

# **SoftSand Fax Add-On for Rave**

*Copyright © 2003 by John Sandage*



# Table of Contents

|  |    |
|--|----|
| Table of Contents .....                          | 3  |
| Chapter 1: Getting Started.....                  | 5  |
| License Agreement.....                           | 5  |
| Overview of "SoftSand Fax Add-On for Rave" ..... | 5  |
| Installation.....                                | 6  |
| Chapter 2: Tutorial .....                        | 7  |
| The FaxDemo Program .....                        | 7  |
| FaxDemo as a C++ Builder Example.....            | 7  |
| Creating the Main Form .....                     | 7  |
| Creating the Program Code .....                  | 10 |
| Chapter 3: Component Reference .....             | 13 |
| The TSSRenderFax Component.....                  | 13 |
| Using TSSRenderFax with C++ Builder.....         | 14 |
| TSSRenderFax Properties .....                    | 14 |
| TSSRenderFax Methods .....                       | 14 |
| TSSRenderFax Events .....                        | 14 |
| Chapter 4: Alphabetic Reference.....             | 15 |
| AfterSetupDlg .....                              | 15 |
| BeforeSetupDlg.....                              | 15 |
| CoverPageInfo .....                              | 15 |
| CoverPageType .....                              | 16 |
| DocumentName.....                                | 17 |
| ESSRenderFaxError .....                          | 17 |
| FaxingSupported .....                            | 17 |
| FaxJobID.....                                    | 17 |
| FaxPrinterIndex.....                             | 18 |
| FaxPrinterName .....                             | 18 |
| LoadFromIniFile .....                            | 18 |
| OnDocBegin.....                                  | 18 |
| OnDocEnd.....                                    | 19 |
| Printing.....                                    | 19 |
| ReceiptInfo .....                                | 19 |
| SaveToIniFile.....                               | 20 |
| SenderInfo .....                                 | 20 |
| ServerName.....                                  | 20 |
| SetupInfo.....                                   | 21 |
| TCoverPageType.....                              | 21 |
| TFaxCoverPageInfo .....                          | 21 |
| TFaxReceiptInfo .....                            | 22 |
| TFaxSenderInfo.....                              | 22 |
| TFaxSetupAction .....                            | 23 |
| TFaxSetupInfo .....                              | 23 |
| TFaxSetupTabs .....                              | 23 |
| TFaxSetupVisibleTabs .....                       | 23 |
| Appendix A: Contact Information.....             | 25 |
| Index.....                                       | 27 |



# Chapter 1: Getting Started

## **License Agreement**

### **1. RIGHTS GRANTED**

John Sandage, d.b.a. Sandage and Associates, having a business address of 1255 Tropical Ave., Pasadena, CA 91107 ("Licensor") grants to you (the "Licensee") a non-exclusive, non-sublicensable, non-transferable, license to use this version of "SoftSand Fax Add-On for Rave" and the accompanying User Documentation (the "Software") which consists of the right to utilize the compiled components of the Software to prepare derivative works of other programs which the Licensee owns or controls with no further license or payment made to the Licensor, provided that the Software cannot be separated out in whole or in part from such derivative works, and provided that the derivative works do not constitute a programming library or tool for resale to other software developers. The right does not extend to the redistribution of the component source code of the Software. The Licensee may not rent, lease, loan, resell for profit, sublicense or distribute the Software. The Licensee agrees to abide by the copyright law and all other applicable laws of the United States including export control laws.

### **2. RIGHTS RETAINED BY THE LICENSOR**

The Licensor retains all title, ownership rights, and intellectual property rights in and to the Software. The Licensee acknowledges that the Software in source code form remains a confidential trade secret of the Licensor.

### **3. LIMITED WARRANTY**

The Licensor will replace defective Software or correct substantial Software errors at no charge, provided that the Licensee return the Software with dated proof of payment to the Licensor within 30 days of the date of purchase. If the Licensor is unable to replace defective Software or correct substantial Software errors, the Licensor will refund the license fee. These are the Licensee's sole remedies for any breach of warranty. Except as specifically provided above, the Licensor makes no warranty or representation, either express or implied, with respect to the Software, including its quality, performance, merchantability, or fitness for a particular purpose. Because software is inherently complex and may not be completely free of errors, the Licensee is advised to verify his or her work. In no event will the Licensor be liable for direct, indirect, special, incidental, or consequential damages arising out of the use of or inability to use the Software, even if advised of the possibility of such damages. In particular, the Licensor is not responsible for any costs including, but not limited to, those incurred as a result of lost profits or revenue, loss of use of the software, loss of data, the cost of recovering such software or data, the cost of substitute software, or claims by third parties. In no case shall the Licensor's liability exceed the amount of the license fee. The warranty and remedies set forth above are exclusive and in lieu of all others, oral or written, express or implied. No dealer, distributor, agent, or employee is authorized to make any modification or addition to this warranty. Some states do not allow the exclusion or limitation of implied warranties or limitation of liability for incidental or consequential damages, so the above limitation or exclusion may not apply to the Licensee.

### **4. TERM AND GOVERNING LAW**

This Licensing Agreement will immediately and automatically terminate without notice if the Licensee fails to comply with any term or condition of this Licensing Agreement. The Licensee may terminate this Licensing Agreement at any time by destroying or erasing all copies of the Software. In the event of termination of this Licensing Agreement, clauses 2 and 3 survive. This License shall be governed by and construed in accordance with the laws of the State of California and, as to matters affecting copyrights, trademarks, and patents, by United States federal law. This Licensing Agreement sets forth the entire agreement between the parties.

