Table of Contents

1 Snappy Fax Automation Server

Snappy Fax Automation Server supplies a set of properties and methods that can be called from your program to control snappy fax as an automation server much as you would control Microsoft Word or Excel through their object model.

The following is a description of the object model that is exposed by snappy fax automation server.

We have provided in some sections code examples in C#, Delphi and VB. Code samples have not been provided for all properties and procedures, since there would be a lot of duplication and the declarations and usages are very similar.

Note: Snappy fax automation is provided in the snappy fax desktop/client software. No automation methods are exposed in the snappy fax server software since all needed functionality can be accomplished within the desktop/client automation model, regardless of whether snappy fax is configured as a desktop application or a client to the fax server.

Registering the Automation Server

Before you can use the Automation server's object model in your program you must register it with Windows using regserver. After installing the Automation Server edition, start a command prompt running as administrator:

1. Click Start button and type cmd in search box
2. When cmd.exe appears in the list above, right click and select 'Run As Administrator'

In the command box, navigate to the program files folder where it was installed and type: sf5.exe /regserver

If the registration is successful, there will be no message

2 Object Model

2.1 Addcontact

Purpose: Add a contact to the address book indicated by the Address book name parameter.
Return value: boolean
Parameters:

Address book name, string
Contact Full Name, string
Fax Number, string
Contact Full Address, string
Group, string
Email, string

2.1.1 C# Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated with Snappy = CreateObject("SF5.SnappyFaxIntf")

```
bool AddAddressBookItem(string AName, string AContactName, string
AFaxNumber, string AAddress, string AGroup)
{
   if (Snappy.AddContact(AName, AContactName, AFaxNumber, AAddress,
AGroup))
   {
      MessageBox.Show(string.Format("A new contact: {0} has been added to
the {1} Address Book", AContactName, AName));
      return true;
   }
   else
   {
      MessageBox.Show("Add Address Book Item Failed");
      return false;
   }
}
```

2.1.2 Delphi Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated
Snappy = CreateOleObject("SF5.SnappyFaxIntf")

```
function AddAddressBookItem(const
AName,AContactName,AFaxNumber,AAddress,AGroup : String): WordBool;
begin
        if
Snappy.AddContact(AName,AContactName,AFaxNumber,AAddress,AGroup)
then
```

```
        MessageBox(Format('A new contact: %s has been added to the %s
Address Book,[AContactName,AName]))
      else
        MessageBox('Add Address Book Item Failed');
end;
```

2.1.3 VB Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated (Snappy = CreateObject("SF5.SnappyFaxIntf")**)**

```
Function AddAddressBookItem(ByVal AName As String, ByVal AContactName
As String, ByVal AFaxNumber As String, ByVal AAddress As String, ByVal
AGroup As String) As Boolean
  If Snappy.AddContact(AName, AContactName, AFaxNumber, AAddress,
AGroup) Then
    MessageBox.Show(String.Format("A new contact: {0} has been added to
the {1} Address Book", AContactName, AName))
    Return True
  Else
    MessageBox.Show("Add Address Book Item Failed")
    Return False
  End If
End Function
```

2.2 Connect

Purpose: Provide your application with a connection to Snappy Fax Automation Server
Return value: None
Parameters:
HWnd : integer - Your Application window handle

Use the Connect procedure to establish a connection with Snappy Fax in situations where you are interested in receiving status messages but don't intend to necessarily process any new faxes. Snappy Fax will use your Windows handle to Post messages when status conditions of previously processed faxes changes. This is most useful if Snappy Fax is configured as a client to Snappy Fax Server or it is configured to use either FaxAge or SRFax internet fax service. You should call this procedure immediately after creating the Snappy Fax Application object.

2.3 DeleteContact

Purpose: Delete a contact from the address book indicated by the Address book name parameter.
Return value: boolean

Parameters:
Address book name, string
Key, integer

2.3.1 C# Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated (Snappy = CreateObject("SF5.SnappyFaxIntf")**)**

```
void DeleteContact(string ABName, int AContactNum)
{
   if (Snappy.DeleteContact(ABName,AContactNum))
   {
      MessageBox.Show(string.Format("Contact Number: {0} was
deleted from the {1} Address Book", AContactNum, ABName));
   }
}
```

2.3.2 Delphi Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated with Snappy := CreateOleObject("SF5.SnappyFaxIntf")

```
procedure DeleteContact(const ABName : String;AContactNum :
integer);
begin
       if Snappy.DeleteContact(ABName,AContactNum) then
         ShowMessage(Format('Contact Number: %d was deleted from
the %s Address Book',[AContactNum,ABName]));
end;
```

2.3.3 VB Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated with Snappy = CreateObject("SF5.SnappyFaxIntf")

```
Sub DeleteContact(ByVal ABName As String, ByVal AContactNum As
Integer)
```

```
    If Snappy.DeleteContact(ABName,AContactNum) Then
        MessageBox.Show(String.Format("Contact Number: {0} was
deleted from the {1} Address Book", AContactNum, ABName))
    End If
End Sub
```

2.4 ConvertFaxImage

Purpose: You can use this function to convert an Input file to a file (outputfile) in either .tif or .pdf format
Return value: boolean
Parameters:
InputFile, String
OutputFile, String

Note: The inputfile can be one of the following file types: Bmp,Tiff,Jpg,Pdf,Png,Dcx,Pcx
The outputfile can only be .tiff (.tif) or .pdf

Note: if the inputfile and outputfile are of the same filetype a simple copy is performed.

Possible Exceptions that may be raised:

Invalid Input File type
Invalid Output File type
Filenames must include fully qualified path

2.4.1 C# Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated (Snappy = CreateObject("SF5.SnappyFaxIntf")**)**

```
bool ConvertFaxImage(string SourceFile, string DestFile)
{
    bool result = Snappy.ConvertFaxImage(SourceFile, DestFile);
    if (result)
    {
        MessageBox.Show(string.Format("{0} File was created", DestFile));
    }
    else
    {
        MessageBox.Show("Conversion Failed");
    }
```

```
    return result;
}
```

2.4.2 Delphi Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated with Snappy := CreateOleObject("SF5.SnappyFaxIntf")

```
function ConvertFaxImage(const SourceFile, DestFile : String) :
boolean;
begin
        result := Snappy.ConvertFaxImage(SourceFile,DestFile);
        if result then
          ShowMessage(Format('%s File was created',[DestFile]))
        else
          ShowMessage('Conversion Failed');
end;
```

2.4.3 VB Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated (Snappy = CreateObject("SF5.SnappyFaxIntf")**)**.

```
Function ConvertFaxImage(ByVal SourceFile As String, ByVal DestFile
As String) As Boolean
    Dim result As Boolean = Snappy.ConvertFaxImage(SourceFile,
DestFile)
    If result Then
        MessageBox.Show(String.Format("{0} File was created", DestFile))
    Else
        MessageBox.Show("Conversion Failed")
    End If
    Return result
End Function
```

2.5 DeleteInboxRecord

Purpose: Delete a record from the Incoming Fax Table.
Return value: boolean
Parameters:

Key, integer

2.5.1 C# Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated (Snappy = CreateObject("SF5.SnappyFaxIntf")**)**

```
void DeleteRecord()
{
   int Key = int.Parse(EditBox.Text);
   if (Snappy.DeleteInboxRecord(Key))
   {
      MessageBox.Show($"Inbox Key: {Key} was deleted");
   }
   else
   {
      MessageBox.Show("Delete Record failed");
   }
}
```

2.5.2 Delphi Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated with Snappy := CreateOleObject("SF5.SnappyFaxIntf")

```
TForm1.DeleteRecord;
var
   Key : integer;
begin
   Key := StrToInt(EditBox.Text);
   if Snappy.DeleteInboxRecord(Key) then
     ShowMessage(Format('Inbox Key: %d was deleted',[Key]))
   else
     ShowMessage('Delete Record failed');
end;
```

2.5.3 VB Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated (Snappy = CreateObject("SF5.SnappyFaxIntf")**)**

```
Sub DeleteRecord()
   Dim Key As Integer
   Key = Integer.Parse(EditBox.Text)
   If Snappy.DeleteInboxRecord(Key) Then
      MessageBox.Show(String.Format("Inbox Key: {0} was deleted",
Key))
   Else
      MessageBox.Show("Delete Record failed")
   End If
End Sub
```

2.6 DeleteOutboxRecord

Purpose: Delete a record from the Outgoing Fax Table.
Return value: boolean
Parameters:
Key, integer

Sample Usage (Visual Basic):

```
Dim Key as long

Snappy = CreateObject("SF5.SnappyFaxIntf")
Key = 1023
if Snappy.DeleteOutboxRecord(Key) then
  'success.
```

2.6.1 C# Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated (Snappy = CreateObject("SF5.SnappyFaxIntf")**)**

```
void DeleteRecord()
{
   int Key = int.Parse(EditBox.Text);
   if (Snappy.DeleteOutboxRecord(Key))
```

```
    {
        MessageBox.Show($"Inbox Key: {Key} was deleted");
    }
    else
    {
        MessageBox.Show("Delete Record failed");
    }
}
```

2.6.2 Delphi Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated with Snappy := CreateOleObject("SF5.SnappyFaxIntf")

```
TForm1.DeleteRecord;
var
    Key : integer;
begin
    Key := StrToInt(EditBox.Text);
    if Snappy.DeleteOutboxRecord(Key) then
      ShowMessage(Format('Inbox Key: %d was deleted',[Key]))
    else
      ShowMessage('Delete Record failed');
end;
```

2.6.3 VB Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated (Snappy = CreateObject("SF5.SnappyFaxIntf")**)**

```
Sub DeleteRecord()
    Dim Key As Integer
    Key = Integer.Parse(EditBox.Text)
    If Snappy.DeleteOutboxRecord(Key) Then
        MessageBox.Show(String.Format("Inbox Key: {0} was deleted",
Key))
    Else
        MessageBox.Show("Delete Record failed")
    End If
```

```
End Sub
```

2.7 DeleteAnnotationRecordByKey

Purpose: To delete a record in Snappy Fax's Automation History Table
Return Value: boolean
Parameters:
  Key : integer;

Key is the key value of the record you want to delete.

