Projeto Open Source – ACBrNFe

Manual ACBrNFe

Manual de Orientação ao Desenvolvedor – Configurações, Alimentação e Métodos

Versão 1.04
Dezembro / 2014

Controle de Versões

Versão	Data	Autor	Resumo
1.00	24/10/2014	Italo Jurisato Junior	
1.01	28/10/2014	Elton M Barbosa	Revisão nos textos.
1.02	30/10/2014	Italo Jurisato Junior	Inclusão do item Recomendações
1.02	13/11/2014	Juliomar Marchetti	Revisão no item Recomendações
1.03	02/12/2014	Italo Jurisato Junior	Revisão nos textos.
1.04	18/12/2014	Italo Jurisato Junior	Inclusão de novas propriedades de configuração.

1. Introdução

Este manual traz informações sobre o componente ACBrNFe, tais como a sua configuração, passagem de dados e métodos.

O ACBrNFe é um componente destinado a emissão da NF-e – Nota Fiscal Eletrônica, contempla toda a estrutura do XML publicada nos manuais e notas técnicas publicadas pelo ENCAT no Portal Nacional da NF-e.

É compatível com os certificados A1 e A3 (formato cartão e token) para realizar a assinatura digital. Utiliza para isso as DLLs: CAPICOM e OpenSSL, distribuídas juntamente com os fontes do componente.

Possui métodos que atende todos os *Webservices* disponibilizados pelas SEFAZ-Autorizadoras, SEFAZ-Virtuais e SEFAZ Ambiente Nacional.

A impressão do DANFE, fica a cargo de um segundo componente ACBrNFeDANFExxx (xxx = indica o gerenciador de relatórios). Atualmente esse componente é distribuído para os seguintes gerenciadores de relatórios (*Reports*): Rave, RaveCB, Quick Report, Fast Report e Fortes Report.

O ACBrNFe possui uma propriedade de configuração onde é feita a associação com o componente de impressão do DANFE.

2. Configuração

O componente ACBrNFe possui dezenas de propriedades de configurações que podem ser definidas no Object Inspector ou através de linhas de código.

Duas dessas propriedades são de extrema importância: ModeloDF e VersaoDF, pois é através delas que configuramos o componente para emitir NF-e ou NFC-e e a versão do XML desejada: 2.00 ou 3.10 para a NF-e e 3.00 ou 3.10 para NFC-e.

Observação: a versão 3.00 para a NFC-e não é mais aceita pela SEFAZ desde 01/08/2014 e a versão 2.00 para a NF-e deixara de ser aceita pela SEFAZ a partir de 01/04/2015.

As configurações no componente permitem que os arquivos XML de envio e de retorno sejam salvos em disco ou não, e possam ser organizados em pastas ou não, de acordo com a preferência do desenvolvedor.

Propriedades:

Algumas delas só vão aparecer no Object Inspector dependendo de como o componente foi compilado, ou seja, versão Capicom ou OpenSSL.

Nome	**Valor Padrão**	**Descrição**
AboutACBrNFe	Versão: 0.5.0a	Apresenta a versão do componente
Configuracoes		
Arquivos		
AdicionarLiteral	False	Se True adiciona o literal NFe ao Path.
EmissaoPathNFe	False	Se True se baseia pela data de Emissão.
Name	ArquivosConf	
PastaMensal	False	Se True cria uma estrutura de pastas AnoMes ao Path.
PathCan		Path destinado aos arquivos de cancelamento (em desuso).
PathCCe		Path destinado aos arquivos da Carta de Correção.
PathDPEC		Path destinado aos arquivos de DPEC (em desuso).
PathEvento		
PathInu		Path destinado aos arquivos de Inutilização de numeração.
PathMDe		Path destinado aos arquivos de Manifestação de Documentos eletrônicos (em desuso).
PathNFe		Path destinado aos arquivos de NF-e.
Salvar	False	Se True salva os arquivos XML referente a documentos.
SalvarApenasNFeProcessadas	False	Se True salva as NF-e processadas, ou seja com protocolo de Autorização ou Denegação.