## **Overview of "SoftSand Fax Add-On for Rave"**

This native VCL component will allow you to FAX any Rave or Report Printer report through the Windows NT/2000/XP faxing platform. Simply drop the component onto your report form, set the component properties, adjust Rave's "report destination" property, and you're ready to go!

Rave already allows you to create dazzling, eye-catching reports that can be previewed and printed. Now with this component, your Delphi or C++ Builder applications can FAX these reports to any standard FAX machine or FAX-enabled computer. It's great for sending invoices, sales orders, purchase orders, contracts, or any other automated correspondence.

**Features:**

- Allows your application to populate the Sender, Recipient, and Cover Page properties to automate the faxing process.
- Links to the Microsoft standard ".COV" cover page templates created by the Microsoft Cover Page Editor (ships with Win 2000 and XP).
- The optional embedded setup form allows the user to verify and/or change faxing parameters at fax-time. (This dialog may be enabled, disabled, or shown only if required parameters are left blank).

**Requirements:**

- Delphi 5 (or above), or C++ Builder 5 (or above).
- Rave 5 BEX (for Delphi 5,6,7 or C++ Builder 5,6) or Rave 5 standard (that ships with Delphi 7).
- Windows NT, 2000, or XP for both development and deployment. (Windows 95, 98, and ME are not supported).
- Any Windows compatible Fax/modem card.

**Installation**

To begin, you must first install the component files onto your hard disk. This is accomplished through the provided setup program. When the "SSFaxXXX.exe" program is run, (where XXX refers the specific version number) you will be presented with the "SoftSand Fax Add-On for Rave Setup Wizard".

The setup wizard will prompt you for various information as required, including the installation directory name. For purposes of this discussion, we suggest that you specify "C:\SSFax" for this entry. If this directory name is not appropriate for you, please make the appropriate directory name substitutions in the discussion below. The setup program will automatically build the component, and install it into the IDE. The new component is named, "TSSRenderFax" and will be placed on the "Rave" pallet page.

**What the Setup Wizard Really Does**

Many of the tasks necessary to copy, build, and configure the fax component, and install it in your development environment have been automatically accomplished by the setup wizard. It is important to know, however, what is really going on in the background in case you need to perform these tasks manually in the future.

First, the source code, help, and example files are copied onto your hard disk. The source code is copied in its entirety -- all versions of the product may be made from this source code.

The compilation batch files are modified by the setup wizard to reflect the directory where the product is installed. The modified lines are clearly commented as such at the beginning of each batch file. These batch files are then run at install-time by the setup wizard in order to create the required library files. This ensures that the resulting files exactly match your specific compiler and RAVE versions and patches. If, in the future, you need to re-compile the product, you may use the compilation batch files: MakeDx.bat and MakeCx.bat (where "x" refers to the particular version of Delphi or C++ Builder).

Your development environment has also been configured to recognize the new component package. This, again, is automatically accomplished by the setup program. To accomplish this manually, you would need to select "Component|Install Packages" from the IDE's main menu, click "Add", browse to find the appropriate package library, and then click "Open". This procedure is clearly documented in your compilers' on-line help or documentation.

The compilers' library search path has also been modified to include the appropriate \SSFax\Lib directory. This may be accomplished manually by selecting "Tools|Environment Options" from the IDE's main menu, selecting the "Library" tab, then editing the "Library Path" as appropriate.

Finally, this help file has been integrated with your compilers' help system. This is accomplished through the program "UpHelp32.exe" located in the \SSFax\Help directory.

# Chapter 2: Tutorial

## **The FaxDemo Program**

The "FaxDemo" program illustrates the basic functionality of the Fax Add-On component. The component works by re-directing the native output of Rave's TRvSystem through the TSSRenderFax component; therefore, every project that uses TSSRenderFax must also contain a TRvSystem. If you are using the TRvProject component to render reports generated by the Rave Visual Designer, you must still include a TRvSystem component and link them by TRvProject's "Engine" property. (More on this later).

Sender, Recipient, and Cover Page information is collected through various data controls placed on the Main Form. These setup parameters are then transferred to the properties of TSSRenderFax at run-time, before the report is actually rendered. The FaxDemo program can also save and load these parameters so that the user does not need to re-enter them each time the project is run. This is accomplished through the "LoadFromIniFile" and "SaveToIniFile" methods. Of course, you could alternatively save and load this information through a database if you choose.

For this example, the report content is generated from code placed in the TRvSystem's "OnPrint" event handler. Here, only the most basic "report" has been coded -- (the classic "Hello World!" example). The program has the capability of printing, previewing, or faxing this "report" from the "Print", "Preview", and "Fax" buttons respectively.