Note: The AutoJobId is NOT the key to this data table. The key is an autoincrement integer value.

2.8 DeleteAutomationRecordByJobID

Purpose: To delete a record in Snappy Fax's Automation History Table
Return Value: boolean
Parameters:
  JobId : integer;

JobId is the Auto job id you have assigned

2.9 AddressBookNames

Purpose: Obtain a list of address book names. The return value is a string with address book names separated by semi-colons
Return value: string
Parameters: None

2.9.1 C# Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated
(Snappy = CreateObject("SF5.SnappyFaxIntf")**)**

```
private void btnGetNamesClick(object sender, EventArgs e)
{
   List<string> T = new List<string>();
   string ABNames;

   try
```

```
    {
        ABNames = Snappy.AddressBookNames;
        T.AddRange(ABNames.Split(','));
        ListBox1.DataSource = T;
    }
    finally
    {
        T.Clear();
    }
}
```

2.9.2 Delphi Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated with Snappy := CreateOleObject("SF5.SnappyFaxIntf")

```
TForm1.btnGetNamesClick(Sender : TObject);
var
  T : TStringList;
  ABNames : String;
begin
   T := TStringList.Create;
   try
     ABNames := Snappy.AddressBookNames;
     T.CommaText := ABNames;
     ListBox1.Assign(T);
   finally
     T.Free;
   end;
end;
```

2.9.3 VB Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated (Snappy = CreateObject("SF5.SnappyFaxIntf")**)**

```
Private Sub btnGetNamesClick(ByVal Sender As Object, ByVal e As
EventArgs) Handles btnGetNames.Click
    Dim T As New List(Of String)()
```

```
    Dim ABNames As String

    Try
       ABNames = Snappy.AddressBookNames
       T.AddRange(ABNames.Split(","c))
       ListBox1.DataSource = T
    Finally
       T.Clear()
    End Try
End Sub
```

2.10 AddressBookPathOf

Purpose: Obtain the data folder location of the address book indicated by the string parameter
Return value: string
Parameters:

Address book name, string [in]

2.10.1 C# Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated (Snappy = CreateObject("SF5.SnappyFaxIntf")**)**

```
string GetDefaultAddressBookPath()
{
   return Snappy.AddressBookPathOf("Default");
}
```

2.10.2 Delphi Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated with Snappy := CreateOleObject("SF5.SnappyFaxIntf")

```
function GetDefaultAddressBookPath : String;
begin
      result := Snappy.AddressBookPathOf('Default');
end;
```

2.10.3 VB Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated (Snappy = CreateObject("SF5.SnappyFaxIntf")**)**

```
Function GetDefaultAddressBookPath() As String
   Return Snappy.AddressBookPathOf("Default")
End Function
```

2.11 AddressBookRecs

Purpose: Obtain the number of records in address book data table for the named address book indicated by the string parameter
Return value: long integer
Parameters: Address Book Name, string

Note: Snappy fax can have multiple address books, if the address book name is blank (blank string as parameter), the Default address book will be assumed.

2.11.1 C# Example:

```
private void btnGetRecs(object sender, EventArgs e)
{
   string ABName;
   object Snappy = null;
   bool Connected = false;

   try
   {
      Snappy = GetActiveObject("SF5.snappyFaxIntf");
      Connected = true;
   }
   catch (Exception ex1)
   {
      try
      {
         Snappy = CreateObject("SF5.snappyFaxIntf");
         Connected = true;
      }
      catch (Exception ex2)
      {
         MessageBox.Show("Could not bind to OleObject");
```

```
        }
    }

    if (Connected)
    {
        ABName = "Default";
        this.FNumABRecords = Snappy.AddressBookRecs(ABName);
        Snappy = null;
    }
}
```

2.11.2 Delphi Example:

```
TForm1.btnGetRecs(Sender : TObject);
var
  ABName : String;
  Snappy : OleVariant;
  Connected : boolean;
begin
        Connected := false;
        try
          Snappy := := GetActiveOleObject('SF5.snappyFaxIntf');
          Connected := true;
        except
          On E: Exception do
          begin
            try
              Snappy := CreateOleObject('SF5.snappyFaxIntf');
              Connected := true;
            except
              on E: Exception do
              begin
                ShowMessage('Could not bind to OleObject');
              end;
            end;
          end;
```

```
        end;
        if Connected then
        begin
            ABName := 'Default';
            Self.FNumABRecords :=
Snappy.AddressBookRecs(ABName);
            Snappy := Unassigned;
        end;
end;
```

2.11.3 VB Example:

```
Private Sub btnGetRecs(ByVal Sender As Object, ByVal e As
EventArgs) Handles btnGetRecs.Click
   Dim ABName As String
   Dim Snappy As Object = Nothing
   Dim Connected As Boolean = False

   Try
      Snappy = GetActiveObject("SF5.snappyFaxIntf")
      Connected = True
   Catch ex1 As Exception
      Try
         Snappy = CreateObject("SF5.snappyFaxIntf")
         Connected = True
      Catch ex2 As Exception
         MessageBox.Show("Could not bind to OleObject")
      End Try
   End Try

   If Connected Then
      ABName = "Default"
      Me.FNumABRecords = Snappy.AddressBookRecs(ABName)
      Snappy = Nothing
   End If
End Sub
```

2.12 AppMsgNumber

Purpose: Set the Windows message number your application will use to process status messages from snappy fax

Return value : None
Parameters: None

Sample Usage (Visual Basic):

```
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.AppWindow = Me.Handle
Snappy.AppMsgNumber = WM_APP + 1000
```

2.13 AppWindow

Purpose: Set the Windows handle of your application window to receive status messages about a fax job
Return value : None
Parameters: None

Sample Usage (Visual Basic):

```
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.AppWindow = Me.Handle
```

2.14 AutoJobId

Purpose: Set a job id of your choice to be used to track progress of fax job
Return value : None
Parameters: None

You will need to ensure that the AutoJobId is unique for each fax you send, otherwise the results may not be reliable when obtaining job status

See Also:

GetOutboxFaxDetailByAutoJobID

Sample Usage (Visual Basic):

```
  Dim Snappy as Object
  Dim JOB as integer
  Snappy = CreateObject("SF5.SnappyFaxIntf")
  Snappy.AppWindow = Me.Handle
  Snappy.AppMsgNumber = WM_APP + 1000
  JOB = 1
  Snappy.AutoJobId = JOB
{prepare send}
 Snappy.SendFax
```

2.15 AutomationRecs

Purpose: To obtain a count of the number of records in Snappy Fax's Automation History Table (Automation Queue)
Return value : integer
Parameters: None

See Also: GetAutomationCountOfItemsNotProcessed

2.16 Billcode

Purpose: Set text to display in snappy fax's billing code field in the outbox data table
Return value : None
Parameters: None

Sample Usage (Visual Basic):

```
  Dim Snappy as Object
  Dim JOB as integer
  Snappy = CreateObject("SF5.SnappyFaxIntf")
  Snappy.BillCode = "B10929"
```

2.17 ConfigurationType

Purpose: Obtain the type of configuration of snappy fax
Return value : long integer (read only)
Parameters: None

return values:

0 = configured as desktop only
1 = configured as client to fax server

Sample Usage (Visual Basic):

```
Dim config as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
config = Snappy.ConfigurationType
if config = 1 then
  MessageBox("Configured as client to server")
else
  MessageBox("Configured as desktop")
end if
```

2.18 FaxNumber

Purpose: Set this property to fax number of the intended recipient
Return value: None
Parameters: None

Sample Usage (Visual Basic):

```
Dim FaxNumber as string
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.FaxNumber = "18009871626"
```

You must specify this property, if not the exception 'No Fax Number Specified' will be raised when attempting to start the transmission.

Note: The fax number should always be specified exactly as it should be dialed. Snappy fax does not use dialing rules.

