decade wheel if last decade be only 0 or 5
        End If

    strFixZeros = String(Config.FixedZeros, "0")  'A "0" for each
    'fixed zero
    strPrintImage = strPrintImage + strFixZeros  'Add the zeros
    'to the decade wheels

    If Config.DecPointPosition > 1 Then  'If FM uses a
decimal point
        strPostPoint = Right(strPrintImage, Config.DecPointPosition - 1)  'Copy the characters that follow the decimal point
        strPrePoint = Left(strPrintImage, Len(strPrintImage) - Config.DecPointPosition + 1)  'Copy the characters that are before the
        strPrintImage = strPrePoint + "." + strPostPoint  'Insert the decimal point
    End If
Conversions

CreatePrintImage = strPrintImage 'Return composed string

Set Config = Nothing 'Disassociate Config object
End Function

'Creates a format mask that can be used to format money values for displaying.
'Attention: the "," (thousand separator) and the "." (decimal point) will be ' automatically replaced by the signs defined in the Windows country ' settings (e.g. "," and ",").

Function CreateCurrencyFormatMask() As String
  Dim Config As New FMCTRLLib.FMConfig 'Create an new FMC config object
  Dim strMask As String 'Define needed variables

  '!!! Error handling must be done in the calling function !!!

  Config.ActiveConnection = fMainForm.Con 'Define the connection to be used by Config
   strMask = "##,##0" 'Define the initial mask
   If Config.DecPointPosition > 1 Then 'If FM uses a decimal point
     'Add a "." and as many "0" as there are decades in the FM after the point
     strMask = strMask + "." + String(Config.DecPointPosition - 1, "0")
   End If

  CreateCurrencyFormatMask = strMask 'Return mask
  Set Config = Nothing 'Disassociate Config object
End Function

'Converts a True to a "Yes" and a False to a "No"

Function BoolToYesNo(bBool As Boolean) As String
  If bBool Then
    BoolToYesNo = "Yes"
  Else
Conversions

BoolToYesNo = "No"
End If
End Function

'Composes a string containing a list of the names of all available Franking Modes

Public Function FrankModesToString() As String
    Dim Config As New FMCTRLLib.FMConfig 'Create an new FMC config object
    Dim strFrkModes As String 'Define needed variables

    '!!! Error handling must be done in the calling function !!!

    Config.ActiveConnection = f MainForm.Con 'Define the connection to be used by Config
    strFrkModes = ""

    If Config.FrankModeAvailNorm Then 'If FRK_NORM is available
        strFrkModes = "Normal"
        'Set string
    End If
    If Config.FrankModeAvailTape Then 'If FRK_TAPE is available
        If Len(strFrkModes) > 0 Then 'If there is already an entry
            strFrkModes = strFrkModes + " / " 'Insert a /
            End If
            strFrkModes = strFrkModes + "Tapes" 'Add Tapes
        End If
        If Config.FrankModeAvailLetter Then '...
            If Len(strFrkModes) > 0 Then 'Insert a /
                strFrkModes = strFrkModes + " / " 'Add Letters
            End If
            strFrkModes = strFrkModes + "Letters"
        End If
        If Config.FrankModeAvailItem Then '...
            If Len(strFrkModes) > 0 Then 'Insert a /
                strFrkModes = strFrkModes + " / " 'Add Items
            End If
            strFrkModes = strFrkModes + "Items"
        End If
    End If

    FrankModesToString = strFrkModes 'Return string

    Set Config = Nothing 'Disassociate Config object
Conversions

End Function

' Converts an index (used to identify the Franking Mode in a ComboBox.ItemData)
' to enumerated type FRANKMODES

Public Function IndexToFrankMode(Index As Integer) As FRANKMODES
    Select Case Index
        Case 0
            IndexToFrankMode = FRK_NORM
        Case 1
            IndexToFrankMode = FRK_TAPE
        Case 2
            IndexToFrankMode = FRK_LETTER
        Case 3
            IndexToFrankMode = FRK_ITEM
    End Select
End Function

'Select the whole text of the given TextBox.

Public Sub SelectAll(txtBox As TextBox)
    txtBox.SelStart = 0
    txtBox.SelLength = Len(txtBox.Text)
End Sub

' Get the appropriate text message to a return value from the FM Control library.

Public Function RetValToText(RetVal As RETVALS) As String
    Dim s As String

    Select Case RetVal
        Case E_OK
            s = "OK, function successful."
        Case E_NOK
            s = "Not OK, general error."
        Case R_NEWDATE
            s = "Print date has changed."
        Case R QUIET
            s = "FM in QUIET state."
        Case R_HVLIM
            s = "Postage is above High Value limit."
        Case E_TIMEOUT
            s = "Timeout has run down."
        Case E_CALLSERVICE
            s = "Call a service technician."
    Case E_ROTOR
        s = "Rotation error."
    End Select

    RetValToText = s
End Function
Conversions

    s = "Error on rotor."
    Case E_CROSS
        s = "Cross error."
    Case E_BASESPEED
        s = "Speed of base to high."
    Case E_DECade
        s = "Decade wheels cannot be set."
    Case E_PRINTER
        s = "Printer error."
    Case W_DESCREG
        s = "Low Credit warning (soon no more funds)."
    Case W_ASCREG
        s = "Ascending counter warning level reached."
    Case W_MAXITEMS
        s = "Item counter warning level reached."
    Case W_READING
        s = "Reading warning level reached."
    Case W_BATTERY
        s = "Battery expiration date reached."
    Case L_DESCREG
        s = "No more funds (descending counter is zero)."
    Case L_ASCREG
        s = "Ascending register limit reached."
    Case L_MAXITEMS
        s = "Item counter limit reached."
    Case L_READING
        s = "Reading limit reached."
    Case L_BATCHCOUNTER
        s = "Batch counter full."
    Case L_DPT
        s = "Department counter full."
    Case L_TOT
        s = "Since Total statistics counter full."
    Case E_HSB_KEY
        'RetVals for H
    SB (high speed base)
        s = "Key of high speed base is in wrong position."
    Case E_HSB_HOT
        s = "High speed base too hot."
    Case E_HSB_STANDBY
        s = "High speed base is on standby."
    Case E_HSB_INKCOVER
        s = "Ink cover of high speed base is open."
    Case Else
        s = "Error: unhandledRetVal!"
End Select

RetValToText = s
End Function
'Get the appropriate short text message to a return value from the FM Control library. These short texts are used for displaying in the Frankings list of frmFranking

Public Function RetValToShortText(RetVal As RETVALS) As String
    Dim s As String
    Select Case RetVal
    Case R_OK
        s = "OK"
    Case E_NOK
        s = "Error"
    Case R_NEWDATE
        s = "New Date"
    Case R_QUIET
        s = "QUIET"
    Case R_HVLIM
        s = "HV limit"
    Case E_TIMEOUT
        s = "Timeout"
    Case E_CALLSERVICE
        s = "Call service"
    Case E_ROTOR
        s = "Rotor error"
    Case E_CROSS
        s = "Cross error"
    Case E_BASESPEED
        s = "Base speed"
    Case E_DECAY
        s = "Decade error"
    Case E_PRINTER
        s = "Printer error"
    Case W_DESCREG
        s = "Low Credit"
    Case W_ASCREG
        s = "Warning: Ascending"
    Case W_MAXITEMS
        s = "Warning: Items"
    Case W_READING
        s = "Warning: Reading"
    Case W_BATTERY
        s = "Warning: Battery"
    Case L_DESCREG
        s = "No more funds"
    Case L_ASCREG
        s = "Limit: Ascending counter"
Conversions

Case L_MAXITEMS
  s = "Limit: Item counter"
Case L_READING
  s = "Limit: Reading"
Case L_BATCHCOUNTER
  s = "Batch counter full"
Case L_DPT
  s = "Department full"
Case L_TOT
  s = "Statistics full"
Case Else
  s = "Error: unhandled RetVal!"
End Select

RetValToShortText = s
End Function

'Displays the appropriate message box to a return value from the FM Control library.
'Some of the messages need to be answered and an action must be taken.

Public Sub ProcessRetVal(RetVal As RETVALS)
  Dim msg As String
  Dim MsgBoxStyle As VbMsgBoxStyle
  Dim bShowMsgBox As Boolean
  Dim NL As String

  NL = Chr(13)                                'Define newline character
  bShowMsgBox = True                           'Get the text message for
  msg = RetValToText(RetVal)                   the RetVal

  Select Case RetVal
  Case R_OK
    bShowMsgBox = False                        'Do not show message box
    frmFranking.sbStatusBar.SimpleText = "Ready for Franking"
    'Write to StatusBar of frmFranking
  Case E_NOK
    MsgBoxStyle = vbCritical
    msg = msg + NL + "Confirm new date?"     'Message box with Yes and No buttons
    MsgBoxStyle = vbYesNo + vbQuestion
    If vbYes = MsgBox(msg, MsgBoxStyle) Then  'If Yes clicked
      ConfirmNewDate
      'Confirmation is sent to FM
Conversions

End If
bShowMsgBox = False 'Do not show a

ditional message box
Case R_QUIET
frmFranking.sbStatusBar.SimpleText = "Quiet" 'Write to
StatusBar of frmFranking
MsgBoxStyle = vbInformation
bShowMsgBox = False 'Do not show m
essage box
Case R_HVLIM
msg = msg + NL + "Accept high postage?"
MsgBoxStyle = vbYesNo + vbQuestion 'Message box w
ith Yes and No buttons
AnswerHVLimit (MsgBox(msg, MsgBoxStyle)) 'Send message
to FM depending on the button pressed
bShowMsgBox = False 'Do not show a
dditional message box
Case E_TIMEOUT
msg = msg + NL + "Decades not set."
MsgBoxStyle = vbInformation
Case E_CALLSERVICE, E_ROTOR, E_CROSS, E_BASESPEED, E_DECADE, E _PRINTER
msg = msg + NL + "Decades not set."
MsgBoxStyle = vbCritical
Case W_DESCREG, W_ASCREG, W_MAXITEMS, W_READING, W_BATTERY
msg = msg + NL + "Decades not set."
MsgBoxStyle = vbInformation
Case L_DESCREG, L_ASCREG, L_MAXITEMS, L_READING, L_BATCHCOUNTEN
R, L_DPT, L_TOT
msg = msg + NL + "Decades not set."
MsgBoxStyle = vbExclamation
Case E_HSB_KEY, E_HSB_HOT, E_HSB_STANDBY, E_HSB_INKCOVER
msg = msg + NL + "Decades not set."
MsgBoxStyle = vbInformation
Case Else
MsgBoxStyle = vbCritical
End Select

If bShowMsgBox Then
MsgBox msg, MsgBoxStyle 'Show message
box with parameters declared above
End If
End Sub

'Send HVDeblock or HVAbort depending on the message box result

Private Sub AnswerHVLimit(MsgBoxResult As VbMsgBoxResult)
    Dim Actions As New FMCTRLLib.FMActions 'Create an new FMA
Conversions

Dim Ret As RETVALS 'Define needed variables

'!!! Error handling must be done in the calling function !!!

'Define the connection to be used by Actions

Actions.ActiveConnection = fMainForm.Connection

If MsgBoxResult = vbYes Then 'If Yes clicked in message box
    Ret = Actions.HVLimitDeblock 'Send High Value deblock message
Else 'No clicked in message box
    Ret = Actions.HVLimitAbort 'Send High Value abort message
End If 'Disassociate Actions
    Set Actions = Nothing 'Process Ret (show)

ProcessRetVal Ret message if necessary)

End Sub

'Sends New Date Confirmation to FM (the FM then adjusts date if it has automatic date, or 'supposes that the user has adjusted the date manually.

Private Sub ConfirmNewDate() 'Create a new FMActions object

Dim Actions As New FMCTRLLib.FMActions

'!!! Error handling must be done in the calling function !!!

Actions.ActiveConnection = fMainForm.Connection 'Define the connection to be used by Actions
Actions.NewDateConfirm 'Send confirmation
Set Actions = Nothing 'Disassociate Actions

End Sub
End If ' (AutoTaxFM)
If Config.MarginAvailable Then
  .txtMargin.Enabled = True
  .txtMargin.Text = "20"
Else
  .txtMargin.Enabled = False
  .txtMargin.Text = "0"
End If
End With
Screen.MousePointer = vbDefault
Set Config = Nothing
frmFranking.Show vbModal, Me

Exit Sub
ErrorHandler:
  Screen.MousePointer = vbDefault
  ErrorLine = Screen.MousePointer = vbHourglass
  Set Config = Nothing
End Sub

'Prepare GUI for MailClassAdjust depending on the FM configuration and open window.
Private Sub mnuActionsMailClassAdjust_Click()
  Dim Config As New FMCTRLLib.FMConfig
  Dim MCTxts() As String
  Dim nMCCylPos As Integer
  Dim nEmptyPos As Integer
  Dim i As Integer
  On Error GoTo ErrorHandler
  Screen.MousePointer = vbHourglass
  Config.ActiveConnection = fMainForm.Config.ActiveConnection
  If Config.SettableMailClasses > 0 Then
    nMCCylPos = Config.MailClassCylinderPositions
    MCTxts = Config.MailClassTexts
    frmMailClassAdjust.cmbMailClass.Clear
    For i = LBound(MCTxts) To UBound(MCTxts)
      frmMailClassAdjust.cmbMailClass.AddItem MCTxts(i), i
    Next i
    nEmptyPos = nMCCylPos - (UBound(MCTxts) - LBound(MCTxts)) - 1
    'Calculate the number of positions on Mail
    Class cylinder that are not assigned to a
'Mail Class text
For i = 1 To nEmptyPos
    frmMailClassAdjust.cmbMailClass.AddItem "(no text)" 'to
make all Mail Class cylinder positions available
Next i
    Screen.MousePointer = vbDefault 'Show default mouse
    Set Config = Nothing 'Disassociate Config
object
    frmMailClassAdjust.Show vbModal, Me 'Open the window
Else
    Screen.MousePointer = vbDefault 'Show default mouse
    Set Config = Nothing 'Disassociate Config
object
    MsgBox "Mail Classes not available", vbCritical 'Show
message
End If
Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault 'Show default mouse
    ErrorHandler 'Handle errors (show
appropriate message)
    Set Config = Nothing 'Disassociate Config
object
End Sub
Private Sub mmuActionsTextToDisp_Click()
    frmTextToDisplay.Show vbModal, Me
End Sub
Private Sub mmuConnectionConnect_Click()
    Dim strConnectionString As String
    On Error GoTo ErrorHandler:
    in case of an error
    Screen.MousePointer = vbHourglass 'Show hourglass mouse
pointer
    strConnectionString = "COMPORT" + CStr(nComPort) + ";
    PROTOCOL=MLPVS" 'Compose ConnectionString
    Con.Connect (strConnectionString) 'Connect to FM
    nFrankingCounter = 0 'Initialize Franking
Counter (used to number the Frankings)
    SetFormState (fsConnected) 'Set frmMain all
controls on frmMain for state Connected
    Screen.MousePointer = vbDefault 'Show default mouse
pointer
Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault 'Show default mouse
    ErrorHandler 'Handle errors (show
appropriate message)
    SetFormState (fsDisconnected) 'Set frmMain all
controls on frmMain for state Disconnected
End Sub
Private Sub mmuConnectionDisconnect_Click()
On Error GoTo ErrorHandler:
    Screen.MousePointer = vbHourglass
    Con.Disconnect
    SetFormState (fsDisconnected)
    Screen.MousePointer = vbDefault
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault
    ErrorHandler
    appropriate message) does usually not happen here
    End Sub

'Executes appropriate mnuXXX_Click routine depending on the clicked button

Private Sub tbToolBar_ButtonClick(ByVal Button As MSComctlLib.Button)
    On Error Resume Next
    Select Case Button.Key
        Case "Interface"
            mnuSystemInterface_Click
        Case "Connect"
            mnuConnectionConnect_Click
        Case "Disconnect"
            mnuConnectionDisconnect_Click
        Case "Properties"
            mnuConfigProperties_Click
        Case "Settings"
            mnuConfigSettings_Click
        Case "Franking"
            mnuActionsFranking_Click
        Case "Counters"
            mnuActionsCounters_Click
        Case "DateTime"
            mnuActionsDateTime_Click
        Case "MailClass"
            mnuActionsMailClassAdjust_Click
        Case "TextChanged"
            mnuActionsTextChanged_Click
        Case "About"
            mnuHelpAbout_Click
    Case Else
        MsgBox "Add ToolBar_ButtonClick code. " + Button.Key
    End Select
End Sub

Private Sub mnuConfigProperties_Click()
    LoadfrmConfig 0
    "Properties"
End Sub

Private Sub mnuConfigSettings_Click()
    LoadfrmConfig 1
    "Settings"
End Sub

'Information is filled into the fields and window is opened

Private Sub LoadfrmConfig(nTab As Integer)
    Dim Config As New FMCTRLLib.FMConfig
    'Create an new
FMConfig object
On Error GoTo ErrorHandler: 'Jump to ErrorHandler
in case of an error
Screen.MousePointer = vbHourglass 'Show hourglass mouse pointer
Config.ActiveConnection = fMainForm.Connection 'Define the connection to be used by Config

With frmConfig
 'General information
 .lbFMType.Caption = FMTypeToString(Config.FMType) 'Read and display FM Type
 .lbSerialNr.Caption = Config.SerialNr 'Read and display Serial Number
 .lbFMSWVer.Caption = Config.FMSWVer 'Read and display FM Software version
 If Config.AutoTaxFM Then 'If it is a special Auto Tax app. (Japan)
   .lbApplication.Caption = "Auto Tax" 'Compose the description of the application
   If Config.AcquisitionTaxAvailable Then 'Does the Auto Tax app. support Acquisition Tax?
     .lbApplication.Caption = .lbApplication.Caption + " with Acquisition Tax"
   End If
 Else
   .lbApplication.Caption = "Standard" 'Non Auto Tax application
 End If

 'Print image
 .lbNrDecades.Caption = CStr(Config.DecadeNumber) 'Read and display Number of Decade wheels
 .lbDecPos.Caption = CStr(Config.DecPointPosition) 'Read and display position of the decimal point
 .lbFixZeros.Caption = CStr(Config.FixedZeros) 'Read and display number of fixed zeros
 If Config.LastDecadeType = LD09 Then 'Read and display type of the last Decade wheel
   .lbLastDecade.Caption = "0..9" 'Can be all 0..9
 Else
   .lbLastDecade.Caption = "0/5" 'Can only be 0 or 5
 End If
 .lbPrintImage.Caption = CreatePrintImage 'Compose the print image (e.g. 99.99 or 99999900)

 'Information about the Base
 .lbBaseModel.Caption = BaseModelToString(Config.BaseModel) 'Read and display the Base model
 .lbBaseSoftware.Caption = Config.BaseSWVer 'Read and display Base software version

 'Features
 .lbAutoDate = BoolToYesNo(Config.AutoDate) 'Read and display whether FM has automatic date
 .lbMargin.Caption = BoolToYesNo(Config.MarginAvailable) 'Read and display whether margin is changable
 .lbNrMailClass.Caption = CStr(Config.SettableMailClasses) 'Read and display the number of available Mail Classes

 'Franking Modes
 .lbFrankModes = FrankModesToString 'Compose a String containing a list of the available Franking Modes
End With
Screen.MousePointer = vbDefault 'Show default mouse
pointer
Set Config = Nothing 'Disassociate Config
object

frmConfig.tabConfig.Tab = nTab 'Display the
appropriate tab
frmConfig.Show vbModal, Me 'Show the window

Exit Sub
ErrorHandler:
Screen.MousePointer = vbDefault 'Show default mouse
pointer
ErrorHandler 'Handle errors (show
appropriate message) 'Disassociate Config
Set Config = Nothing
object
End Sub

Private Sub mnuHelpAbout_Click()
    frmAbout.Show vbModal, Me 'Show default mouse
End Sub

Private Sub mnuSystemInterface_Click()
    frmInterface.Show vbModal, Me 'Handle errors (show
End Sub

Private Sub mnuViewStatusBar_Click()
    mnuViewStatusBar_Checked = Not mnuViewStatusBar_Checked 'Disassociate Config
    sbStatusBar.Visible = mnuViewStatusBar_Checked 'Display the
End Sub

Private Sub mnuViewToolbar_Click()
    mnuViewToolbar_Checked = Not mnuViewToolbar_Checked 'Define the
    lbToolbar.Visible = mnuViewToolbar_Checked 'Read and
display FM Type
End Sub

Private Sub mnuFileExit_Click()
    Unload Me 'Write to status bar
End Sub

'Makes all the GUI elements appear in the appropriate way depending
on the FromState

Sub SetFormState(State As FORMSTATES)
    Dim Config As New FMCtrlLib.FMConfig 'Read and
    'Create an new
    FMConfig object
    On Error GoTo ErrorHandler: 'Disassociate Config
    in case of an error

    Select Case State
        Case fsConnected
            Config.ActiveConnection = Con 'Define the
            connection to be used by Config
            lbFMTYPE.Caption = FMTypeToString(Config.FMType) 'Read and
display FM Type
            lbSerialNr.Caption = Config.SerialNr 'Write to status bar
            Set Config = Nothing 'Enable or disable
            the GUI elements
            sbStatusBar.SimpleText = "Connected"
            mnuSystemInterface.Enabled = False
    End Select
End Sub
mmuFileExit.Enabled = True
mmuConnectionConnect.Enabled = False
mmuConnectionDisconnect.Enabled = True
mmuConfigProperties.Enabled = True
mmuConfigSettings.Enabled = True
mmuActionsFranking.Enabled = True
mmuActionsCounters.Enabled = True
mmuActionsDateTime.Enabled = True
mmuActionsMailClassAdjust.Enabled = True
mmuActionsTextToDisp.Enabled = True
mmuViewToolbar.Enabled = True
mmuViewStatusBar.Enabled = True
mmuHelpAbout.Enabled = True
With tbToolbar
  .Buttons("Interface").Enabled = False
  .Buttons("Connect").Enabled = False
  .Buttons("Disconnect").Enabled = True
  .Buttons("Properties").Enabled = True
  .Buttons("Settings").Enabled = True
  .Buttons("Franking").Enabled = True
  .Buttons("Counters").Enabled = True
  .Buttons("DateTime").Enabled = True
  .Buttons("MailClass").Enabled = True
  .Buttons("TextToDisplay").Enabled = True
  .Buttons("About").Enabled = True
End With
End With
Case fsDisconnected
  lbFMType.Caption = "(none)"
  lbSerialNr.Caption = "(none)"
  sbStatusBar.SimpleText = "Disconnected" 'Write to status bar

mmuSystemInterface.Enabled = True 'Enable or disable the GUI elements
mmuFileExit.Enabled = True
mmuConnectionConnect.Enabled = True
mmuConnectionDisconnect.Enabled = False
mmuConfigProperties.Enabled = False
mmuConfigSettings.Enabled = False
mmuActionsFranking.Enabled = False
mmuActionsCounters.Enabled = False
mmuActionsDateTime.Enabled = False
mmuActionsMailClassAdjust.Enabled = False
mmuActionsTextToDisp.Enabled = False
mmuViewToolbar.Enabled = True
mmuViewStatusBar.Enabled = True
mmuHelpAbout.Enabled = True
With tbToolbar
  .Buttons("Interface").Enabled = True
  .Buttons("Connect").Enabled = True
  .Buttons("Disconnect").Enabled = False
  .Buttons("Properties").Enabled = False
  .Buttons("Settings").Enabled = False
  .Buttons("Franking").Enabled = False
  .Buttons("Counters").Enabled = False
  .Buttons("DateTime").Enabled = False
  .Buttons("MailClass").Enabled = False
  .Buttons("TextToDisplay").Enabled = False
  .Buttons("About").Enabled = True
End With
End With
End Select

Exit Sub
ErrorHandler:
Screen.MousePointer = vbDefault 'Show default mouse pointer
ErrorHandler 'Handle errors (show
appropriate message)
    Set Config = Nothing
object
End Sub