SalvarCCeCanEvento	False	Se True cria uma estrutura de pastas de eventos para os cancelamentos, Carta de correção e Manifestação do Destinatário.
SepararPorCNPJ	False	Se True cria uma estrutura de pastas para separar os XML por CNPJ do Emitente.
SepararPorModelo	False	Se True cria uma estrutura de pastas para separar os XML por Modelo (55 e 65).
Tag	0	Sem funcionalidade predefinida, pode ser usado livremente.
Certificados		
Certificado		Path do Certificado, somente usado no OpenSSL.
CNPJ		Retorna o CNPJ do Certificado.
DataVenc		Retorna a Data de Vencimento do Certificado, somente Capicom.
Name	CertificadosConf	
NumeroSerie		Numero de Série do Certificado, somente Capicom.
Senha		Senha do Certificado (normalmente não é informado).
SubjectName		Retorna o Nome da Entidade do Certificado, somente Capicom.
Tag	0	Sem funcionalidade predefinida, pode ser usado livremente.
Geral		
AtualizarXMLCancelado	True	Se True faz com que o XML da NF-e seja atualizado após o cancelamento substituindo o protocolo de autorização pelo de cancelamento (em desuso).
ExibirErroSchema	False	Se True exibe as mensagem de erro ao validar o XML antes do seu envio.

FormaEmissao	teNormal	Formato ou tipo de emissão da NF-e o valor padrão é teNormal, mas deve ser alterado para emitir em contingência.
FormatoAlerta	TAG:%TAGNIVEL% ID:%ID%/%TAG%(%DESCRICAO%) - %MSG%.	Formatação da mensagem de erro a ser exibida ao validar o XML.
IdToken		Identificador do CSC – Código de Segurança do Contribuinte (tamanho de 6 dígitos).
IniFinXMLSECAutomatico	True	Se True se utilizada xmlsec library.
ModeloDF	moNFe	Modelo do documento fiscal: moNFe para NF-e ou moNFCe para NFC-e.
Name	GeralConf	
PathSalvar		Path destinado aos arquivos de envio e retorno da SEFAZ.
PathSchemas		Path destinado aos arquivos XSD (Schemas) utilizados para validar o XML.
RetirarAcentos	True	Remove os acentos das vogais ao gerar o XML.
Salvar	False	Se True salva os arquivos de envio e retorno da SEFAZ.
Tag	0	Sem funcionalidade predefinida, pode ser usado livremente.
Token		Código de Segurança do Contribuinte (tamanho de 36 alfanumerico).
ValidarDigest	True	Se True compara o Value Digest da assinatura com o do protocolo de autorização, se forem iguais o protocolo será adicionado ao XML da NF-e.
VersaoDF	ve200	Versão do documento fiscal: ve200 para 2.00 e ve310 para 3.10

Name	Configuracoes	
Tag	0	Sem funcionalidade predefinida, pode ser usado livremente.
WebServices		
AguardarConsultaRet	0	Tempo em milissegundos de espera antes de realizar a primeira consulta após o envio da NF-e para SEFAZ.
AjustaAguardarConsultaRet	False	Se True ajusta o tempo de espera baseado no tempo de retorno do numero do recibo após o envio.
Ambiente	taHomologacao	Defino o ambiente a ser utilizado para o envio (Homologação ou Produção).
IntervaloTentativas	1000	Tempo em milissegundos de espera entre uma consulta e outra após o envio da NF-e para SEFAZ.
Name	WebServicesConf	
ProxyHost		Define o Host do Proxy
ProxyPass		Define a Senha do Proxy
ProxyPort		Define a Porta do Proxy
ProxyUser		Define o nome do usuário do Proxy
Salvar	False	Se True salva os arquivos de envio e retorno da SEFAZ com a estrutura Soap.
Tag	0	Sem funcionalidade predefinida, pode ser usado livremente.
Tentativas	5	Numero de tentativas de consultas a serem realizadas após o envio da NF-e para SEFAZ.
UF	SP	Sigla da UF da SEFAZ-Autorizadora
Visualizar	False	Se True será apresentado na tela um form com os dados dos retornos ao enviar uma solicitação a SEFAZ.
DANFE		Usado para associar o componente ACBrNFe ao

		componente de impressão de DANFE.
Name	ACBrNFe1	Nome do componente podendo ser alterado conforme a necessidade.
Tag	0	Sem funcionalidade predefinida, pode ser usado livremente.