2.19 CoverAction_Comment

Purpose: Set to true to have snappy fax check the 'Comment' cover page action item
Return value : None
Parameters: None

Sample Usage (Visual Basic):

```
Dim Snappy as Object
Dim JOB as integer
Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.UseCoverPage = true
```

```
Snappy.FileToFax = "ThursdayMenu.tif"
Snappy.CoverPageTemplate = "thursdaymenu.fct"
Snappy.CoverPageMemo = "This is the Bistro's menu for Thursday, Oct. 31"
Snappy.CoverPageSubject = "New Jersey Bistro Menu"
Snappy.CoverAction_Comment = true
```

2.20 CoverAction_Recycle

Purpose: Set to true to have snappy fax check the 'Recycle' cover page action item
Return value : None
Parameters: None

Sample Usage (Visual Basic):

```
Dim Snappy as Object
Dim JOB as integer
Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.UseCoverPage = true
Snappy.FileToFax = "ThursdayMenu.tif"
Snappy.CoverPageTemplate = "thursdaymenu.fct"
Snappy.CoverPageMemo = "This is the Bistro's menu for Thursday, Oct. 31"
Snappy.CoverPageSubject = "New Jersey Bistro Menu"
Snappy.CoverAction_Recycle = true
```

2.21 CoverAction_Reply

Purpose: Set to true to have snappy fax check the 'Reply' cover page action item
Return value : None
Parameters: None

Sample Usage (Visual Basic):

```
Dim Snappy as Object
Dim JOB as integer
Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.UseCoverPage = true
Snappy.FileToFax = "ThursdayMenu.tif"
Snappy.CoverPageTemplate = "thursdaymenu.fct"
Snappy.CoverPageMemo = "This is the Bistro's menu for Thursday, Oct. 31"
Snappy.CoverPageSubject = "New Jersey Bistro Menu"
Snappy.CoverAction_Reply = true
```

2.22 CoverAction_Review

Purpose: Set to true to have snappy fax check the 'Review' cover page action item
Return value : None
Parameters: None

Sample Usage (Visual Basic):

```
Dim Snappy as Object
Dim JOB as integer
Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.UseCoverPage = true
Snappy.FileToFax = "ThursdayMenu.tif"
Snappy.CoverPageTemplate = "thursdaymenu.fct"
Snappy.CoverPageMemo = "This is the Bistro's menu for Thursday, Oct. 31"
Snappy.CoverPageSubject = "New Jersey Bistro Menu"
Snappy.CoverAction_Review = true
```

2.23 CoverAction_Urgent

Purpose: Set to true to have snappy fax check the 'Urgent' cover page action item
Return value : None
Parameters: None

Sample Usage (Visual Basic):

```
Dim Snappy as Object
Dim JOB as integer
Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.UseCoverPage = true
Snappy.FileToFax = "ThursdayMenu.tif"
Snappy.CoverPageTemplate = "thursdaymenu.fct"
Snappy.CoverPageMemo = "This is the Bistro's menu for Thursday, Oct. 31"
Snappy.CoverPageSubject = "New Jersey Bistro Menu"
Snappy.CoverAction_Urgent = true
```

2.24 CoverOnlyFax

Purpose: Set to true if the fax will consist of a cover page only
Return value : None
Parameters: None

Sample Usage (Visual Basic):

```
Dim Snappy as Object
Dim JOB as integer
Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.UseCoverPage = true
Snappy.CoverOnlyFax = true
Snappy.CoverPageTemplate = "mytemplate.fct"
Snappy.CoverPageMemo = "This is the Bistro's menu for Thursday, Oct. 31"
Snappy.CoverPageSubject = "New Jersey Bistro Menu"
```

2.25 CoverPageMemo

Purpose: Set to the desired text to appear in the memo area of the cover page
Return value : None
Parameters: None

Sample Usage (Visual Basic):

```
Dim Snappy as Object
Dim JOB as integer
Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.UseCoverPage = true
Snappy.FileToFax = "ThursdayMenu.tif"
Snappy.CoverPageTemplate = "thursdaymenu.fct"
Snappy.CoverPageMemo = "This is the Bistro's menu for Thursday, Oct. 31"
Snappy.CoverPageSubject = "New Jersey Bistro Menu"
```

2.26 CoverPageSubject

Purpose: Set to the desired text to appear on the subject line of the cover page
Return value : None
Parameters: None

Sample Usage (Visual Basic):

```
Dim Snappy as Object
Dim JOB as integer
```

```
Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.UseCoverPage = true
Snappy.FileToFax = "ThursdayMenu.tif"
Snappy.CoverPageTemplate = "thursdaymenu.fct"
Snappy.CoverPageMemo = "This is the Bistro's menu for Thursday, Oct. 31"
Snappy.CoverPageSubject = "New Jersey Bistro Menu"
```

2.27 CoverPageTemplate

Purpose: Set to the file name of the desired cover page template
Return value : None
Parameters: None

Sample Usage (Visual Basic):

```
Dim Snappy as Object
Dim JOB as integer
Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.UseCoverPage = true
Snappy.FileToFax = "ThursdayMenu.tif"
Snappy.CoverPageTemplate = "thursdaymenu.fct"
Snappy.CoverPageMemo = "This is the Bistro's menu for Thursday, Oct. 31"
Snappy.CoverPageSubject = "New Jersey Bistro Menu"
```

Note: if CoverPageTemplate is left blank then snappy fax will use the default cover page template defined in its settings. If left blank and no default template is defined in its settings, the exception "No Cover Page Template defined" will be raised. If the template designated here does not exist in snappy fax's cover page folder the exception 'cover page template does not exist' will be raised.

2.28 FaxRecipient

Purpose: Set this property to name of the fax recipient
Return value: None
Parameters: None

Sample Usage (Visual Basic):

```
Dim FaxRecipient as string
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
```

```
Snappy.FaxRecipient = "John Doe"
```

You must specify a recipient name, if this property is not specified, the exception 'No Fax Recipient Name Specified' will be raised then attempting to start the fax transmission

2.29 FaxRouting

Purpose: Set this property to indicate the fax routing of the fax file to be transmitted
Return value: None
Parameters: None

Sample Usage (Visual Basic):

```
Dim Faxrouting as string
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.FaxRouting = "R"
```

When starting a fax job via automation, set the FaxRouting to either "R", "S" or "I".

Fax routing of "R" indicates send the fax normally (Regular fax) via modem. Fax routing of "S" is only valid if snappy fax is configured as a client to the fax server software.

If you set this value to "S" and snappy fax is not configured as a client to the fax server, the exception 'Server Routing not supported in configuration' will be raised.

If you set this value to "I" and snappy fax is not configured to use either FaxAge or SRFax internet fax server then the exception 'Invalid Fax routing Method specified' will be raised

If you set this value to anything other than "S" , "R" or "I" then the exception 'Invalid Fax routing Method specified' will be raised

See Also:

IsIFaxEnabled
ConfigurationType

2.30 SendASAP

Purpose: Set this property to indicate the whether the fax should be sent as soon as possible
Return value: None
Parameters: None

Note this property should only be used when the Faxrouting is set to 'S'.

Sample Usage (Visual Basic):

```
Dim Faxrouting as string
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.FaxRouting = "S"
Snappy.SendASAP = true
```

If nothing is specified for this property, true will be assumed

2.31 SendDate

Purpose: Set this property to indicate the target date when a fax job is not to be sent immediately
Return value: None
Parameters: None

Note this property should only be used when the Faxrouting is set to 'S' and the SendASAP property has been set to false

Sample Usage (Visual Basic):

```
Dim Faxrouting as string
Dim DateToSend as Date
Dim TimeToSend as Date
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.FaxRouting = "S"
Snappy.SendASAP = false
DateToSend = #4/30/2015#
TimeToSend = #5:00:00 PM#
Snappy.SendDate = DateToSend
Snappy.Sendtime = TimeToSend
```

If you set SendASAP to false and do not set the SendDate *and* SendTime properties, SendASAP will revert to true

2.32 SendTime

Purpose: Set this property to indicate the target date when a fax job is not to be sent immediately
Return value: None
Parameters: None

Note this property should only be used when the Faxrouting is set to 'S' and the SendASAP property has been set to false

Sample Usage (Visual Basic):

```
Dim Faxrouting as string
Dim DateToSend as Date
Dim TimeToSend as Date
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.FaxRouting = "S"
Snappy.SendASAP = false
DateToSend = #4/30/2015#
TimeToSend = #5:00:00 PM#
Snappy.SendDate = DateToSend
Snappy.Sendtime = TimeToSend
```

If you set SendASAP to false and do not set the SendDate *and* SendTime properties, SendASAP will revert to true

2.33 FileToFax

Purpose: Set this property to indicate the file that is to be transmitted
Return value: string (read, write)
Parameters: None

Sample Usage (Visual Basic):

```
Dim FaxFile as string
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.FileToFax = "c:\FaxImage.tif"
```

When starting a fax job via automation, set the FileToFax to the file you want to transmit.

Supported file types:

.txt (text files)
.tif, .tiff (tif files)
.jpg, .jpeg (jpg files)
.bmp (bitmap files)
.pdf (pdf files)
.doc, .docx (Microsoft Word Files) Note: Word must be installed on system

.xls, .xlsx (Microsoft Excel Files) Note: Excel must be installed on system
.odt (Open Office document files) Note: Open Office must be installed on system
.ods (Open Office spreadsheet files) Note: Open Office must be installed on system

Note: Attempting to set the FileToFax to any other file type will result in an exception 'File Type is not Supported for Fax Automation'. If the file does not exist the exception 'File Specified does not exist' will be raised.