' Disassociate Config
VERSION 5.00
Begin VB.Form frmTextToDisplay
  BorderStyle = 3 'Fixed Dialog
  Caption = "Text to display"
  ClientHeight = 1440
  ClientLeft = 2760
  ClientTop = 3750
  ClientWidth = 5355
  Icon = "frmTextToDisplay.frx":0000
  LinkTopic = "Form1"
  MaxButton = 0 'False
  MinButton = 0 'False
  ScaleHeight = 1440
  ScaleWidth = 5355
  ShowInTaskbar = 0 'False
  StartUpPosition = 1 'CenterOwner
Begin VB.TextBox txtStartPos
  Alignment = 1 'Right Justify
  Height = 285
  Left = 1200
  MaxLength = 2
  TabIndex = 1
  Text = "0"
  Top = 840
  Width = 375
End
Begin VB.TextBox txtText
  BeginProperty Font
    Name = "Fixedsys"
    Size = 9
    Charset = 0
    Weight = 400
    Underline = 0 'False
    Italic = 0 'False
    Strikethrough = 0 'False
  EndProperty
  Height = 375
  Left = 1200
  MaxLength = 32
  TabIndex = 0
  Text = "Ascom Mailsys FM Control"
  ToolTipText = "Text to display"
  Top = 360
  Width = 3975
End
Begin VB.CommandButton cmdClose
  Cancel = -1 'True
  Caption = "Close"
  Height = 375
  Left = 3960
  TabIndex = 3
  Top = 960
  Width = 1215
End
Begin VB.CommandButton cmdSend
  Caption = "&Send"
  Default = -1 'True
  Height = 375
  Left = 2640
  TabIndex = 2
  Top = 960
  Width = 1215
End
Begin VB.Label Label5
  Alignment = 2 'Center
  BackStyle = 0 'Transparent
Caption = "16"
Height = 255
Left = 3000
TabIndex = 10
Top = 0
Width = 255
End
Begin VB.Line Line5
  X1 = 3120
  X2 = 3120
  Y1 = 240
  Y2 = 360
End
Begin VB.Label Label4
  Alignment = 2 'Center
  BackStyle = 0 'Transparent
  Caption = "30"
  Height = 255
  Left = 4680
  TabIndex = 9
  Top = 0
  Width = 255
End
Begin VB.Label Label3
  Alignment = 2 'Center
  BackStyle = 0 'Transparent
  Caption = "20"
  Height = 255
  Left = 3480
  TabIndex = 8
  Top = 0
  Width = 255
End
Begin VB.Label Label2
  Alignment = 2 'Center
  BackStyle = 0 'Transparent
  Caption = "10"
  Height = 255
  Left = 2280
  TabIndex = 7
  Top = 0
  Width = 255
End
Begin VB.Label Label1
  Alignment = 2 'Center
  BackStyle = 0 'Transparent
  Caption = "1"
  Height = 255
  Left = 1200
  TabIndex = 6
  Top = 0
  Width = 255
End
Begin VB.Line Line4
  X1 = 4780
  X2 = 4780
  Y1 = 240
  Y2 = 360
End
Begin VB.Line Line3
  X1 = 3600
  X2 = 3600
  Y1 = 240
  Y2 = 360
End
Begin VB.Line Line2
X1 = 2400  
X2 = 2400  
Y1 = 240  
Y2 = 360  

End  
Begin VB.Line Line1  
X1 = 1320  
X2 = 1320  
Y1 = 240  
Y2 = 360  
End  
Begin VB.Label lbStartPos  
Alignment = 1 'Right Justify  
Caption = "Start Position:"  
Height = 255  
Left = 120  
TabIndex = 5  
Top = 900  
Width = 975  
End  
Begin VB.Label lbText  
Alignment = 1 'Right Justify  
Caption = "Text:"  
Height = 255  
Left = 120  
TabIndex = 4  
Top = 420  
Width = 975  
End  

End  
Attribute VB_Name = "frmTextToDisplay"  
Attribute VB_GlobalNameSpace = False  
Attribute VB_Creatable = False  
Attribute VB_PredeclaredId = True  
Attribute VB_Exposed = False  
' frmTextToDisplay provides an interface to send a text to the  
' FM that is written to its display  

Option Explicit  

Private Sub cmdClose_Click()  
Unload Me  
End Sub  

Private Sub cmdSend_Click()  
Dim Actions As New FMCTRLLib.FMActions  
'Create an new  
FMActions object  
On Error GoTo ErrorHandler:  
'Jump to  
ErrorHandler in case of an error  
Screen.MousePointer = vbHourglass  
'mouse pointer  
Actions.ActiveConnection = fMainForm.Con  
'define the  
connection to be used by Actions  

Actions.TextEditToDisplay txtText.Text, CInt(txtStartPos.Text)  
'display text on FM  

Screen.MousePointer = vbDefault  
'mouse pointer  
Set Actions = Nothing  
'Disassociate  
Actions object  
Exit Sub  
ErrorHandler:  
Screen.MousePointer = vbDefault  
'mouse pointer  
ErrorHandler
(show appropriate message)
    Set Actions = Nothing
Actions object
End Sub

' Disassociate

Private Sub txtStartPos_GotFocus()
    SelectAll txtStartPos
    text in the TextBox
End Sub

' Select the whole

Private Sub txtText_GotFocus()
    SelectAll txtText
    text in the TextBox
End Sub

' Select the whole
Line 19: Class TabDlg.SS: TabConfig was not a loaded control class.
[SCC]
SCC=This is a source code control file
[FMCtrlDemo.vbp]
SCC_Project_Name=this project is not under source code control
SCC_Aux_Path=<This is an empty string for the mssccprj.scc file>
Type=Exe
Reference="\G\00020430-0000-0000-C000-000000000046)#2.0#0#C:\WINNT\System32\STDOLB2.TLB#OLE Automation
Reference="\G\6B263850-900B-11D0-9484-00A0C91100ED)#1.0#0#C:\WINNT\System32\MSSTDFMT.DLL#Microsoft Data Formatting Object Library
Reference="\G\853A99F1-9752-11D3-819E-0050048C727A)#1.0#0#..\FMCtrlLib\FMCtrl\Debug\FMCtrl1_d.exe#FMCtrl1 1.0 Type Library
Object={F9043C88-F6P2-101A-A3C9-08002B2F49FB)#1.2#0; cmdlg32.ocx
Object={811FDD16-0C5C-11D2-A9FC-0000F8754DA1)#2.0#0; Mscnctl.ocx
Object={BDC217C8-ED16-11CD-956C-000004E4C0A)#1.1#0; Tabctl32.ocx
Object={A8B3B723-0B5A-101B-B228-00A0037B2FC)#1.0#0; Grid32.OCX
Module=Start; Start.bat
Form=frmMain.frm
Form=formAbout.frm
Form=formConfig.frm
Form=formInterface.frm
Form=formTextToDisplay.frm
Form=formMailClassAdjust.frm
Form=formFranking.frm
Module=ErrorHandler; Handler.frm
Module=ErrorInd; ErrorHandler.bas
IconForm="\frmMain"
Startup="Sub Main"
HelpFile=""
Title="FM Control User Application Demo"
ExeName32="FMCtrlDemo.exe"
Command32=""
Name="FMCtrlDemo"
HelpContextID="#0"
Description="Demonstrates the usage of the FM Control library."
CompatibleMode="#0"
MajorVer=1
MinorVer=2
RevisionVer=0
AutoIncrementVer=0
ServerSupportFiles=0
VersionComments="Demonstrates the usage of the FM Control library."
VersionCompanyName="Ascom Autelco AG Mailing Systems, Switzerland"
VersionFileDescription="Demonstrates the usage of the FM Control library."
VersionProductName="FM Control User Application Demo"
CompilationType=0
OptimizationType=0
FavorPentiumPro(tm)=0
CodeViewDebugInfo=0
NoAliasing=0
BoundsCheck=0
OverflowCheck=0
FIPCheck=0
FDIVCheck=0
UnroundedFP=0
StartMode=0
Unattended=0
Retained=0
ThreadPerObject=0
MaxNumberOfThreads=0
DebugStartUpOption=0
Start = 197, 107, 828, 629, C
frmMain = 33, 7, 903, 748, I, 267, 58, 792, 463, C
frmAbout = -2, 6, 789, 751, C, 205, 209, 662, 560, C
frmConfig = 191, 3, 927, 732, C, 266, 42, 723, 480, C
frmInterface = 49, 13, 848, 535, C, 15, 29, 246, 208, C
frmCounters = 258, 28, 846, 626, C, 379, 4, 813, 271, C
frmDateTime = 330, 97, 910, 671, C, 535, 21, 785, 240, C
frmTextToDisplay = 27, 17, 838, 693, C, 377, 7, 783, 203, C
frmMailClassAdjust = 84, 16, 915, 643, I, 470, 10, 781, 192, C
frmFranKing = 12, 8, 799, 703, C, 180, 116, 691, 496, C
ErrorHnd = 9, 9, 897, 651, C
Conversions = 71, 2, 830, 735, C
AutoSize = 2
TextSave = "08.09.2000"
EndProperty
BeginProperty Panel3 {8E3867AB-8586-11D1-B16A-00C0F0283628}
  Style = 5
  AutoSize = 2
  TextSave = "16:29"
EndProperty
End
Begin MSComctlLib.ImageList imlToolbarIcons
  Left = 6480
  Top = 480
  _ExtentX = 1005
  _ExtentY = 1005
  BackColor = 16777215
  ImageWidth = 16
  ImageHeight = 16
  MaskColor = 12632256
  Version = 393216
BeginProperty Images {2C247F25-8591-11D1-B16A-00C0F0283628}
  NumListImages = 11
BeginProperty ListImage1 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture = "frmMain.frx";0B94
  Key = "Help"
EndProperty
BeginProperty ListImage2 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture = "frmMain.frx";0CF2
  Key = "Counters"
EndProperty
BeginProperty ListImage3 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture = "frmMain.frx";100C
  Key = "MailClass"
EndProperty
BeginProperty ListImage4 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture = "frmMain.frx";145E
  Key = "Franking"
EndProperty
BeginProperty ListImage5 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture = "frmMain.frx";18B0
  Key = "DateTime"
EndProperty
BeginProperty ListImage6 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture = "frmMain.frx";1D02
  Key = "Properties"
EndProperty
BeginProperty ListImage7 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture = "frmMain.frx";2154
  Key = "TextToDisplay"
EndProperty
BeginProperty ListImage8 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture = "frmMain.frx";25A6
  Key = "Connect"
EndProperty
BeginProperty ListImage9 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture = "frmMain.frx";2704
  Key = "Disconnect"
BeginProperty ListImage10 {2C247F27-8591-11D1-B16A-00C0F0283628}
  .Picture = "frmMain.frm":2862
  .Key = "Interface"
EndProperty

BeginProperty ListImage11 {2C247F27-8591-11D1-B16A-00C0F0283628}
  .Picture = "frmMain.frm":2974
  .Key = "Settings"
EndProperty

Begin VB.PictureBox picAscomLogo
  Appearance = 0 'Flat
  BackColor = &H80000005&
  BorderStyle = 0 'None
  FillColor = &H00FFFFFF&
  ForeColor = &H80000005&
  Height = 592
  Left = 2520
  Picture = "frmMain.frm":2A86
  ScaleHeight = 585
  ScaleWidth = 2010
  TabIndex = 2
  Top = 3300
  Width = 2010
End

Begin VB.Label lbSerialNr
  BackColor = &H80000005&
  BackStyle = 0 'Transparent
  Caption = "654321"
  BeginProperty Font
    Name = "MS Sans Serif"
    Size = 12
    CharSet = 0
    Weight = 700
    Underline = 0 'False
    Italic = 0 'False
    Strikethrough = 0 'False
  EndProperty
  Height = 375
  Left = 3480
  TabIndex = 6
  Top = 1680
  Width = 1455
End

Begin VB.Label lbSerialNrLabel
  BackColor = &H80000005&
  BackStyle = 0 'Transparent
  Caption = "Serial Nr.:
  BeginProperty Font
    Name = "MS Sans Serif"
    Size = 12
    CharSet = 0
    Weight = 700
    Underline = 0 'False
    Italic = 0 'False
    Strikethrough = 0 'False
  EndProperty
  Height = 375
  Left = .2160
  TabIndex = 5
  Top = 1680
  Width = 1215
End
Begin VB.Label lbFMType
  BackColor = &H80000005&
  BackStyle = 0 'Transparent
  Caption = "F3XXPLUS"
  BeginProperty Font
    Name = "MS Sans Serif"
    Size = 12
    CharSet = 0
    Weight = 700
    Underline = 0 'False
    Italic = 0 'False
    Strikethrough = 0 'False
  EndProperty
  Height = 375
  Left = 3480
  TabIndex = 4
  Top = 1200
  Width = 1455
End

Begin VB.Label lbFMTypeLabel
  BackColor = &H80000005&
  BackStyle = 0 'Transparent
  Caption = "FM Type:"
  BeginProperty Font
    Name = "MS Sans Serif"
    Size = 12
    CharSet = 0
    Weight = 700
    Underline = 0 'False
    Italic = 0 'False
    Strikethrough = 0 'False
  EndProperty
  Height = 375
  Left = 2160
  TabIndex = 3
  Top = 1200
  Width = 1215
End

Begin VB.Menu mnuSystem
  Caption = "&System"
  Begin VB.Menu mnuSystemInterface
    Caption = "&Interface..."
    Shortcut = "^I"
  End
  Begin VB.Menu mnuSystemBarItem0
    Caption = "..-"
  End
  Begin VB.Menu mnuFileExit
    Caption = "E&xit"
  End
End

Begin VB.Menu mnuConnection
  Caption = "&Connection"
  Begin VB.Menu mnuConnectionConnect
    Caption = "&Connect*"
    Shortcut = {F3}
  End
  Begin VB.Menu mnuConnectionDisconnect
    Caption = "&Disconnect"
    Shortcut = {F4}
  End
End

Begin VB.Menu mnuConfig
  Caption = "C&onfiguration"
  Begin VB.Menu mnuConfigProperties
    Caption = "&Properties..."
Shortcut = ^P
End

Begin VB.Menu mnuConfigSettings
 Caption = "&Settings..."
Shortcut = ^S
End

Begin VB.Menu mnuActions
 Caption = "&Actions"
Begin VB.Menu mnuActionsPranking
 Caption = "&Pranking..."
Shortcut = {F5}
End
Begin VB.Menu mnuActionsCounters
 Caption = "&Counters..."
Shortcut = {F6}
End
Begin VB.Menu mnuActionsBar0
 Caption = "_"
End
Begin VB.Menu mnuActionsDate_Time
 Caption = "&Date & Time..."
Shortcut = {F7}
End
Begin VB.Menu mnuActionsMailClassAdjust
 Caption = "&Mail Class adjust..."
Shortcut = {F8}
End
Begin VB.Menu mnuActionsTextToDisp
 Caption = "&Text to display..."
Shortcut = {F9}
End

Begin VB.Menu mnuView
 Caption = "&View"
Begin VB.Menu mnuViewToolBar
 Caption = "&ToolBar"
Checked = -1 'True
End
Begin VB.Menu mnuViewStatusBar
 Caption = "Status &Bar"
Checked = -1 'True
End

Begin VB.Menu mnuHelp
 Caption = "&Help"
Begin VB.Menu mnuHelpAbout
 Caption = "&About..."
End
End

Attribute VB_Name = "frmMain"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_F redeclaredId = True
Attribute VB_Exposed = False

' frmMain is the main window of this application.
' It contains a menu and a toolbar that allow to establish connection,
' disconnect and open the sub forms.
' frmMain contains the Connection object that is used to communicate
' with the FM.
' Therefore also all events raised by the FM Control Library
' (FMCTRLLib) are
' processed in this form.

Option Explicit
Public WithEvents Con As FMCTRLLib.Connection  'Define Connection object variable
Attribute Con.VB_VarHelpID = -1

Public Enum FORMSTATES  'Enumerator for the different states of the frmMain
  fsConnected
  fsDisconnected
End Enum

Dim nFrankingCounter As Long  'Counter to number the Frankings
Private bDieCoverOpen As Boolean  'Flag indicating whether the Die Cover is open

'Event, raised whenever the Die Cover is closed
Private Sub Con_OnDieCoverClosed()
  bDieCoverOpen = False  'Reset flag
  sbStatusBar.SimpleText = "Connected"  'Write to statusbar of frmMain
  frmFranking.sbStatusBar.SimpleText = "Ready"  'Write to statusbar of frmFranking (does not need to be open)
End Sub

'Event, raised whenever the Die Cover is closed
Private Sub Con_OnDieCoverOpen()
  bDieCoverOpen = True  'Set flag
  sbStatusBar.SimpleText = "Die Cover open"  'Write to statusbar of frmMain
  frmFranking.sbStatusBar.SimpleText = "Die Cover open"  'Write to statusbar of frmFranking (does not need to be open)
End Sub

'Event, raised whenever an unexpected disconnection is detected
Private Sub Con_OnDisconnect()
  Dim i As Integer
  SetFormState (fsDisconnected)  'Set frmMain all controls on frmMain for state Disconnected
  For i = Forms.Count - 1 To 1 Step -1  'Close all sub forms
    Unload Forms(i)
  Next
End Sub

'Event, raised whenever FM goes to QUIET state without having received an explicit command to do so.
'This happens e.g. when the user timeout has run down.
Private Sub Con_OnQuiet(ByVal StatusCode As FMCTRLLib.RETVALS)
  If Not bDieCoverOpen Then  'Do not overwrite StatusBar if Die Cover is open
    frmFranking.sbStatusBar.SimpleText = "Quiet"  'Write to statusbar of frmFranking (does not need to be open)
  End If
  If Not (StatusCode = R_OK) Then  'Display a message for with the message from the high speed base
    ProcessRetVal (StatusCode)
  End If
End Sub

'Event, raised whenever a Franking is released
Private Sub Con_OnFranking(ByVal Postage As Currency, ByVal JobRest
As Integer, ByVal StatusCode As FMCTRLLib.RETVALS)
   Dim liItem As ListItem  'Define the needed variables
   Dim strJobRest As String
   nFrankingCounter = nFrankingCounter + 1  'Increment the Franking counter
   Set liItem = frmFranking.lvFrankings.ListItems.Add(1)  'Create a new list item on 1st position in the Frankings list on frmFranking

   'Column 0 is not used (1st column must be left aligne, but # should be right

   'aligned, so column 0 has width 0)
   liItem.SubItems(1) = CStr(nFrankingCounter)
   'Write # to list item
   liItem.SubItems(2) = Format(Postage, CreateCurrencyFormatMask)
   'Write formatted Postage to list item
   If JobRest = -1 Then  'If no Preselection (-1)
      strJobRest = "(none)"
   Else
      strJobRest = CStr(JobRest)
   End If
   'Write remaining number of items to list item
   liItem.SubItems(3) = strJobRest
   liItem.SubItems(4) = RetValToShortText(StatusCode)  'Write short description of state to list item

   ProcessRetVal (StatusCode)  'Process StatusCode
   (show message if necessary)
End Sub

'Event, raised when no more Tapes are available in Franking Mode FRK_TAPE

Private Sub Con_OnNoMoreTapes()
   Dim Actions As New FMCTRLLib.FMActions  'Create an new FMActions object
   Dim msg As String  'Define needed variables
   Dim MsgBoxStyle As VbMsgBoxStyle
   On Error GoTo ErrorHandler  'Jump to ErrorHandler in case of an error
   Actions.ActiveConnection = fMainForm.Con  'Define the connection to be used by Actions

   msg = "No more Tapes. Please refill."  'Prepare MsgBox text
   MsgBoxStyle = vbExclamation + vbRetryCancel  'MsgBox has Retry and Cancel button

   If vbRetry = MsgBox(msg, MsgBoxStyle) Then  'Show message if Retry button is pressed
      Screen.MousePointer = vbHourglass  'Show hourglass mouse pointer
      Actions.TapesPresent = False  'Tell FM that tapes are ready again
   Else
      Screen.MousePointer = vbHourglass  'Show hourglass mouse pointer
      Actions.GoQuiet  'Set FM to Quiet state (not ready for Franking)
   End If
   frmFranking.sbStatusBar.SimpleText = "Quiet"  'Write to statusbar of frmFranking (does not need to be open)

   fMainForm.fMessageText.Text = "Finished"  'Set message text
   fMainForm.fMessageText.Visible = True  'Show message text
End Sub

ErrorHandler:
    Resume Next
End If
Screen.MousePointer = vbDefault pointer
Set Actions = Nothing
Exit Sub
ErrorHandler:
Screen.MousePointer = vbDefault pointer
ErrorHandler
appropriate message)
Set Actions = Nothing
Exit Sub

'Event, raised whenever a rotor error occurs
Private Sub Con_OnRotorError(ByVal StatusCode As FMCTRLLib.RETVALS)
    Dim msg As String
    sbStatusBar.SimpleText = "Connected"
    'Write to statusbar of frmMain
    frmFranking.sbStatusBar.SimpleText = "Quiet"
    'Write to statusbar of frmFranking (does not need to be open)
    msg = "Rotor error!" + Chr(13)
    msg = msg + RetValToText(StatusCode)
    MsgBox msg, vbCritical
End Sub

Private Sub Form_Load()
    'frmMain is opene centred on the screen
    Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)
    Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)
    Me.Width = 7200
    'Define width of frmMain
    Me.Height = 4800
    'Define hight of frmMain
    Set Con = New FMCTRLLib.Connection object
    SetFormState (fsDisconnected)
    'Set frmMain all controls on frmMain for state Disconnected
    sbStatusBar.SimpleText = "Ready"
    'Write to statusbar of frmMain
    nComPort = 1
    'Initialize COM Port number
End Sub

'Called before frmMain is closed
Private Sub Form_QueryUnload(Cancel As Integer, UnloadModel As Integer)
    If Con.ConnectionString <> "" Then
        Cancel = True
        'If connection exists
        MsgBox "Can not exit while connection exists." + Chr(13) +
        "Please disconnect.", vbCritical
    End If
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Dim i As Integer
    For i = Forms.Count - 1 To 1 Step -1
        'Close all sub forms
Unload Forms(i)
Next
If Me.WindowState <> vbMinimized Then                 'Save window position
  SaveSetting App.Title, "Settings", "MainLeft", Me.Left
  SaveSetting App.Title, "Settings", "MainTop", Me.Top
End If
Set Con = Nothing                                     'Disassociate
Connection object
End Sub

'Information is filled into the fields and window is opened.

