Date: Mon, 07 Apr 1997 11:21:11 -0600 (CST)
From: Ben.R.Schneider@LAWRENCE.EDU
Subject: bib capture macro
To: Kathleen.Isaacson@LAWRENCE.EDU

```
// Version KEA! 4.10

// ------------------------------------------------------------
// Predefined variables follow. Change to suit your needs.
// ------------------------------------------------------------


Var IdleTime = 5                // Idle time after a received prompt for match t


// ----------------------------------------------
// Recorded script follows. Modify as necessary.
// ----------------------------------------------


Function Main( )
        Connect( )
        If Not IsConnected( ) Or Not GetInfo( I
nfo_PortOk )
                MsgBox( "Unable to connect to host.", Button_Ok )
                Cancel( )
        End


        TryMatch( "^M°", MaxTime=100, Idle=IdleTime )
        SendKey( "Right" )

        TryMatch( ".       ", MaxTime=100, Idle=IdleTime )
        Send( "^M" )

        TryMatch( "CALL°", MaxTime=100, Idle=IdleTime )
        SendKey( "Down" )

        TryMatch( "°", MaxTime=100, Idle=IdleTime )
        SendKey( "Down" )

        TryMatch( "°F", MaxTime=100, Idle=IdleTime )
        SendKey( "Down" )

        TryMatch( "°", MaxTime=100, Idle=IdleTime )
        SendKey( "Down" )

        Try
Match( "°", MaxTime=100, Idle=IdleTime )
        SendKey( "Down" )

        TryMatch( "to ", MaxTime=100, Idle=IdleTime )
        Send( "^M" )

        TryMatch( "to ", MaxTime=100, Idle=IdleTime )
        SendKey( "Down" )

        TryMatch( "°Print ", MaxTime=100, Idle=IdleTime )
        Send( "^M" )
        TryMatch( "°", MaxTime=100, Idle=IdleTime )
        SendKey( "Down" )

        TryMatch( " ", MaxTime=100, Idle=IdleTime )
        SendKey( "Down" )

        TryMatch( "°", MaxTime=100, Idle=IdleTime )
        Send( "^M" )

        TryMatch( "°", MaxTime=100, Idle=IdleTime )

        SendKey( "Down" )

        TryMatch( "ûáááááááááááááÇ", MaxTime=100, Idle=IdleTime )
        SendKey( "Down" )

        TryMatch( "NUMB°", MaxTime=100, Idle=IdleTime )
        Send( "^M" )

        TryMatch( "user. ", MaxTime=100, Idle=IdleTime )
        EchoSend( "ben.r.schneider@lawrence.edu" )
        Send( "^M" )
End


// ---------------------------------------------
// Predefined functions follow. Modify with care.
// ---------------------------------------------


// Sends the specified string checking for echoed characters (c
ase insensitive)
// while providing the ability to abort, retry or ignore on error.
Function EchoSend( Text )
        Var msg, result, pos, echo, uppertext
        If Len( Text ) = 0 Return End
        pos = GetCursor( )
        Send( Text )
        uppertext = UpperCase( Text )
        Begin
                When Match( "?", Pat_RegExp )
                        echo = UpperCase( Screen( pos.Page, pos.Row, pos.Col, Le
                        If echo <> uppertext
                                Continue
                        End

                When Timeout( 20 * Len( Text ) )
                        msg = "Match failed on echo while sending \""+Text+"\""

                        result = MsgBox( msg, Button_Abort+Button_Retry+Button_I
                        If result = Button_Abort
                                Cancel( )
                        Elseif result = Button_Retry
                                Continue
                        End
        End
End

// Searches for a pattern match to the left of the cursor when idle
// while providing the ability to abort, retry or ignore on error.
// The pattern may contain ^M characters which indicate the beginning
// of a line.
Function TryMatch( Pat, MaxTime, Idle )
        Function Trim( S )
                Return SubStr( S, 1, InStr( S+
"^0", " +-^0", Options=PAT_REGEXP )-1 )
        End

        Var msg, result, pos, prompt, count, count2, length, temp

        // count the number of CRs in the string skipping position 1
        count = 0
        pos = 1
        temp = 1
        while temp <> 0
                pos = pos+temp
                count = count+1
                temp = InStr( SubStr( Pat, pos ), "^M" )
        End

        Begin
                When Match( Idle=Idle )
                        If Len( Pat ) = 0
                                Return
                        End

                        // read the cursor line up to (but not including) the cu
                        pos = GetCursor( )
                        prompt = "^M"
                        I
f pos.Col > 1
                                prompt = prompt+Screen( pos.Page, pos.Row, 1, po
                        End

                        // create a string containing the cursor line from above
                        // line from the screen for each CR counted, seperated b
                        // trailing spaces removed
                        count2 = count
                        while pos.Row > 1 And count2 > 1
                                pos.Row = pos.Row-1
                                prompt = "^M"+Trim( Screen( pos.Page, pos.Row, 1
                                count2 = count2-1
                        End

                        // see if the tail end of our prompt string matches the
d pattern
                        If SubStr( prompt, -Len( Pat ) ) = Pat
                                Return
                        End
                        Continue

                When Timeout( MaxTime )
                        msg = "Match failed \""+Pat+"\""
                        If Idle > 0
                                msg = msg + " (with idle)"
                        End
                        result = MsgBox( msg, Button_Abort+Button_Retry+Button_I
                        If result = Button_Abort
                                Cancel( )
                        Elseif result = Button_Retry
```

```
                                Continue
                        End
        End
End

// Specifies the information for transferring a file or files to the host
// while providing the ability to
abort, retry or ignore on error.
Function TryTransmit( Files, Options, Path, HostName )
        Var msg, result
        Var xfer = TransferTransmit( Files, Options, Path, HostName )
        While xfer = XFER_ENDFAIL Or xfer = XFER_ENDCANCEL
                msg = "Transmit failed"
                result = MsgBox( msg, Button_Abort+Button_Retry+Button_Ignore, B
                If result = Button_Abort
                        Cancel( )
                Elseif result = Button_Ignore
                        Break
                Else
                        xfer = TransferTransmit( Files, Options, Path, HostName
                End
        End
End

// S
pecifies the information for transferring a file or files from a host
// while providing the ability to abort, retry or ignore on error.
Function TryReceive( Files, Options, Path )
        Var msg, result
        Var xfer = TransferReceive( Files, Options, Path )
        While xfer = XFER_ENDFAIL Or xfer = XFER_ENDCANCEL
                msg = "Receive failed"
                result = MsgBox( msg, Button_Abort+Button_Retry+Button_Ignore, B
                If result = Button_Abort
                        Cancel( )
                Elseif result = Button_Ignore
                        Break
                Else
                        x
fer = TransferReceive( Files, Options, Path )
                End
        End
End

// Specifies the commands to send to a Kermit server
// while providing the ability to abort, retry or ignore on error.
Function TryCommand( Command, Options )
        Var msg, result
        Var file = TransferCommand( Command, Options )
        While FileRead( file ) = "TransferCommand Failed"
                msg = "Command failed"
                result = MsgBox( msg, Button_Abort+Button_Retry+Button_Ignore, B
                If result = Button_Abort
                        Cancel( )
                Elseif resul
t = Button_Ignore
                        Break
                Else
                        FileClose( file )
                        file = TransferCommand( Command, Options )
                End
        End
        FileSetPos( file, 0 )
        Return file
End


// ------------------------
// Run the recorded script.
// ------------------------


Main( )
```