2.34 FirstAddressBookKey

Purpose: Obtain the Key of the first (oldest) record in the address book indicated by the string parameter
Return value: long integer
Parameters: Address book name, string

Sample Usage (Visual Basic):

```
Dim Key as integer
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Key = Snappy.FirstAddressBookKey("Default")
```

Note: if the address book name is passed as a blank string, the Default address book will be assumed

2.35 FirstInBoxKey

Purpose: Obtain the key of the first (oldest) record in the inbox data table.
Return value: long integer
Parameters: None

Sample Usage (Visual Basic):

```
Dim FirstInKey as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
FirstInKey = Snappy.FirstInboxKey
```

2.36 FirstOutBoxKey

Purpose: Obtain the key of the first (oldest) record in the outbox data table.
Return value: long integer

Parameters: None

Sample Usage (Visual Basic):

```
Dim FirstOutKey as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
FirstOutKey = Snappy.FirstOutboxKey
```

2.37 FirstAutomationKey

Purpose: Obtain the key of the first (oldest) record in Snappy Fax's Automation History Table
Return value: long integer
Parameters: None

Sample Usage (Visual Basic):

```
Dim FirstAutoKey as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
FirstAutoKey = Snappy.FirstAutomationKey
```

Note: The return value is the KEY of the last record NOT the AutoJobId

2.38 FirstServerInBoxKey

Purpose: Obtain the key of the first (oldest) record in the fax Server's inbox data table.
Return value: long integer
Parameters: None

Sample Usage (Visual Basic):

```
Dim FirstInKey as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
FirstInKey = Snappy.FirstServerInboxKey
```

Note: Only applies if snappy fax is configured as a client to snappy fax server, otherwise the exception 'Client not configured for fax server' will be raised

2.39 FirstServerJobKey

Purpose: Obtain the key of the first (oldest) record in the fax Server's pending jobs data table.
Return value: long integer
Parameters: None

Sample Usage (Visual Basic):

```
Dim FirstKey as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
FirstKey = Snappy.FirstServerJobKey
```

Note: Only applies if snappy fax is configured as a client to snappy fax server, otherwise the exception 'Client not configured for fax server' will be raised

2.40 FirstServerOutBoxKey

Purpose: Obtain the key of the first (oldest) record in the fax Server's outbox data table.
Return value: long integer
Parameters: None

Sample Usage (Visual Basic):

```
Dim FirstKey as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
FirstKey = Snappy.FirstServerOutboxKey
```

Note: Only applies if snappy fax is configured as a client to snappy fax server, otherwise the exception 'Client not configured for fax server' will be raised

2.41 GetAutomationCountOfItemsNotProcessed

Purpose: Obtain the count of the number of the items in Automation Queue that have not been processed
Return value: long integer
Parameters: None

Sample Usage (Visual Basic):

```
Dim Count as long
```

```
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Count = Snappy.GetAutomationCountOfItemsNotProcessed
```

Note: When you use the SendFax procedure of Snappy Fax, it deposits the record in the Automation History Table and the items are then processed in succession. This function can be useful to see how many items remain to be processed.

2.42 GetAutomationRecordDetailsByAutoJobID

Purpose: Retrieve detail information of a record in the Automation History Table
Return value: True if record located, false if not
Parameters:

- AutoJobID : integer [in]
- Recipient : string [out]
- FaxNumber : string [out]
- Routing : string [out]
- FaxResult: string [out]
- FileName : string [out]
- OutboxKey : long integer [out] Note: this will be -1 if there is no representative record in the Outbox table, which will be the case for an item that has not yet been processed
- DateTimeSent : Date [out] The date and time the item was sent or 0 if it has not been sent
- ErrorMsg : String [out] If the function result is false, examine ErrorMsg to determine why the record could not be retrieved

AutoJobID is the Job id of the item you are interested in

Sample Usage (Visual Basic):

```
Dim AutoJobID as long
Dim Recipient as String
Dim FaxNumber as String
Dim Routing As String
Dim FaxResult as String
Dim FileName as String;
Dim OutboxKey as long integer
Dim DateTimeSent as Date
Dim ErrMsg as String
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Sub MyGetAutomationRecordDetails(AutoJobID As Integer, ByRef Recipient As String,
ByRef FaxNumber As String, ByRef Routing As String,ByRef FaxResult As String, ByRef
FileName As String
```

```
                          ByRef Outboxkey as integer, ByRef DateTimeSent As Integer,
ByRef ErrMsg As String)
        AutoJobID = 10988
        if
Snappy.GetAutomationRecordDetails(AutoJobID,Recipient,FaxNumber,Routing,FaxResult,File
Name,Outboxkey,DateTimeSent,ErrMsg) then
           'success
        else
           'ErrMsg will contain information like 'Record not found'
End Sub
```

2.43 GetAutomationRecordByKey

Purpose: Retrieve detail information of a record in the Automation History Table
Return value: True if record located, false if not
Parameters:

Key: integer [in]
AutoJobID: integer [out]
Recipient: string [out]
FileName: string [out]
FaxNumber: string [out]
FaxResult: string [out]
Routing: string [out]
TimeStamp: Date [out]
OutBoxKey: integer [out]
Complete: boolean [out]
NextKey: integer [out]

Note: the Automation History Table's key is an auto-incremented integer, NOT the AutoJobId. This function takes that index as the key and returns with the indicated information.

2.43.1 C# Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated with Snappy = CreateObject("SF5.SnappyFaxIntf")

```
void CheckAutomationRecord()
{
   int Key = 1909;
   int AutoJobID;
   string Recipient;
   string FileName;
```

```
    string FaxNumber;
    string FaxResult;
    string Routing;
    DateTime TimeStamp;
    int OutBoxKey;
    bool Complete;
    int NextKey;

    if (!((OleObject)Snappy).GetAutomationRecordByKey(Key, out
AutoJobID, out Recipient, out FileName, out FaxNumber, out FaxResult,
out Routing, out TimeStamp, out OutBoxKey, out Complete, out
NextKey))
    {
        MessageBox.Show(string.Format("Record Key: {0} not found",
Key));
    }
    else
    {
        MessageBox.Show(string.Format("Record with key: {0} found.
Recipient Name was: {1}", Key, Recipient));
    }
}
```

2.43.2 Delphi Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated with Snappy := CreateOleObject("SF5.SnappyFaxIntf")

```
procedure CheckAutomationRecord;
var
        Key : integer;
        AutoJobID : integer;
        Recipient : String;
        FileName : String;
        FaxNumber : String;
        FaxResult : String;
        Routing : String;
        TimeStamp : TDateTime;
        OutBoxKey : integer;
        Complete : WordBool;
```

```
        NextKey : integer;
begin
        Key := 1909;
        if not
Snappy.GetAutomationRecordByKey(Key,AutoJobID,Recipient,FileNa
me,FaxNumber,FaxResult,Routing,TimeStamp,OutBoxKey,Complete,N
extKey) then
          ShowMessage(Format('Record Key: %d not found',[Key]))
        else
           ShowMessage(Format('Record with key: %d found.  Recipient
Name was: %s',[Key,Recipient]));
end;
```

2.43.3 VB Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated with Snappy = CreateObject("SF5.SnappyFaxIntf")

```
Sub CheckAutomationRecord()
   Dim Key As Integer
   Dim AutoJobID As Integer
   Dim Recipient As String
   Dim FileName As String
   Dim FaxNumber As String
   Dim FaxResult As String
   Dim Routing As String
   Dim TimeStamp As Date
   Dim OutBoxKey As Integer
   Dim Complete As Boolean
   Dim NextKey As Integer

   Key = 1909
   If Not Snappy.GetAutomationRecordByKey Key, ByRef AutoJobID,
ByRef Recipient, ByRef FileName, ByRef FaxNumber, ByRef
FaxResult, ByRef Routing, ByRef TimeStamp, ByRef OutBoxKey,
ByRef Complete, ByRef NextKey) Then
      MessageBox.Show(String.Format("Record Key: {0} not found",
Key))
   Else
```

```
        MessageBox.Show(String.Format("Record with key: {0} found.
Recipient Name was: {1}", Key, Recipient))
    End If
End Sub
```

2.44 GetAddressBookDetails

Purpose: Retrieve detail information of a record in the outbox data table
Return value: None
Parameters:

- ABname : string [in]
- Key : integer [in]
- sFirstName : string [out]
- sLastName : string [out]
- sFullName : string [out]
- sFaxNumber : string [out]
- sAddress : string [out]
- sEmail : string [out]
- NextKey : long integer [out]

Abname is the name of the desired address book, if left blank the default address book is assumed. Key is the key of the record in the address book table you are interested in.

Sample Usage (Visual Basic):

Refer to the example for GetInboxFaxDetails

2.45 GetInboxFax

Purpose: Retrieve the fax image of a record in the inbox data table and save to a .tif file or a .pdf file
Return value: boolean (true if successful)
Parameters:

- Key : integer
- sFile : string

Key is the key of the record in the inbox data table you want to retrieve the fax image for sFile is the desired file name to save the fax image to. The file extension must either be .tif or .pdf, if a file with another file extension is passed in this parameter an exception will be raised.