Private Sub mnuActionsCounters_Click()
  Dim Actions As New FMCTRLLib.FMActions                'Create an new
FMActions object
  Dim strFormatMask As String                           'Define needed
variables
  Dim cAscending As Currency                            Dim cDescending As Currency
  Dim nItems As Long                                     Dim bOn As Boolean
  Dim cValue As Currency                                 Dim cValue As Currency
On Error GoTo ErrorHandler:                            'Jump to ErrorHandler
in case of an error
  Screen.MousePointer = vbHourglass                     'Show hourglass mouse
  Actions.ActiveConnection = fMainForm.Con              'Define the
pointer
  Actions.ActiveConnection = fMainForm.Con              'connection to be used by Actions
strFormatMask = CreateCurrencyFormatMask               'Create format mask
for displaying money values
  Actions.GetCounterValues cAscending, cDescending, nItems 'Read
postal counters form FM

  With frmCounters                                      With frmCounters
    .lbAscending.Caption = Format(cAscending, strFormatMask) 'Display postal counters
    .lbDescending.Caption = Format(cDescending, strFormatMask)
    .lbItems.Caption = Format(nItems, "##,##0")            'Read Batch
Actions.BatchCounterRead bOn, nItems, cValue             Counter from FM
  .lbBatchActive = BoolToYesNo(bOn)                      'Display
Batch Counter values
    .lbBatchValue = Format(cValue, strFormatMask)
    .lbBatchItems = Format(nItems, "##,##0")             'Show default mouse
End With                                               'Display
  Screen.MousePointer = vbDefault                      'Show default mouse
  Set Actions = Nothing                                'Handle errors (show
  frmCounters.Show vbModal, Me                         'Disassociate Actions
End Sub
ErrorHandler:
  Screen.MousePointer = vbDefault                        'Show default mouse
  ErrorHandle appropriate message)                      'Handle errors (show
  Set Actions = Nothing                                  'Disassociate Actions
End Sub

'Read and display the current FM system time and open window

Private Sub mnuActionsDateTime_Click()
  Dim Actions As New FMCTRLLib.FMActions
  'Create an new
  FMActions object
  On Error GoTo ErrorHandler:
  'Jump to ErrorHandler
  in case of an error
  Screen.MousePointer = vbHourglass
  'Show hourglass mouse
  pointer
  Actions.ActiveConnection = fMainForm.Con
  'Define the
  connection to be used by Actions
  frmDateTime.lbDateTime.Caption = Actions.GetTimeDate
  'read
  Date and time from FM
  Screen.MousePointer = vbDefault
  'Show default mouse
  pointer
  Set Actions = Nothing
  'Disassociate Actions
  object
  frmDateTime.Show vbModal, Me
  Exit Sub
End Sub

ErrorHandler:
  Screen.MousePointer = vbDefault
  'Show default mouse
  pointer
  ErrorHandler
  'Handle errors (show
  appropriate message)
  Set Actions = Nothing
  'Disassociate Actions
  object
End Sub

'Prepare frmFranking for the application used by the connected FM.
'Some TextBoxes are used for different data entry depending on the
'application (Standard / Auto Tax)

Private Sub mnuActionsFranking_Click()
  Dim Config As New FMCTRLLib.FMConfig
  'Create an new
  FMConfig object
  Dim MCTxts() As String
  'Define the needed
  variables
  Dim i As Integer
  On Error GoTo ErrorHandler
  'Jump to ErrorHandler
  in case of an error
  Screen.MousePointer = vbHourglass
  'Show hourglass mouse
  pointer
  Config.ActiveConnection = fMainForm.Con
  'Define the
  connection to be used by Config
  With frmFranking
    .txtPostage.Text = "0"
    'Initialize
    Postage/AutoTax field
    .txtDptAcq.Text = "0"
    'Initialize
    Department/AcquisitionTax field
    If Config.AutoTaxFM = False Then
      'If it is a
      Standard application
      .lbPostage.Caption = "Postage:"
      'Label the fields
      according to usage
      .lbDptAcq.Caption = "Department:"
      .lbPreselectionDpt.Caption = "Preselection:"
    If Config.SettableMailClasses > 0 Then
      'If any
      Mail Classes are available
      MCTxts = Config.MailClassTexts
      'Read
Mail Class texts form FM
  .cmbMailClass.Clear 'Clear
the ComboBox
  For i = LBound(MCTxts) To UBound(MCTxts) 'Fill
Mail Class texts into the ComboBox
  .cmbMailClass.AddItem MCTxts(i), i
  Next i
  .cmbMailClass.ListIndex = 0 'Initialize Mail Class ComboBox (choose 1st entry)
  Else 'No
Mailclasses available
  .cmbMailClass.Enabled = False 'Disable
Mail Class ComboBox
  End If
.
  .cmbFrankMode.Clear 'Clear Mail Class ComboBox
  If Config.FrankModeAvailNorm Then 'If Franking Mode Normal is available
    .cmbFrankMode.AddItem "Normal"
    'Insert entry into ComboBox
    .cmbFrankMode.ItemData(.cmbFrankMode.NewIndex) = 0 'Add item data to be able to identify the choosen entry
    End If
    If Config.FrankModeAvailTape Then 'Same as above for Franking Mode Tape
      .cmbFrankMode.AddItem "Tapes"
      .cmbFrankMode.ItemData(.cmbFrankMode.NewIndex) = 1
      End If
      If Config.FrankModeAvailLetter Then 'Same as above for Franking Mode Letter
        .cmbFrankMode.AddItem "Letters"
        .cmbFrankMode.ItemData(.cmbFrankMode.NewIndex) = 2
        End If
        If Config.FrankModeAvailItem Then 'Same as above for Franking Mode Item
          .cmbFrankMode.AddItem "Items"
          .cmbFrankMode.ItemData(.cmbFrankMode.NewIndex) = 3
          End If
          .cmbFrankMode.ListIndex = 0 'Initialize Mail Class ComboBox (choose 1st entry)
          .cmbFrankMode.Enabled = .cmbFrankMode.ListIndex > 1 'Enable Franking Mode ComboBox if there is more than one entry
          'Limit the text length of the Preselection field to 3 (max = "999")
          txtPreselectionDpt.MaxLength = 3 'Auto Tax application
          .lbPostage.Caption = "Auto Tax:" 'Label the fields according to usage
          .lbDptAcq.Caption = "Acquisition Tax:" .lbPreselectionDpt.Caption = "Department:
          .txtPreselectionDpt.Enabled = True 'Enable the Department field
          .txtPreselectionDpt.Width = 975 'Set
width of Department field (more space necessary than for
Preselection).txtPreselectionDpt.MaxLength = 0 'No
text length limit
  .txtDptAcq.Enabled = Config.AcquisitionTaxAvailable 'Enable Acquisition Tax field if application supports Acquisition Tax
  .cmbMailClass.Enabled = False 'Disable Mail Class field (Auto Tax does not support Mail Classes)
  .cmbFrankMode.Enabled = False 'Disable Franking Mode field (Auto Tax does not support Franking Modes)
ErrorHandler

Attribute VB_Name = "ErrorHandler"
Option Explicit

'Displays an error message for the last error occurred.
'The exceptions of the FM Control library are converted using vbCrLf
'as offset.

Sub ErrorHandler()
    Dim msg As String
    Dim MsgBoxStyle As VbMsgBoxStyle

    MsgBoxStyle = vbCritical

    Select Case Err.Number
        Case NOCONNECTION + vbObjectError To UNKNOWN + vbObjectError
            'All FM Control ExceptionIDs
            msg = Err.Description + " (" + CStr(Err.Number - vbObjectError) + ")".
            'Convert error numbers using vbObjectError
        Case 13
            'Type mismatch - occurs when trying to convert a text into a number
            msg = "One or more of the entered values are invalid" + " (" + CStr(Err.Number) + ")." "
        Case Else
            'Display all other messages as they are
            msg = Err.Description + " (" + CStr(Err.Number) + ")." "
    End Select

    'msg = msg + Chr(13) + "Source: " + Err.Source
    'Display source on a new line
    MsgBox msg, MsgBoxStyle

End Sub
Attribute VB_Name = "Start"
'Application starts with this module

Option Explicit

Public fMainForm As frmMain
Global nComPort As Integer

'Application starts with this Sub

Sub Main()
    Set fMainForm = New frmMain        .'Create new frmMain
    fMainForm.Show
End Sub
VERSION 5.00
Begin VB.Form frmAbout
    BorderStyle = 3 'Fixed Dialog
    Caption = "About"
    ClientHeight = 3630
    ClientLeft = 45
    ClientTop = 330
    ClientWidth = 5865
    ClipControls = 0 'False
    Icon = "frmAbout.frx":0000
    LinkTopic = "Form1"
    MaxButton = 0 'False
    MinButton = 0 'False
    ScaleHeight = 3630
    ScaleWidth = 5865
    ShowInTaskbar = 0 'False
    StartUpPosition = 1 'CenterOwner
    Tag = "About Project"
End
Begin VB.PictureBox picIcon
    AutoSize = -1 'True
    BackColor = &H00C0C0C0&
    ClipControls = 0 'False
    Height = 540
    Left = 240
    Picture = "frmAbout.frx":0442
    ScaleHeight = 480
    ScaleMode = 0 'User
    ScaleWidth = 480
    TabIndex = 2
    TabStop = 0 'False
    Top = 240
    Width = 540
End
Begin VB.CommandButton cmdOK
    Cancel = -1 'True
    Caption = "OK"
    Default = -1 'True
    Height = 345
    Left = 4245
    TabIndex = 0
    Tag = "OK"
    Top = 2625
    Width = 1467
End
Begin VB.CommandButton cmdSysInfo
    Caption = "&System Info..."
    Height = 345
    Left = 4260
    TabIndex = 1
    Tag = "&System Info..."
    Top = 3075
    Width = 1452
End
Begin VB.Label lbLibraryVerison
    Caption = "Library Version"
    Height = 255
    Left = 1050
    TabIndex = 8
    Top = 960
    Width = 4095
End
Begin VB.Label Label1
Caption = "Ascom Autelca AG Mailing Systems"
Height = 255
Left = 1050
TabIndex = 7
Top = 1320
Width = 4095
End
Begin VB.LabelLabel1Description
Caption = $"frmAbout.frx":0884
ForeColor = &H00000000&
Height = 650
Left = 1050
TabIndex = 6
Tag = "App Description"
Top = 1680
Width = 4095
End
Begin VB.Label lblTitle
Caption = "Application Title"
ForeColor = &H00000000&
Height = 240
Left = 1050
TabIndex = 5
Tag = "Application Title"
Top = 240
Width = 4095
End
Begin VB.Line Line1
BorderColor = &H00808080&
BorderStyle = 6 'Inside Solid
Index = 1
X1 = 225
X2 = 5657
Y1 = 2430
Y2 = 2430
End
Begin VB.Line Line1
BorderColor = &H00FFFFFD&
BorderWidth = 2
Index = 0
X1 = 240
X2 = 5657
Y1 = 2445
Y2 = 2445
End
Begin VB.Label Label1Version
Caption = "Version"
Height = 255
Left = 1050
TabIndex = 4
Tag = "Version"
Top = 600
Width = 4095
End
Begin VB.Label lblDisclaimer
ForeColor = &H00000000&
Height = 825
Left = 255
TabIndex = 3
Tag = "Warning: ..."]
Top = 2625
Width = 3750
End
End
Attribute VB_Name = "frmAbout"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

' Reg Key Security Options...
Const KEY_ALL_ACCESS = &H2000F

' Reg Key ROOT Types...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1
Const REG_DWORD = 4

Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"

Private Declare Function RegOpenKeyEx Lib "advapi32" Alias
"RegOpenKeyExA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal ulOptions As Long, ByVal samDesired As Long, ByRef phkResult As Long) As Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias
"RegQueryValueExA" (ByVal hKey As Long, ByVal lpValueName As String, ByVal lpReserved As Long, ByVal lpType As Long, ByVal lpData As String, ByVal lpReserved2 As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As Long

Private Sub Form_Load()
    & App.Revision
    lblTitle.Caption = App.Title
    lblLibraryVersion.Caption = "FM Control library version " + fMainForm.Con.LibraryVersion
    'Display Version of FM Control Library
End Sub

Private Sub cmdSysInfo_Click()
    Call StartSysInfo
End Sub

Private Sub cmdOK_Click()
    Unload Me
End Sub

Public Sub StartSysInfo()
    On Error GoTo SysInfoErr

    Dim rc As Long
    Dim SysInfoPath As String

' Try To Get System Info Program Path\Name From Registry...
If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO,
gREGVALSYSINFO, SysInfoPath) Then
  ' Try To Get System Info Program Path Only From Registry...
  ElseIf GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC,
gREGVALSYSINFOLOC, SysInfoPath) Then
    ' Validate Existance Of Known 32 Bit File Version
    If (Dir(SysInfoPath & "\MSINFO32.EXE") <> ") Then
      SysInfoPath = SysInfoPath & "\MSINFO32.EXE"
    End If
  End If
  ' Error - File Can Not Be Found...
  Else
    GoTo SysInfoErr
  End If
  ' Error - Registry Entry Can Not Be Found...
  Else
    GoTo SysInfoErr
  End If

Call Shell(SysInfoPath, vbNormalFocus)

Exit Sub

SysInfoErr:
  MsgBox "System Information Is Unavailable At This Time",
  vbOKOnly
End Sub

Public Function GetKeyValue(KeyRoot As Long, KeyName As String,
  SubKeyRef As String, ByRef KeyVal As String) As Boolean
  Dim i As Long ' Loop
  Dim rc As Long ' Return
  Dim hKey As Long ' Handle
  Dim hDepth As Long ' Data
  Dim KeyValType As Long ' Size
  Dim tmpVal As String
  Dim KeyValSize As Long

  ' Tempory Storage For A Registry Key Value
  ' Of Registry Key Variable

  ' Open RegKey Under KeyRoot {HKEY_LOCAL_MACHINE...}
  '--------------------------------------------------------
  rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey) ' Open Registry Key

  If (rc <> ERROR_SUCCESS) Then GoTo GetKeyErr

  ' Handle Error...
  tmpVal = String$(1024, 0) ' Allocate Variable Space
KeyValSize = 1024
Variable Size

' Retrieve Registry Key Value...
rc = RegQueryValueEx(hKey, SubKeyRef, 0, KeyValType, tmpVal, KeyValSize)  ' Get/Create Key Value

If (rc <> ERROR_SUCCESS) Then GoTo GetKeyErrror  ' Handle Errors

tmpVal = VBA.Left(tmpVal, InStr(tmpVal, VBA.Chr(0)) - 1)

' Determine Key Value Type For Conversion...
Select Case KeyValType
  Case REG_SZ
    String
    KeyVal = tmpVal
  Case REG_DWORD
    Double
    Word Registry Key Data Type
    For i = Len(tmpVal) To 1 Step -1
      Convert Each Bit
      KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1)))
    Next
    Build Value Char. By Char.
    KeyVal = Format$("&h" + KeyVal)
  Case Else
End Select

getKeyValue = True

Success
rc = RegCloseKey(hKey)
Registry Key
Exit Function

getKeyError:  ' Cleanup After An Error Has Occured...
KeyVal = ""
Return Val To Empty String
getKeyValue = False
Return
Failure
rc = RegCloseKey(hKey)
Registry Key
End Function
VERSION 5.00
Object = "{BDC217C8-ED16-11CD-956C-0000C04E400A}"#1.1#0"; "TABCTL32.OCX"
Begin VB.Form frmConfig
  BorderStyle = 3 'Fixed Dialog
  Caption = "Configuration"
  ClientHeight = 4920
  ClientLeft = 2565
  ClientTop = 1500
  ClientWidth = 5910
  Icon = "frmConfig.frx":0000
  KeyPreview = -1 'True
  LinkTopic = "Form1"
  MaxButton = 0 'False
  MinButton = 0 'False
  ScaleHeight = 4920
  ScaleWidth = 5910
  ShowInTaskbar = 0 'False
  StartUpPosition = 1 'CenterOwner
Begin TabDlg.STab tabConfig
  Height = 4215
  Left = 120
  TabIndex = 13
  Top = 120
  Width = 5655
  _ExtentX = 9975
  _ExtentY = 7435
  _Version = 393216
  Style = 1
  Tabs = 2
  TabsPerRow = 2
  TabHeight = 520
  TabCaption(0) = "Properties"
  TabPicture(0) = "frmConfig.frx":0442
End Tab
End Tab
Begin VB.Form fraLowCreditWarn
  Caption = "Low Credit warning"
  Height = 1095
  Left = -72120
  TabIndex = 53
  Top = 480
End VB
Width = 2535
Begin VB.CommandButton cmdLowCreditWarnDeactivate
 Caption = "Deactivate"
 Height = 495
 Left = 1320
 TabIndex = 7
 Top = 360
 Width = 975
End

Begin VB.CommandButton cmdLowCreditWarnActivate
 Caption = "Activate"
 Height = 495
 Left = 240
 TabIndex = 6
 Top = 360
 Width = 975
End

Begin VB.Frame fraUserTimeout
 Caption = "User timeout"
 Height = 1095
 Left = -72120
 TabIndex = 52
 Top = 2880
 Width = 2535
Begin VB.CommandButton cmdUserTimeoutDeactivate
 Caption = "Deactivate"
 Height = 495
 Left = 1320
 TabIndex = 11
 Top = 360
 Width = 975
End

Begin VB.CommandButton cmdUserTimeoutActivate
 Caption = "Activate"
 Height = 495
 Left = 240
 TabIndex = 10
 Top = 360
 Width = 975
End

Begin VB.Frame fraKeyboard
 Caption = "FM keyboard"
 Height = 1095
 Left = -74760
 TabIndex = 51
 Top = 2880
 Width = 2055
Begin VB.CommandButton cmdKeyboardUnlock
 Caption = "Unlock"
 Height = 495
 Left = 1080
 TabIndex = 5
 Top = 360
 Width = 735
End

Begin VB.CommandButton cmdKeyboardLock
 Caption = "Lock"
 Height = 495
 Left = 240
TabIndex = 4
Top = 360
Width = 735
End

Begin VB.Frame fraHighValue
Caption = "High Value limit"
Height = 1095
Left = -72120
TabIndex = 50
Top = 1680
Width = 2535
Begin VB.CommandButton cmdHVDeactivate
Caption = "Deactivate"
Height = 495
Left = 1320
TabIndex = 9
Top = 360
Width = 975
End

Begin VB.CommandButton cmdHVActivate
Caption = "Activate"
Height = 495
Left = 240
TabIndex = 8
Top = 360
Width = 975
End

End

Begin VB.Frame fraFeatures
Caption = "Features"
Height = 1020
Left = 2880
TabIndex = 43
Top = 2160
Width = 2535
Begin VB.Label lbNrMailClass
Caption = "4"
Height = 255
Left = 1600
TabIndex = 49
Top = 720
Width = 900
End

Begin VB.Label lbMargin
Caption = "Yes"
Height = 255
Left = 1600
TabIndex = 48
Top = 480
Width = 900
End

Begin VB.Label lbAutoDate
Caption = "Yes"
Height = 255
Left = 1600
TabIndex = 47
Top = 240
Width = 900
End

Begin VB.Label lbNrMailClassLabel
Caption = "Nr. of Mail Classes:"
Height = 255
Left = 120
TabIndex = 46
Top = 720
Width = 1400
End

Begin VB.Label lbMarginLabel
Caption = "Margin settable:"
Height = 255
Left = 120
TabIndex = 45
Top = 480
Width = 1400
End

Begin VB.Label lbAutoDateLabel
Caption = "Automatic date:"
Height = 255
Left = 120
TabIndex = 44
Top = 240
Width = 1400
End

End

Begin VB.Frame fraBase
Caption = "Base"
Height = 1020
Left = 240
TabIndex = 38
Top = 2160
Width = 2535

Begin VB.Label lbBaseSoftware
Caption = "0000000B"
Height = 255
Left = 1000
TabIndex = 42
Top = 600
Width = 1500
End

Begin VB.Label lbBaseModel
Caption = "B335P"
Height = 255
Left = 1000
TabIndex = 41
Top = 300
Width = 1500
End

Begin VB.Label lbSoftwareLabel
Caption = "Software:"
Height = 255
Left = 120
TabIndex = 40
Top = 600
Width = 700
End

Begin VB.Label lbModelLabel
Caption = "Model:"
Height = 255
Left = 120
TabIndex = 39
Top = 300
Width = 700

Begin VB.Frame fraPrintImage
Caption = "Print image"
Height = 1500
Left = 2880
TabIndex = 27
Top = 480
Width = 2535
Begin VB.Label lbPrintImage
Caption = "99999.00"
Height = 255
Left = 1500
TabIndex = 37
Top = 1200
Width = 1000
End

Begin VB.Label lbLastDecade
Caption = "0..9"
Height = 255
Left = 1500
TabIndex = 36
Top = 960
Width = 1000
End

Begin VB.Label lbFixZeros
Caption = "0"
Height = 255
Left = 1500
TabIndex = 35
Top = 720
Width = 1000
End

Begin VB.Label lbDecPos
Caption = "2"
Height = 255
Left = 1500
TabIndex = 34
Top = 480
Width = 1000
End

Begin VB.Label lbNrDecades
Caption = "5"
Height = 255
Left = 1500
TabIndex = 33
Top = 240
Width = 1000
End

Begin VB.Label lbPrintImageLabel
Caption = "Print image:
Height = 255
Left = 120
TabIndex = 32
Top = 1200
Width = 1300
End

Begin VB.Label lbLastDecadeLabel
Caption = "Last decade type:
Height = 255
Left = 120
TabIndex = 31
Top = 960
Width = 1300

End
Begin VB.Label lbFixZerosLabel Caption = "Nr. of fixed zeros:
Height = 255
Left = 120
TabIndex = 30
Top = 720
Width = 1300

End
Begin VB.Label lbDecPosLabel Caption = "Decimal position:
Height = 255
Left = 120
TabIndex = 29
Top = 480
Width = 1300

End
Begin VB.Label lbNrDecadesLabel Caption = "Nr. of decades:
Height = 255
Left = 120
TabIndex = 28
Top = 240
Width = 1300

End
End
Begin VB.Frame fraGeneral Caption = "General"
Height = 1500
Left = 240
TabIndex = 18
Top = 480
Width = 2535
Begin VB.Label lbFMSSoftware Caption = "JFAB005H"
Height = 255
Left = 1200
TabIndex = 26
Top = 720
Width = 1300

End
Begin VB.Label lbSerialNr Caption = "123456"
Height = 255
Left = 1200
TabIndex = 25
Top = 480
Width = 1300

End
Begin VB.Label lbFMTYPE Caption = "F3XXPLUS"
Height = 255
Left = 1200
TabIndex = 24
Top = 240
Width = 1300

End
Begin VB.Label lbApplication
    Caption = "Auto Tax with Acquisition Tax"
    Height  = 495
    Left    = 1200
    TabIndex = 23
    Top     = 960
    Width   = 1300
End

Begin VB.Label lbApplicationLabel
    Caption = "Application:"
    Height  = 255
    Left    = 120
    TabIndex = 22
    Top     = 960
    Width   = 1000
End

Begin VB.Label lbFMSLabel
    Caption = "FM Software:"
    Height  = 255
    Left    = 120
    TabIndex = 21
    Top     = 720
    Width   = 1000
End

Begin VB.Label lbSerialNrLabel
    Caption = "Serial Nr.:"
    Height  = 255
    Left    = 120
    TabIndex = 20
    Top     = 480
    Width   = 1000
End

Begin VB.Label lbFMTypelabel
    Caption = "FM Type:"
    Height  = 255
    Left    = 120
    TabIndex = 19
    Top     = 240
    Width   = 1000
End

End

Begin VB.Frame fraFrankingModes
    Caption = "Franking Modes"
    Height  = 600
    Left    = 2880
    TabIndex = 16
    Top     = 3360
    Width   = 2535
Begin VB.Label lbFrankModes
    Caption = "Normal / Tapes / Letters / Items"
    Height  = 255
    Left    = 120
    TabIndex = 17
    Top     = 300
    Width   = 2380
End

End

Begin VB.Frame fraFrankMenu
    Caption = "Franking menu"
    Height  = 1095
    Left    = -74760
TabIndex = 15
Top = 1680
Width = 2055
Begin VB.CommandButton cmdFrankMenuUnlock
  Caption = "Unlock"
  Height = 495
  Left = 1080
  TabIndex = 3
  Top = 360
  Width = 735
End

Begin VB.CommandButton cmdFrankMenuLock
  Caption = "Lock"
  Height = 495
  Left = 240
  TabIndex = 2
  Top = 360
  Width = 735
End

Begin VB.Frame fraDescReg
  Caption = "Descending register"
  Height = 1095
  Left = -74760
  TabIndex = 14
  Top = 480
  Width = 2055
Begin VB.CommandButton cmdDescRegHide
  Caption = "Hide"
  Height = 495
  Left = 1080
  TabIndex = 1
  Top = 360
  Width = 735
End

Begin VB.CommandButton cmdDescRegShow
  Caption = "Show"
  Height = 495
  Left = 240
  TabIndex = 0
  Top = 360
  Width = 735
End
End

Begin VB.CommandButton cmdClose
  Cancel = -1 'True
  Caption = "Close"
  Height = 375
  Left = 4680
  TabIndex = 12
  Top = 4455
  Width = 1095
End

Attribute VB_Name = "frmConfig"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'
frmConfig displays the properties of the connected
Option Explicit