## **FaxDemo as a C++ Builder Example**

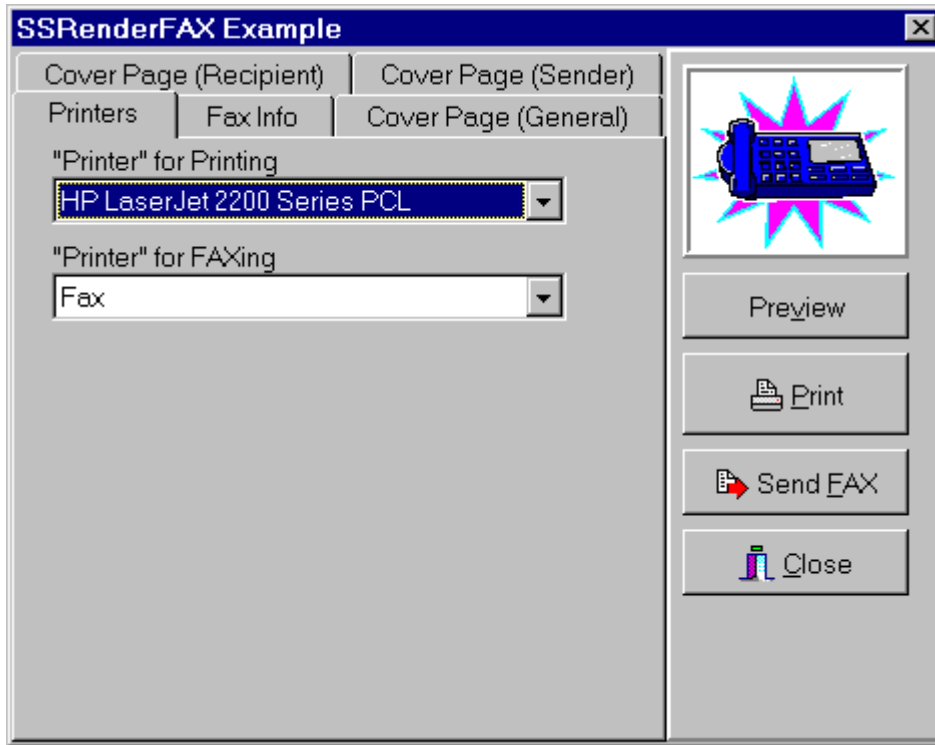
Beneath the \SSFax\Example folder, you will find separate folders for each supported version of Delphi and C++ Builder.. Under the CBuild folders are C++ versions of the FaxDemo program. Although the example source in this manual refers to Object Pascal code, the concepts directly apply to the provided C++ Builder example.

It is also important to note that in order for C++ Builder to properly locate the necessary header and library files, include and library path information must be provided in the "Include Path" and "Library Path" fields of the "Project Options|Directories and Conditions" dialog. In the case of this demo program, you will find that these paths have been already setup for you; however, they should be verified for your particular installation of Rave. (It is assumed here that Rave has been installed in the "C:\Rave5" directory. If this is not correct, these paths must be changed before the FaxDemo project will compile.

The correct include and library paths must be entered into the Project Options dialog for each project where Rave and the SoftSand Fax Add-On are used.

## **Creating the Main Form**

The FaxDemo program consists of a single form, -- the Main Form, -- containing a button panel on the right and a TPageControl component in the client portion of the form. From the page control, the user can select various tabs to enter the required fax setup information. This form is illustrated below:



### The Printers Page

The Printers Page contains two combo boxes that allows the user to select the Windows printer that will be used for printing, and the Windows printer driver that should be used to capture the fax output. These controls are populated in the form's "OnCreate" event handler and contain all printers installed on the workstation.

### The Fax Info Page

The Fax Info Page contains controls for the standard Recipient and Sender information. Only the "Recipient Fax Number" is required to send the fax. If this field is left blank, a setup dialog will appear at fax-time allowing the user to enter the fax number. The Fax Info Page is shown below:

The screenshot shows the 'SSRenderFAX Example' dialog box with the 'Fax Info' tab selected. The dialog has a title bar with a close button. Below the title bar are four tabs: 'Cover Page (Recipient)', 'Cover Page (Sender)', 'Printers', and 'Cover Page (General)'. The 'Fax Info' tab is active. On the right side of the dialog is a preview area containing a fax machine icon and four buttons: 'Preview', 'Print', 'Send FAX', and 'Close'.

**Recipient's Info**

Name:

Fax:

**Sender's Info**

Name:

Company:

Dept:

Bill Code:

### Cover Page Information

The remaining three tabs are devoted to cover page information. Of particular note is the "Cover Page (General)" page as shown below:

The screenshot shows the 'SSRenderFAX Example' dialog box with the 'Cover Page (General)' tab selected. The dialog has a title bar with a close button. Below the title bar are four tabs: 'Cover Page (Recipient)', 'Cover Page (Sender)', 'Printers', and 'Cover Page (General)'. The 'Cover Page (General)' tab is active. On the right side of the dialog is a preview area containing a fax machine icon and four buttons: 'Preview', 'Print', 'Send FAX', and 'Close'.

Type:

File Name:

Subject:

Note:

The cover page "Type" combo box contains the options: "None", "Local", and "Server". If "None" is selected, no cover page will be included in the fax transmission. If "Local" is selected, the TSSRenderFax component will

look for the cover page template file on a local or shared drive.(as specified in the "File Name" edit box). If "Server" is selected, the "File Name" specified in the file name edit box will reference a file located on a dedicated fax server (such as the Microsoft Small Business Server).

Other controls allow the user to specify the fax "Subject", and "Cover Page Note". This information will only appear on the fax cover page, and are for descriptive purposes only.

Finally, are the "Cover Page (Recipient)" and "Cover Page (Sender)" tabs. As expected, these contain the recipient and sender information that will be shown on an included cover page. These fields correspond to fields included on a "Cover Page Template". The FaxDemo program uses a cover page template named "Generic.cov" stored in the example directory. Other templates may be created using the Microsoft Cover Page Editor that ships with Windows. Some standard templates may be found in the "\My Documents\Faxes\Personal Cover Pages", and the "\Documents and Settings\All Users\Application Data\Microsoft\Windows NT\MSFax\Common Coverpages" directories.

The program name for the Microsoft Cover Page Editor is "fxscov.exe" and is stored in the Windows "System 32" directory. By default, it is associated with the ".COV" file extension, so that, by simply double-clicking a ".COV" file, you should be allowed to view and modify it in the Cover Page Editor.