3. Alimentação

O componente ACBrNFe possui uma classe que é uma coleção de itens e cada item se refere a uma NF-e. A princípio, podemos adicionar centenas de notas a essa coleção, mas o componente somente vai gerar e enviar um lote com no máximo 50 notas (limite máximo estabelecido pela SEFAZ).

Cada item da coleção possui centenas de propriedades que tem como objetivo receber os dados a serem utilizados na geração do XML.

Cada propriedade representa uma TAG do XML e segue a mesma nomenclatura definida nos manuais e notas técnicas publicadas pelo ENCAT no Portal Nacional da NF-e.

O programa exemplo: ACBrNFe_demo que encontra-se na pasta: *...\Exemplos\ACBrNFe2\Delphi* possui uma procedure chamada *GerarNFe* que exemplifica a alimentação dessas propriedades com os dados pertinentes a venda.

A procedure *GerarNFe* adiciona apenas uma nota para a coleção de itens chamada *NotasFiscais*. Mas se a mesma estiver dentro de um *loop*, serão adicionadas quantas notas for o número de vezes desse loop.

4. Métodos

O desenvolvedor pode optar por utilizar uma *procedure* ou *function* que automatiza dois ou mais métodos ou utilizar os métodos diretamente. Neste caso fica a cargo do desenvolvedor efetuar o efetivo controle da execução dos mesmos.

Os métodos possuem diversas propriedades de entrada e de retorno, onde o desenvolvedor terá que passar alguns dados para que o mesmo execute a sua função corretamente, por fim ler as de retorno para que seja feito o tratamento adequado, apresentando para o usuário e ou armazenando no banco de dados.

a. Funções

ACBrNFe1.Enviar(ALote: Integer/String; Imprimir: Boolean = True; Sincrono: Boolean = False): Boolean;

A função Enviar é capaz de gerar o XML baseado nos dados que foram alimentados ao componente, dados estes pertinentes a venda do(s) produto(s), assinar o XML digitalmente através do certificado digital do emitente previamente instalado e configurado (vide configuração), validar o XML com base nos arquivos XSD (Schemas) e enviar para SEFAZ caso a validação esteja OK, caso contrario apresenta o erro de validação a aborta o envio.

Se enviado aguarda o retorno com o numero do recibo, realiza a consulta e aguarda o retorno com o resultado do processamento da NF-e pela SEFAZ.

Se constar no retorno que a NF-e foi autorizada, o XML assinado recebe as TAGs retornadas pela SEFAZ juntamente com o protocolo de autorização.

A função Enviar possui três parâmetros:

ALote que pode ser um numero (integer) ou (String) utilizado para identificar o numero do lote a ser enviado. Um lote pode conter de 1 até 50 NF-e.

Imprimir pode ser omitido uma vez que possui um valor padrão igual a True, faz com que no final de todo o processo o DANFE seja impresso no papel. Se desejar informe False como sendo o valor do segundo parâmetro para que o DANFE não seja impresso automaticamente.

Síncrono pode ser omitido uma vez que possui um valor padrão igual a False, faz com que o envio neste caso seja no modo assíncrono. Só passe o valor True como terceiro parâmetro caso o lote a ser enviado tenha apenas **UMA** NFC-e.

Para Lotes de NF-e devemos deixar como False.

ACBrNFe1.Consultar: Boolean;

Essa função é muito útil quando ocorre algum problema após o envio da NF-e para SEFAZ e o XML assinado fica sem o protocolo de autorização e na SEFAZ a mesma encontra-se autorizada.

A função Consultar realiza uma consulta a SEFAZ e obtém como resposta a situação atual da NF-e informada.

É aconselhável alimentar o componente com os dados da NF-e a ser consultada, lendo o XML da mesma através do LoadFromFile (por exemplo), desta forma se o XML estiver assinado, receberá as TAGs referentes ao protocolo de autorização caso esta tenha sido autorizada.

ACBrNFe1.EnviarEventoNFe(idLote : Integer): Boolean;

Através dessa função podemos enviar para SEFAZ qualquer tipo de evento disponível para a NF-e e NFC-e, por exempo: Cancelamento, Carta de Correção, EPEC, Manifestação do Destinatário.

É preciso inicialmente alimentar o componente com os dados pertinentes ao evento desejado.