Private Sub cmdClose_Click()
    Unload Me
End Sub

Private Sub cmdDescRegHide_Click()
    Dim Config As New FMCtrlLib.FMConfig
    'Create an new FMConfig object
    On Error GoTo ErrorHandler:
    'Jump to ErrorHandler in case of an error
    Screen.MousePointer = vbHourglass
    'Show hourglass mouse pointer
    Config.ActiveConnection = fMainForm.Con
to be used by Config
    'Define the connection
    Config.DescRegHide
    'Hide Descending register
    Screen.MousePointer = vbDefault
    'Show default mouse pointer
    Set Config = Nothing
    'Disassociate Config object
    Exit Sub
    ErrorHandler:
    Screen.MousePointer = vbDefault
    'Show default mouse pointer
    ErrorHandelr
    'Handle errors (show appropriate message)
    Set Config = Nothing
    'Disassociate Config object
End Sub

Private Sub cmdDescRegShow_Click()
    'See Sub cmdDescRegHide_Click for commentary (same structure)
    Dim Config As New FMCtrlLib.FMConfig
    On Error GoTo ErrorHandler:
    Screen.MousePointer = vbHourglass
    Config.ActiveConnection = fMainForm.Con
    'Show Descending register
    Screen.MousePointer = vbDefault
    Set Config = Nothing
    Exit Sub
    ErrorHandler:
    Screen.MousePointer = vbDefault
    ErrorHandelr
    Set Config = Nothing
End Sub

Private Sub cmdFrankMenuLock_Click()
    'See Sub cmdDescRegHide_Click for commentary (same structure)
    Dim Config As New FMCtrlLib.FMConfig
    On Error GoTo ErrorHandler:
    Screen.MousePointer = vbHourglass
    Config.ActiveConnection = fMainForm.Con
Config.FrankMenuLock = vbDefault
Set Config = Nothing
Exit Sub
ErrorHandler:
Screen.MousePointer = vbDefault
ErrorHandler
Set Config = Nothing
End Sub
Private Sub cmdFrankMenuUnlock_Click()
'Unlock Franking menu
Screen.MousePointer = vbDefault
Set Config = Nothing
Exit Sub
ErrorHandler:
Screen.MousePointer = vbDefault
ErrorHandler
Set Config = Nothing
End Sub
Private Sub cmdHVActivate_Click()
'Activate High Value limit
Screen.MousePointer = vbDefault
Set Config = Nothing
Exit Sub
ErrorHandler:
Screen.MousePointer = vbDefault
ErrorHandler
Set Config = Nothing
End Sub
Private Sub cmdHVDeactivate_Click()
'Deactivate High Value limit
Screen.MousePointer = vbDefault
Set Config = Nothing
Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault
    ErrorHandler.
    Set Config = Nothing
End Sub

Private Sub cmdKeyboardLock_Click() 'See Sub cmdDescRegHide_Click for commentary (same structure)
    Dim Config As New FMCTRLLib.FMConfig
    On Error GoTo ErrorHandler:
    Screen.MousePointer = vbHourglass
    Config.ActiveConnection = fMainForm.Con
    Config.KeyboardLock 'Lock FM keyboard
    Screen.MousePointer = vbDefault
    Set Config = Nothing
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault
    ErrorHandler.
    Set Config = Nothing
End Sub

Private Sub cmdKeyboardUnlock_Click() 'See Sub cmdDescRegHide_Click for commentary (same structure)
    Dim Config As New FMCTRLLib.FMConfig
    On Error GoTo ErrorHandler:
    Screen.MousePointer = vbHourglass
    Config.ActiveConnection = fMainForm.Con
    Config.KeyboardUnlock 'Unlock FM keyboard
    Screen.MousePointer = vbDefault
    Set Config = Nothing
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault
    ErrorHandler.
    Set Config = Nothing
End Sub

Private Sub cmdLowCreditWarnActivate_Click() 'See Sub cmdDescRegHide_Click for commentary (same structure)
    Dim Config As New FMCTRLLib.FMConfig
    On Error GoTo ErrorHandler:
    Screen.MousePointer = vbHourglass
    Config.ActiveConnection = fMainForm.Con
    Config.WarningLowCreditActivate 'Activate Low Credit warning
    Screen.MousePointer = vbDefault
    Set Config = Nothing
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault
    ErrorHandler.
    Set Config = Nothing
End Sub
Private Sub cmdLowCreditWarnDeactivate_Click() 'See Sub cmdDescRegHide_Click for commentary (same structure)
    Dim Config As New FMCTRLLib.FMConfig
    On Error GoTo ErrorHandler:
    Screen.MousePointer = vbHourglass
    Config.ActiveConnection = fMainForm.Con

    Config.WarningLowCreditDeactivate 'Deactivate Low Credit warning
    Screen.MousePointer = vbDefault
    Set Config = Nothing
    Exit Sub
    ErrorHandler:
    Screen.MousePointer = vbDefault
    ErrorHandler
    Set Config = Nothing
End Sub

Private Sub cmdUserTimeOutActivate_Click() 'See Sub cmdDescRegHide_Click for commentary (same structure)
    Dim Config As New FMCTRLLib.FMConfig
    On Error GoTo ErrorHandler:
    Screen.MousePointer = vbHourglass
    Config.ActiveConnection = fMainForm.Con

    Config.UserTimeoutActivate 'Activate User Timeout
    Screen.MousePointer = vbDefault
    Set Config = Nothing
    Exit Sub
    ErrorHandler:
    Screen.MousePointer = vbDefault
    ErrorHandler
    Set Config = Nothing
End Sub

Private Sub cmdUserTimeOutDeactivate_Click() 'See Sub cmdDescRegHide_Click for commentary (same structure)
    Dim Config As New FMCTRLLib.FMConfig
    On Error GoTo ErrorHandler:
    Screen.MousePointer = vbHourglass
    Config.ActiveConnection = fMainForm.Con

    Config.UserTimeoutDeactivate 'Deactivate User Timeout
    Screen.MousePointer = vbDefault
    Set Config = Nothing
    Exit Sub
    ErrorHandler:
    Screen.MousePointer = vbDefault
    ErrorHandler
    Set Config = Nothing
End Sub
VERSION 5.00

Begin VB.Form frmCounters
    BorderStyle = 3 "Fixed Dialog"
    Caption = "Dialog Caption"
    ClientHeight = 2520
    ClientLeft = 2760
    ClientTop = 3750
    ClientWidth = 5775
    Icon = "frmCounters.frx":0000
    LinkTopic = "Form1"
    MaxButton = 0 'False
    MinButton = 0 'False
    ScaleHeight = 2520
    ScaleWidth = 5775
    ShowInTaskbar = 0 'False
    StartUpPosition = 1 'CenterOwner

Begin VB.Frame Frame2
    Caption = "Batch counter"
    Height = 1695
    Left = 2760
    TabIndex = 5
    Top = 120
    Width = 2895

Begin VB.CommandButton cmdBatchActivate
    Caption = "&Activate"
    Height = 375
    Left = 1920
    TabIndex = 1
    Top = 240
    Width = 855
End

Begin VB.CommandButton cmdBatchClear
    Caption = "&Clear"
    Height = 375
    Left = 1920
    TabIndex = 3
    Top = 1200
    Width = 855
End

Begin VB.CommandButton cmdBatchRead
    Caption = "&Read"
    Height = 375
    Left = 1920
    TabIndex = 2
    Top = 720
    Width = 855
End

Begin VB.Label lbBatchItems
    Alignment = 1 "Right Justify"
    Caption = "4578"
    Height = 255
    Left = 840
    TabIndex = 17
    Top = 1200
    Width = 855
End

Begin VB.Label lbBatchActiveLabel
    Caption = "Active:"
    Height = 255
    Left = 240
    TabIndex = 16
End
Top = 480
Width = 495
End

Begin VB.Label lbBatchValue
Alignment = 1 'Right Justify
Caption = "4567899"
Height = 255
Left = 840
TabIndex = 15
Top = 840
Width = 855
End

Begin VB.Label lbBatchActive
Alignment = 1 'Right Justify
Caption = "Yes"
Height = 255
Left = 840
TabIndex = 14
Top = 480
Width = 855
End

Begin VB.Label lbBatchValueLabel
Caption = "Value:"
Height = 255
Left = 240
TabIndex = 7
Top = 840
Width = 495
End

Begin VB.Label lbBatchItemsLabel
Caption = "Items:"
Height = 255
Left = 240
TabIndex = 6
Top = 1200
Width = 495
End

End

Begin VB.Frame Framel
Caption = "Postal counter"
Height = 1695
Left = 120
TabIndex = 4
Top = 120
Width = 2535

Begin VB.Label lbItems
Alignment = 1 'Right Justify
Caption = "9'999'999"
Height = 255
Left = 1200
TabIndex = 13
Top = 1200
Width = 1095
End

Begin VB.Label lbDescending
Alignment = 1 'Right Justify
Caption = "9'999'999.00"
Height = 255
Left = 1200
TabIndex = 12
Top = 840
Width = 1095

End

Begin VB.Label lbAscending
  Alignment = 1 'Right Justify
  Caption = "9,999,999.00"
  Height = 255
  Left = 1200
  TabIndex = 11
  Top = 480
  Width = 1095

End

Begin VB.Label lbItemsLabel
  Caption = "Items:"
  Height = 255
  Left = 240
  TabIndex = 10
  Top = 1200
  Width = 975

End

Begin VB.Label lbDescendingLabel
  Caption = "Descending:"
  Height = 255
  Left = 240
  TabIndex = 9
  Top = 840
  Width = 975

End

Begin VB.Label lbAscendingLabel
  Caption = "Ascending:"
  Height = 255
  Left = 240
  TabIndex = 8
  Top = 480
  Width = 975

End

End

Begin VB.CommandButton cmdClose
  Cancel = -1 'True
  Caption = "Close"
  Height = 375
  Left = 4440
  TabIndex = 0
  Top = 2040
  Width = 1215

End

End

Attribute VB_Name = "frmCounters"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False

'frmCounters displays the current values of the counters
'and allows to control the Batch Counter of the FM.

Option Explicit

'Activates Batch Counter (switch it ON)

Private Sub cmdBatchActivate_Click()
  Dim Actions As New FMCTRLLib.FMActions
  'Create an new FMActions object
On Error Goto ErrorHandler:
case of an error
   Screen.MousePointer = vbHourglass
pointer
   Actions.ActiveConnection = fMainForm.Con
to be used by Actions

   Actions.BatchCounterActivate
Counter in the FM
   Screen.MousePointer = vbDefault
pointer
   Set Actions = Nothing
object
   Exit Sub
ErrorHandler:
   Screen.MousePointer = vbDefault
pointer
   ErrorHandler
appropriate message
   Set Actions = Nothing
object
   End Sub

'Clears the Batch Counter and displays the counter values and state
(ON/OFF) before the clearing

Private Sub cmdBatchClear_Click()
   Dim Actions As New FMCTRLLib.FMActions
object
   Dim strFormatMask As String
   Dim bOn As Boolean
   Dim nItems As Long
   Dim cValue As Currency
   On Error Goto ErrorHandler:
   case of an error
      Screen.MousePointer = vbHourglass
   pointer
      Actions.ActiveConnection = fMainForm.Con
   to be used by Actions
   strFormatMask = CreateCurrencyFormatMask
   'Create format mask for displaying money values
   Actions.BatchCounterClear bOn, nItems, cValue
   'Clear Batch Couter and receive its values before the clearing
   lbBatchActive = BoolToYesNo(bOn)
   'Display the last counter values before clear
   lbBatchValue = Format(cValue, strFormatMask)
   lbBatchItems = Format(nItems, "##,##0")
   Screen.MousePointer = vbDefault
pointer
   Set Actions = Nothing
object
   Exit Sub
ErrorHandler:
   Screen.MousePointer = vbDefault
pointer
   ErrorHandler
       'Show default mouse
   'Disassociate Actions
   'Show default mouse
   'Handle errors (show
   'Disassociate Actions
appropriate message)
  Set Actions = Nothing  'Disassociate Actions
End Sub

'Reads the current counter values and state (ON/OFF)

Private Sub cmdBatchRead_Click()
  'See Sub cmdBatchClear_Click for commentary (same structure)
  Dim Actions As New FMCTRLLib.FMActions
  Dim strFormatMask As String
  Dim bOn As Boolean
  Dim nItems As Long
  Dim cValue As Currency
  On Error GoTo ErrorHandler:
    Screen.MousePointer = vbHourglass
    Actions.ActiveConnection = fMainForm.Con
    strFormatMask = CreateCurrencyFormatMask
    Actions.BatchCounterRead bOn, nItems, cValue  'Read Batch Counter from FM
    lbBatchActive = BoolToYesNo(bOn)  'Display Batch Counter values
    lbBatchValue = Format(cValue, strFormatMask)
    lbBatchItems = Format(nItems, "##,##0")
    Screen.MousePointer = vbDefault
    Set Actions = Nothing
  Exit Sub
ErrorHandler:
  Screen.MousePointer = vbDefault
  ErrorHandler
  Set Actions = Nothing
End Sub

Private Sub cmdClose_Click()
  Unload Me
End Sub
VERSION 5.00
Begin VB.Form frmDateTime
    BorderStyle = 3 'Fixed Dialog
    Caption = "Date & Time"
    ClientHeight = 1680
    ClientLeft = 2760
    ClientTop = 3750
    ClientWidth = 2535
    Icon = "frmDateTime.frx":0000
    LinkTopic = "Form1"
    MaxButton = 0 'False
    MinButton = 0 'False
    ScaleHeight = 1680
    ScaleWidth = 2535
    ShowlnTaskbar = 0 'False
    StartUpPosition = 1 'CenterOwner
End VB.Frame fraDateTime
    Caption = "FM system date & time (local)"
    Height = 975
    Left = 120
    TabIndex = 2
    Top = 120
    Width = 2295
Begin VB.Label lbDateTime
    Alignment = 2 'Center
    BackStyle = 0 'Transparent
    Caption = "25.12.00 13:25:34"
    Height = 255
    Left = 120
    TabIndex = 3
    Top = 480
    Width = 2055
End

Begin VB.CommandButton cmdRefresh
    Caption = "&Refresh"
    Default = -1 'True
    Height = 375
    Left = 120
    TabIndex = 0
    Top = 1200
    Width = 1095
End

Begin VB.CommandButton cmdClose
    Cancel = -1 'True
    Caption = "Close"
    Height = 375
    Left = 1320
    TabIndex = 1
    Top = 1200
    Width = 1095
End

Attribute VB_Name = "frmDateTime"
Attribute VB_GlobalsNamespace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'frmDateTime displays the current local system time and date.
'The display can be updated by clicking a button.

Option Explicit

Private Sub cmdClose_Click()
    Unload Me
End Sub
'Read and display the current FM system time

Public Sub cmdRefresh_Click()
    Dim Actions As New FMCTRLLib.FMActions
    FMActions object
    On Error GoTo ErrorHandler:
    in case of an error
    Screen.MousePointer = vbHourglass
    pointer
    Actions.ActiveConnection = fMainForm.Con
    connection to be used by Actions

    gbDateTime.Caption = Actions.GetTimeDate
    from FM

    Screen.MousePointer = vbDefault
    pointer
    Set Actions = Nothing
    object
    Exit Sub
    ErrorHandler:
    Unload Me
    Screen.MousePointer = vbDefault
    pointer
    ErrorHandler
    appropriate message)
    Set Actions = Nothing
    object
    End Sub

'Create an new
'Jump to ErrorHandler
'Show hourglass mouse
'Define the
'read Date and time
'Show default mouse
'Disassociate Actions
'Show default mouse
'Handle errors (show
'Disassociate Actions
VERSION 5.00
Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1)#2.0#0"
"MSCOMCTL.OCX"
Begin VB.Form frmFranking
BorderStyle = 3 'Fixed Dialog
Caption = "Franking"
ClientHeight = 4440
ClientLeft = 2760
ClientTop = 3750
ClientWidth = 6975
Icon = "frmFranking.frx":0000
LinkTopic = "Forml"
MaxButton = 0 'False
MinButton = 0 'False
ScaleHeight = 4440
ScaleWidth = 6975
ShowlnTaskbar = 0 'False
StartUpPosition = 1 'CenterOwner
Begin MSComctlLib.StatusBar sStatusBar
Align = 2 'Align Bottom
Height = 255
Left = 0
TabIndex = 19
Top = 4185
Width = 6975
ExtentX = 12303
ExtentY = 450
Style = 1
SimpleText = "Ready"
_Version = 393216
BeginProperty Panels (8E3867A5-8586-11D1-B16A-00C0F0283628)
NumPanels = 1
BeginProperty Paneli (8E3867AB-8586-11D1-B16A-00C0F0283628)
EndProperty
EndProperty
End
Begin VB.CommandButton cmdClearList
Caption = "&Clear list"
Height = 375
Left = 5640
TabIndex = 9
Top = 2280
Width = 1215
End
Begin MSComctlLib.ListView lvFrankings
Height = 1815
Left = 120
TabIndex = 8
TabStop = 0 'False
ToolTipText = "Franckings"
Top = 2280
Width = 5415
ExtentX = 9551
ExtentY = 3201
View = 3
LabelWrap = 0 'False
HideSelection = -1 'True
FullRowSelect = -1 'True
GridLines = -1 'True
_Version = 393217
ForeColor = -2147483640
BackColor = -2147483643
BorderStyle = 1
Appearance = 1
Numltem = 5
BeginProperty ColumnHeader(1) (BDD1F052-858B-11D1-B16A-
Key = "Dummy"
Text = "Dummy"
Object.Width = 0
EndProperty
BeginProperty ColumnHeader (2) [BDD1F052-858B-11D1-B16A-
00C0F0283628]
Alignment = 1
SubItemIndex = 1
Key = "Number"
Text = "#"
Object.Width = 1058
EndProperty
BeginProperty ColumnHeader (3) [BDD1F052-858B-11D1-B16A-
00C0F0283628]
Alignment = 1
SubItemIndex = 2
Key = "Postage"
Text = "Postage"
Object.Width = 1764
EndProperty
BeginProperty ColumnHeader (4) [BDD1F052-858B-11D1-B16A-
00C0F0283628]
Alignment = 1
SubItemIndex = 3
Key = "JobRest"
Text = "Presel. rest"
Object.Width = 1764
EndProperty
BeginProperty ColumnHeader (5) [BDD1F052-858B-11D1-B16A-
00C0F0283628]
SubItemIndex = 4
Key = "Status"
Text = "Status"
Object.Width = 3528
EndProperty
End
Begin VB.CommandButton cmdGoQuiet
Caption = "Go &Quiet"
Height = 375
Left = 5640
TabIndex = 7
Top = 720
Width = 1215
End
Begin VB.Frame fraSettings
Caption = "Settings"
Height = 1815
Left = 120
TabIndex = 11
Top = 120
Width = 5415
Begin VB.TextBox txtPreselectionDpt
Alignment = 1 'Right Justify
Height = 285
Left = 4200
MaxLength = 3
TabIndex = 5
Text = "0"
Top = 1320
Width = 375
End
Begin VB.TextBox txtMargin
Alignment = 1 'Right Justify
Height = 285
Left = 4200
MaxLength = 3
TabIndex = 3
Text = "20"
Top = 840
Width = 375
End
Begin VB.TextBox txtDptAcq
Alignment = 1 'Right Justify
Height = 285
Left = 4200
TabIndex = 1
Text = "9'999'900"
Top = 360
Width = 975
End
Begin VB.TextBox txtPostage
Alignment = 1 'Right Justify
Height = 285
Left = 1560
TabIndex = 0
Text = "9'999'900"
Top = 360
Width = 975
End
Begin VB.ComboBox cmbFrankMode
Height = 315
ItemData = "frmFranking.frx":0442
Left = 1560
List = "frmFranking.frx":0444
Style = 2 'Dropdown List
TabIndex = 4
Top = 1320
Width = 1215
End
Begin VB.ComboBox cmbMailClass
Height = 315
Left = 1560
Style = 2 'Dropdown List
TabIndex = 2
Top = 840
Width = 1215
End
Begin VB.Label lbPreseletionDpt
Alignment = 1 'Right Justify
BackStyle = 0 'Transparent
Caption = "Preselection:
Height = 255
Left = 2880
TabIndex = 17
Top = 1380
Width = 1215
End
Begin VB.Label lbMargin
Alignment = 1 'Right Justify
BackStyle = 0 'Transparent
Caption = "Margin:
Height = 255
Left = 2880
TabIndex = 16
Top = 900
Width = 1215
End
Begin VB.Label lbFrankMode
Alignment = 1 'Right Justify
BackStyle = 0 'Transparent
Caption = "Franking Mode:"
Begin VB.Label lbMailClass
Alignment = 1 'Right Justify
BackStyle = 0 'Transparent
Caption = "Mail Class:"
Height = 255
Left = 240
TabIndex = 15
Top = 1380
Width = 1215
End

Begin VB.Label lbDptAcq
Alignment = 1 'Right Justify
BackStyle = 0 'Transparent
Caption = "Acquisition Tax:"
Height = 255
Left = 240
TabIndex = 14
Top = 900
Width = 1215
End

Begin VB.Label lbPostage
Alignment = 1 'Right Justify
BackStyle = 0 'Transparent
Caption = "Auto Tax:"
Height = 255
Left = 240
TabIndex = 13
Top = 420
Width = 1215
End

End

Begin VB.CommandButton cmdClose
Cancel = -1 'True
Caption = "Close"
Height = 375
Left = 5640
TabIndex = 10
Top = 3720
Width = 1215
End

Begin VB.CommandButton cmdSetDecades
Caption = "&Set Decades"
Default = -1 'True
Height = 375
Left = 5640
TabIndex = 6
Top = 240
Width = 1215
End

Begin VB.Label lbFrankings
BackStyle = 0 'Transparent
Caption = "Frankings:"
Height = 255
Left = 120
TabIndex = 18
Top = 2040
Width = 855
End

End

Attribute VB_Name = "frmFranking"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredID = True
Attribute VB_Exposed = False
'frmFranking provides the GUI for setting the FM ready for Franking.
'In addition it contains a list of the Frankings that have been released.