### **Creating the Program Code**

All of the source code necessary to run the FaxDemo program is contained in the file, "Main.pas". Its various methods may be divided into the following categories: Form Events, Supporting Procedures, Component Events, and Button Events.

#### **Form Events**

Of particular importance is the form's "OnCreate" event. In this method, the printer combos are populated, the cover page path and file name are specified, the INI file object is created, and the form's components are loaded from that INI file. Loading the form controls is accomplished through a supporting procedure called, "LoadForm" and is discussed later. Finally, since the TSSRenderFax component is not supported by Win9x, it is important to disable the faxing capabilities if the program is run under that platform. The method, "FaxingSupported" was implemented specifically for this purpose. It is used to enable or disable the fax button. The form's "OnCreate" method is presented below:

```

procedure TMainForm.FormCreate(Sender: TObject);
begin
    {Fill the printer combos from the registered system printers}
    PrinterCombo.Items.Assign(Printer.Printers);
    if PrinterCombo.Items.Count > 0 then begin
        PrinterCombo.ItemIndex := 0;
    end;
    FaxPrinterCombo.Items.Assign(Printer.Printers);
    if FaxPrinterCombo.Items.Count > 0 then begin
        FaxPrinterCombo.ItemIndex := 0;
    end;

    {Set the component defaults}
    RenderFax.CoverPageInfo.CoverPageName :=
    ExtractFilePath(Application.Exename)+'Generic.cov';

    {Load form settings from an INI file}
    AppIniFile :=
    TIniFile.Create(ExtractFilePath(Application.ExeName)+'\FAXDemo.INI');
    LoadForm;

    {Set the button enable properties}
    FaxButton.Enabled := RenderFax.FaxingSupported;

end; {FormCreate}

```

Similarly, the form needs an "OnDestroy" event to save the user entered information and free the memory allocated for the INI file object. The "FormDestroy" method is presented below:

```
procedure TMainForm.FormDestroy(Sender: TObject);
begin

    {Save the form settings to an INI file}
    SaveForm;
    AppIniFile.Free;

end; {FormDestroy}
```

### Supporting Procedures

In order to load the form controls from the TSSRenderFax component, and load the TSSRenderFax component from the controls, we rely on the supporting procedures, "LoadForm", and "LoadComponent". They simply copy the information field-by-field from and to the TSSRenderFax component. Its code is straightforward and is not presented here.

### Component Events

If the user selects "None" for the cover page type, we would like to disable all other cover page controls on the form. This is accomplished through the "CoverPageTypeComboBoxChange" event presented below:

```
procedure TMainForm.CoverPageTypeComboBoxChange(Sender: TObject);
begin
    if CoverPageTypeCombo.ItemIndex = 0 then begin
        CoverPageNameEdit.Enabled := false;
        SubjectEdit.Enabled := false;
        NoteEdit.Enabled := false;
        CPRecipientSheet.TabVisible := false;
        CPSenderSheet.TabVisible := false;
    end else begin
        CoverPageNameEdit.Enabled := true;
        SubjectEdit.Enabled := true;
        NoteEdit.Enabled := true;
        CPRecipientSheet.TabVisible := true;
        CPSenderSheet.TabVisible := true;
    end;
end; {CoverPageTypeComboBoxChange}
```

In order to generate the content of the report, we use the ReportSystem's "BeforePrint" and "OnPrint" events. In this example, the report content consists of the trivial "Hello World!" text; however, in a real application, the "OnPrint" event would typically be much more complex.

```
procedure TMainForm.ReportSystemBeforePrint(Sender: TObject);
begin
    with Sender as TBaseReport do begin
        if ReportSystem.RenderObject = nil then begin
            ReportSystem.BaseReport.PrinterIndex := PrinterCombo.ItemIndex;
        end;
        MarginTop := 0.5;
        MarginBottom := 0.5;
        MarginLeft := 0.5;
        MarginRight := 0.5;
    end; {with ...}
end; {BeforePrint}

procedure TMainForm.ReportSystemPrint(Sender: TObject);
begin
    with Sender as TBaseReport do begin
        SetFont('Arial', 24);
        Print('Hello World!');
    end; {with ...}
```

```
end; {ReportSystemPrint}
```

### Button Events

Finally, all that is left is to code for the "Preview", "Print", and "Fax" buttons. In each case, we specify the report destination and the rendering object. In the case of the "Preview" and "Print" events, the rendering object is set to nil, causing the report system to be rendered by TRvSystem itself. In the "Fax" case, we must set the rendering object to "RenderFax" causing the TRvSystem component to render the report through the TSSRenderFax component. Code for these events are presented below:

```
procedure TMainForm.PreviewButtonClick(Sender: TObject);
begin
  ReportSystem.DefaultDest := rdPreview;
  ReportSystem.RenderObject := nil;
  ReportSystem.Execute;
end; {PreviewButtonClick}

procedure TMainForm.PrintButtonClick(Sender: TObject);
begin
  ReportSystem.DefaultDest := rdPrinter;
  ReportSystem.RenderObject := nil;
  ReportSystem.Execute;
end; {PrintButtonClick}

procedure TMainForm.FaxButtonClick(Sender: TObject);
begin
  LoadComponent;
  ReportSystem.DefaultDest := rdFile;
  ReportSystem.DoNativeOutput := false;
  ReportSystem.RenderObject := RenderFAX;
  ReportSystem.Execute;
end; {FAXButtonClick}
```

# Chapter 3: Component Reference

## The TSSRenderFax Component

The TSSRenderFax component is a rendering component that allows the output from Rave's TRvSystem component to be redirected to the Windows NT/2000/XP faxing platform. Therefore, the component is only supported by Windows NT, 2000, and XP. (Windows 95, 98, and ME are not supported).