Sample Usage (Visual Basic):

```
Dim Key as long
Dim sFile as string
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Key = 1909
sFile = "c:\MyTifFile.tif"
if Snappy.GetInboxFax(Key,sFile) then
  'do something with image file
```

2.46 GetCoverPageTemplateNames

Purpose: Obtain a list of cover page template file names. The return value is a string with the names separated by semi-colons
Return value: string
Parameters: None

Sample Usage (Visual Basic):

```
Dim TemplateNames() as string
Dim TemplateName as string
Dim i as integer
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")

TemplaateName = Snappy.GetCoverPageTemplateNames
TemplateNames = Split(TemplateName,";")
for i = 0 to TemplateNames.Length - 1
   cbDropDownListBox.Items.Add(TemplateNames(i))
next
```

Note: Above example shows how to populate a drop down list box with cover page template names

2.47 GetDefaultCoverPageTemplateName

Purpose: Obtain the default cover page template file name.
Return value: string
Parameters: None

Sample Usage (Visual Basic):

```
Dim TemplateName as string
Dim i as integer
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")

TemplateName = Snappy.GetDefaultCoverPageTemplateName
```

2.48 GetInboxFaxDetails

Purpose: Retrieve detail information of a record in the inbox data table
Return value: None
Parameters:

- Key : integer [in]
- Pages : integer [in]
- Date : Date [out]
- sFax : string [out]
- sFaxResult : string [out]
- NextKey : long integer [out]

Key is the key of the record in the inbox data table you want to retrieve the details for
The Number of pages in the fax, the date /time of the fax, the fax number of the caller, the result of the fax and the key of the next record in the inbox data table are returned in the Pages, Date, sFax, sFaxResult and NextKey parameters.

Sample Usage (Visual Basic):

Here is a detailed example showing how to iterate through all inbox records and populate a grid with the results...

```
Sub MyGetInboxFaxDetails(Key As Integer, ByRef Pages As Integer, ByRef sDate As Date,
ByRef sFax As String, ByRef sResult As String, ByRef NextKey As Integer)
        Snappy.getInboxFaxDetails(Key, Pages, sDate, sFax, sResult, NextKey)
End Sub
Private Sub btnGetLastDetail_Click(sender As Object, e As EventArgs) Handles
btnGetLastDetail.Click
        Dim Pages As Long
        Dim Key As Long
        Dim LastKey As Long
        Dim FirstKey As Long
        Dim count As Long
        Dim NumRecs As Long
        Dim sFax As String
        Dim sResult As String
        Dim NextKey As Integer
        Dim fDate As Date

        sTime = ""
        sFax = ""
```

```
        Snappy = CreateObject("SF5.SnappyFaxIntf")
        NumRecs = Snappy.InboxRecs
        LastKey = Snappy.LastInboxKey
        FirstKey = Snappy.FirstInboxKey
        count = 1
        Key = FirstKey
        NextKey = 0
        MyGetInboxFaxDetails(Key, Pages, fDate, sFax, sResult, NextKey)
        Grid1.Rows.Add(Key, sFax, fDate, Pages, sResult)
        Do While NextKey <> -1
            count = count + 1
            Key = NextKey
            NextKey = -1
            sTime = ""
            sFax = ""
            sResult = ""
            MyGetInboxFaxDetails(Key, Pages, fDate, sFax, sResult, NextKey)
            Grid1.Rows.Add(Key, sFax, fDate, Pages, sResult)
            inProgressBar.Value = (count * 100) / NumRecs
        Loop
    End Sub
```

2.48.1 C# Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated with Snappy = CreateObject("SF5.SnappyFaxIntf")

```
private void FetchInboxDetail()
{
   int Key, NextKey;
   DateTime dDate;
   string sFax, sFaxResult;
   int Pages, count, Percent, Total;

   Action AddNode = () =>
   {
      TreeListNode Node = InBoxTree.Add();
      Node[colInBoxKey.ItemIndex] = Key.ToString();
      Node[colInboxFaxNumber.ItemIndex] = sFax;
      Node[colInboxDate.ItemIndex] = dDate.ToString("MM/dd/yyyy");
      Node[colInboxPages.ItemIndex] = Pages.ToString();
      Node[colInboxResult.ItemIndex] = sFaxResult;
   };

   count = 0;
   Total = FSF5.InboxRecs;
```

```
    InBoxTree.Clear();
    InBoxTree.BeginUpdate();
    Screen.Cursor = Cursors.WaitCursor;

    try
    {
      Key = FSF5.FirstInboxKey;
      FSF5.GetInboxFaxDetails(Key, out Pages, out dDate, out sFax, out
sFaxResult, out NextKey);

      do
      {
        AddNode();
        Key = NextKey;
        FSF5.GetInboxFaxDetails(Key, out Pages, out dDate, out sFax, out
sFaxResult, out NextKey);
        count++;
        Percent = (count * 100) / Total;
        ProgressPane.Percent = Percent;
      } while (NextKey != -1);

      AddNode();
    }
    finally
    {
      Screen.Cursor = Cursors.Default;
      InBoxTree.EndUpdate();
      ProgressPane.Percent = 0;
    }
}
```

2.48.2 Delphi Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated with Snappy := CreateOleObject("SF5.SnappyFaxIntf")

```
procedure TFmMain.FetchInboxDetail;
var
```

```
  Key, NextKey : integer;
  dDate : TDateTime;
  sFax, sFaxResult : String;
  Pages : integer;
  count : integer;
  Percent : integer;
  Total : integer;
  procedure AddNode;
  var
    Node : TcxTreeListNode;
  begin
    Node := InBoxTree.Add;
    With Node do
    begin
      Texts[colInBoxKey.ItemIndex] := IntToStr(Key);
      Texts[colInboxFaxNumber.ItemIndex] := sFax;
      Texts[colInboxDate.ItemIndex] :=
FormatDatetime('mm/dd/yyyy',dDate);
      Texts[colInboxPages.ItemIndex] := IntToStr(Pages);
      Texts[colInboxResult.Itemindex] := sFaxResult;
    end;
  end;
begin
    count := 0;
    Total := Snappy.InboxRecs;
    InBoxTree.Clear;
    InBoxTree.BeginUpdate;
    Screen.Cursor := crHourglass;
    try
      Key := Snappy.FirstInboxKey;
      FSF5.GetInboxFaxDetails(Key,Pages,dDate,sFax,sFaxResult,Next
Key);
      repeat
        AddNode;
        Key := NextKey;
        Snappy.GetInboxFaxDetails(Key,Pages,dDate,sFax,sFaxResult,N
extKey);
        inc(count);
```

```
        Percent := (count * 100) div Total;
        ProgressPane.Percent := Percent;
      until (NextKey = -1);
      AddNode;
    finally
      Screen.Cursor := crDefault;
      InBoxTree.EndUpdate;
      ProgressPane.Percent := 0;
    end;
end;
```

2.48.3 VB Example:

Note: The "Snappy" variable is an OleObject that has already been instantiated with Snappy = CreateObject("SF5.SnappyFaxIntf")

```
Private Sub FetchInboxDetail()
   Dim Key, NextKey As Integer
   Dim dDate As DateTime
   Dim sFax, sFaxResult As String
   Dim Pages, count, Percent, Total As Integer

   Dim AddNode As Action = Sub()
                              Dim Node As TreeListNode = InBoxTree.Add()
                              With Node
                                 .Texts(colInBoxKey.ItemIndex) = Key.ToString()
                                 .Texts(colInboxFaxNumber.ItemIndex) = sFax
                                 .Texts(colInboxDate.ItemIndex) =
dDate.ToString("MM/dd/yyyy")
                                 .Texts(colInboxPages.ItemIndex) =
Pages.ToString()
                                 .Texts(colInboxResult.ItemIndex) = sFaxResult
                              End With
                           End Sub

   count = 0
   Total = FSF5.InboxRecs
   InBoxTree.Clear()
   InBoxTree.BeginUpdate()
```

```
    Screen.Cursor = Cursors.WaitCursor

    Try
        Key = FSF5.FirstInboxKey
        FSF5.GetInboxFaxDetails(Key, Pages, dDate, sFax, sFaxResult,
NextKey)

        Do
            AddNode.Invoke()
            Key = NextKey
            FSF5.GetInboxFaxDetails(Key, Pages, dDate, sFax,
sFaxResult, NextKey)
            count += 1
            Percent = (count * 100) \ Total
            ProgressPane.Percent = Percent
        Loop Until (NextKey = -1)

        AddNode.Invoke()
    Finally
        Screen.Cursor = Cursors.Default
        InBoxTree.EndUpdate()
        ProgressPane.Percent = 0
    End Try
End Sub
```

2.49 GetServerInboxFax

Purpose: Retrieve the fax image of a record in the fax server's inbox data table and save to a .tif file or a .pdf file
Return value: boolean (true if successful)
Parameters:

- Key : integer
- sFile : string

Key is the key of the record in the inbox data table you want to retrieve the fax image for sFile is the desired file name to save the fax image to. The file extension must either be .tif or .pdf, if a file with another file extension is passed in this parameter an exception will be raised.

2.50 GetServerInboxFaxDetails

Purpose: Retrieve detail information of a record in the fax server's inbox data table
Return value: boolean
Parameters:

- Key : integer [in]
- Date : Date [out]
- sFax : string [out]
- sFaxResult : string [out]
- sCheckedOutby : string [out]
- Pages : integer [out]
- NextKey : integer [out]
- CheckedOut : boolean [out]

The key of the desired record is passed in the Key parameter. Snappy fax will pass back the other parameters from the fax detail record. The CheckedOut parameter will indicate if the record is checked out and the sCheckedOutby parameter will contain the client computer name that has checked out the record if it is checked out.