Option Explicit

'Enable or disable the Preselection field depending on the chosen
'Franking Mode and set the field to a valid value.
Private Sub cmbFrankMode_Click()
    If cmbFrankMode.ItemData(cmbFrankMode.ListIndex) = 0 Then 'If
        Franking Mode = Normal
        txtPreselectionDpt.Enabled = False
    'Disable Preselection field
    txtPreselectionDpt.Text = 0
    Else
        txtPreselectionDpt.Enabled = True
        txtPreselectionDpt.Text = 3
    End If
End Sub

Private Sub cmdClearList_Click()
    lvFrankings.ListItems.Clear 'Clear the Frankings
End Sub

Private Sub cmdClose_Click()
    Unload Me
End Sub

'Set FM to QUIET state (not ready for Franking)
Private Sub cmdGoQuiet_Click()
    Dim Actions As New FMCTRLLib.FMActions 'Create an new
    FMActions.object
    On Error GoTo ErrorHandler: 'Jump to ErrorHandler
    In case of an error
        Screen.MousePointer = vbHourglass 'Show hourglass mouse
        Actions.ActiveConnection = fMainForm.Connection 'Define the
        connection to be used by Actions
        Actions.GoQuiet state (not ready for Franking)
        frmFranking.StatusBar.SimpleText = "Quiet" 'Write to
        StatusBar
        Screen.MousePointer = vbDefault 'Show default mouse
        Set Actions = Nothing 'Disassociate Actions
    Exit Sub
End Sub

ErrorHandler:
    Screen.MousePointer = vbDefault 'Show default mouse
    ErrorHandler 'Handle errors (show
    appropriate message)
    Set Actions = Nothing 'Disassociate Actions
End Sub

'Set the FM ready for Franking
Private Sub cmdSetDecades_Click()
    Dim Actions As New FMCTRLLib.FMActions  
    'Create an new FMActions object
    Dim Config As New FMCTRLLib.FMConfig     
    'Create an new FMConfig object
    Dim Postage As Currency                   
    'Define needed variables
    Dim AcquisitionTax As Currency
    Dim Department As Long                    
    Dim MailClass As Integer                  
    Dim FrankMode As FRANKMODES
    Dim Preselection As Integer
    Dim Margin As Integer                     
    Dim Ret As RETVALS
    Dim msg As String
    On Error GoTo ErrorHandler:               
    'Jump to ErrorHandler in case of an error
    Screen.MousePointer = vbHourglass         
    'Show hourglass mouse pointer
    Actions.ActiveConnection = fMainForm.Con  
    'Define the connection to be used by Actions
    Config.ActiveConnection = fMainForm.Con   
    'Define the connection to be used by Config
    If Config.AutoTaxFM Then                  
        'If FM is a special Auto Tax application (Japan)
        Postage = CCur(txtPostage.Text)       
        'Convert the entered values
        AcquisitionTax = CCur(txtDptAcq.Text)
        Department = CLng(txtPreselectionDpt.Text)
        Margin = CInt(txtMargin.Text)
        Ret = Actions.SetDecadesTax(Postage, AcquisitionTax,
        Department, Margin)  
        'Set Decades for Auto Tax FM
    Else                                     
        Postage = CCur(txtPostage.Text)       
        'Convert the entered values
        Department = CLng(txtDptAcq.Text)
        MailClass = cmbMailClass.ListIndex + 1 
        FrankMode = IndexToFrankMode(cmbFrankMode.ItemData
        (cmbFrankMode.ListIndex))  
        'Convert Franking Mode Preselection = CInt(txtPreselectionDpt.Text)
        Margin = CInt(txtMargin.Text)
        Ret = Actions.SetDecades(Postage, Department, MailClass,
        FrankMode, Preselection, Margin)  
        'Set Decades for Standard FM
    End If

    ProcessRetVal Ret      
    'Process the result of the Set Decades command (show message if necessary)
    Screen.MousePointer = vbDefault         
    'Show default mouse pointer
    Set Actions = Nothing                  
    'Disassociate Actions
    Set Config = Nothing                   
    'Disassociate Config
    Exit Sub                                
    ErrorHandler:                          
        Screen.MousePointer = vbDefault     
        'Show default mouse pointer
        ErrorHandler appropriate message)    
        'Handle errors (show
        Set Actions = Nothing
        'Disassociate Actions
        Set Config = Nothing
        'Disassociate Config
End Sub

'Prepare frmFranking for the application used by the connected FM.
'Some TextBoxes are used for different data entry depending on the
'application (Standard / Auto Tax)

Private Sub Form_Load()
    Dim Config As New FMCTRLLib.FMConfig
    'Create an new FMConfig object
    Dim MCTxts() As String
    'Define the needed variables
    Dim i As Integer
    On Error GoTo ErrorHandler
    'Jump to ErrorHandler in case of an error
    Screen.MousePointer = vbHourglass
    'Show hourglass mouse pointer
    Config.ActiveConnection = fMainForm.Connection
    'Define the connection to be used by Config

    txtPostage.Text = "0"
    'Initialize Postage/AutoTax field
    txtDptAcq.Text = "0"
    'Initialize Department/ AcquisitionTax field
    If Config.AutoTaxFM = False Then
        'If it is a Standard application
        lbPostage.Caption = "Postage:"
        'Label the fields according to usage
        lbDptAcq.Caption = "Department;"
        lbPreselectionDpt.Caption = "Preselection;"
        
        If Config.SettableMailClasses > 0 Then
            'If any Mail Classes are available
            MCTxts = Config.MailClassTexts
            'Read Mail Class text from FM
            cmbMailClass.Clear
            'Clear the ComboBox
            For i = LBound(MCTxts) To UBound(MCTxts)
                'Fill Mail Class texts into the ComboBox
                cmbMailClass.AddItem MCTxts(i), i
            Next i
            cmbMailClass.ListIndex = 0
            'Initialize Mail Class ComboBox (choose 1st entry)
            Else
                'No MailClasses available
                cmbMailClass.Enabled = False
                'Disable Mail Class ComboBox
            End If

            cmbFrankMode.Clear
            'Clear Mail Class ComboBox
            If Config.FrankModeAvailNorm Then
                'If Franking Mode Normal is available
                cmbFrankMode.AddItem "Normal"
                'Insert entry into ComboBox
                cmbFrankMode.ItemData(cmbFrankMode.NewIndex) = 0
                'Add item data to be able to identify the chosen entry
                End If
            If Config.FrankModeAvailTape Then
                'Same as above for Franking Mode Tape
                cmbFrankMode.AddItem "Tapes"
                cmbFrankMode.ItemData(cmbFrankMode.NewIndex) = 1
                End If
            If Config.FrankModeAvailLetter Then
                'Same as above for Franking Mode Letter
                cmbFrankMode.AddItem "Letters"
                cmbFrankMode.ItemData(cmbFrankMode.NewIndex) = 2
                End If
    End Else
End Sub
End If
If Config.FrankModeAvailItem Then
    'Same as above for Franking Mode Item
    cmbFrankMode.AddItem "Items"
    cmbFrankMode.ItemData(cmbFrankMode.NewIndex) = 3
End If
cmbFrankMode.ListIndex = 0
'Initialize Mail Class ComboBox (choose 1st entry)
cmbMailClass.Enabled = cmbFrankMode.ListCount > 1
'Enable Franking Mode ComboBox if there is more than one entry
txtPreselListBox.MaxLength = 3
'Limit the text length of the Preselection field to 3 (max = "999")
Else ' (AutoTaxFM)
    'Auto Tax application
    lbPostage.Caption = "Auto Tax:"
    'Label the fields according to usage
    lbDptAcq.Caption = "Acquisition Tax:"
    lbPreselListBox.Caption = "Department:"
    txtPreselListBox.Enabled = True
    'Enable the Department field
    txtPreselListBox.Width = 975
    'Set width of Department field (more space necessary than for Preselection)
    txtPreselListBox.MaxLength = 0
    'No text length limit
    txtDptAcq.Enabled = Config.AcquisitionTaxAvailable
    'Enable Acquisition Tax field if application supports Acquisition Tax
    cmbMailClass.Enabled = False
    'Disable Mail Class field (Auto Tax does not support Mail Classes)
    cmbFrankMode.Enabled = False
    'Disable Franking Mode field (Auto Tax does not support Franking Modes)
End If ' (AutoTaxFM)
If Config.MarginAvailable Then
    txtMargin.Enabled = True
    txtMargin.Text = "20"
Else
    txtMargin.Enabled = False
    txtMargin.Text = "0"
End If
Screen.MousePointer = vbDefault
'Show default mouse pointer
Set Config = Nothing
'Disassociate Config object
Exit Sub
ErrorHandler:
Unload Me
Screen.MousePointer = vbDefault
'Set valid value
ErrorReason
appropriate message
'Show default mouse pointer
Set Config = Nothing
'Disassociate Config object
End Sub
Private Sub Unload(Cancel As Integer)
    If Len(fMainForm.Con.ConnectionString) <> 0 Then
        'If connection exists
        cmdGoQuiet_Click
        QUIET state (not ready for Franking)
    End If
End Sub
Private Sub txtDptAcq_GotFocus()
    SelectAll txtDptAcq
    'Select the whole
text in the TextBox
End Sub

Private Sub txtMargin_GotFocus()
    SelectAll txtMargin
text in the TextBox
End Sub

Private Sub txtPostage_GotFocus()
    SelectAll txtPostage
text in the TextBox
End Sub

Private Sub txtPreselectionDpt_GotFocus()
    SelectAll txtPreselectionDpt
text in the TextBox
End Sub

'Select the whole
VERSION 5.00
Begin VB.Form frmInterface
  BorderStyle = 3 "Fixed Dialog"
  Caption = "Interface"
  ClientHeight = 1320
  ClientLeft = 2760
  ClientTop = 3750
  ClientWidth = 2775
  Icon = "frmInterface.frx":0000
  LinkTopic = "Form1"
  MaxButton = 0 'False
  MinButton = 0 'False
  ScaleHeight = 1320
  ScaleWidth = 2775
  ShowInTaskbar = 0 'False
  StartUpPosition = 1 'CenterOwner
Begin VB.TextBox txtComPort
  Alignment = 1 'Right Justify
  Height = 285
  Left = 1680
  MaxLength = 2
  TabIndex = 0
  Text = "1"
  Top = 240
  Width = 375
End
Begin VB.CommandButton cmdCancel
  Cancel = -1 'True
  Caption = "Cancel"
  Height = 375
  Left = 120
  TabIndex = 2
  Top = 840
  Width = 1215
End
Begin VB.CommandButton cmdOK
  Caption = "OK"
  Default = -1 'True
  Height = 375
  Left = 1440
  TabIndex = 1
  Top = 840
  Width = 1215
End
Begin VB.Label lbCOMPort
  Caption = "COM Port ":
  Height = 255
  Left = 720
  TabIndex = 3
  Top = 300
  Width = 975
End
Attribute VB_Name = "frmInterface"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'frmInterface allows changing the COM Port that is used to 'connect to the FM.

Option Explicit

Private Sub cmdCancel_Click()
  Unload Me
End Sub
Private Sub cmdOK_Click()
On Error GoTo ErrorHandler:
in case of an error
    nComPort = CInt(txtComPort.Text)
Global variable
Unload Me
frmInterface
Exit Sub
ErrorHandler:
    ErrorHandler
    'Handle errors (show
appropriate message - here usually conversion error)
    txtComPort.SetFocus
End Sub

Private Sub Form_Load()
    txtComPort.Text = CStr(nComPort)
    'Write current COM
Port # into TextBox
End Sub

Private Sub txtComPort_GotFocus()
    SelectAll txtComPort
text in the TextBox
End Sub
VERSION 5.00

Begin VB.Form frmMailClassAdjust
  BorderStyle = 3 'Fixed Dialog
  Caption = "Mail Class adjust"
  ClientHeight = 1320
  ClientLeft = 2760
  ClientTop = 3750
  ClientWidth = 3240
  Icon = "frmMailClassAdjust.frx":0000
  LinkTopic = "Form1"
  MaxButton = 0 'False
  MinButton = 0 'False
  ScaleHeight = 1320
  ScaleWidth = 3240
  ShowInTaskbar = 0 'False
  StartUpPosition = 1 'CenterOwner
End VB.Form

Begin VB.ComboBox cmbMailClass
  Height = 315
  ItemData = "frmMailClassAdjust.frx":0442
  Left = 1680
  List = "frmMailClassAdjust.frx":0455
  Style = 2 'Dropdown List
  TabIndex = 0
  ToolTipText = "Printed Mail Class"
  Top = 240
  Width = 1455
End VB.ComboBox

Begin VB.CommandButton CancelButton
  Cancel = -1 'True
  Caption = "Close"
  Height = 375
  Left = 1920
  TabIndex = 3
  Top = 840
  Width = 1215
End VB.CommandButton

Begin VB.CommandButton OKButton
  Caption = "&Adjust"
  Default = -1 'True
  Height = 375
  Left = 600
  TabIndex = 1
  Top = 840
  Width = 1215
End VB.CommandButton

Begin VB.Label lbPrintedMC
  Alignment = 1 'Right Justify
  BackStyle = 0 'Transparent
  Caption = "Printed Mail Class:"
  Height = 255
  Left = 240
  TabIndex = 2
  Top = 300
  Width = 1335
End VB.Label

Attribute VB_Name = "frmMailClassAdjust"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False

' frmMailClassAdjust allows correcting the Mail Class assignment to
' the Mail Class Cylinder position.
' In rare failure cases the assignment can get wrong. With this
' function the assignment can be adjusted.
Option Explicit

Private Sub CancelButton_Click()
    Unload Me
End Sub

'Executes Mail Class adjusting

Private Sub OKButton_Click()
    Dim Actions As New FMCTRLLib.FMActions 'Create an new
    FMConfig object
    Dim MCTxts() As String 'Define needed
    variables
    Dim nMCYlPos As Integer
    Dim nEmptyPos As Integer
    Dim i As Integer
    On Error GoTo ErrorHandler 'Jump to ErrorHandler
    in case of an error
    Screen.MousePointer = vbHourglass 'Show hourglass mouse
    pointer
    Actions.ActiveConnection = fMainForm.Con 'Define the
    connection to be used by Config
    If cmbMailClass.ListIndex < 0 Then 'If no Mail Class is
        chosen
        MsgBox "Please choose a Mail Class.", vbExclamation 'Show
        message
    Else
        Actions.MailClassAdjust cmbMailClass.ListIndex + 1 'Adjust Mail Class on FM
        Exit Sub
    End If
    Screen.MousePointer = vbDefault 'Show default mouse
    Set Actions = Nothing 'Disassociate Config
    object
    Exit Sub
ErrorHandler:
    Unload Me
    Screen.MousePointer = vbDefault 'Show default mouse
    pointer
    ErrorHandler 'Handle errors (show
    appropriate message)
    Set Actions = Nothing 'Disassociate Config
    object
End Sub
VERSION 5.00
Object = "{83FDD16-0C5C-11D2-A9FC-0000F8754DA1}"#2.0#0"
"Mxcomcl1.ocx"
Begin VB.Form frmMain
BackColor = &H80000005&
BorderStyle = 1 'Fixed Single
Caption = "FM Control User Application Demo"
ClientHeight = 4140
ClientLeft = 1410
ClientTop = 1410
ClientWidth = 7110
Icon = "frmMain.frx":0000
LinkTopic = "Form1"
MaxButton = 0 'False
ScaleHeight = 4140
ScaleWidth = 7110
StartUpPosition = 2 'CenterScreen
Begin MSComctlLib.Toolbar tbToolbar
Align = 1 'Align Top
Height = 360
Left = 0
TabIndex = 1
Top = 0
Width = 7110
_ExtentX = 12541
_ExtentY = 635
ButtonWidth = 609
ButtonHeight = 582
Appearance = 1
Style = 1
ImageList = "imlToolbarIcons"
_Version = 393216
BeginProperty Buttons {66833FE8-8583-11D1-B16A-00C0F0283628}
NumButtons = 15
BeginProperty Button1 {66833FE8-8583-11D1-B16A-00C0F0283628}
Key = "Interface"
ObjectName = "Interface (Ctrl+I)"
KeyName = "Interface"
EndProperty
BeginProperty Button2 {66833FE8-8583-11D1-B16A-00C0F0283628}
Style = 3
EndProperty
BeginProperty Button3 {66833FE8-8583-11D1-B16A-00C0F0283628}
Key = "Connect"
ObjectName = "Connect (F3)"
KeyName = "Connect"
EndProperty
BeginProperty Button4 {66833FE8-8583-11D1-B16A-00C0F0283628}
Key = "Disconnect"
ObjectName = "Disconnect (F4)"
KeyName = "Disconnect"
EndProperty
BeginProperty Button5 {66833FE8-8583-11D1-B16A-00C0F0283628}
Style = 3
EndProperty
BeginProperty Button6 {66833FE8-8583-11D1-B16A-00C0F0283628}
Key = "Properties"
ObjectName = "Properties (Ctrl+P)"
KeyName = "Properties"
EndProperty
BeginProperty Button7 {66833FE8-8583-11D1-B16A-00C0F0283628}
Key = "Settings"
ObjectName = "Settings (Ctrl+S)"
KeyName = "Settings"
EndProperty
BeginProperty Button8 {66833FE8-8583-11D1-B16A-00C0F0283628}
Style = 3
EndProperty
BeginProperty Button9 {66833FEA-8583-11D1-B16A-00C0F0283628}
  Key = "Franking"
  Object.ToolTipText = "Franking (F5)"
  ImageKey = "Franking"
EndProperty
BeginProperty Button10 {66833FEA-8583-11D1-B16A-00C0F0283628}
  Key = "Counters"
  Object.ToolTipText = "Counters (F6)"
  ImageKey = "Counters"
EndProperty
BeginProperty Button11 {66833FEA-8583-11D1-B16A-00C0F0283628}
  Key = "DateTime"
  Object.ToolTipText = "Date & Time (F7)"
  ImageKey = "DateTime"
EndProperty
BeginProperty Button12 {66833FEA-8583-11D1-B16A-00C0F0283628}
  Key = "MailClass"
  Object.ToolTipText = "Mail Class adjust (F8)"
  ImageKey = "MailClass"
EndProperty
BeginProperty Button13 {66833FEA-8583-11D1-B16A-00C0F0283628}
  Key = "TextToDisplay"
  Object.ToolTipText = "Text to display (F9)"
  ImageKey = "TextToDisplay"
EndProperty
BeginProperty Button14 {66833FEA-8583-11D1-B16A-00C0F0283628}
  Style = 3
EndProperty
BeginProperty Button15 {66833FEA-8583-11D1-B16A-00C0F0283628}
  Key = "About"
  Object.ToolTipText = "About"
  ImageKey = "Help"
EndProperty
EndProperty
MouseIcon = "frmMain.frx":0442
End
Begin MSComctlLib.StatusBar sbStatusBar
  Align = 2 'Align Bottom
  Height = 255
  Left = 0
  TabIndex = 0
  Top = 3885
  Width = 7110
  _ExtentX = 12541
  _ExtentY = 450
  Style = 1
  SimpleText = "Ready"
  Version = 393216
EndProperty
BeginProperty Panels {8E3867A5-8586-11D1-B16A-00C0F0283628}
  NumPanels = 3
  BeginProperty Panel1 {8E3867AB-8586-11D1-B16A-00C0F0283628}
    AutoSize = 1
    Object.Width = 7355
    Text = "Status"
    TextSave = "Status"
  EndProperty
  BeginProperty Panel2 {8E3867AB-8586-11D1-B16A-00C0F0283628}
    Style = 6
Connection (Connection)

Properties

ConnectionString – Connection string

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSTR</td>
<td>String</td>
<td>Read-Only</td>
<td>valid strings</td>
</tr>
</tbody>
</table>

Returns the ConnectionString for the current connection. All characters of this string are always upper case. If the string is not empty (length ≠ 0) the connection does exist. On disconnection the string is reset to 0 length.

LibraryVersion – Version number

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSTR</td>
<td>String</td>
<td>Read-Only</td>
<td>#.#.#.# where # is a number 0...9999</td>
</tr>
</tbody>
</table>

Returns the version of the program library that is used.
Methods

Connect – Establish connection to FM
Connect (ConnectionString)

<table>
<thead>
<tr>
<th>ConnectionString</th>
<th>BSTR</th>
<th>String</th>
<th>In valid strings</th>
<th>Defines connection device, protocol etc.</th>
</tr>
</thead>
</table>

Establishes connection to the FM and loads all properties.

ConnectionString contains the necessary information to establish the connection. It consists of the connection device definition and the protocol name. This version of the library supports serial communication via the COM-Ports using the MLPV6 protocol only. So the only valid syntax looks as follows: COMPORT=#; PROTOCOL=MLPV6 while # represents the number of the COM-Port. The fields have to be separated by a semicolon (;). The MLPV6 protocol is set as default so that the PROTOCOL field can be omitted. Examples for valid ConnectionStrings (ConnectionString is NOT case sensitive):

- COMPORT=1; PROTOCOL=MLPV6
- COMPORT=4
- COMPORT=12; PORTOCOL=MLPV6

Disconnect – Disconnect from FM
Disconnect()

After disconnection all properties are invalid.
Events

OnDieCoverClose – Die cover closed

Triggered when the FM’s Die cover is closed. The FM always goes to QUIET state (not ready for Franking) when the Die cover is opened and can not be set ready for Franking until the Die cover is closed.

OnDieCoverOpen – Die cover opened

Triggered when the FM’s Die cover is opened. The FM always goes to QUIET state (not ready for Franking) when the Die cover is opened and can not be set ready for Franking until the Die cover is closed.

OnDisconnect – Unexpected disconnection

Triggered when the connection between PC and FM is unexpectedly broken.

OnFranking – A Franking is released

<table>
<thead>
<tr>
<th>Postage</th>
<th>JobRest</th>
<th>StatusCode</th>
<th>Currency</th>
<th>In</th>
<th>any</th>
<th>any</th>
<th>any</th>
<th>any</th>
</tr>
</thead>
<tbody>
<tr>
<td>Postage value Franked</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remaining number of items in this Preselection session. -1 if no Preselection was chosen on SetDecades(Tax).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Returns information about the state of the machine.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This event is triggered every time a Franking is released on the FM. This may also happen without the previous call of SetDecades, as the user may perform this action via the FM keyboard. For Autotax applications the postage represents the sum of Autotax and Acquisition tax. See also SetDecades – Set FM ready for Franking on page 16, SetDecadesTax – Set Autotax FM ready for Franking (for Autotax applications only) on page 17 and RETVALS on page 26.

OnNoMoreTapes – No more Tapes

Triggered when in Franking Mode FRK_TAPE and there are no more Tapes in the FM. Franking can be continued by calling TapesPresent. See also TapesPresent – Tapes present again on page 17.
**OnQuiet – FM goes to QUIET state (not ready for Franking)**

**OnQuiet(StatusCode)**

<table>
<thead>
<tr>
<th>StatusCode</th>
<th>RETVALS</th>
<th>RRTVALS</th>
<th>In</th>
<th>Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>R_OK, E_HSB_KEY, E_HSB_HOT, E_HSB_STANDBY, E_HSB_INKCOVER</td>
<td>Returns information about the state of the machine.</td>
</tr>
</tbody>
</table>

Triggered when the FM goes to QUIET state without a corresponding command (GoQuiet) from the PC.

**OnRotorError – Rotor error**

**OnRotorError(StatusCode)**

<table>
<thead>
<tr>
<th>StatusCode</th>
<th>RETVALS</th>
<th>RRTVALS</th>
<th>In</th>
<th>Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>E_CALL_SERVICE, E_ROTOR, E_CROSS, E_BASE_SPEED, E_DECade, E_PRINTER</td>
<td>Returns information about the state of the machine.</td>
</tr>
</tbody>
</table>

Triggered when the FM has a problem that does not allow further Frankings. This may be a problem setting decade wheels or the rotor position etc. See also *RETVALS* on page 26.
OnTMSEnd – End of TMS procedure

<table>
<thead>
<tr>
<th>OnTMSEnd(Descending, Message, StatusCode)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Descending</strong></td>
</tr>
<tr>
<td><strong>Message</strong></td>
</tr>
<tr>
<td><strong>StatusCode</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Informs that the TMS procedure has finished either successful carried out or with an error. In the case of an error the message received from the TMS center may contain additional information about the cause of the error. If one calls the function to start the TMS process on an FM that does not support TMS, this event will be triggered with StatusCode = E_TMS_FAILED.

The TMS procedure can be started by calling the function TMSSStart. See also **TMSSStart – Start TMS reset** on page 18. See also **RETVALS** on page 26.
Configuration (FMConfig)

Important: Before any of the properties or methods of FMConfig can be used the property ActiveConnection must be set.

Properties

The following applies to all properties except for ActiveConnection:

All properties are READ-ONLY. The properties are loaded from the FM when the connection is established. None of the properties does ever change, so there is no need to reload them. On disconnection all properties become invalid.

ActiveConnection – Relation to Connection

| IConnection* | Connection | Read-Write | Any |

Has to be set before the first use of any property or method of FMConfig. It defines the Connection that FMConfig uses to perform the functions in the FM.

AcquisitionTaxAvailable – Availability of Acquisitiontax

| VARIANT_BOOL | AcquisitionTaxAvailable |

Determines whether the FM features Acquisitiontax. Acquisitiontax is possible on Autotax FMs only. See also SetDecadesTax – Set Autotax FM ready for Franking (for Autotax applications only) on page 17.

AutoDate – Presence of Automatic Date

| VARIANT_BOOL | AutoDate |

Determines whether the FM has automatic date wheels.
AutoTaxFM – Autotax / Standard FM

VARIANT BOOL AutoDate

<table>
<thead>
<tr>
<th>Boolean</th>
<th>Read-Only</th>
<th>TRUE/FALSE</th>
</tr>
</thead>
</table>

TRUE if FM is a special Autotax FM. FALSE if FM is a standard (non Autotax) FM. This property allows deciding whether SetDecadesTax or SetDecades must be used to set the FM ready for Franking. The following table shows which function must be used for which FM application:

<table>
<thead>
<tr>
<th>FM</th>
<th>Function to set ready for Franking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autotax</td>
<td>SetDecadesTax</td>
</tr>
<tr>
<td>Standard</td>
<td>SetDecades</td>
</tr>
</tbody>
</table>

BaseModel – Base model

BASSES BaseModel

<table>
<thead>
<tr>
<th>BASSES</th>
<th>BASSES</th>
<th>Read-Only</th>
<th>All of BASSES</th>
</tr>
</thead>
</table>

See also BASSES on page 25.