In order to use the component, place it on the same form where you have the TRvSystem component, then implement the following code before the ReportSystem's "Execute" method is called:

```
ReportSystem.DefaultDest := rdFile;  
ReportSystem.DoNativeOutput := false;  
ReportSystem.RenderObject := RenderFAX;
```

Where, "ReportSystem" refers to the "TRvSystem" component, and "RenderFAX" refers to the "TSSRenderFax" component. When the ReportSystem's "Execute" method is called, the report will be faxed according to the "RecipientInfo", "SenderInfo", and "CoverPageInfo" properties. If the "RecipientInfo.FaxNumber" property is blank, a setup dialog will optionally appear prompting the user to enter the missing information. An example of this dialog is presented below:

The screenshot shows a "Fax Setup" dialog box with a blue title bar and a close button. The dialog is divided into two main sections: "Cover Page (Recipient)" and "Cover Page (Sender)". The "Cover Page (Sender)" section is active and contains a "General" sub-tab and a "Cover Page (General)" sub-tab. The "Recipient's Info" section has a "Name" field with "Ralph Recipient" and an empty "Fax" field. The "Sender's Info" section has "Name" (Suzi Sender), "Company" (XYZ Company), "Dept" (Programming), and an empty "Bill Code" field. At the bottom, there is a dropdown menu for "Printer for Fax Capture" with "Fax" selected. On the right side, there are "OK" and "Cancel" buttons.

The behavior of this dialog may be controlled by the "SetupInfo" property.

If you are using the TRvProject component to generate reports created by the Rave Visual Designer, you must still include a "TRvSystem" component on the form, and assign the TRvProject.Engine property to that component. Then, set the RvSystem's "RenderObject" property as before.

### **Using TSSRenderFax with C++ Builder**

As mentioned in the tutorial, pathing information must be included in the "Project Options" dialog for each project using Rave and the SoftSand Fax Add-On. Specifically, the Rave library directory and the SSFax library directory must be included in both the "Include Path" and the "Library Path" edit boxes found under the "Directories and Conditionals" tab of the "Project Options" dialog.

### **TSSRenderFax Properties**

```
CoverPageInfo: TFaxCoverPageInfo; {published, read/write}
CoverPageType: TCoverPageType; {published, read/write}
DocumentName: String; {published, read/write}
FaxingSupported: Boolean; {public, read-only}
FaxJobID: DWORD; {public, read-only}
FaxPrinterIndex: Integer; {published, read/write}
FaxPrinterName: String; {published, read/write}
Printing: Boolean; {public, read-only}
RecipientInfo: TFaxRecipientInfo; {published, read/write}
SenderInfo: TFaxSenderInfo; {published, read/write}
ServerName: String; {public, read-only}
SetupInfo: TFaxSetupInfo; {published, read/write}
```

### **TSSRenderFax Methods**

```
procedure LoadFromIniFile(AppIniFile: TIniFile;
    SectionName: String; LoadRecipientInfo: Boolean);

procedure SaveToIniFile(AppIniFile: TIniFile;
    SectionName: String;
    SaveRecipientInfo, UserModified: Boolean);
```

### **TSSRenderFax Events**

```
BeforeSetupDlg: TNotifyEvent;
AfterSetupDlg: TNotifyEvent;
OnDocBegin: TNotifyEvent;
OnDocEnd: TNotifyEvent;
```

# Chapter 4: Alphabetic Reference

## **AfterSetupDlg**

**Applies To:**

TSSRenderFax

**Declaration:**

```
AfterSetupDlg: TNotifyEvent;
```

**Scope:**

Published, Read/Write

**Description:**

The AfterSetupDlg event is triggered after the user has clicked the "OK" button on the embedded setup dialog. This event is not fired if the user clicks "Cancel".

## **BeforeSetupDlg**

**Applies To:**

TSSRenderFax

**Declaration:**

```
BeforeSetupDlg: TNotifyEvent;
```

**Scope:**

Published, Read/Write

**Description:**

BeforeSetupDlg is triggered before the setup dialog would be displayed. It is only triggered under conditions where the setup dialog would appear, i.e. SetupInfo.SetupAction is set to "saAlways" or "saBlank" (and the RecipientInfo.FaxNumber is blank).

## **CoverPageInfo**

**Applies To:**

TSSRenderFax

**Declaration:**

```
property CoverPageInfo: TFaxCoverPageInfo;
```

**Scope:**

Published, Read/Write

**Description:**

CoverPageInfo is a composite property allowing you to specify the information that would appear on an included cover page. Each sub-property corresponds to a field that may be included in the Microsoft Cover Page Template file (.COV) specified in the "CoverPageName" property. Below are listed the sub-properties that comprise "CoverPageInfo":

**CoverPageName:** Specifies the file name and path of the Microsoft cover page template file (.COV) to use for an included cover page. If "CoverPageType" is set to "cptLocal", the included path refers to a location

with respect to the local machine. Alternatively, if "CoverPageType" is set to "cptServer", it refers to a location on the fax server.

**RecName:** Specifies the recipient's name as it would appear on the cover page.

**RecFaxNumber:** Specifies the recipient's fax number as it would appear on the cover page. Note that this may differ from the "RecipientInfo.FaxNumber" field which is actually used to send the fax.