Refer to the example visual basic code under the topic GetInBoxFaxDetails for proper usage of this method

2.51 GetOutboxFax

Purpose: Retrieve the fax image of a record in the outbox data table and save to a .tif file
Return value: boolean (true if successful)
Parameters:

- Key : integer
- sFile : string

Key is the key of the record in the outbox data table you want to retrieve the fax image for
sFile is the desired file name to save the fax image to, *must* be a .tif file

2.52 GetOutboxFaxDetails

Purpose: Retrieve detail information of a record in the outbox data table
Return value: boolean
Parameters:

- Key : integer [in]
- Pages : integer [out]
- Date : Date [out]
- sFax : string [out]

- sSentTo : string [out]
- sFaxResult : string [out]
- NextKey : long integer [out]

Key is the key of the record in the inbox data table you want to retrieve the details for The Number of pages in the fax, the date /time of the fax, the fax number called, the recipient's name to whom the fax was sent, the result of the fax and the key of the next record in the inbox data table are returned in the Pages, Date, sFax, sFaxResult and NextKey parameters.

Sample Usage (Visual Basic):

Here is a detailed example showing how to iterate through all outbox records and populate a grid with the results...

```
Function MyGetOutboxFaxDetails(Key As Integer, ByRef Pages As Integer, ByRef sDate As
Date, ByRef sFax As String,ByRef sSentTo As String, ByRef sResult As String, ByRef
NextKey As Integer)
        {Note: Snappy has already been instantiated to an object with global scope}
        Snappy.getOutboxFaxDetails(Key, Pages, sDate, sFax,sSentTo, sResult, NextKey)
End Sub
Private Sub btnGetLastDetail_Click(sender As Object, e As EventArgs) Handles
btnGetLastDetail.Click
        Dim Pages As Long
        Dim Key As Long
        Dim LastKey As Long
        Dim FirstKey As Long
        Dim count As Long
        Dim NumRecs As Long
        Dim sSentTo As String
        Dim sFax As String
        Dim sResult As String
        Dim NextKey As Integer
        Dim fDate As Date
        sFax = ""
        sResult = ""
        Snappy = CreateObject("SF5.SnappyFaxIntf")
        NumRecs = Snappy.InboxRecs
        LastKey = Snappy.LastInboxKey
        FirstKey = Snappy.FirstInboxKey
        count = 1
        Key = FirstKey
        NextKey = 0
        MyGetInboxFaxDetails(Key, Pages, fDate, sFax,sSentto, sResult, NextKey)
        Grid1.Rows.Add(Key, sFax,sSentto, fDate, Pages, sResult)
        Do While NextKey <> -1
            count = count + 1
            Key = NextKey
            NextKey = -1
            sFax = ""
            sResult = ""
            MyGetOutboxFaxDetails(Key, Pages, fDate, sFax,sSentto, sResult, NextKey)
            Grid1.Rows.Add(Key, sFax,sSentTo, fDate, Pages, sResult)
```

```
            OutProgressBar.Value = (count * 100) / NumRecs
        Loop
    End Sub
```

2.53 GetOutboxFaxDetailByAutoJobID

Purpose: Retrieve detail information of a record in the outbox data table referencing it by the AutoJobId
Return value: None
Parameters:

- Key : integer [in]
- Pages : integer [out]
- Date : Date [out]
- sFax : string [out]
- sSentTo : string [out]
- sFaxResult : string [out]

Key is the AutoJobID used when you originally made a call to SendFax. This number (integer) should always be unique, it is up to you to enforce that. If the number is not unique then the call to this method will produce unreliable results.
The Number of pages in the fax, the date /time of the fax, the fax number called, the recipient's name to whom the fax was sent, the result of the fax and the key of the next record in the inbox data table are returned in the Pages, Date, sFax, sFaxResult parameters.

Here is a detailed example showing how to iterate through all outbox records and populate a grid with the results...

```
Sub MyGetOutboxFaxDetailByAutoJobID(Key As Integer, ByRef Pages As Integer, ByRef
sDate As Date, ByRef sFax As String,ByRef sSentTo As String, ByRef sResult As String)
        {Note: Snappy has already been instantiated to an object with global scope}
        Snappy.getOutboxFaxDetailByAutoJobID(Key, Pages, sDate, sFax,sSentTo, sResult)
End Sub
Private Sub btnGetLastDetail_Click(sender As Object, e As EventArgs) Handles
btnGetLastDetail.Click
        Dim Pages As Long
        Dim Key As Long
        Dim sSentTo As String
        Dim sFax As String
        Dim sResult As String
        Dim fDate As Date
        sFax = ""
        sResult = ""
        Snappy = CreateObject("SF5.SnappyFaxIntf")
        MyGetInboxFaxDetailByAutoJobID(Key, Pages, fDate, sFax,sSentto, sResult)
        Grid1.Rows.Add(Key, sFax,sSentto, fDate, Pages, sResult)
    End Sub
```

2.54 GetServerOutboxFax

Purpose: Retrieve the fax image of a record in the fax server's outbox data table and save to a .tif file or a .pdf file
Return value: boolean (true if successful)
Parameters:

- Key : integer
- sFile : string

Key is the key of the record in the inbox data table you want to retrieve the fax image for sFile is the desired file name to save the fax image to. The file extension must either be .tif or .pdf, if a file with another file extension is passed in this parameter an exception will be raised.

Sample Usage (Visual Basic):

```
Dim Key as long
Dim sFile as string
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Key = 1909
sFile = "c:\MyTifFile.tif"
if Snappy.GetInServerOutboxFax(Key,sFile) then
  'do something with image file
```

2.55 GetServerJobDetails

Purpose: Retrieve detail information of a record in the fax server's pending jobs data table
Return value: boolean
Parameters:

- Key : integer [in]
- sSubmitter : string [out]
- sClient : string [out]
- sSendto : string [out]
- DateTimeToSend : Date [out]
- SendImmediately : boolean [out]
- InProgress : boolean [out]
- NextKey : integer [out]

The key of the desired record is passed in the Key parameter. Snappy fax will pass back the other parameters from the job detail record.

See Also:

GetServerOutBoxFaxDetails

GetServerInBoxFaxDetails

2.56 GetServerOutboxFaxDetails

Purpose: Retrieve detail information of a record in the fax server's outbox data table
Return value: boolean
Parameters:

- Key : integer [in]
- Date : Date [out]
- sSentto : string [out]
- sFax : string [out]
- sFaxResult : string [out]
- sSubmitter : string [out]
- Pages : integer [out]
- NextKey : integer [out]

The key of the desired record is passed in the Key parameter. Snappy fax will pass back the other parameters from the fax detail record. The sSubmitter parameter will contain the network name of the client that submitted the job to the fax server.

Refer to the example visual basic code under the topic GetOutboxFaxDetails for proper usage of this method

2.57 InboxRecs

Purpose: Obtain the number of records in the inbox data table.
Return value : long integer
Parameters: None

Sample Usage (Visual Basic):

```
Dim NumInboxRecs as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
NumInboxRecs = Snappy.InBoxRecs
```

2.58 IsInboxImageValid

Purpose: Test if the image data that is stored in the inbox record indicated by Key is valid
Return value: boolean
Parameters:

Key , integer [in]

Sample Usage (Visual Basic):

```
Dim Key as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Key = 10902
if Snappy.IsInboxImageValid(Key) then
  'yes do something
```

2.59 IsOutboxImageValid

Purpose: Test if the image data that is stored in the outbox record indicated by Key is valid
Return value: boolean
Parameters:

Key , integer [in]

Sample Usage (Visual Basic):

```
Dim Key as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Key = 10902
if Snappy.IsOutboxImageValid(Key) then
  'yes do something
```

2.60 IsServerInboxImageValid

Purpose: Test if the image data that is stored in the fax server's inbox record indicated by Key is valid
Return value: boolean
Parameters:

Key , integer [in]

Sample Usage (Visual Basic):

```
Dim Key as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Key = 10902
if Snappy.IsServerInboxImageValid(Key) then
  'yes do something
```

2.61 IsServerOutboxImageValid

Purpose: Test if the image data that is stored in the fax server's outbox record indicated by Key is valid
Return value: boolean
Parameters:

Key , integer [in]

Sample Usage (Visual Basic):

```
Dim Key as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Key = 10902
if Snappy.IsServerOutboxImageValid(Key) then
  'yes do something
```

2.62 IsValidInboxKey

Purpose: Test if Key is a valid index key for the inbox table
Return value: boolean
Parameters:

Key , integer [in]

Sample Usage (Visual Basic):

```
Dim Key as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Key = 10902
if Snappy.IsValidInBoxKey(Key) then
  'yes do something
```

2.63 IsValidOutboxKey

Purpose: Test if Key is a valid index key for the outbox table
Return value: boolean
Parameters:

Key , integer [in]

Sample Usage (Visual Basic):

```
Dim Key as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Key = 10
if Snappy.IsValidOutBoxKey(Key) then
  'yes do something
```

2.64 IsIFaxEnabled

Purpose: Test if an Internet Fax Service has been configured in Snappy Fax
Return value: boolean
Parameters: None

Sample Usage (Visual Basic):

```
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
if Snappy.IsIFaxEnabled then
  'yes do something like select the 'I' fax routing
```

2.65 OutboxRecs

Purpose: Obtain the number of records in the outbox data table.
Return value: long integer
Parameters: None

Sample Usage (Visual Basic):

```
Dim NumOutboxRecs as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
NumOutboxRecs = Snappy.OutBoxRecs
```

2.66 LastAddressBookKey

Purpose: Obtain the Key of the last (newest) record in the address book indicated by the string parameter
Return value: long integer
Parameters: Address book name, string