BaseSWVer – Base software version

BSTR BaseSWVer

<table>
<thead>
<tr>
<th>BSTR</th>
<th>String</th>
<th>Read-Only</th>
<th>Any text</th>
</tr>
</thead>
</table>

DecadeNumber – Number of decade wheels

short DecadeNumber

<table>
<thead>
<tr>
<th>short</th>
<th>Integer</th>
<th>Read-Only</th>
<th>4 or 5</th>
</tr>
</thead>
</table>

See also Examples for print image configuration on page 9.
DecPointPosition – Position of decimal point

<table>
<thead>
<tr>
<th>DecPointPosition</th>
</tr>
</thead>
<tbody>
<tr>
<td>short DecPointPosition</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>value</th>
<th>access</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>short</td>
<td>Integer</td>
<td>Read-Only</td>
<td>0..5</td>
</tr>
</tbody>
</table>

Position of decimal point in money values, right aligned. This property is necessary to format money values correctly to match FM's configuration. It may be used to generate an entry mask on the user interface. FixedZeros do also count as a position. See also Examples for print image configuration on page 9.

FixedZeros – Number of fixed zeros

<table>
<thead>
<tr>
<th>FixedZeros</th>
</tr>
</thead>
<tbody>
<tr>
<td>short FixedZeros</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>value</th>
<th>access</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>short</td>
<td>Integer</td>
<td>Read-Only</td>
<td>0..3</td>
</tr>
</tbody>
</table>

Number of fixed zeros after the decade wheels. This property is necessary to format money values correctly to match FM's configuration. It may be used to generate an entry mask on the user interface. See also Examples for print image configuration on page 9.

FMSWVer – FM software version

<table>
<thead>
<tr>
<th>FMSWVer</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSTR FMSWVer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>value</th>
<th>access</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSTR</td>
<td>String</td>
<td>Read-Only</td>
<td>Any</td>
</tr>
</tbody>
</table>

FMType – FM type

<table>
<thead>
<tr>
<th>FMTYPES FMType</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>value</th>
<th>access</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>FMTYPES</td>
<td>FMTYPES</td>
<td>Read-Only</td>
<td>All of FMTYPES</td>
</tr>
</tbody>
</table>

Determinates what type of FM is connected to the PC. See also FMTYPES on page 25.

FrankModeAvailItem – Availability of Franking Mode FRK_ITEMS

<table>
<thead>
<tr>
<th>FrankModeAvailItem</th>
</tr>
</thead>
</table>

| VARIANT BOOL FrankModeAvailItem |

<table>
<thead>
<tr>
<th>type</th>
<th>value</th>
<th>access</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>VARIANT BOOL</td>
<td>Boolean</td>
<td>Read-Only</td>
<td>TRUE/FALSE</td>
</tr>
</tbody>
</table>

Determinates whether the Item Franking Mode (FRK_ITEM) is available. One of the available Franking Modes has to be given as parameter when calling the SetDecades function. The available Franking Modes depend on the Base model. For more details on Franking Modes see SetDecades – Set FM ready for Franking on page 16.
FrankModeAvailLetter – Availability of Franking Mode FRK_LETTER

Determinates whether the Letter Franking Mode (FRK_LETTER) is available. One of the available Franking Modes has to be given as parameter when calling the SetDecades function. The available Franking Modes depend on the Base model. For more details on Franking Modes see SetDecades – Set FM ready for Franking on page 16.

FrankModeAvailNorm – Availability of Franking Mode FRK_NORM

Determinates whether the Normal Franking Mode (FRK_NORM) is available. One of the available Franking Modes has to be given as parameter when calling the SetDecades function. The available Franking Modes depend on the Base model. For more details on Franking Modes see SetDecades – Set FM ready for Franking on page 16.

FrankModeAvailTape – Availability of Franking Mode FRK_TAPE

Determinates whether the Tape Franking Mode (FRK_TAPE) is available. One of the available Franking Modes has to be given as parameter when calling the SetDecades function. The available Franking Modes depend on the Base model. For more details on Franking Modes see SetDecades – Set FM ready for Franking on page 16.

LastDecadeType – Type of the last decade wheel

Determinates whether the last decade wheel can have all values (0...9 = LD09) or only the values 0 and 5 (= LD05). This property is necessary to format money values correctly to match FM's configuration. It may be used to generate an entry mask on the user interface. See Examples for print image configuration on page 11 and LASTDECTYPE on page 25.
MailClassCylinderPositions – Number of positions on Mail Class cylinder

<table>
<thead>
<tr>
<th>short</th>
<th>Integer</th>
<th>Read-Only</th>
<th>Value Range</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>0, 4, 5</td>
</tr>
</tbody>
</table>

Often not all positions on the Mail Class cylinder are in use. So there are less Mail Classes available than positions on the cylinder. This value is important for MailClassAdjust as it is possible that the printed Mail Class is one of the unused positions on the cylinder. See also SettableMailClasses – Number of available Mail Classes on page 9 and MailClassAdjust – Adjust Mail Class assignment on page 15.

MailClassTexts – Texts assigned to Mail Classes

| SAFEARRAY(BSTR) | String() | Read-Only | One-dimensional array of up to 5 texts of max. 10 chars |

Each settable Mail Class has a text assigned that is shown on the FM display when a Mail Class is chosen using the FMs keyboard. These texts usually correspond to the Mail Class Dies. Each text is max. 10 chars long. There can be up to 5 texts as there are up to 5 positions on Mail Class cylinder. Possibly not all cylinder positions are available for printing. So this array contains as many texts as there are settable Mail Classes. If there are zero settable Mail Classes the array is empty (no elements). To set the Mail Class for Franking, it is necessary to pass the index of the Mail Class. The index starts from 1 to the number of settable Mail Classes. See also SettableMailClasses – Number of available Mail Classes on page 9.

MarginAvailable – Availability of margin

| VARIANT_BOOL MarginAvailable |

The margin indicates the distance between the edge of the letter and the beginning of the print. Margin can be set (is available) on FMs optical release only.

SerialNr – Serial number of FM

| BSTR SerialNr |

Serial number that identifies the FM.
SettableMailClasses – Number of available Mail Classes

<table>
<thead>
<tr>
<th>short SettableMailClasses</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>short</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Number of Mail Classes that are available. FMs that do not have Mail Classes return 0 (zero).

Examples for print image configuration

The labels of column 2 through 5 are the names of the properties, which specify the print image configuration.

<table>
<thead>
<tr>
<th>Mail</th>
<th>Position</th>
<th>DateRoomPast</th>
<th>MaskZero</th>
<th>TrashDecayed</th>
<th>Suppress</th>
</tr>
</thead>
<tbody>
<tr>
<td>99.999</td>
<td>5</td>
<td>4</td>
<td>0</td>
<td>LD09</td>
<td></td>
</tr>
<tr>
<td>999.90</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>LD09</td>
<td></td>
</tr>
<tr>
<td>999.95</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>LD05</td>
<td></td>
</tr>
<tr>
<td>999.99</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>LD09</td>
<td></td>
</tr>
<tr>
<td>9999.99</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>LD09</td>
<td></td>
</tr>
<tr>
<td>9995</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>LD05</td>
<td></td>
</tr>
<tr>
<td>9999</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>LD09</td>
<td></td>
</tr>
<tr>
<td>9999.900</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>LD09</td>
<td></td>
</tr>
<tr>
<td>99990</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>LD09</td>
<td></td>
</tr>
<tr>
<td>99999</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>LD09</td>
<td></td>
</tr>
<tr>
<td>999500</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>LD05</td>
<td></td>
</tr>
<tr>
<td>9999900</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>LD09</td>
<td></td>
</tr>
</tbody>
</table>

Meaning of column 'Mask':

- 9 Can be anything 0..9
- 5 Can be 0 or 5
- 0 Fixed Zero
Methods

DescRegHide – Hide (do not show) Descending Register
DescRegHide()
The Descending Register contains the amount of money that is currently stored inside the FM. This value can be shown on the FM display. This function hides the Descending Register on FM display.

DescRegShow – Show Descending Register
DescRegShow()
The Descending Register contains the amount of money that is currently stored inside the FM. This function shows the Descending Register on FM display.

FrankMenuLock – Lock the Franking Menu
FrankMenuLock()
Locks the Franking Menu so that Decade setting and Franking can not be done from FM keyboard anymore. All other functions stay available from FM keyboard.

FrankMenuUnlock – Unlock the Franking Menu
FrankMenuUnlock()
Unlocks the Franking Menu so that Decade setting and Franking are available directly from FM keyboard again.

HVLimitActivate – Activate High Value limit
HVLimitActivate()
Activates High Value limit so that Frankings with a higher value than specified can only be carried out after confirmation.

HVLimitDeactivate – Deactivate High Value limit
HVLimitDeactivate()
Deactivates High Value limit so that all Frankings can be carried out without confirmation.

KeyboardLock – Lock the FM keyboard
KeyboardLock()
Locks the FM keyboard so that it is accessible via the PC interface only.
**KeyboardUnlock – Unlock the FM keyboard**

KeyboardUnlock()

Unlocks the FM keyboard so that it is accessible via the PC interface and from the FM keyboard at the same time.

**UserTimeoutActivate – Activate User Timeout**

UserTimeoutActivate()

Activates the User Timeout. After a specified time of inactivity the FM goes automatically into sleep mode (turns off Base motor and display and goes into QUIET state). The duration of the User Timeout is defined in the FM software and can be changed by the service technician.

**UserTimeoutDeactivate – Deactivate User Timeout**

UserTimeoutDeactivate()

Deactivates the User Timeout. The FM never goes automatically into sleep mode (turns off Base motor and display and goes into QUIET state).

**WarningLowCreditActivate – Activate Low Credit warning**

UserTimeoutActivate()

Activates the Low Credit warning. If the Descending register drops below a specified value the user receives a warning. The warning level is defined in the FM software and can be changed by the service technician.

**WarningLowCreditDeactivate – Deactivate Low Credit warning**

UserTimeoutDeactivate()

Deactivates the Low Credit warning. No warning is issued concerning the Descending register.
Actions (FMActions)

Important: Before any of the methods of FMActions can be used the property ActiveConnection must be set.

Properties

ActiveConnection – Relation to Connection

<table>
<thead>
<tr>
<th>ActiveConnection</th>
<th>Connection</th>
<th>Read-Write</th>
<th>Any</th>
</tr>
</thead>
<tbody>
<tr>
<td>IConnection</td>
<td>Connection</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Has to be set before the first use of any method of FMActions. It defines the Connection that FMActions uses to perform the functions in the FM.
Methods

BatchCounterActivate – Activates batch counter

BatchCounterActivate()

Activates the batch counter. The batch counter counts all non zero postage Frankings while it is active. It can only be stopped by clearing (see below).

BatchCounterClear – Read, clear and deactivate batch counter

BatchCounterClear(On, Items, Value)

<table>
<thead>
<tr>
<th>On</th>
<th>VARIANT_BOOL</th>
<th>Boolean</th>
<th>Out</th>
<th>TRUE/FALSE</th>
<th>TRUE if batch counter is (was) active</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items</td>
<td>long</td>
<td>Long</td>
<td>Out</td>
<td>Any</td>
<td>Number of items franked since batch counter was activated</td>
</tr>
<tr>
<td>Value</td>
<td>CURRENCY</td>
<td>Currency</td>
<td>Out</td>
<td>Any</td>
<td>Total value franked since batch counter was activated</td>
</tr>
</tbody>
</table>

Reads, clears and deactivates batch counter. The items and the money value of the batch counter, are returned as well as the information whether the batch counter was active before this method was called. The batch counter is deactivated and its values are reset to zero respective FALSE.

BatchCounterRead – Read batch counter

BatchCounterRead(On, Items, Value)

<table>
<thead>
<tr>
<th>On</th>
<th>VARIANT_BOOL</th>
<th>Boolean</th>
<th>Out</th>
<th>TRUE/FALSE</th>
<th>TRUE if batch counter is active</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items</td>
<td>long</td>
<td>Long</td>
<td>Out</td>
<td>Any</td>
<td>Number of items franked since batch counter was activated</td>
</tr>
<tr>
<td>Value</td>
<td>CURRENCY</td>
<td>Currency</td>
<td>Out</td>
<td>Any</td>
<td>Total value franked since batch counter was activated</td>
</tr>
</tbody>
</table>

Items and the money value of the batch counter, are returned as well as the information whether the batch counter is active.

GetCounterValues – Read postal counters

GetCounterValues(Ascending, Descending, Items)

<table>
<thead>
<tr>
<th>Ascending</th>
<th>CURRENCY</th>
<th>Currency</th>
<th>Out</th>
<th>Any</th>
<th>Value of the Ascending counter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Descending</td>
<td>CURRENCY</td>
<td>Currency</td>
<td>Out</td>
<td>Any</td>
<td>Value of the Descending counter</td>
</tr>
<tr>
<td>Items</td>
<td>long</td>
<td>Long</td>
<td>Out</td>
<td>Any</td>
<td>Value of the Item counter</td>
</tr>
</tbody>
</table>

The current values of the postal counters are returned.
Appendix 1 - Page 16

GetTimeDate – Current time and date of the FM

<table>
<thead>
<tr>
<th>DATE</th>
<th>GetTimeDate</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>DATE</th>
<th>Date</th>
<th>Ret</th>
<th>Any date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Current FM system time and date.</td>
</tr>
</tbody>
</table>

The current system time and date (local) of the FM.

GoQuiet – Set FM to QUIET state (not ready for Franking)

GoQuiet

Sets the FM to QUIET state. In QUIET state the machine is not ready for Franking and the decade wheels are not set.

HVLimitAbort – Not accept postage above High Value limit

<table>
<thead>
<tr>
<th>RETVALS</th>
<th>HVLimitAbort()</th>
</tr>
</thead>
</table>

| RetVal | RETVALS | RETVALS | Ret | R_OK, E_TIMEOUT | Returns information about the state of the machine. |

A SetDecades method can return with a RetVal = R_HVLIM saying that the given postage is higher than the High Value limit. This means that the decades are not set yet and the FM is not ready for Franking yet. To carry out the SetDecades command a confirmation is necessary. With HVLimitAbort the SetDecades command is aborted, meaning the high postage is not accepted.

One of the methods HVLimitAbort or HVLimitDeblock has to be called before the timeout (for most FMs around 8 sec.) has run down. The timeout starts to run when SetDecades returns R_HVLIM. If the HVLimitAbort is called too late, E_TIMEOUT is returned and the FM goes to QUIET state. See also SetDecades – Set FM ready for Franking on page 16 and RETVALS on page 25.

HVLimitDeblock – Accept postage above High Value limit

<table>
<thead>
<tr>
<th>RETVALS</th>
<th>HVLimitDeblock()</th>
</tr>
</thead>
</table>

| RetVal | RETVALS | RETVALS | Ret | R_OK, E_TIMEOUT | Returns information about the state of the machine. |

A SetDecades method can return with a RetVal = R_HVLIM saying that the given postage is higher than the High Value Limit. This means that the decades are not set yet and the FM is not ready for Franking yet. To carry out the SetDecades command a confirmation is necessary. With HVLimitDeblock the SetDecades command is carried out, meaning that the high postage is accepted.

One of the methods HVLimitAbort or HVLimitDeblock has to be called before the timeout (for most FMs around 8 sec.) has run down. The timeout starts to run when SetDecades returns R_HVLIM. If the HVLimitDeblock is called too late, E_TIMEOUT is returned and the FM goes to QUIET state. See also SetDecades – Set FM ready for Franking on page 16 and RETVALS on page 25.
**MailClassAdjust – Adjust Mail Class assignment**

```
<table>
<thead>
<tr>
<th>PrintedMailClass</th>
<th>short</th>
<th>Integer</th>
<th>In 1_MailClassCylinderPositions</th>
<th>Index of the printed Mail Class</th>
</tr>
</thead>
</table>
```

Under very rare circumstances it can happen that the Mail Class assignment between the Mail Class texts and the positions of the Mail Class cylinder are shifted. Because of that, the wrong Mail Class is stamped. With this function it is possible to reassign the Mail Classes. The parameter PrintedMailClass has to be the index of the wrongly printed Mail Class. So the current position of the Mail Class cylinder will be reassigned with the Mail Class text of the given index. See also MailClassCylinderPositions – Number of positions on Mail Class cylinder on page 8 and MailClassCylinderPositions – Number of positions on Mail Class cylinder on page 8.

Example:

The following Mail Class texts are available:

<table>
<thead>
<tr>
<th>Text</th>
<th>Index</th>
<th>Chosen:</th>
<th>Adjustment:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground</td>
<td>1</td>
<td>Ground</td>
<td>MailClassAdjust(3)</td>
</tr>
<tr>
<td>Express</td>
<td>2</td>
<td>Printed: Airmail</td>
<td></td>
</tr>
<tr>
<td>Airmail</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(empty)</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NewDateConfirm – Confirm new date request**

NewDateConfirm

After the FM has requested a new print date (SetDecades(Tax) returned R_NEWDATE) this function needs to be called to confirm the request. On FMs with automatic date the date wheels are adjusted automatically as soon as this function is called. FMs without automatic date assume that the date wheels have been adjusted manually when NewDateConfirm is called.
<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Postage</td>
<td>Currency</td>
<td>Value to be Franked.</td>
</tr>
<tr>
<td>DPT</td>
<td>Long</td>
<td>Department to book Franking to (0 if no DPTs exist). If DPTs exist a valid number has to be given (not 0)</td>
</tr>
<tr>
<td>MailClass</td>
<td>Integer</td>
<td>Index of MailClass to be printed (ignored if no MailClass is settable).</td>
</tr>
<tr>
<td>FrankMode</td>
<td>FRANKMODES</td>
<td>Franking mode. Check properties for availability. Must be FRK_NORM.</td>
</tr>
<tr>
<td>Preselection</td>
<td>Integer</td>
<td>Max. number of items to be Franked with this settings (0 = no limit). If Preselection is 0, FrankMode must be FRK_NORM.</td>
</tr>
<tr>
<td>Margin</td>
<td>Integer</td>
<td>Margin from the edge of the envelope to the beginning of the print.</td>
</tr>
<tr>
<td>RetVal</td>
<td>RETVALS</td>
<td>Returns information about the state of the machine.</td>
</tr>
</tbody>
</table>

This is probably the most important function. It sets the FM ready for Franking by setting the postage value along with a range of other settings.

For each Franking that is released the OnFranking event is called. See OnFranking – A Franking is released on page 3.

This function can NOT be used for Autotax FMs. See also AutoTaxFM – Autotax / Standard FM on page 5.

If the RetVal is R_HVLIM, the postage given is higher than the defined High Value Limit (and HV limit is active). After that, it is necessary to call either HVLimitAbort to abort the decade setting or HVLimitDeblock to accept the high postage. With HVLimitDeblock the decade wheels are set for one Franking only. After the Franking the FM goes back into QUIET state and the SetDecades function has to be repeated. See also HVLimitAbort – Not accept postage above High Value limit on page 14, HVLimitDeblock – Accept postage above High Value limit on page 14 and the following sequence diagrams.
For more information about allowed postage values see Examples for print image configuration on page 9.
For more information about Mail Classes SettableMailClasses – Number of available Mail Classes on page 9 and MailClassTexts – Texts assigned to Mail Classes on page 8.
For more information about Franking Modes see
FRANKMODES on page 25, FrankModeAvailItem – Availability of Franking Mode FRK_ITEMS on page 7 and the three following.
The relation between the value of margin and the real distance on the printed item depends on several other factors like Base speed etc.
More detailed information can be found in chapter 10 of the Service Manual for PLUS FMs. See also MarginAvailable – Availability of margin on page 8.
For all RetVals apart from R_OK and R_HVLIM the FM goes back into QUIET state (not ready for Franking). See also RETVALS on page 25.

SetDecadesTax – Set Autotax FM ready for Franking (for Autotax applications only)
RETVALS SetDecadesTax(Autotax, Acquisitiontax, DPT, Margin)

<table>
<thead>
<tr>
<th>Function</th>
<th>CURRENCY</th>
<th>Currency</th>
<th>In</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autotax</td>
<td>Currency</td>
<td>In 0...max. of print image configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acquisitiontax</td>
<td>Currency</td>
<td>In 0...max. of print image configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DPT</td>
<td>Long</td>
<td>In 0 or any existing DPT number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Margin</td>
<td>Short</td>
<td>In 1...255</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RetVal</td>
<td>RETVALS</td>
<td>In R_OK, R_HVLIM, R_QUIET, R_MERGE, all W, all L ...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SetDecadesTax can NOT be used for standard (non Autotax) FMs. See also AutoTaxFM – Autotax / Standard FM on page 5.
This function does the same as SetDecades but for Autotax applications only. It sets the FM ready for Franking by setting the Autotax and Acquisitiontax value along with other settings. The sum of Autotax and Acquisitiontax is printed. For Autotax applications that do not feature Acquisitiontax the Acquisitiontax value has to be set to zero. See also AcquisitionTaxAvailable – Availability of Acquisitiontax on page 5.
Mail Classes, Franking Modes and Preselection are not available for Autotax applications. Apart from that, the behavior of SetDecadesTax SetDecades is identical.
See also SetDecades – Set FM ready for Franking on page 16.
For all RetVals apart from R_OK and R_HVLIM the FM goes back into QUIET state (not ready for Franking). See also RETVALS on page 25.
**TapesPresent – Tapes present again**

TapesPresent

Can be sent to continue Franking Tapes after OnNoMoreTapes has been executed. See also *OnNoMoreTapes – No more Tapes* on page 3.

**TextToDisplay – Write text to FM display**

<table>
<thead>
<tr>
<th>Text</th>
<th>BSTR</th>
<th>String</th>
<th>In</th>
<th>Any text with max. length = 32</th>
<th>Text to display on the FM</th>
</tr>
</thead>
<tbody>
<tr>
<td>StartPos</td>
<td>Short</td>
<td>Integer</td>
<td>In</td>
<td>0..31</td>
<td>Position where the text starts</td>
</tr>
</tbody>
</table>

Displays any text on the FM display. If the text goes beyond the display the an exception with the ID INCORRDATA is raised. The text on the FMs display may be overwritten by "*" if User Timeout is active. The positions on the FM display are numbered as follows:

<table>
<thead>
<tr>
<th>0</th>
<th></th>
<th></th>
<th></th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td>31</td>
</tr>
</tbody>
</table>

**TMSStart – Start TMS reset**

TMSStart(Amount)

<table>
<thead>
<tr>
<th>Amount</th>
<th>CURRENCY</th>
<th>Currency</th>
<th>In</th>
<th>any</th>
<th>Amount of money to load into the FM. The value range depends on the FM's configuration.</th>
</tr>
</thead>
</table>

Starts the TMS process. The FM must be configured for TMS (center phone number, account number etc.) before calling this method. This function will return immediately so that the user application is not blocked while the TMS process is running. When the TMS process finishes the event OnTMSEnd is triggered. The whole TMS process can take up to about 150 sec. It is recommended that the user application starts a timer of about 3 minutes upon calling TMSStart. If the timer runs down without OnTMSEnd triggered it can be assumed that an error occurred (e.g. connection between PC and FM broken).

See also *OnTMSEnd – End of TMS procedure* on page 4.
Departments (FMDPT)

Important: Before any of the properties or methods of FMDPT can be used, the property `ActiveConnection` must be set.

**Properties**
The following applies to all properties except for `ActiveConnection`:
All properties are READ-ONLY. The properties are loaded from the FM when the connection is established. None of the properties ever change, so there is no need to reload them. On disconnection all properties become invalid.

**ActiveConnection – Relation to Connection**

<table>
<thead>
<tr>
<th>IConnection*</th>
<th>Connection</th>
<th>Read-Write</th>
<th>Any</th>
</tr>
</thead>
</table>

Has to be set before the first use of any property or method of FMDPT. It defines the Connection that FMDPT uses to perform the functions in the FM.