**RecCompany:** Specifies the recipient's company as it would appear on the cover page.

**RecStreetAddress:** Specifies the recipient's street address as it would appear on the cover page.

**RecCity:** Specifies the recipient's city as it would appear on the cover page.

**RecState:** Specifies the recipient's state as it would appear on the cover page.

**RecZip:** Specifies the recipient's zip or postal code as it would appear on the cover page.

**RecCountry:** Specifies the recipient's country as it would appear on the cover page.

**RecTitle:** Specifies the recipient's title or position as it would appear on the cover page.

**RecDepartment:** Specifies the recipient's department as it would appear on the cover page.

**RecOfficeLocation:** Specifies the recipient's office location as it would appear on the cover page.

**RecHomePhone:** Specifies the recipient's home phone as would appear on the cover page.

**RecOfficePhone:** Specifies the recipient's office phone as it would appear on the cover page.

**SdrName:** Specifies the sender's name as it would appear on the cover page.

**SdrFaxNumber:** Specifies the sender's fax number as it would appear on the cover page.

**SdrCompany:** Specifies the sender's company as it would appear on the cover page.

**SdrAddress:** Specifies the sender's address as it would appear on the cover page.

**SdrTitle:** Specifies the sender's title or position as it would appear on the cover page.

**SdrDepartment:** Specifies the sender's department as it would appear on the cover page.

**SdrOfficeLocation:** Specifies the sender's office location as it would appear on the cover page.

**SdrHomePhone:** Specifies the sender's home phone as it would appear on the cover page.

**SdrOfficePhone:** Specifies the sender's office phone as it would appear on the cover page.

**Subject:** Specifies the subject of the fax as it would appear on the cover page.

**Note:** Specifies a multi-line note as it would appear on the cover page. This property is defined as a TStrings component.

### **CoverPageType**

#### **Applies To:**

TSSRenderFax

#### **Declaration:**

```
property CoverPageType: TCoverPageType;
```

#### **Scope:**

Published, Read/Write

#### **Description:**

CoverPageType determines whether a cover page will be included in the fax transmission, and if so, on what machine (local or server) the cover page template is located. It may take on the enumerated values of "cptNone", "cptLocal", and "cptServer" as described below:

**cptNone:** Specifies that no cover page should be included in the fax transmission.

**cptLocal:** Specifies that a cover page should be included, and that the CoverPageName property refers to a file with respect to the local machine.

**cptServer:** Specifies that a cover page should be included and that the CoverPageName property refers to a file with respect to the fax server.

**DocumentName**

**Applies To:**  
TSSRenderFax

**Declaration:**  
property DocumentName: String;

**Scope:**  
Published, Read/Write

**Description:**  
DocumentName specifies the user-friendly name as it will appear in the fax manager queue.

**ESSRenderFaxError**

**Applies To:**  
TSSRenderFax

**Declaration:**  
ESSRenderFaxError = Exception;

**Description:**  
ESSRenderFaxError is the exception type that will be raised upon any error occurring within TSSRenderFax.

**FaxingSupported**

**Applies To:**  
TSSRenderFax

**Declaration:**  
property FaxingSupported: Boolean;

**Scope:**  
Public, Read-Only

**Description:**  
FaxingSupported is a boolean property that may be used to determine whether faxing is supported by the workstation running the parent application. It will return true if it is being run on a Windows NT, 2000, or XP platform and if the "WinFax.dll" can be found. "WinFax.dll" ships with Windows NT/2000/XP and is installed in the System32 directory when faxing support is included in the Windows setup.

**FaxJobID**

**Applies To:**  
TSSRenderFax

**Declaration:**  
property FaxJobID: DWORD;

**Scope:**  
Public, Read-Only

**Description:**  
FaxJobID is the DWORD parameter returned by the Windows operating system when the fax job is submitted to the faxing platform. It remains valid until the component is destroyed or until another fax job is submitted through TSSRenderFax.

**FaxPrinterIndex****Applies To:**

TSSRenderFax

**Declaration:**

```
property FaxPrinterIndex: Integer;
```

**Scope:**

Published, Read/Write

**Description:**

FaxPrinterIndex determines the printer driver used to capture the print information to be faxed. It refers to the position of the desired printer in the VCL's "TPrinter.Printers" list (as defined in "printers.pas"). You may also specify the desired printer driver by the "FaxPrinterName" property. If FaxPrinterName is specified, FaxPrinterIndex will not be used. A value of -1 will use the default fax printer assigned to the local machine.

**FaxPrinterName****Applies To:**

TSSRenderFax

**Declaration:**

```
property FaxPrinterName: String;
```

**Scope:**

Published, Read/Write

**Description:**

FaxPrinterName determines the printer driver used to capture the print information to be faxed. If this property is used, "FaxPrinterIndex" is ignored.

**LoadFromIniFile****Applies To:**

TSSRenderFax

**Declaration:**

```
procedure LoadFromIniFile(AppIniFile: TIniFile;  
    SectionName: String; LoadRecipientInfo: Boolean);
```

**Description:**

LoadFromIniFile will load the TSSRenderFax properties from an .INI file specified by "AppIniFile". The "AppIniFile" object must have already been created before LoadFromIniFile is called. "SectionName" will determine the INI file section that will be used, Properties will be loaded from entries stored beneath this section using the property name as an identifier. "LoadRecipientInfo" will determine whether recipient information is also loaded into the component.

**OnDocBegin****Applies To:**

TSSRenderFax

**Declaration:**

```
OnDocBegin: TNotifyEvent;
```

**Scope:**

Published, Read/Write

**Description:**

OnDocBegin is triggered before the fax rendering process begins. This is, however, after the TRvSystem component has rendered the report.