Sample Usage (Visual Basic):

```
Dim Key as integer
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Key = Snappy.LastAddressBookKey("Default")
```

Note: if the address book name is passed as a blank string, the Default address book will be assumed

2.67 LastInboxKey

Purpose: Obtain the key of the last (most recent) record in the Inbox data table.
Return value: long integer
Parameters: None

Sample Usage (Visual Basic):

```
Dim LastInKey as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
LastInKey = Snappy.LastInKey
```

2.68 LastOutboxKey

Purpose: Obtain the key of the last (most recent) record in the outbox data table.
Return value: long integer
Parameters: None

Sample Usage (Visual Basic):

```
Dim LastOutKey as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
LastOutKey = Snappy.LastOutboxKey
```

2.69 LastAutomationKey

Purpose: Obtain the key of the last (most recent) record in the automation history data table.
Return value: long integer
Parameters: None

Sample Usage (Visual Basic):

```
Dim LastAutoKey as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
LastAutoKey = Snappy.LastAutomationKey
```

Note: The return value is the KEY of the last record NOT the AutoJobId

2.70 LastServerInboxKey

Purpose: Obtain the key of the last (most recent) record in the fax Server's inbox data table.
Return value: long integer
Parameters: None

Sample Usage (Visual Basic):

```
Dim LastInKey as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
LastInKey = Snappy.LastServerInboxKey
```

Note: Only applies if snappy fax is configured as a client to snappy fax server, otherwise the exception 'Client not configured for fax server' will be raised

2.71 LastServerJobKey

Purpose: Obtain the key of the last (most recent) record in the fax Server's pending jobs data table.
Return value: long integer
Parameters: None

Sample Usage (Visual Basic):

```
Dim LastKey as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
LastKey = Snappy.LastServerJobKey
```

Note: Only applies if snappy fax is configured as a client to snappy fax server, otherwise the exception 'Client not configured for fax server' will be raised

2.72 LastServerOutboxKey

Purpose: Obtain the key of the last (most recent) record in the fax Server's outbox data table.
Return value: long integer
Parameters: None

Sample Usage (Visual Basic):

```
Dim LastKey as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
LastKey = Snappy.LastServerOutboxKey
```

Note: Only applies if snappy fax is configured as a client to snappy fax server, otherwise the exception 'Client not configured for fax server' will be raised

2.73 MemoField

Purpose: Set text to display in snappy fax's memo field in the outbox data table
Return value : None
Parameters: None

Sample Usage (Visual Basic):

```
Dim Snappy as Object
Dim JOB as integer
Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.MemoField = "Project 7"
```

2.74 NextInboxKey

Purpose: Obtain the next key that will be assigned to a new inbox record
Return value: long integer
Parameters: None

Sample Usage (Visual Basic):

```
Dim NextKey as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
```

```
NextKey = Snappy.NextInBoxKey
```

2.75 NextOutboxKey

Purpose: Obtain the next key that will be assigned to a new outbox record
Return value: long integer
Parameters: None

Sample Usage (Visual Basic):

```
Dim NextKey as long
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
NextKey = Snappy.NextOutBoxKey
```

2.76 SendFax

Purpose: Call this method to start the fax transmission after setting the required property values
Return value: None
Parameters: None

Sample Usage (Visual Basic):

```
Dim FaxNumber as string
Dim FaxRecipient as string
Dim FaxFile as string
Dim Snappy as Object

Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.FaxNumber = "18009871626"
Snappy.FaxRouting = "R"
Snappy.FiletoFax = "c:\mytiffile.tif"
Snappy.FaxRecipient = "John Doe"
Snappy.AutoJobId = 121098  {some unique integer value}
Snappy.SendFax
```

Snappy fax will raise an exception if any of the properties are incorrect:

The AutoJobID in not unique, please supply a unique ID
No Image file specified
No Fax Number Specified
No Fax Recipient Name Specified
No Fax Routing Specified

Invalid Fax Routing Method Specified
Fax Number is invalid, must be at least 10 characters

2.77 UseCoverPage

Purpose: Set to true if the fax will contain a cover page
Return value : None
Parameters: None

Sample Usage (Visual Basic):

```
Dim Snappy as Object
Dim JOB as integer
Snappy = CreateObject("SF5.SnappyFaxIntf")
Snappy.UseCoverPage = true
Snappy.FileToFax = "ThursdayMenu.tif"
Snappy.CoverPageTemplate = "mytemplate.fct"
Snappy.CoverPageMemo = "This is the Bistro's menu for Thursday, Oct. 31"
Snappy.CoverPageSubject = "New Jersey Bistro Menu"
```

2.78 Windows Messages

Snappy Fax sends the following windows messages to alert your application that certain events have occurred. If you are interested in handling these messages then you should set up a windows message procedure in your application.

Message: **WM_AUTOMSG_FAXSTARTED**
Definition: WM_APP + 10001
wParam: 0
lParam: your auto job id of the fax job
Event triggering: The fax job has started

Message: **WM_AUTOMSG_FAXFINISHED**
Definition: WM_APP + 10002
wParam: Error code (see error codes topic under Advanced Topics)
lParam: your auto job id of the fax job
Event triggering: The fax job has finished

Message: **WM_AUTOMSG_STARTPAGE**
Definition: WM_APP + 10004;

wParam: Page number
lParam your auto job id of the fax job

Message: **WM_AUTOMSG_PAGEOK**
Definition: WM_APP + 10003;
wParam: Page number
lParam your auto job id of the fax job

Message: **WM_AUTOMSG_PAGEBYTES_SENT**
Definition: WM_APP + 10005;
wParam: Bytes sent for current page
lParam your auto job id of the fax job

Message: **WM_AUTOMSG_TOTALBYTES_SENT**
Definition: WM_APP + 10006;
wParam: Bytes sent for sessioni
lParam your auto job id of the fax job

Message: **WM_AUTOMSG_DOCUMENTSIZE_INBYTES**
Definition: WM_APP + 10007;
wParam: Document (fax file) size in Bytes
lParam your auto job id of the fax job

Message: **WM_AUTOMSG_PAGESIZE_INBYTES**
Definition: WM_APP + 10008;
wParam: Size of current page
lParam your auto job id of the fax job

Message: **WM_AUTOMSG_PERCENT_DONE**
Definition: WM_APP + 10009;
wParam: Percent Done (all pages)
lParam your auto job id of the fax job

Message: **WM_AUTOMSG_PAGESINFAX**
Definition: WM_APP + 100010;
wParam: The number of pages in this transmission
lParam: your auto job id of the fax job

Message: **WM_AUTOMSG_JOBACCEPTED**
Definition: WM_APP + 10011;
wParam: Sent when the job has been added to Snappy Fax's Automation Queue
lParam your auto job id of the fax job
Note: You can submit a number of fax jobs in succession. Snappy Fax will add each one to the queue and process them in the order submitted

Message: **WM_AUTOMSG_FAXSTATUS**
Definition: WM_APP + 10012;
wParam: Fax Status Code See: Fax Progress Codes
lParam your auto job id of the fax job

Message: **WM_AUTOMSG_SERVERSTATUS_UPDATED**
Definition: WM_APP + 10013;
wParam: 0

lParam your auto job id of the fax job
Note: This message is sent when Snappy Fax has updated the status of a fax job submitted to snappy fax server. This only occurs if the fax routing was 'S' and Snappy Fax is configured as a client to snappy fax server.

Message: **WM_AUTOMSG_SUBMITTED_TO_FAXSERVER**
Definition: WM_APP + 10014;
wParam: 0
lParam your auto job id of the fax job
Note: This message is sent when Snappy Fax has submitted your job to snappy fax server. This only occurs when Snappy fax is configured as a client to snappy fax server.

Message: **WM_AUTOMSG_SUBMITTED_TO_IFAXSERVICE**
Definition: WM_APP + 10015;
wParam: 0
lParam your auto job id of the fax job
Note: This message is sent when Snappy Fax has submitted the fax job to the internet fax service. This only occurs when Snappy Fax is configured to use either FaxAge or SRFax internet fax service and the fax routing method was set to 'I'.

Message: **WM_AUTOMSG_IFAXSTATUS_UPDATED**
Definition: WM_APP + 10016;
wParam: 0
lParam your auto job id of the fax job
Note: This message is sent when Snappy Fax has updated the status of the fax job that was submitted to the internet fax service. This only occurs when Snappy Fax is configured to use either FaxAge or SRFax internet fax service and the fax routing method was set to 'I'.

3 Appendix

3.1 Tracking Fax Progress

You can track the progress of a fax that is submitted to snappy fax for processing by using the AppWindow, AppMsgNumber and AutoJobID properties.

- Set the AppWindow property to the window handle of your application's window.
- Set the AppMsgNumber to your desired message number, usually WM_APP + (some number of your choosing)
- Set the AutoJobID property to a unique integer id that you can use internally in your application to know which fax job is being referred to in the status messages snappy fax sends to your application.
- Setup a message handler in your application to receive and process the message number set in AppMsgNumber sent from snappy fax to your window handle specified in AppWindow.

When a fax job has been ***started*** by snappy fax it will send a message to your application as follows:

PostMessage(AppWindow,AppMsgNumber,1,LParam(AutoJobId))

Note the wParam is 1 indicates that the job is now being started (transmission will begin), the job being referred to is sent in the lparam

When the fax job has ***completed*** with or without success, snappy fax will send a message to your application as follows:

PostMessage(AppWindow,AppMsgNumber,ErrorCode,LParam(AutojobId))

Note that in the completion message the wParam is the ErrorCode which can never be 1, so a 1 in this message parameter indicates the start of the fax job transmission. Any other value indicates this is a completion message and the wParam contains the error code. Refer to the topic Error Codes for a complete listing of all error codes.