**NrAccounts – Max. number of DPTs**

<table>
<thead>
<tr>
<th>short <code>ActiveConnection</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>short Integer Read-Only  0...max. Number of departments (currently = 800)</td>
</tr>
</tbody>
</table>

Maximal number of department accounts that can be opened in the FM.

**NrDigits – Length of department numbers**

<table>
<thead>
<tr>
<th>short <code>ActiveConnection</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>short Integer Read-Only Any</td>
</tr>
</tbody>
</table>

Max. number of digits a department number can have.
Methods

Clear – Read and clear one department

Clear(DPT, Value, Items)

<table>
<thead>
<tr>
<th>DPT</th>
<th>long</th>
<th>long</th>
<th>In</th>
<th>Number of a existing department</th>
<th>Number of the department to be cleared</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>CURRENCY</td>
<td>Currency</td>
<td>Out</td>
<td>any</td>
<td>Value Franked to this department</td>
</tr>
<tr>
<td>Items</td>
<td>long</td>
<td>Long</td>
<td>Out</td>
<td>any</td>
<td>Sum of all values Franked to departments</td>
</tr>
</tbody>
</table>

Reads the value and the number of items Franked to a department and resets them to zero.

ClearAll – Read total and clear all departments

ClearAll(Value, Items)

<table>
<thead>
<tr>
<th>Value</th>
<th>CURRENCY</th>
<th>Currency</th>
<th>Out</th>
<th>any</th>
<th>Sum of all values Franked to departments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items</td>
<td>long</td>
<td>Long</td>
<td>Out</td>
<td>any</td>
<td>Sum of items Franked to departments</td>
</tr>
</tbody>
</table>

Reads the total value and the number of items Franked to departments and resets all departments to zero.

Close – Close department

RETVALS Close(DPT)

<table>
<thead>
<tr>
<th>DPT</th>
<th>long</th>
<th>long</th>
<th>In</th>
<th>Number of a existing department</th>
<th>Number of the department to be closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>RetVal</td>
<td>RETVALS</td>
<td>RETVALS</td>
<td>Ret</td>
<td>R_OK, E_NOTCLEAR</td>
<td>Returns information about the state of the machine</td>
</tr>
</tbody>
</table>

Closes an existing department. The department must be clear to be closed. See also RETVALS on page 25.

Open – Open new department

RETVALS Open(DPT)

<table>
<thead>
<tr>
<th>DPT</th>
<th>long</th>
<th>long</th>
<th>In</th>
<th>Number of a not existing department</th>
<th>Number of the department to be opened</th>
</tr>
</thead>
<tbody>
<tr>
<td>RetVal</td>
<td>RETVALS</td>
<td>RETVALS</td>
<td>Ret</td>
<td>R_OK, E_EXIST, E_FULL</td>
<td>Returns information about the state of the machine</td>
</tr>
</tbody>
</table>

Opens a new department with the given number. The department must not exist yet. DPT must not be longer than specified by the NrDigits property. See also NrDigits – Length of department numbers on page 19 and RETVALS on page 25.
### APPENDIX 1 · PAGE 24

#### Read – Read one department

**Read***(DPT, Value, Items)**

<table>
<thead>
<tr>
<th>DPT</th>
<th>Type (Long)</th>
<th>Value Type (Currency)</th>
<th>Type (Out)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>long</td>
<td>Currency</td>
<td>out</td>
<td>Any</td>
<td>Number of a existing department</td>
</tr>
<tr>
<td>long</td>
<td>Currency</td>
<td>out</td>
<td>Any</td>
<td>Number of the department to be read</td>
</tr>
</tbody>
</table>

Reads the value and the number of items Franked to a department.

#### ReadAll – Read all departments

**ReadAll***(DPT, Value, Items)**

<table>
<thead>
<tr>
<th>DPT</th>
<th>Type (Long)</th>
<th>Value Type (Currency)</th>
<th>Type (Out)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAFARRAY (long)</td>
<td>Long()</td>
<td>out</td>
<td>Any</td>
<td>Number of an existing department</td>
</tr>
<tr>
<td>SAFARRAY (CURRENCY)</td>
<td>Currency()</td>
<td>out</td>
<td>Any</td>
<td>One-dimensional array of department numbers</td>
</tr>
</tbody>
</table>

Provides a table containing all open departments. For each department the department number, the value Franked to the department and the items Franked to the department are supplied.

All arrays contain the same number of elements. The elements with the same index belong together and represent the data of one department.

#### ReadTotal – Read total of all departments

**ReadNext***(Value, Items)**

<table>
<thead>
<tr>
<th>Value</th>
<th>Type (Currency)</th>
<th>Type (Out)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>long</td>
<td>Currency</td>
<td>out</td>
<td>Sum of all values Franked to departments</td>
</tr>
<tr>
<td>long</td>
<td>Currency</td>
<td>out</td>
<td>Sum of items Franked to departments</td>
</tr>
</tbody>
</table>

Reads the total value and the number of items Franked to departments.
Statistics (FMStatistic)

Important: Before any of the methods of FMStatistic can be used the property ActiveConnection must be set.

Properties

ActiveConnection — Relation to Connection

<table>
<thead>
<tr>
<th>IConnection</th>
<th>ActiveConnection</th>
</tr>
</thead>
<tbody>
<tr>
<td>IConnection</td>
<td>Connection</td>
</tr>
</tbody>
</table>

Has to be set before the first use of any method of FMStatistic. It defines the Connection that FMStatistic uses to perform the functions in the FM.
**Methods**

**DailyTotal – Read daily statistics**

DailyTotal (Value, Items)

<table>
<thead>
<tr>
<th>Value</th>
<th>Currency</th>
<th>Currency</th>
<th>Out</th>
<th>any</th>
<th>Gradation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items</td>
<td>long</td>
<td>Long</td>
<td>Out</td>
<td>any</td>
<td>Value Franked on current day</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Number of items Franked on current day</td>
</tr>
</tbody>
</table>

Reads value and number of items Franked on current day.

**MonthlyTotal – Read 'Monthly Total' statistics**

MonthlyTotal (Value, Items, Day, Month, Year)

<table>
<thead>
<tr>
<th>Value</th>
<th>SAFEARRAY (Currency)</th>
<th>Currency()</th>
<th>Out</th>
<th>any</th>
<th>Gradation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items</td>
<td>SAFEARRAY (long)</td>
<td>Long()</td>
<td>Out</td>
<td>any</td>
<td>One-dimensional array of values Franked</td>
</tr>
<tr>
<td>Day</td>
<td>SAFEARRAY (short)</td>
<td>Integer()</td>
<td>Out</td>
<td>1...31</td>
<td>One-dimensional array of Days</td>
</tr>
<tr>
<td>Month</td>
<td>SAFEARRAY (short)</td>
<td>Integer()</td>
<td>Out</td>
<td>1..12</td>
<td>One-dimensional array of Months</td>
</tr>
<tr>
<td>Year</td>
<td>SAFEARRAY (short)</td>
<td>Integer()</td>
<td>Out</td>
<td>any (1980..2099)</td>
<td>One-dimensional array of Years (4 digit format)</td>
</tr>
</tbody>
</table>

Provides a table containing the values and number of items franked each day. All arrays contain the same number of elements. The elements with the same index belong together and represent the statistics of one day (value, items and date). The entries are sorted by date (oldest first). MonthlyTotal does not include the current day.

With this data a monthly report of daily consumption can be generated.

**SinceTotalClear – Read and clear 'Since Total' statistics**

SinceTotalClear (Value, Items, Day, Month, Year)

<table>
<thead>
<tr>
<th>Value</th>
<th>Currency</th>
<th>Currency</th>
<th>Out</th>
<th>any</th>
<th>Gradation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items</td>
<td>long</td>
<td>Long</td>
<td>Out</td>
<td>any</td>
<td>Total value Franked since last clear</td>
</tr>
<tr>
<td>Day</td>
<td>short</td>
<td>Integer</td>
<td>Out</td>
<td>1...31</td>
<td>Day of last clear</td>
</tr>
<tr>
<td>Month</td>
<td>short</td>
<td>Integer</td>
<td>Out</td>
<td>1..12</td>
<td>Month of last clear</td>
</tr>
<tr>
<td>Year</td>
<td>short</td>
<td>Integer</td>
<td>Out</td>
<td>any (1980..2099)</td>
<td>Year (4 digit format) of last clear</td>
</tr>
</tbody>
</table>

Reads total value and number of items that have been Franked since the last clear of 'Since Total' counter and resets the counters to zero.
SinceTotalRead - Read 'Since Total' statistics

SinceTotalRead(Value, Items, Day, Month, Year)

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>CURRENCY</td>
<td>Total value Franked since last clear</td>
</tr>
<tr>
<td>Items</td>
<td>long</td>
<td>Total number of items Franked since last clear</td>
</tr>
<tr>
<td>Day</td>
<td>short</td>
<td>Day of last clear</td>
</tr>
<tr>
<td>Month</td>
<td>short</td>
<td>Month of last clear</td>
</tr>
<tr>
<td>Year</td>
<td>short</td>
<td>Year (4 digit format) of last clear</td>
</tr>
</tbody>
</table>

Reads total value and number of items that have been Franked since the last clear of 'Since Total' counter.
**TariffGroups – Tariff group statistics**

<table>
<thead>
<tr>
<th>Label</th>
<th>Type</th>
<th>String()</th>
<th>Out</th>
<th>any</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>SAFEARRAY(CURRENCY)</td>
<td>Currency()</td>
<td>Out</td>
<td>any</td>
<td>Two-dimensional array of values Franked. 1st dim. = values for different tariff groups. 2nd dim. = values for different months</td>
</tr>
<tr>
<td>Items</td>
<td>SAFEARRAY(long)</td>
<td>Long()</td>
<td>Out</td>
<td>any</td>
<td>Two-dimensional array of number of items Franked. 1st dim. = items for different tariff groups. 2nd dim. = items for different months</td>
</tr>
</tbody>
</table>

The tariff group statistics always contain data for the last 12 months. This function provides a data structure that contains the following for each tariff group: one label, 12 values and 12 numbers of items (12 because there is one for each month).

The data is organized as shown below:

Month 0 is always the current month whereas month 1 to 11 are the previous months. E.g.: on November 15 1999, month 0 is November 1999, month 1 is October 1999 and month 11 is December 1998.
Type Definitions

The following definitions are all enumerated types. Do not use the discrete values of the constants, as they might change in later versions of the program library.

<table>
<thead>
<tr>
<th>BASES</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>B120</td>
<td></td>
</tr>
<tr>
<td>B150</td>
<td></td>
</tr>
<tr>
<td>B220</td>
<td></td>
</tr>
<tr>
<td>B220P</td>
<td></td>
</tr>
<tr>
<td>B250P</td>
<td></td>
</tr>
<tr>
<td>B320</td>
<td></td>
</tr>
<tr>
<td>B320P</td>
<td></td>
</tr>
<tr>
<td>B335</td>
<td></td>
</tr>
<tr>
<td>B335P</td>
<td>Low cost base</td>
</tr>
<tr>
<td>B335R</td>
<td></td>
</tr>
<tr>
<td>B337</td>
<td></td>
</tr>
<tr>
<td>B337P</td>
<td></td>
</tr>
<tr>
<td>B340P</td>
<td>High speed base</td>
</tr>
<tr>
<td>BTEST</td>
<td>Special base for duration tests</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FMTYPES</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>F3XXPLUS</td>
<td>FM$s$ of the F3XXPLUS series – the only FM-type that library version 1.0 does work with</td>
</tr>
</tbody>
</table>
FRANKMODES

<table>
<thead>
<tr>
<th>FRK_NORM</th>
<th>Normal Franking without Preselection. Check property for availability.</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRK_TAPE</td>
<td>Preselected tape Franking. Possible with Bases that have a tape motor only. Check property for availability.</td>
</tr>
<tr>
<td>FRK_LETTER</td>
<td>Preselected letter Franking. Available with some Bases only. Check property for availability.</td>
</tr>
<tr>
<td>FRK_ITEM</td>
<td>Preselected item Franking. Tapes or letters can be Franked. Check property for availability.</td>
</tr>
</tbody>
</table>

See also FrankModeAvailItem – Availability of Franking Mode FRK_ITEMS on page 7 as well as the three following properties.

LASTDECTYPES

<table>
<thead>
<tr>
<th>LD05</th>
<th>Last decade wheel can be 0 or 5 only</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD09</td>
<td>Last decade wheel can be anything from 0..9</td>
</tr>
</tbody>
</table>

See also LastDecadeType – Type of the last decade wheel on page 8.

REVALS

The first letter indicates the type of the return value:

R: Normal return value
E: Error
W: Warning
L: Limit

General

Standard return values.
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_OK</td>
<td>OK, function successful</td>
</tr>
<tr>
<td>E_NOK</td>
<td>Not OK, General error. Usually such an error causes an exception.</td>
</tr>
<tr>
<td>R_NEWDATE</td>
<td>Print date has changed and must be adjusted either by hand or by pushing a button on the FM keyboard and the print must be checked. Note: An offset of a few hours between the print date and the FM's system date can be defined in the FM.</td>
</tr>
<tr>
<td>R_QUICK</td>
<td>FM in QUIET state, not ready for Franking (anymore).</td>
</tr>
<tr>
<td>R_HVLIM</td>
<td>Postage higher than High Value limit. Call HVLimitDeblock or HVLimitAbort within the timeout.</td>
</tr>
<tr>
<td>E_TIMEOUT</td>
<td>High Value limit timeout has run down. SetDecades must be repeated.</td>
</tr>
</tbody>
</table>

**Warnings**

Warning limit reached. A warning indicates that soon a limit will be reached which can cause the FM to be blocked. If a warning is returned from a SetDecade command, the FM is NOT ready for Franking. In that case repeat the SetDecade command.

| W_DESCREG   | Descending register warning limit reached (soon no more funds).             |
| W_ASCREG    | Ascending register warning limit reached.                                   |
| W_MAXITEMS  | Item warning limit reached.                                                 |
| W_READING   | Reading warning limit reached. Soon a TMS connection must be established. (Occurs on TMS meters only) |
| W_BATTERY   | Battery expiration date reached.                                            |

**Limits**

Franking is not possible anymore and an action has to be taken to make Franking available again.

| L_DESCREG   | No more funds (descending register is zero).                                |
| L_ASCREG    | Ascending register limit reached.                                           |
| L_MAXITEMS  | Item limit reached.                                                        |
| L_READING   | Reading limit reached. A TMS connection must be established. (Occurs on TMS meters only) |
| L_BATCHCOUNTER | Batch counter full. Clear batch counter.                               |
| L_DPT       | Department counter full. Clear content of department.                      |
| L_TOT       | Since Total statistics counter full. Clear Since Total statistics.          |
### OnRotorError event

These are return values used by OnRotorError event only.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_CALLSERVICE</td>
<td>An error that demands to call a service technician.</td>
</tr>
<tr>
<td>E_ROTOR</td>
<td>Error on rotor e.g. position.</td>
</tr>
<tr>
<td>E_CROSS</td>
<td>Cross error</td>
</tr>
<tr>
<td>E_BASESPEED</td>
<td>Speed of base to high.</td>
</tr>
<tr>
<td>E_DECADE</td>
<td>Decade wheels can not be set.</td>
</tr>
<tr>
<td>E_PRINTER</td>
<td>Printer error, e.g. not ready or not present.</td>
</tr>
</tbody>
</table>

### Departments

These return values are used in department functions only.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_EXIST</td>
<td>Department does exist already. Can not open department.</td>
</tr>
<tr>
<td>E_NOTCLEAR</td>
<td>Department must be clear before it can be closed.</td>
</tr>
<tr>
<td>E_FULL</td>
<td>No more room for departments. Can not open department.</td>
</tr>
</tbody>
</table>

### TMS

These return values are used in TMSEnd only.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_DESC_MAX</td>
<td>Descending register exceeded maximum.</td>
</tr>
<tr>
<td>E_SUM_OFL</td>
<td>Control Total or Descending register overflow.</td>
</tr>
<tr>
<td>E_AMOUNT_MAX</td>
<td>Amount too high.</td>
</tr>
<tr>
<td>E_AMOUNT_STEP</td>
<td>Amount is not a multiple of the defined step. Round amount.</td>
</tr>
<tr>
<td>E_AMOUNT_MIN</td>
<td>Amount too low.</td>
</tr>
<tr>
<td>E_TMS_ABORT</td>
<td>TMS host has aborted.</td>
</tr>
<tr>
<td>E_TMS_TRANSM</td>
<td>Error in TMS transmission.</td>
</tr>
<tr>
<td>E_TMS_CONNECT</td>
<td>No TMS connection.</td>
</tr>
<tr>
<td>E_TMS_MODEM</td>
<td>TMS modem error.</td>
</tr>
<tr>
<td>E_TMS_LINE_BUSY</td>
<td>Phone line is busy.</td>
</tr>
<tr>
<td>E_TMS_MODEM_INIT</td>
<td>Error initializing modem.</td>
</tr>
<tr>
<td>E_TMS_FAILED</td>
<td>TMS failed (internal problem or not a TMS FM).</td>
</tr>
<tr>
<td>E_FMSTATE</td>
<td>FM is in wrong state for carrying out this function. Go into QUIET state first.</td>
</tr>
</tbody>
</table>
HSB (High Speed Base – S340 plus)
These return values indicate the state of the HSB.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_HSB_KEY</td>
<td>Key of high speed base is in wrong position.</td>
</tr>
<tr>
<td>E_HSB_HOT</td>
<td>High speed base too hot.</td>
</tr>
<tr>
<td>E_HSB_STANDBY</td>
<td>High speed base is on standby.</td>
</tr>
<tr>
<td>E_HSB_INKCOVER</td>
<td>Ink cover of high speed base is open.</td>
</tr>
</tbody>
</table>
**Exceptions (EXCEPTIONIDS)**
The exceptions thrown by the FM Control library can be identified by the ID listed below. (Hint: In Visual Basic the exception IDs returned by the Err object do have an offset of the value of vbObjectError. Use vbObjectError to calculate the ID before comparing Err.Number with the IDs.)

<table>
<thead>
<tr>
<th>ID</th>
<th>Description (also returned as Description in the Err object)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOCONNECTION</td>
<td>Connection could not be established</td>
</tr>
<tr>
<td>INVALIDACTCON</td>
<td>No connection (invalid ActiveConnection)</td>
</tr>
<tr>
<td>INVCONSTRING</td>
<td>Invalid Connection String</td>
</tr>
<tr>
<td>CONEXIST</td>
<td>Connection exists already</td>
</tr>
<tr>
<td>LOSTCONNECTION</td>
<td>Connection lost</td>
</tr>
<tr>
<td>NOANSWER</td>
<td>FM does not answer</td>
</tr>
<tr>
<td>SENDFAIL</td>
<td>Send failed</td>
</tr>
<tr>
<td>INVSENDDATA</td>
<td>Invalid data to send to FM</td>
</tr>
<tr>
<td>INCORRDATA</td>
<td>FM said 'data is incorrect'</td>
</tr>
<tr>
<td>INVFMDATA</td>
<td>Invalid data received from FM</td>
</tr>
<tr>
<td>INVPOSTAGE</td>
<td>Invalid Postage</td>
</tr>
<tr>
<td>INVDPT</td>
<td>Invalid DPT number (negative or too many digits)</td>
</tr>
<tr>
<td>DPTNEXIST</td>
<td>DPT does not exist</td>
</tr>
<tr>
<td>INVMAILCLASS</td>
<td>Invalid Mail Class</td>
</tr>
<tr>
<td>INVFRKMODE</td>
<td>Invalid Franking Mode</td>
</tr>
<tr>
<td>INVPRESELECTION</td>
<td>Invalid Preselection value</td>
</tr>
<tr>
<td>INVMARGIN</td>
<td>Invalid Margin value</td>
</tr>
<tr>
<td>WRONGSTATE</td>
<td>FM is in wrong state for carrying out this function</td>
</tr>
<tr>
<td>STANDBY</td>
<td>Rotor task is on standby</td>
</tr>
<tr>
<td>DCOPEN</td>
<td>Die Cover is open</td>
</tr>
<tr>
<td>FCNOTSUPPORTED</td>
<td>Function not supported for this FM type</td>
</tr>
<tr>
<td>ABORT</td>
<td>Current action aborted</td>
</tr>
<tr>
<td>NOK</td>
<td>FM returned NOT OK</td>
</tr>
<tr>
<td>UNKNOWN</td>
<td>Unknown exception</td>
</tr>
</tbody>
</table>
Requirements

PC system requirements
Operating system: Windows95 or higher or WindowsNT 4.0 with Service Pack 4 or higher.
Software: MSVCR.DLL version 6.0 or higher. See below for instructions for checking version and updating. Distributed COM (DCOM) installed. This is included in WindowsNT 4.0 and Windows98 or higher. If you are using Windows95 and are not sure whether DCOM is installed on your system please install DCOM by starting DCOM95.EXE that comes together with FMCtrl.
Hardware: At least one free serial COM-Port (V.24 / RS232) – one per FM to be controlled simultaneously. A SCSI Terminal Server can be used to expand the PC with up to 32 additional COM-Ports. Such SCSI Terminal Servers are available from Digi (www.digi.com).
V.24 / RS232 link cable to connect the PC with the FM.

FM requirements
FM-Type: F3XXPLUS with one free V.24 / RS232 port.

FM using an external modem for TMS can be equipped with an INFAC interface board. INFAC provides two V.24 / RS232 ports. One port to be used for the connection to the PC, the other for the optional external modem.
Remark: While in FM Control mode (Remote Control = ON) no other devices than the PC or the external modem can be connected to the FM. Additional periphery like scales would have to be interfaced to the PC.

FM-Software: Remote control enabled software. FMs having such a software installed do have the option to switch remote control ON or OFF in the Service menu.
Before using the program library

Version 6.0 of MSVCRT.DLL
MSVCRT.DLL is an important part of the WindowsNT system and is often used in Windows95 too. To make sure that the FMCtrl library works correctly, version 6.0 or higher of MSVCRT.DLL is necessary. So before the first use of the FMCtrl library check the existence and version of MSVCRT.DLL and update it if necessary. To do this proceed as follows:

- Use Windows Explorer to open the directory where MSVCRT.DLL is located (WindowsNT: C:\WinNT\System32  Windows95: C:\Windows\System)

If MSVCRT.DLL does not exist:
- Copy the file delivered together with the FMCtrl library to the location mentioned above.

If MSVCRT.DLL does exist:
- Select MSVCRT.DLL and choose Properties form File menu.
- In the appearing window select Version. On the first line the version number is shown.

If the existing MSVCRT.DLL is older than version 6.0:
- Open a DOS console window or switch into DOS mode.
- Change to directory where MSVCRT.DLL is located (see above).
- Rename MSVCRT.DLL to MSVCRT.OLD by entering rename msvcrtd.dll msvcrtd.old (this does work from the DOS console window or DOS mode only)
- Copy MSVCRT.DLL that comes together with FMCtrl to the directory where you found MSVCRT.DLL.
- Reboot your system
Registering the library
Before the FMCtrl library can be used it must be registered in Windows:
• Open a DOS console window.
• Change to directory where FMCtrl.exe is located (this can be any directory).
• Enter the following commands (the first command unregisters a possibly registered older version of the library – always do this before registering):
  FMCtrl-unregserver
  FMCtrl-regserver
• Now FMCtrl is ready for use e.g. from Visual Basic. To use it from Visual Basic 6.0 open a new project, choose References from Project menu and tick the FMCtrl entry. Open the object browser by pressing F2. In the object browser select FMCTRL.LIB from the drop down list. Now the interface of the library is shown in the object browser.

Using the library

Establishing connection
Before any of the functions in the library can be used a Connection object has to be created. The next step is usually calling the Connect method that establishes the connection to the FM.
Each of the other objects has an ActiveConnection property. It defines the Connection that the object uses to perform the functions in the FM. The ActiveConnection must be assigned with a Connection before any property or method of the object can be used.