**OnDocEnd****Applies To:**

TSSRenderFax

**Declaration:**

```
OnDocEnd: TNotifyEvent;
```

**Scope:**

Published, Read/Write

**Description:**

OnDocEnd is triggered after the fax rendering process has completed and been sent to the Windows faxing platform.

**Printing****Applies To:**

TSSRenderFax

**Declaration:**

```
property Printing: Boolean;
```

**Scope:**

Public, Read-Only

**Description:**

Printing is a read-only, run-time property that is set to true while the fax is being rendered from the report data. It is then automatically set to false when the fax job is submitted to the Windows faxing platform.

**RecipientInfo****Applies To:**

TSSRenderFax

**Declaration:**

```
property RecipientInfo: TFaxRecipientInfo;
```

**Scope:**

Published, Read/Write

**Description:**

RecipientInfo is a composite property that specifies the name and fax number of the recipient. The fax number is required to transmit the fax. If RecipientInfo.FaxNumber is blank, and SetupInfo.SetupAction is set to saBlank, the setup dialog will appear at faxing-time for the user to enter the required fax number. The sub-properties of RecipientInfo are presented below:

**Name:** Specifies the recipient's name as it will appear in the status window and faxing logs.

**FaxNumber:** Specifies the fax number where the document will be faxed. The standard dialing rules as set up in the Windows operating system will apply.

### **SaveToIniFile**

**Applies To:**

TSSRenderFax

**Declaration:**

```
procedure SaveToIniFile(AppIniFile: TIniFile;  
    SectionName: String;  
    SaveRecipientInfo, UserModified: Boolean);
```

**Description:**

SaveToIniFile will save the TSSRenderFax properties to an INI file specified by "AppIniFile". The "AppIniFile" object must have already been created before SaveToIniFile is called. "SectionName" will determine the INI file section that will be used. Properties will be saved to entries beneath this section using the property name as an identifier. "SaveRecipientInfo" will determine whether recipient information is saved along with the other properties of TSSRenderFax. If "UserModified" is true, saved values will be taken from those presented in the setup dialog. (These may have been changed by the user at fax-time). If "UserModified" is false, saved values will be taken from the component properties themselves.

### **SenderInfo**

**Applies To:**

TSSRenderFax

**Declaration:**

```
property SenderInfo: TFaxSenderInfo;
```

**Scope:**

Published, Read/Write

**Description:**

SenderInfo is a composite property that specifies the sender information as it will appear in the status window and faxing logs. The sub-properties of SenderInfo are presented below:

**Name:** Specifies the sender's name.

**Company:** Specifies the sender's company.

**Dept:** Specifies the sender's department.

**BillingCode:** Specifies the billing code for the fax transmission.

### **ServerName**

**Applies To:**

TSSRenderFax

**Declaration:**

```
property ServerName: String;
```

**Scope:**

Public, Read-Only

**Description:**

If a fax server is used, this property will return the server name of the fax server used in the last fax job submitted.

**SetupInfo****Applies To:**

TSSRenderFax

**Declaration:**

property SetupInfo: TFaxSetupInfo;

**Scope:**

Published, Read/Write

**Description:**

SetupInfo will determine the behavior and appearance of the embedded setup dialog. The setup dialog will appear as a modal form at faxing-time if SetupAction is set to "saAlways", or "saBlank" (and the RecipientInfo.FaxNumber is blank). Sub-properties of SetupInfo are as follows:

**VisibleTabs:** Specifies the tabs that are visible on the setup dialog. This is a set property that may contain one or more of the following: (vtGeneral, vtCoverPageGeneral, vtCoverPageRecipient, and vtCoverPageSender). These correspond to the "General", "Cover Page (General)", "Cover Page (Recipient)" and "Cover Page (Sender)" tabs of the setup form respectively.

**SetupAction:** Specifies under what conditions the setup dialog should appear. It may take on the enumerated values of "saAlways", "saNever", and "saBlank". As expected, if set to "saAlways", the setup dialog will always appear; if set to "saNever", the setup dialog will never appear; and if set to "saBlank", the setup dialog will only appear if the RecipientInfo.FaxNumber property is blank.

**CoverPageDir:** Specifies the initial directory where the "Open File Dialog" will begin when browsing for cover pages.

**TCoverPageType****Applies To:**

TSSRenderFax

**Declaration:**

TCoverPageType = (cptNone, cptLocal, cptServer);

**Description:**

TCoverPageType is the type definition for the CoverPageType property.

**TFaxCoverPageInfo****Applies To:**

TSSRenderFax

**Declaration:**

```
TFaxCoverPageInfo = class(TPersistent)
published
  property CoverPageName: String; {read/write}
  property RecName: String; {read/write}
  property RecFaxNumber: String; {read/write}
  property RecCompany: String; {read/write}
  property RecStreetAddress: String; {read/write}
  property RecCity: String; {read/write}
  property RecState: String; {read/write}
  property RecZip: String; {read/write}
  property RecCountry: String; {read/write}
  property RecTitle: String; {read/write}
  property RecDepartment: String; {read/write}
```

```

property RecOfficeLocation: String; {read/write}
property RecHomePhone: String; {read/write}
property RecOfficePhone: String; {read/write}
property SdrName: String; {read/write}
property SdrFaxNumber: String; {read/write}
property SdrCompany: String; {read/write}
property SdrAddress: String; {read/write}
property SdrTitle: String; {read/write}
property SdrDepartment: String; {read/write}
property SdrOfficeLocation: String; {read/write}
property SdrHomePhone: String; {read/write}
property SdrOfficePhone: String; {read/write}
property Subject: String; {read/write}
property Note: TStrings; {read/write}
end; {TFaxCoverPageInfo}

```

**Description:**

TFaxCoverPageInfo is the type definition of the CoverPageInfo property.