See Also: GetOutboxFaxDetailsByAutoJobID

See also: Windows Messages for more effective ways to track your fax progress.

3.2 Error Codes

All possible error codes and their meaning are presented below. Error code is the wParam sent by snappy fax in the completion message when a fax job has been completed either successfully or not. The error code is also sent in the wParam of the WM_AUTOMSG_FAXFINISHED message. Refer to Windows Messages topic.

Note: you can ignore the constant designator as only the integer value in parenthesis will be of interest to you.

Error Code	Meaning
ecOk (0)	OK
ecFaxWidthNotSupported (8078)	Remote Does Not Support Fax Width
ecHighResNotSupported (8079)	Remote Does Not Support High Resolution
ecHungSessionAbort (6028)	Hung Fax Session Aborted
ecNoFilename (6010)	No Fax File Found
ecCancelRequested (6005)	Cancel requested
ecTimeout (6006)	Fatal time out
ecFileRejected (6011)	File Rejected
ecCantWriteFile (6012)	Cannot write file
ecAbortNoCarrier (6014)	Abort carrier lost
ecDCNSeen (6017)	Remote sent Disconnect
ecRTNSeen (6049)	Fax Failed RTN Received
ecFaxBadMachine (8060)	Not compatible with remote fax
ecFaxBadModemResult (8061)	Modem said "ERROR"
ecFaxTrainError (8062)	Devices could not complete training
ecafNextFaxFailed (8059)	afNextFax Failed
ecFaxInitError (8063)	Error while initializing modem

ecFaxInitTimedOut (8076)	No Modem response during initialization
ecFaxBusy (8064)	Called fax number was busy
ecFaxVoiceCall (8065)	Called fax number answered with voice
ecFaxDataCall (8066)	Incoming data call
ecFaxNoDialTone (8067)	No dial tone
ecClassNotSupported (8077)	Fax Class Not Supported
ecUnknownPhaseBError (6058)	Unknown Phase B Error
ecUnknownPhaseCError (6059)	Unknown Phase C Error
ecMissingEOL (6060)	Missing EOL
ecBufferOverFlow (6061)	Buffer Overflow
ecReceivedDCNFromRemote (6062)	Remote Disconnected
ecInvalidCommandFromRemote (6063)	Invalid command from remote
ecRemoteControlFrameError (6064)	Remote control frame error
ecDataUnderFlow : result (6065)	Send Data underflow
ecTooManyBlockRetransmits (6066)	Too many block retransmits
ecInvalidResponsetoEOR (6067)	Invalid response to EOR
ecInvalidPPSFrame (6068)	Invalid PPS frame
ecInvalidEORFrame (6069)	Invalid EOR frame
ecFaxNoCarrier (8068)	No fax frames detected (possibly a voice call
ecFaxSessionError (8069)	Fax failed in mid-session
ecFaxPageError (8070)	Fax failed at page end
ecFaxMixedResolution (8072)	Mixed resolution not allowed
ecNoAnswer (8074)	Remote fax did not answer
ecFaxMCFNoAnswer (6018)	Remote Did Not Reply to MCF
ecNOCFRReceived (6023)	Remote did not send CFR
ecExtractTemplateFail (6019)	Extract Cover Template Failed
ecCoverTemplateMissing (6026)	Cover Template Missing from Job Record
ecMakeCoverFail (6024)	Make Cover Page Failed
ecExtractImageFail (6020)	Extract Image Failed
ecOtherMakeJobFail (6021)	Make Fax Job Failed
ecAPFNoExist (6022)	APF File does not exist
ecInvalidFaxImage (6043)	Invalid Fax Image
ecConvertAPFToTifFail (6044)	Convert Fax Image Failed
ecImageBlobNull (6035)	Image Blob Null
ecUnexpectedDCN (6030)	Unexpected Disconnect
ecDCS3TimesNoResponse (6031)	DCS sent 3 times, no response
ecDCSNotRecognized (6032)	DIS/DTC received 3 time, DCS not Recognized
ecPSPRECInvalidResponse (6033)	PSPREC Invalid Response
ecMPS3TimesNoResponse (6034)	No response to MPS, repeated 3 times
ecMPSInvalidResponse (6035)	Invalid response to MPS
ecEOP3TimesNoResponse (6036)	No response to EOP, repeated 3 times
ecEOPInvalidResponse (6037)	Invalid response to EOP
ecEOM3TimesNoResponse (6038)	No response to EOM, repeated 3 times
ecEOMInvalidResponse (6039)	Invalid response to EOM
ecExpectedPageNotReceived (6040)	Expected Page not Received
ecEOMTimeout (6041)	Timeout after EOM Received
ecBadFrame (6042)	Bad CRC or frame in ECM mode
ecBadNumberFormat (6047)	Fax # Formatted Incorrectly
ecNoPDFConvert (6050)	PDF File Conversion failure
ecNoTifConvert (6051)	Tif File Conversion failure
ecFaxNumberMissing (6053)	Fax Number is Missing
ecBadModulation (6054)	Remote's Modulation Incorrect
ecPortFailedToOpen (6055)	Com Port Failed to Open

ecPrepFaxFailed (6056)	Prep Fax Failed
ecUnknownPhaseDError (6057)	Unspecified Phase D Error

3.3 Fax Progress Codes

These are the fax progress codes sent in the wParam of the WM_AUTOMSG_FAXSTATUS Message:
See: Windows Messages

This is a comprehensive list of snappy fax progress code for both incoming and outgoing faxes, you will not necessarily be interested in many of them, they are listed here for completeness. Note: you can ignore the constant designator as only the integer value in parenthesis will be of interest to you.

Numeric Code	Meaning
fpRightDRON (4868)	DRON MATCH
fpWrongDRON (4867)	DRON MISMATCH
fpRewindingPage (4813)	Preparing to retransmit page
fpDrainPage (4865)	Draining Buffer...
fpRTPReceived (4809)	RTP Frame Received, Page OK
fpMCFReceived (4810)	MCF Frame Received, Page OK
fpRTNReceived (4811)	RTN Frame Received, Page Rejected
fpUnexpectedFrame (4812)	??? Frame Received, expected RTP,MCF or RTN
fpNoDataRecv (4862)	Waiting on more page data...
fpRewindPage (4869)	Resending Current Page
fpGotRing (4844)	The Line is Ringing...
fpInitModem (4801)	Initializing Modem...
fpDialing (4801)	Dialing...
fpBusyWait (4803)	Busy Wait...
fpSendPage (4804)	Sending Fax Page Data...
fpSendPageStatus (4805)	Getting Page Status...
fpPageError (4806)	Fax Page Error
fpPageOK (4807)	Fax Page OK
fpWaiting (4820)	Waiting on Call...
fpNoConnect (4821)	No Connection
fpAnswer (4822)	Answering Call...
fpIncoming (4823)	Incoming Fax Call
fpGetPage (4824)	Getting Page Data...
fpGetPageResult (4825)	Getting Fax Page Result...
fpCheckMorePages (4826)	Checking for more pages...
fpGetHangup (4827)	Getting Hangup Code...
fpSessionParams (4840)	Getting Session Parameters
fpGotRemoteID (4841)	Get Remote ID...
fpCancel (4842)	Cancel
fpFinished (4843)	Finished
fpBeginTrain (4843)	Begin Training...
fpEndTrain (4846)	End Training...
fpSendCFR (4847)	Sending CFR...
fpSendMCF (4848)	Sending MCF...
fpWaitMCF (4849)	Waiting on MCF from Remote...
fpRetrain (4850)	Retraining...
fpSendCSI (4851)	Sending CSI...

fpSendDIS (4852)	Sending DIS...
fpWritePage (4853)	Writing Page Data...
fpPageWriteError (4863)	Error Writing Page Data
fpPageWriteComplete (4864)	Write Page Done
fpSendDCN (4854)	Sending Disconnect...
fpWaitCFR (4855)	Waiting on CFR from Remote...
fpWaitDCN (4856)	Waiting on DCN from Remote...
fpSendTSI (4857)	Waiting on TSI from Remote...
fpWaitHangup (4848)	Wait Hangup...
fpRecvFail (4850)	Incoming Fax Session Failed
fpGotHangup (4828)	Got Hangup
fpRejectingFax (4814)	Rejecting Junk Fax
fpHandShake (4900)	HandShaking...
fpHandShakeOK (4901)	HandShaking OK
fpConnecting (4808)	Call Connect
fpHandShakeFail (4902)	Handshake Failed
fpConnectOK (4903)	Connect OK
fpConnectFail (4904)	Connect Failed
fpMorePages (4905)	More Pages
fpNoMorePages (4906)	No More Pages
fpGotEOPSignal (4907)	Got EOP Signal
fpUpdateDatabase (4908)	Updating Database
fpSessionEnd (4909)	Fax Session End
fpEndProcess (4910)	End Process

Index

- 6 -

- 8 -

- A -

- B -

- C -

- E -

- F -

- G -

- I -

- L -

- M -

- N -

- O -

- P -

- R -

- S -

- T -

- U -