Essa função possui um parâmetro chamado **idLote** onde devemos informar o numero do lote de eventos a ser enviado para SEFAZ.

Se o componente estiver configurado para salvar os arquivos de documentos, será salvo em disco o arquivo <ID do Evento>-procEventoNFe.xml. Neste arquivo temos a solicitação e o retorno da SEFAZ com o protocolo de autorização e o status que o evento foi registrado.

ACBrNFe1.ConsultaNFeDest(CNPJ: String; IndNFe: TpcnIndicadorNFe; IndEmi: TpcnIndicadorEmissor; ultNSU: String): Boolean;

Através dessa função podemos realizar uma consulta para obter uma lista de notas emitidas contra o CNPJ do destinatário. Desta forma é possível realizar a manifestação, lembrando que a manifestação do destinatário é um evento, logo devemos utilizar a função **EnviarEventoNFe** para realizar a manifestação.

A função possui quatro parâmetros:

CNPJ é o CNPJ do destinatário.

IndNFe é o indicador de NF-e consultadas, que pode receber 3 valores: inTodas (todas as notas), inSemManifestacaoComCiencia (somente as NF-e que ainda não tiveram manifestação do destinatário) e inSemManifestacaoSemCiencia (idem ao anterior, incluído as NF-e que também não tiveram a ciência da operação).

IndEmi é o indicador do Emissor da NF-e, que pode receber 2 valores: ieTodos (todos os emitentes / remetentes) e ieRaizCNPJDiferente (somente as NF-e emitidas por emissores / remetentes que não tenham o mesmo CNPJ-base do destinatário desta forma é excluído as notas de transferencia entre as filiais).

ultNSU é o últino NSU recebido, caso seja informado zero, ou um NSU muito antigo, a consulta retornará unicamente as notas que tenham sido recepcionadas nos últimos 15 dias.

ACBrNFe1.Download: Boolean;

Esta função tem como objeto realizar o Download de até 10 NF-e, devemos alimentar o componente com o CNPJ do destinatário e com a(s) chave(s) da(s) NF-e.

Mas a SEFAZ só libera para Download as notas que foram manifestadas pelo destinatário e a mesma recomenda que não se deve realizar o Download da totalidade das notas recebidas.

ACBrNFe1.AdministrarCSC(ARaizCNPJ: String; AIndOP: TpcnIndOperacao; AIdCSC: Integer; ACodigoCSC: String): Boolean;

Essa função tem como objeto administrar do CSC – Código de Segurança do Contribuinte, utilizado para gerar o QR-Code no DANFE da NFC-e, portanto é destinado a somente as empresas que emitem a NFC-e – Nota Fiscal ao Consumidor Eletrônica.

Com ela podemos consultar um CSC ativo, solicitar um novo e revogar um CSC ativo.

A função possui quatro parâmetros:

ARaizCNPJ é a raiz do CNPJ do emitente.

AIndOP é o indicador de operação, pode receber 3 valores: ioConsultaCSC (Consulta CSC Ativos), ioNovoCSC (Solicita um novo CSC) e ioRevogaCSC (Revoga CSC Ativo).

AIdCSC é o numero do identificador do CSC a ser revogado.

ACodigoCSC é o código alfanumérico do CSC a ser revogado.

ACBrNFe1.DistribuicaoDFe(AcUFAutor: Integer; ACNPJCPF, AultNSU, ANSU: String): Boolean;

Essa função tem como objeto realizar uma consulta das NF-e emitidas, ela vem para substituir o **ConsultaNFeDes**, pois não se restringe somente ao destinatário, ou seja, tem como serviço a distribuição de informações resumidas e documentos fiscais eletrônicos de interesse de um ator, seja esta pessoa física ou jurídica.

Ela permite que um ator da NF-e tenha acesso aos documentos fiscais eletrônicos (DF-e) e informações resumidas que não tenham sido gerados por ele e que sejam de seu interesse. Pode ser executado por qualquer ator de NF-e, Pessoa Física ou Jurídica, que possua um certificado digital. No caso de Pessoa Jurídica, a empresa será autenticada pelo CNPJ base e poderá realizar a consulta com qualquer CNPJ da empresa desde que o CNPJ base consultado seja o mesmo do certificado digital.

A função possui quatro parâmetros:

AcUFAutor é o código da UF do ator.