**TFaxRecipientInfo****Applies To:**

TSSRenderFax

**Declaration:**

```

TFaxRecipientInfo = class(TPersistent)
published
    property Name: String; {read/write}
    property FaxNumber: String; {read/write}
end; {TFaxRecipientInfo}

```

**Description:**

TFaxRecipientInfo is the type definition for the RecipientInfo property.

**TFaxSenderInfo****Applies To:**

TSSRenderFax

**Declaration:**

```

TFaxSenderInfo = class(TPersistent)
published
    property Name: String; {read/write}
    property Company: String; {read/write}
    property Dept: String; {read/write}
    property BillingCode: String; {read/write}
end; {TFaxSenderInfo}

```

**Description:**

TFaxSenderInfo is the type definition of the SenderInfo property.

**TFaxSetupAction****Applies To:**

TSSRenderFax

**Declaration:**

```
TFaxSetupAction = (saAlways, saBlank, saNever);
```

**Description:**

TFaxSetupAction is the type definition for the SetupInfo.SetupAction property.

**TFaxSetupInfo****Applies To:**

TSSRenderFax

**Declaration:**

```
TFaxSetupInfo = class(TPersistent)
published
  property VisibleTabs: TFaxSetupVisibleTabs; {read/write}
  property SetupAction: TFaxSetupAction; {read/write}
  property CoverPageDir: String; {read/write}
end; {TSetupInfo}
```

**Description:**

TFaxSetupInfo is the type definition of the SetupInfo property.

**TFaxSetupTabs****Applies To:**

TSSRenderFax

**Declaration:**

```
TFaxSetupTabs = (vtGeneral, vtCoverPageGeneral,
  vtCoverPageRecipient, vtCoverPageSender);
```

**Description:**

TFaxSetupTabs is the type definition used in the TFaxSetupVisibleTabs set definition of the SetupInfo.VisibleTabs property.

**TFaxSetupVisibleTabs****Applies To:**

TSSRenderFax

**Declaration:**

```
TFaxSetupVisibleTabs = set of TFaxSetupTabs;
```

**Description:**

TFaxSetupVisibleTabs is the type definition of the SetupInfo.VisibleTabs property.



---

# Appendix A: Contact Information

John Sandage may be reached at:

John Sandage  
SoftSand  
1255 Tropical Ave.  
Pasadena, CA 91107  
USA

Phone: (626) 351-1299

E-Mail: [John@softsand.com](mailto:John@softsand.com)



# Index

## -- A --

AfterSetupDlg, 15  
Appendix A, 25

## -- B --

BeforeSetupDlg, 15

## -- C --

Chapter 2, 7  
Chapter 3, 13  
Chapter 4, 15  
Contact Information, 25  
CoverPageDir, 21  
CoverPageInfo, 15  
CoverPageName, 15  
CoverPageType, 16  
cptLocal, 16  
cptNone, 16  
cptServer, 16  
Creating the Main Form, 7  
Creating the Program Code, 10

## -- D --

DocumentName, 17

## -- E --

ESSRenderFaxError, 17

## -- F --

FaxDemo as a C++ Builder Example, 7  
FaxDemo Program, 7  
FaxingSupported, 17  
FaxJobID, 17  
FaxNumber, 19  
FaxPrinterIndex, 18  
FaxPrinterName, 18

## -- G --

Getting Started, 5

## -- I --

Index, 27  
Installation, 6

## -- L --

License Agreement, 5  
LoadFromIniFile, 18

## -- N --

Note, 15

## -- O --

OnDocBegin, 18  
OnDocEnd, 19  
Overview of "SoftSand Fax Add-On for Rave", 5

## -- P --

Printing, 19

## -- R --

RecCity, 15  
RecCompany, 15  
RecCountry, 15  
RecDepartment, 15  
ReceipientInfo, 19  
RecFaxNumber, 15  
RecHomePhone, 15  
RecName, 15  
RecOfficeLocation, 15  
RecOfficePhone, 15  
RecState, 15  
RecStreetAddress, 15  
RecTitle, 15  
RecZip, 15

## -- S --

saAlways, 21  
saBlank, 21  
saNever, 21  
SaveToIniFile, 20  
SdrAddress, 15  
SdrCompany, 15  
SdrDepartment, 15  
SdrFaxNumber, 15  
SdrHomePhone, 15  
SdrName, 15  
SdrOfficeLocation, 15  
SdrOfficePhone, 15

SdrTitle, 15  
SenderInfo, 20  
ServerName, 20  
SetupAction, 21  
SetupInfo, 21  
SSRenderFax Events, 14  
SSRenderFax Methods, 14  
SSRenderFax Properties, 14  
Subject, 15

**-- T --**

TCoverPageType, 21  
TFaxCoverPageInfo, 21  
TFaxRecipientInfo, 22  
TFaxSenderInfo, 22  
TFaxSetupAction, 23  
TFaxSetupInfo, 23  
TFaxSetupTabs, 23

TFaxSetupVisibleTabs, 23  
TSSRenderFax, 13

**-- U --**

Using TSSRenderFax with C++ Builder, 14

**-- V --**

VisibleTabs, 21  
vtCoverPageGeneral, 21  
vtCoverPageRecipient, 21  
vtCoverPageSender, 21  
vtGeneral, 21

**-- W --**

What the Setup Wizard Really Does, 6