Error handling
Most of the functions in the FM Control library can throw COM-Exceptions in case of an error. COM-Exceptions provide an error number and a description, that can be used in the error handler of the user application.
Some functions do also return a value that determinates the state of the machine. Those values do indicate the reaching of warning levels and limits rather than errors.
Logfile
If you encounter problems using the FM Control library the logfile can provide additional information about the failure. Logfile writing can be enabled by creating a subdirectory named Log in the directory where FMControl.exe is located. The logfile fmctrl.txt will be placed into this subdirectory. Please always create a logfile before contacting technical support!

Connection (Connection)

Properties

ConnectionString – Connection string
BSTR ConnectionString

<table>
<thead>
<tr>
<th>BSTR</th>
<th>String</th>
<th>Read-Only</th>
<th>Valid strings</th>
<th>Defines connection device, protocol etc.</th>
</tr>
</thead>
</table>

Returns the ConnectionString for the current connection. All characters of this string are always upper case. If the string is not empty (length ≠ 0) the connection does exist. On disconnection the string is reset to 0 length.

LibraryVersion – Version number
BSTR LibraryVersion

<table>
<thead>
<tr>
<th>BSTR</th>
<th>String</th>
<th>Read-Only</th>
<th>#.#.#. where # is a number 0...9999</th>
<th>Version of this program library</th>
</tr>
</thead>
</table>

Returns the version of the program library that is used.
### States with reactions

The following list contains all states returned from the Franking Machine with a description and the reaction that is taken by the FM Control library. The FM Control library either throws an exception or returns a certain value to inform the user application.

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
<th>Reaction (return-value</th>
<th>REIVal) or exception</th>
</tr>
</thead>
<tbody>
<tr>
<td>ER_OK</td>
<td>Function successful</td>
<td>R_OK</td>
<td></td>
</tr>
<tr>
<td>ST_QUIET</td>
<td>Error</td>
<td>E_OK</td>
<td>Exception: &quot;FM sent NOT OK&quot; (not for Events)</td>
</tr>
<tr>
<td>ST_NEW_DATE</td>
<td>Print date has changed</td>
<td>R_OK</td>
<td></td>
</tr>
<tr>
<td>ST_Wr_DptNr</td>
<td>Wrong department number</td>
<td>R_QUIET</td>
<td></td>
</tr>
<tr>
<td>ST_Dpt_First</td>
<td>Department number does not exist</td>
<td>R_NEWDATE</td>
<td></td>
</tr>
<tr>
<td>ST_Dpt_Exist</td>
<td>Department exists already (when trying to open DPT)</td>
<td>E_EXIST</td>
<td>Exception: &quot;DPT does not exist&quot;</td>
</tr>
<tr>
<td>ST_NoDpt_empty</td>
<td>No Department open (when reading or clearing total)</td>
<td>E_FULL</td>
<td></td>
</tr>
<tr>
<td>ST_NoDptSpace</td>
<td>Do space for more departments</td>
<td>E_NOTCLEAR</td>
<td></td>
</tr>
<tr>
<td>ST_EoList</td>
<td>Department is not clear (when trying to close a DPT)</td>
<td>remapped to ST_OK</td>
<td></td>
</tr>
<tr>
<td>ER_PARAMETER</td>
<td>Parameter value invalid</td>
<td>remapped to ST_OK</td>
<td>Exception: &quot;FM said 'data is incorrect'&quot;</td>
</tr>
<tr>
<td>ER_DC_OPEN</td>
<td>Please close die cover</td>
<td>Exception: &quot;Die Cover is open&quot;</td>
<td></td>
</tr>
<tr>
<td>ER_FORMAT</td>
<td>Invalid value or invalid mail class or invalid presel mode (neither FRIK_NORM nor FRIK_TEM)</td>
<td>Exception: &quot;FM said 'data is incorrect'&quot;</td>
<td></td>
</tr>
<tr>
<td>ER_CALL_SERVICE</td>
<td>OS appeared or exists already</td>
<td>E_CALLSERVICE</td>
<td>(also used with ROTOR-ERROR event)</td>
</tr>
<tr>
<td>ER_BREAK</td>
<td>Current action aborted / Setting error</td>
<td>E_TIMEOUT</td>
<td></td>
</tr>
<tr>
<td>ER_HV_TIMEOUT</td>
<td>HV timer run down -&gt; meter not ready to frank</td>
<td>E_TIMEOUT</td>
<td></td>
</tr>
<tr>
<td>ER_STANDBY</td>
<td>Please repeat procedure</td>
<td>Exception: &quot;Roter task is on standby&quot;</td>
<td></td>
</tr>
<tr>
<td>ER_ROTOR</td>
<td>Rotor not in base position</td>
<td>E_ROTOR</td>
<td>(also used with ROTOR-ERROR event)</td>
</tr>
<tr>
<td>ER_CROSS</td>
<td>Cross error</td>
<td>E_CROSS</td>
<td>(also used with ROTOR-ERROR event)</td>
</tr>
<tr>
<td>ER_BASE_SPEED</td>
<td>Speed of Base to high</td>
<td>E_BASESPEED</td>
<td>(also used with ROTOR-ERROR event)</td>
</tr>
<tr>
<td>ER_MIDM_HIGH_LOW</td>
<td>Margin too low</td>
<td>E_MIDM_HIGH_LOW</td>
<td></td>
</tr>
<tr>
<td>ER_MIDM_HIGH</td>
<td>Margin too high</td>
<td>E_MIDM_HIGH</td>
<td></td>
</tr>
<tr>
<td>ER_MIDM_LOW</td>
<td>Number of bytes (parameter length) incorrect</td>
<td>E_MIDM_LOW</td>
<td></td>
</tr>
<tr>
<td>ER_NOPRESEL</td>
<td>No preselection entered</td>
<td>E_NOPRESEL</td>
<td></td>
</tr>
<tr>
<td>ER_INVPRESEL</td>
<td>Wrong preselection (0) entered</td>
<td>E_INVPRESEL</td>
<td></td>
</tr>
<tr>
<td>ER_PRINTER_NOT_READY</td>
<td>Journal printer not ready (no paper, no connection etc.)</td>
<td>E_PRINTER</td>
<td>(also used with ROTOR-ERROR event)</td>
</tr>
<tr>
<td>ER_MSTATE</td>
<td>FM in wrong state for carrying out requested function</td>
<td>E_MSTATE</td>
<td>Exception: &quot;FM is in wrong state for carrying out this function&quot; Special: E_FMSTATE for OnTIMEnd event.</td>
</tr>
<tr>
<td>L_HV</td>
<td>Postage above HV limit (state returned with I-Message)</td>
<td>R_HVIM</td>
<td></td>
</tr>
<tr>
<td>L_ASC_REG</td>
<td>Ascending register limit reached</td>
<td>R_ASCREG</td>
<td></td>
</tr>
<tr>
<td>L_DES_REG</td>
<td>No more funds (descending register is zero)</td>
<td>R_DESREG</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>Description</td>
<td>Reaction (return value)</td>
<td>State</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>L_MAX_ITEMS</td>
<td>Item limit reached</td>
<td>L_MAX_ITEMS</td>
<td>L_READING</td>
</tr>
<tr>
<td>L_VAL_READING</td>
<td>Reading limit (items) reached</td>
<td>L_READING</td>
<td>L_READING</td>
</tr>
<tr>
<td>L_ITEM_READING</td>
<td>Clear content if this department</td>
<td>L_OPT</td>
<td>L_TOT_ITEM</td>
</tr>
<tr>
<td>L_TIME_READING</td>
<td>Clear content if this department</td>
<td>L_TOT_VALUE</td>
<td>L_BAT_ITEM</td>
</tr>
<tr>
<td>L_OPT_ITEM</td>
<td>Clear content if this department</td>
<td>L_TOT_VALUE</td>
<td>L_BAT_VALUE</td>
</tr>
<tr>
<td>W_ASC_REG</td>
<td>Warning: ascending register limit</td>
<td>W_DESC_REG</td>
<td>W_READING</td>
</tr>
<tr>
<td>W_MAX_ITEMS</td>
<td>Warning: item limit</td>
<td>W_MAX_ITEMS</td>
<td>W_READING</td>
</tr>
<tr>
<td>W_DESC_REG</td>
<td>Warning: descending register limit (soon no more funds)</td>
<td>W_READING</td>
<td>W_READING</td>
</tr>
<tr>
<td>W_VAL_READING</td>
<td>Warning: battery expiration date</td>
<td>W_READING</td>
<td>W_READING</td>
</tr>
<tr>
<td>W_ITEM_READING</td>
<td>Max. descending register value exceeded</td>
<td>ER_DESC_MAX</td>
<td>W_BATTERY</td>
</tr>
<tr>
<td>W_TIME_READING</td>
<td>(TMS) Amount too high</td>
<td>ER_VORG_MAX</td>
<td>E_AMOUNT_MAX</td>
</tr>
<tr>
<td>W_BATTERY</td>
<td>(TMS) Amount not a multiple of the defined step</td>
<td>ER_VORG_PAR</td>
<td>E_AMOUNT_MIN</td>
</tr>
<tr>
<td>ER_DESC_MAX</td>
<td>(TMS) Amount too low</td>
<td>ER_VORG_MIN</td>
<td>E_AMOUNT_STEP</td>
</tr>
<tr>
<td>ER_VORG_MAX</td>
<td>TMS reset aborted by the user</td>
<td>ER_USER_ABORT</td>
<td>E_AMOUNT_MIN</td>
</tr>
<tr>
<td>ER_VORG_PAR</td>
<td>TMS host has aborted</td>
<td>ER_TMS_ABORT</td>
<td>E_TMS_ABORT</td>
</tr>
<tr>
<td>ER_VORG_MIN</td>
<td>TMS host has aborted (with message)</td>
<td>ER_TMS_ABORT_MSG</td>
<td>E_TMS_ABORT</td>
</tr>
<tr>
<td>ER_USER_ABORT</td>
<td>Error in TMS transmission</td>
<td>ER_TMS_TRANSM</td>
<td>E_TMS_TRANSM</td>
</tr>
<tr>
<td>ER_TMS_ABORT</td>
<td>No TMS connection</td>
<td>ER_TMS_DISCONNECT</td>
<td>E_TMS_DISCONNECT</td>
</tr>
<tr>
<td>ER_TMS_CONNECT</td>
<td>TMS modem error</td>
<td>ER_TMS_MODEM</td>
<td>E_TMS_MODEM</td>
</tr>
<tr>
<td>ER_TMS_MODEM</td>
<td>TMS modem error</td>
<td>ER_TMS_MODEM_INIT</td>
<td>E_TMS_MODEM_INIT</td>
</tr>
<tr>
<td>ER_TMS_PHONE_BUSY</td>
<td>(TMS) Error initializing modem</td>
<td>E_TMS_PHONE_BUSY</td>
<td>E_TMS_PHONE_BUSY</td>
</tr>
<tr>
<td>ER_TMS_PHONE_BUSY</td>
<td>(TMS) Line is busy</td>
<td>E_TMS_PHONE_BUSY</td>
<td>E_TMS_PHONE_BUSY</td>
</tr>
<tr>
<td>ER_HSB_MS</td>
<td>Key of high speed base is in wrong position</td>
<td>E_HSB_MS</td>
<td>E_HSB_MS</td>
</tr>
<tr>
<td>ER_HSB_HOT</td>
<td>High speed base too hot</td>
<td>E_HSB_HOT</td>
<td>E_HSB_HOT</td>
</tr>
<tr>
<td>ER_HSB_SBY</td>
<td>High speed base is on standby</td>
<td>E_HSB_STANDBY</td>
<td>E_HSB_STANDBY</td>
</tr>
<tr>
<td>ER_HSB_FD</td>
<td>Ink cover of high speed base is open</td>
<td>E_HSB_INKCOVER</td>
<td>E_HSB_INKCOVER</td>
</tr>
</tbody>
</table>

Return value to a O message which is sent while the FM is in the wrong STATE, the answer is an R message with ER_MISMATCH (not L CANCEL).
## List of FM functions with returned states

<table>
<thead>
<tr>
<th>ID</th>
<th>Function name</th>
<th>States</th>
</tr>
</thead>
<tbody>
<tr>
<td>1E72</td>
<td>R_SPC_CONNECT</td>
<td>ST_OK, ER_NOK</td>
</tr>
<tr>
<td>1E13</td>
<td>R_ASK_BASE_MODEL_PC</td>
<td>ST_OK, ER_NOK</td>
</tr>
<tr>
<td>1E15</td>
<td>R_ASK_BASE_SW_VERS_PC</td>
<td>ST_OK, ER_NOK</td>
</tr>
<tr>
<td>1E76</td>
<td>R_ROTPARAM_READ</td>
<td>ST_OK, ER_NOK</td>
</tr>
<tr>
<td>1E93</td>
<td>R_READ_DATETIME</td>
<td>ST_OK, ER_NOK</td>
</tr>
<tr>
<td>1E17</td>
<td>R_USER_TIMEOUT_ONOFF_PC</td>
<td>ST_OK, ER_NOK</td>
</tr>
<tr>
<td>1E19</td>
<td>R_LCW_ONOFF_PC</td>
<td>ST_OK, ER_NOK</td>
</tr>
<tr>
<td>1E22</td>
<td>R_READ_MACH_NR</td>
<td>ST_OK, ER_NOK</td>
</tr>
<tr>
<td>1E97</td>
<td>R_READ_NO_MACH</td>
<td>ST_OK, ER_NOK</td>
</tr>
<tr>
<td>1E99</td>
<td>R_READ_NO_REPL</td>
<td>ST_OK, ER_NOK</td>
</tr>
<tr>
<td>1E32</td>
<td>R_HV_ONOFF_PC</td>
<td>ST_OK, ER_NOK</td>
</tr>
<tr>
<td>1E34</td>
<td>R_DISP_DESC_ONOFF_PC</td>
<td>ST_OK, ER_NOK</td>
</tr>
<tr>
<td>1E51</td>
<td>R_KEYBOAORD_ONOFF_PC</td>
<td>ST_OK, ER_NOK</td>
</tr>
<tr>
<td>1E8D</td>
<td>R_BLK_FRANK_MENU_PC</td>
<td>ST_OK, ER_NOK</td>
</tr>
<tr>
<td>1E38</td>
<td>R_BAT_ON_PC</td>
<td>ST_OK, ER_NOK</td>
</tr>
<tr>
<td>1E39</td>
<td>R_BAT_READ_PC</td>
<td>ST_OK, ER_NOK</td>
</tr>
<tr>
<td>1E3A</td>
<td>R_BAT_CLEAR_PC</td>
<td>ST_OK, ER_NOK</td>
</tr>
<tr>
<td>1E70</td>
<td>R_GET_COUNTER</td>
<td>ST_OK, ER_NOK</td>
</tr>
<tr>
<td>1ECC</td>
<td>R_PC_READ_MC_TEXT</td>
<td>ST_OK, ER_NOK, ?ER_MSTATE</td>
</tr>
<tr>
<td>1E11</td>
<td>R_TEXT_TO_DISPLAY_PC</td>
<td>ST_OK, ER_NOK, ER_DATLEN, ER_FORMAT</td>
</tr>
<tr>
<td>1E6A</td>
<td>R_SPC_MC_COR</td>
<td>ST_OK, ER_STANDBY, ER_CALL_SERVICE, ER_DC_OPEN, ER_DATLEN, ER_PARAMETER</td>
</tr>
<tr>
<td>1E83</td>
<td>R_SPC_GO_QUIET</td>
<td>ST_OK, ER_ROTOR, ER_DC_OPEN, ER_CALL_SERVICE</td>
</tr>
<tr>
<td>1E74</td>
<td>R_SPC_DEC_SET</td>
<td>All</td>
</tr>
<tr>
<td>1E8F</td>
<td>R_PC_DEC_SET_TAX</td>
<td>All</td>
</tr>
<tr>
<td>1E58</td>
<td>I_SPC_QUIET</td>
<td>ST_OK, ER_HSB_MS, ER_HSB_HOT, ER_HSB_STBY, ER_HSB_FD</td>
</tr>
<tr>
<td>1E5D</td>
<td>I_PC_FRK_EVENT</td>
<td>ST_OK, ST_QUIET, ST_NEW_DATE?, W,..., L,...</td>
</tr>
<tr>
<td>1E4D</td>
<td>I_SPC_ROT_ERROR</td>
<td>ER_CALL_SERVICE, ER_ROTOR, ER_CROSS, ER_BASE_SPEED, ER_DECADEC, ER_PRINTER_NOTREADY</td>
</tr>
<tr>
<td>1E09</td>
<td>R_SPC_DISCONNECT</td>
<td>ST_OK, ER_ROTOR, ER_DC_OPEN, ER_CALL_SERVICE</td>
</tr>
<tr>
<td>1E8C</td>
<td>R_SET_TMS_PC</td>
<td>ST_OK, ER_TMS_ABORT, ER_TMS_ABORT_MSG, ER_TMS_TRANSM, ER_TMS_CONNECT, ER_TMS_MODEM, ER_TMS_PHONE_BUSY, ER_TMS_MODEM_INIT, ER_TMS_FAILED</td>
</tr>
<tr>
<td>1EAF</td>
<td>R_TotCluster_PC</td>
<td>ST_OK, ER_DATLEN</td>
</tr>
<tr>
<td>1E55</td>
<td>R_TOT_READ_PC</td>
<td>ST_OK, ER_DATLEN</td>
</tr>
<tr>
<td>1E98</td>
<td>R_TOT_CLEAR_PC</td>
<td>ST_OK, ER_DATLEN</td>
</tr>
<tr>
<td>1E43</td>
<td>R_READ_DAY_PC</td>
<td>ST_OK, ER_DATLEN</td>
</tr>
<tr>
<td>1EAF</td>
<td>R_READ_YEAR_PC</td>
<td>ST_OK, ER_DATLEN</td>
</tr>
<tr>
<td>1E57</td>
<td>R_READ_Term_Par</td>
<td>ST_OK, ER_DATLEN</td>
</tr>
<tr>
<td>1E4A</td>
<td>R_1ST_DPT_PC</td>
<td>ST_OK, ER_DATLEN, ST_Eclist</td>
</tr>
<tr>
<td>1E49</td>
<td>R_NEXT_DPT_PC</td>
<td>ST_OK, ER_DATLEN, ST_Eclist</td>
</tr>
<tr>
<td>1E48</td>
<td>R_1ST_DAY_PC</td>
<td>ST_OK, ER_DATLEN, ST_Eclist</td>
</tr>
<tr>
<td>1E4D</td>
<td>R_1ST_TermGroup_PC</td>
<td>ST_OK, ER_DATLEN, ST_Eclist</td>
</tr>
<tr>
<td>1E43</td>
<td>R_1ST_TermGroup_PC</td>
<td>ST_OK, ER_DATLEN, ST_Eclist</td>
</tr>
<tr>
<td>1E3C</td>
<td>R_DPTPARAM_READ</td>
<td>ST_OK, ER_DATLEN</td>
</tr>
<tr>
<td>1E0D</td>
<td>R_DPT_OPEN_PC</td>
<td>ST_OK, ST_W_DPT, ST_D0, Ext, ST_NODISPLAY</td>
</tr>
<tr>
<td>1E6F</td>
<td>R_DPT_CLOSE_PC</td>
<td>ST_OK, ST_W_DPT, ST_D0, Next, ST_ColContent</td>
</tr>
<tr>
<td>1E53</td>
<td>R_SET_DPT_PC</td>
<td>ST_OK, ST_W_DPT, ST_D0, Next</td>
</tr>
<tr>
<td>1E55</td>
<td>R_CLEAR_DPT_PC</td>
<td>ST_OK, ST_W_DPT, ST_D0, Next</td>
</tr>
<tr>
<td>1E57</td>
<td>R_SUM_READ_PC</td>
<td>ST_OK, ST_W_DPT, ST_D0, Next</td>
</tr>
<tr>
<td>1E59</td>
<td>R_SUM_CLEAR_PC</td>
<td>ST_OK, ST_W_DPT, ST_D0, Next</td>
</tr>
</tbody>
</table>
CLAIMS

What is claimed is:

1. A system for controlling a franking machine from a remote computer, comprising:

   control software for controlling said franking machine;

   communications apparatus for said franking machine to communicate with said computer;

   a user application operating on said computer to interface with said control software via said communications apparatus; and

   a programming library for communicating with said user application, said programming library providing said user application with a series of functional controls of said franking machine.

2. The system of claim 1 wherein said functional controls include at least one item selected from the group consisting of connection, configuration, actions, departments and statistics.

3. A computer readable medium having stored therein computer readable code for interacting with a user application running on a computer communicating with a franking machine, said computer readable code comprising:

   a programming library for communicating with said user application, said programming library providing said user application with a series of functional controls of said franking machine.
4. The computer readable medium of claim 3 wherein said functional controls include at least one item selected from the group consisting of connection, configuration, actions, departments and statistics.

5. A method for operating a franking machine which is in communication with a computer, comprising:

running a user application on said computer;

accessing a programming library for communicating with said user application, said programming library providing said user application with a series of functional controls of said franking machine.

6. The method of claim 5 further comprising performing at least one operation using said functional controls, wherein said functional controls include at least one item selected from the group consisting of connection, configuration, actions, departments and statistics.
FIG. 3

USER APPLICATION

WINDOW USER INTERFACE 26

COMMUNICATION WITH OTHER PC PERIPHERY (SCALE, PRINTER, MODEM, BARCODE READER) 32

STATISTIC EVALUATION 28

DATABASE ACCESS 30

PROGRAMMING INTERFACE (COM) 40

PROGRAMMING LIBRARY 24

IN/OUTPUT CHECK PROVIDE ADDITIONAL INFORMATION COMMUNICATION PROTOCOL

DESIGNED BY CUSTOMER OR THIRD PARTY

PROVIDED BY AscomMailng SYSTEMS
○ = QUIET STATE
ALWAYS WHEN THE MACHINE GOES TO QUIET STATE, IT IS NECESSARY TO LET THE USER KNOW WHY (ERROR OR R_GOQUIET)
FIG. 5c

SENDING ANSWER TOO LATE

FRANKING MACHINE

O_SPC_DEC_SET
L_SPC_HV_LIM
L_SPC_HV_DEBLOCK
R_SPC_DEC_SET
(ER_HV_TIMEOUT)

FM CONTROL

SET DECADES = R_HVLIM
DELAY > FM TIMEOUT (8s)

USER APPLICATION

HVLIMITDEBLOCK

HVLIMITDEBLOCK = E_TIMEOUT

SET DECADES = R_HVLIM
## INTERNATIONAL SEARCH REPORT

### A. CLASSIFICATION OF SUBJECT MATTER

<table>
<thead>
<tr>
<th>IPC(7)</th>
<th>US CL</th>
</tr>
</thead>
<tbody>
<tr>
<td>G06F 17/60</td>
<td>705/410</td>
</tr>
</tbody>
</table>

According to International Patent Classification (IPC) or to both national classification and IPC

### B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

| U.S. | 705/410, 401, 406, 411 |

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

### C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>US 5,200,903 A (GILHAM) 06 April 1993 (06.04.1993), See entire document</td>
<td>1-6</td>
</tr>
<tr>
<td>A</td>
<td>US 5,617,519 A (HERBERT) 01 April 1997 (01.04.1997), See entire document</td>
<td>1-6</td>
</tr>
<tr>
<td>A</td>
<td>US 5,953,712 A (HERBERT) 14 September 1999 (14.09.1999), See entire document</td>
<td>1-6</td>
</tr>
<tr>
<td>A</td>
<td>US 5,493,500 A (BOORSA) 20 February 1996 (20.02.1996), See entire document</td>
<td>1-6</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C. See patent family annex.

Date of the actual completion of the international search: 19 October 2001 (19.10.2001)

Date of mailing of the international search report: 05 DEC 2001

Name and mailing address of the ISA/US

| Commissioner of Patents and Trademarks |
| Box PCT |
| Washington, D.C. 20231 |

Facsimile No.: (703)305-3230

Authorized officer: John W. Hayes

Telephone No.: (703)305-9700

Form PCT/ISA/210 (second sheet) (July 1998)