ACNPJCPF é o CNPJ ou CPF do ator.

AultNSU é Último NSU recebido pelo ator. Caso seja informado com zero, ou com um NSU muito antigo, a consulta retornará unicamente as informações resumidas e documentos fiscais eletrônicos que tenham sido recepcionados pelo Ambiente Nacional nos últimos 3 meses.

ANSU é Número Sequencial Único. Geralmente esta consulta será utilizada quando identificado pelo interessado um NSU faltante. O Web Service retornará o documento ou informará que o NSU não existe no Ambiente Nacional. Assim, esta consulta fechará a lacuna do NSU identificado como faltante.

ACBrNFe1.NotasFiscais.ValidaAssinatura(out Msg: String): Boolean;

Essa função valida a assinatura de um XML previamente carregado no componente, retornando a mensagem de erro na propriedade **Msg**.

ACBrNFe1.NotasFiscais.ValidaRegrasdeNegocios: Boolean;

Essa função valida um XML previamente carregado no componente com base as regras de negócio da SEFAZ.

ACBrNFe1.NotasFiscais.Add: NotaFiscal;

Usado para adicionar uma nova nota para ser alimentada com os dados pertinentes a venda, a principio podemos adicionar dezenas de notas, mas a SEFAZ só aceita lotes com no máximo 50 notas. A nota adicionada será a última da lista.

ACBrNFe1.NotasFiscais.Insert(Index: Integer): NotaFiscal;

Usado para inserir em uma determinada posição uma nova nota para ser alimentada com os dados pertinentes a venda.

ACBrNFe1.NotasFiscais.LoadFromFile(CaminhoArquivo: String; AGerarNFe: Boolean = True): Boolean;

Essa função carrega o componente com os dados lidos do XML salvo em disco.

A função possui dois parâmetros:

CaminhoArquivo é o caminho mais o nome do arquivo XML a ser lido.

AGerarNFe é opcional se seu valor padrão é True, isso faz com que após a leitura o XML é gerado novamente, se desejar apenas ler, devemos informar o valor False a esse parâmetro.

ACBrNFe1.NotasFiscais.LoadFromStream(Stream: TStringStream; AGerarNFe: Boolean = True): Boolean;

Essa função carrega o componente com os dados lidos de um Stream útil quando o conteúdo do XML esta armazenado no banco de dados.

A função possui dois parâmetros:

Stream é o conteúdo do campo do banco de dados.

AGerarNFe é opcional se seu valor padrão é True, isso faz com que após a leitura o XML é gerado novamente, se desejar apenas ler, devemos informar o valor False a esse parâmetro.

ACBrNFe1.NotasFiscais.LoadFromString(AString: String; AGerarNFe: Boolean = True): Boolean;

Essa função carrega o componente com os dados lidos de uma String útil quando o conteúdo do XML esta armazenado em uma string.

A função possui dois parâmetros:

String é o conteúdo da variável que contem o conteúdo do XML.

AGerarNFe é opcional se seu valor padrão é True, isso faz com que após a leitura o XML é gerado novamente, se desejar apenas ler, devemos informar o valor False a esse parâmetro.

ACBrNFe1.NotasFiscais.SaveToFile(PathArquivo: String = ''; SalvaTXT : Boolean = False): Boolean;

Essa função salva em disco as nota carregadas no componente no formato XML, opcionalmente salva também no formato TXT.

A função possui dois parâmetros:

PathArquivo é o caminho onde o arquivo será salvo, se informar uma string vazia será salvo segundo a configuração do componente.

SalvaTXT se o valor for True será salvo também as notas no formato TXT, o valor padrão é False.

ACBrNFe1.NotasFiscais.SaveToTXT(PathArquivo: String = ''): Boolean;

Essa função salva em disco as nota carregadas no componente no formato TXT.

A função possui um parâmetro:

PathArquivo é o caminho onde o arquivo será salvo, se informar uma string vazia será salvo segundo a configuração do componente.

ACBrNFe1.NotasFiscais.GetNamePath: String;

Esta função no momento retorna apenas uma string contendo "**NotaFiscal**".

b.Procedimetos

ACBrNFe1.SetStatus(const stNewStatus : TStatusACBrNFe);

O componente possui diversos status listados abaixo, com esse procedimento podemos definir um determinado status. Útil para aqueles desenvolvedores que desejam apresentar na tela uma mensagem de andamento do processo.

Lista de status disponíveis: stIdle, stNFeStatusServico, stNFeRecepcao, stNFeRetRecepcao, stNFeConsulta, stNFeCancelamento, stNFeInutilizacao, stNFeRecibo, stNFeCadastro, stNFeEmail, stNFeEnvDPEC, stNFeConsultaDPEC, stNFeCCe, stNFeEvento, stConsNFeDest, stDownloadNFe, stAdmCSCNFCe, stDistDFeInt, stEnvioWebService

ACBrNFe1.ImprimirEvento;

Imprimi um evento carregado no componente.

ACBrNFe1.ImprimirEventoPDF;

Gera e salva a imagem do evento carregado no componente no formato PDF.

ACBrNFe1.ImprimirInutilizacao;

Imprime a Inutilização de numeração carregada no componente.

ACBrNFe1.ImprimirInutilizacaoPDF;

Gera e salva a imagem da inutilização de numeração carregada no componente no formato PDF.

ACBrNFe1.EnviarEmailEvento(const sSmtpHost, sSmtpPort, sSmtpUser, sSmtpPasswd, sFrom, sTo, sAssunto: String; sMensagem : TStrings; SSL : Boolean; EnviaPDF: Boolean = true; sCC: TStrings = nil; Anexos:TStrings=nil; PedeConfirma: Boolean = False; AguardarEnvio: Boolean = False;

NomeRemetente: String = ''; TLS : Boolean = True; UsarThread: Boolean = True; HTML: Boolean = False);

Envia por email o XML (anexo) de um evento carregado no componente opcionalmente pode-se enviar em anexo a imagem em PDF.

O procedimento possui diversos parâmetros:

sSmtpHost é o endereço de saída do servidor de e-mail.

sSmtpPort é a porta de saída usada pelo servidor de e-mail.

sSmtpUser é o nome do usuário usado pelo servidor de e-mail.

sSmtpPasswd é a senha do usuário usado pelo servidor de e-mail.

sFrom é o endereço de e-mail do remetente.

sTo é o endereço de e-mail do destinatário.

sAssunto é um pequeno texto que identifica o e-mail.

sMensagem é o texto da mensagem que compõe o e-mail.

SSL se True o e-mail será enviado usando o SSL – Secure Socket Layer, troca de mensagens segura.

EnviarPDF Se True anexa automaticamente o PDF do DANFE.

sCC Permite incluir uma lista de endereços de e-mail de outros destinatários – Com Cópia.

Anexos Permite incluir uma lista de Arquivos (Path + Nome) a serem anexados ao e-mail.

PedeConfirma Se True pede confirmação de recebimento do destinatário.

AguardaEnvio Se True só finaliza o procedimento após o fim do envio do e-mail.

NomeRemente Nome do remetente do e-mail.

TLS Se True o e-mail será enviando usando o TLS – Transport Layer Security – protocolo criptográfico usado no envio de e-mail.

UsarThread Se True se utiliza do encadeamento de execução para enviar e-mail.

HTML Se True reconhece o conteúdo de **sMensagem** como sendo um conteúdo no formato HTML.

ACBrNFe1.EnviaEmail(const sSmtpHost, sSmtpPort, sSmtpUser, sSmtpPasswd, sFrom, sTo, sAssunto: String; sMensagem : TStrings; SSL : Boolean; sCC: TStrings = nil; Anexos:TStrings=nil; PedeConfirma: Boolean = False; AguardarEnvio: Boolean = False; NomeRemetente: String = ''; TLS : Boolean = True; StreamNFe : TStringStream = nil; NomeArq : String = ''; UsarThread: Boolean = True; HTML: Boolean = False);

Procedimento genérico para envio de arquivos por e-mail.

O procedimento possui diversos parâmetros:

sSmtpHost é o endereço de saída do servidor de e-mail.

sSmtpPort é a porta de saída usada pelo servidor de e-mail.

sSmtpUser é o nome do usuário usado pelo servidor de e-mail.

sSmtpPasswd é a senha do usuário usado pelo servidor de e-mail.

sFrom é o endereço de e-mail do remetente.

sTo é o endereço de e-mail do destinatário.

sAssunto é um pequeno texto que identifica o e-mail.

sMensagem é o texto da mensagem que compõe o e-mail.

SSL se True o e-mail será enviado usando o SSL – Secure Socket Layer, troca de mensagens segura.

sCC Permite incluir uma lista de endereços de e-mail de outros destinatários – Com Cópia.

Anexos Permite incluir uma lista de Arquivos (Path + Nome) a serem anexados ao e-mail.

PedeConfirma Se True pede confirmação de recebimento do destinatário.

AguardaEnvio Se True só finaliza o procedimento após o fim do envio do e-mail.

NomeRemente Nome do remetente do e-mail.

TLS Se True o e-mail será enviando usando o TLS – Transport Layer Security – protocolo criptográfico usado no envio de e-mail.

StreamNFe Anexa o arquivo lido de um Stream.

NomeArq Nome do arquivo lido de um Stream.

UsarThread Se True se utiliza do encadeamento de execução para enviar e-mail.

HTML Se True reconhece o conteúdo de **sMensagem** como sendo um conteúdo no formato HTML.

ACBrNFe1.NotasFiscais.GerarNFe;

Gera o XML da NF-e, o componente deve estar alimentado com os dados pertinentes a venda.

ACBrNFe1.NotasFiscais.Assinar;

Gera, assina e salva em disco uma NF-e / NFC-e, o componente deve estar alimentado com os dados pertinentes a venda.

ACBrNFe1.NotasFiscais.Valida;

O Valida se utiliza dos arquivos XSD (schemas) para realizar a validação de um XML assinado, se o XML não estiver assinado é executado o procedimento Assinar automaticamente.

ACBrNFe1.NotasFiscais.Imprimir;

Imprime o DANFE de todas as NF-e / NFC-e previamente carregadas no componente.

ACBrNFe1.NotasFiscais.ImprimirResumido;

Imprimi o DANFE NFCe resumido de todas as NFC-e previamente carregadas no componente.

ACBrNFe1.NotasFiscais.ImprimirPDF;

Gera e salva a imagem do DANFE de todas as NF-e / NFC-e previamente carregadas no componente no formato PDF.

ACBrNFe1.NotasFiscais.ImprimirResumidoPDF;

Gera e salva a imagem do DANFE NFC-e resumido de todas as NFC-e previamente carregadas no componente no formato PDF.

ACBrNFe1.NotasFiscais.Items[Index: Integer**].EnviarEmail**(const sSmtpHost, sSmtpPort, sSmtpUser, sSmtpPasswd, sFrom, sTo, sAssunto: String; sMensagem : TStrings; SSL : Boolean; EnviaPDF: Boolean = True; sCC: TStrings = nil; Anexos:TStrings=nil; PedeConfirma: Boolean = False; AguardarEnvio: Boolean = False; NomeRemetente: String = ''; TLS : Boolean = True; UsarThread: Boolean = True; HTML: Boolean = False);

Procedimento para envio por e-mail do XML (anexo) da NF-e ao destinatário carregado no componente, opcionalmente podendo anexar também o PDF do DANFE.

Como o componente comporta uma lista de notas devemos indicar no parâmetro de **Items** qual é a nota que desejamos enviar por e-mail.

O procedimento possui diversos parâmetros:

sSmtpHost é o endereço de saída do servidor de e-mail.

sSmtpPort é a porta de saída usada pelo servidor de e-mail.

sSmtpUser é o nome do usuário usado pelo servidor de e-mail.

sSmtpPasswd é a senha do usuário usado pelo servidor de e-mail.

sFrom é o endereço de e-mail do remetente.

sTo é o endereço de e-mail do destinatário.

sAssunto é um pequeno texto que identifica o e-mail.

sMensagem é o texto da mensagem que compõe o e-mail.

SSL se True o e-mail será enviado usando o SSL – Secure Socket Layer, troca de mensagens segura.

EnviarPDF Se True anexa automaticamente o PDF do DANFE.

sCC Permite incluir uma lista de endereços de e-mail de outros destinatários – Com Cópia.

Anexos Permite incluir uma lista de Arquivos (Path + Nome) a serem anexados ao e-mail.

PedeConfirma Se True pede confirmação de recebimento do destinatário.

AguardaEnvio Se True só finaliza o procedimento após o fim do envio do e-mail.

NomeRemente Nome do remetente do e-mail.

TLS Se True o e-mail será enviando usando o TLS – Transport Layer Security – protocolo criptográfico usado no envio de e-mail.

UsarThread Se True se utiliza do encadeamento de execução para enviar e-mail.

HTML Se True reconhece o conteúdo de **sMensagem** como sendo um conteúdo no formato HTML.

5. Nomes dos Arquivos XML e PDF

Os nomes dos arquivos XML e PDF seguem o padrão de nomes estipulado pelo ENCAT nos manuais e notas técnicas disponibilizados no Portal Nacional da NF-e.

Arquivos XML da NF-e e NFC-e:

Pedido de Consulta do Status do Serviço	<AAAAMMDDHHMMSS>-ped-sta.xml	
Status do Serviço	<AAAAMMDDTHHMMSS>-sta.xml	
Envio de Lote de NF-e	<numLote>-env-lot.xml	
Recibo	<numLote>-rec.xml	
Retorno do processamento síncrono	<numLote>-pro-lot.xml	
Pedido do Resultado do Processamento do Lote	<numRecibo>-ped-rec.xml	
Resultado do Processamento do Lote	<numRecibo>-pro-rec.xml	
NF-e	<chave>-nfe.xml	
Pedido de Consulta Situação Atual	<chave>-ped-sit.xml	
Situação Atual da NF-e	<chave>-sit.xml	
Pedido de Inutilização de Numeração	<ID de inutilização>-ped-inu.xml	
Inutilização de Numeração	<ID de inutilização>-inu.xml	
Compartilhamento de Inutilização de Numeração	<ID de inutilização>-procInutNFe.xml	
Pedido de Registro de Evento	<chave>-ped-eve.xml	
Registro de Evento	<chave>-eve.xml	
Compartilhamento de Registro de Evento	<ID do Evento>-procEventoNFe.xml	
Pedido de Consulta de NF-e Destinadas	<AAAAMMDDHHMMSS>-con-nfe-dest.xml	
Resultado da consulta	<AAAAMMDDHHMMSS>-nfe-dest.xml	
Pedido de Download de NF-e	<AAAAMMDDHHMMSS>-ped-down-nfe.xml	
Retorno do download	<AAAAMMDDHHMMSS>-down-nfe.xml	
Pedido de CSC de NFC-e	<AAAAMMDDHHMMSS>-ped-csc.xml	
Retorno do pedido	<AAAAMMDDHHMMSS>-csc.xml	
Pedido de Consulta de Distribuicao de DF-e	<AAAAMMDDHHMMSS>-con-dist-dfe.xml	
Resultado da consulta	<AAAAMMDDHHMMSS>-dist-dfe.xml	

Arquivos PDF da NF-e e NFC-e:

DANFE	<chave>-nfe.pdf	
DAEvento	<ID do Evento>-procEventoNFe.pdf	
DAInutilizacao	<ID de inutilização>-procInutNFe.pdf	

<ID de inutilização> = <UF + Ano + CNPJ do emitente + Modelo + Série + Núm. Inicial + Núm. Final>
<ID do Evento> = <Tipo de Evento + Chave + Numero Seqüencial do Evento>

Arquivos XML da NF-e e NFC-e (em desuso):

Pedido de Cancelamento	<chave>-ped-can.xml	
Cancelamento de NF-e	<chave>-can.xml	
Compartilhamento de Cancelamento	<chave>-procCancNFe.xml	

Arquivos XML da NF-e e NFC-e (a ser implementado):

Denegação de Uso	<chave>-den.xml	

6. Recomendações

Mantenha todos os fontes de todas as pastas atualizados, procure sempre fazer uma cópia dos fontes atuais e baixar a atualização.

A princípio não há necessidade de desinstalar e instalar novamente os componentes após uma atualização dos fontes.

Aconselhamos sempre compilar as aplicações utilizando-se da opção Build, pois esta recria todas as DCU mesmo dos fontes que não sofreram alteração.

Altamente recomendado utilizar o ACBrInstall para fazer novas instalações e recompilações de pacotes já existentes na IDE após atualizar o SVN! Simples motivo que o instalador já separa os arquivos em pastas corretas para cada Delphi caso exista mais de um no micro! Evitando erros e centralizando em um único local.