

Projeto Open Source – ACBrMDFe



Manual ACBrMDFe

Manual de Orientação ao Desenvolvedor – Configurações,
Alimentação e Métodos

Versão 1.00
Fevereiro / 2015

Controle de Versões

Versão	Data	Autor	Resumo
1.00	02/02/2015	Italo Jurisato Junior	

1. Introdução

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

O ACBrMDFe é um componente destinado a emissão do MDF-e – Manifesto Eletrônico de Documentos Fiscais, contempla toda a estrutura do XML contida nos manuais e notas técnicas publicadas pelo ENCAT no Portal Nacional do MDF-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 DAMDFE, fica a cargo de um segundo componente ACBrMDFeDAMDFExxx (xxx = indica o gerenciador de relatórios). Atualmente esse componente é distribuído para os seguintes gerenciadores de relatórios (*Reports*): Quick Report, Fast Report e Fortes Report.

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

2. Configuração

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

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
AboutACBrMDFe	Versão: 0.8.0	Apresenta a versão do componente
Configuracoes		
Arquivos		
AdicionarLiteral	False	Se True adiciona o literal MDFe ao Path.
EmissaoPathMDFe	False	Se True se baseia pela data de Emissão.
Name	ArquivosConf	
PastaMensal	False	Se True cria uma estrutura de pastas AnoMes ao Path.
PathEvento		Path destinado aos arquivos de Eventos.
PathMDFe		Path destinado aos arquivos de MDF-e.
Salvar	False	Se True salva os arquivos XML referente a documentos.
SalvarApenasMDFeProcessados	False	Se True salva os MDF-e processados, ou seja, com protocolo de Autorização ou Denegação.
SalvarEvento	False	Se True cria uma estrutura de pastas de eventos para os cancelamentos e Encerramentos.
SepararPorCNPJ	False	Se True cria uma estrutura de pastas para separar os XML por CNPJ do Emitente.
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 MDF-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 MDF-e o valor padrão é teNormal, mas deve ser alterado para emitir em contingência.
FormatoAlerta	TAG:%TAGNIVEL% ID:%ID%/TAG%(%DESCRICA0%) - %MSG%.	Formatação da mensagem de erro a ser exibida ao validar o XML.
IniFinXMLSECAutomatico	True	Se True se utilizada xmlsec library, somente usado no OpenSSL.
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.
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 do MDF-e.
VersaoDF	ve100	Versão do documento fiscal: ve100 para 1.00
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 do MDF-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	Define 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 do MDF-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 do MDF-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.
DAMDFE		Usado para associar o componente ACBrMDFe ao componente de impressão de DAMDFE.
Name	ACBrMDFe1	Nome do componente podendo ser alterado conforme a necessidade.
Tag	0	Sem funcionalidade predefinida, pode ser usado livremente.

3. Alimentação

O componente ACBrMDFe possui uma classe que é uma coleção de itens e cada item se refere a um MDF-e. A princípio, podemos adicionar centenas de Manifestos a essa coleção, mas o componente somente vai gerar e enviar um lote com no máximo 1 Manifesto (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 do MDF-e.

O programa exemplo: ACBrMDFe_demo que encontra-se na pasta: ...\\Exemplos\\ACBrMDFe\\Delphi possui uma procedure chamada *GerarMDFe* que exemplifica a alimentação dessas propriedades com os dados pertinentes ao manifesto.

A procedure *GerarMDFe* adiciona apenas um Manifesto para a coleção de itens chamada *Manifestos*.

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

ACBrMDFe1.Enviar(ALote: Integer/String; Imprimir: Boolean = True): Boolean;

A função Enviar é capaz de gerar o XML baseado nos dados que foram alimentados ao componente, dados estes pertinentes ao manifesto, 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 apresentará o erro de validação e abortará o envio.

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

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

A função Enviar possui dois parâmetros:

ALote que pode ser um numero (integer) ou (String) utilizado para identificar o numero do lote a ser enviado. Um lote pode conter apenas 1 MDF-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 DAMDFE seja impresso no papel. Se desejar informe False como sendo o valor do segundo parâmetro para que o DAMDFE não seja impresso automaticamente.

ACBrMDFe1.Consultar: Boolean;

Essa função é muito útil quando ocorre algum problema após o envio do MDF-e para SEFAZ e o XML assinado fica sem o protocolo de autorização e na SEFAZ o mesmo encontra-se autorizado.

A função Consultar realiza uma consulta a SEFAZ e obtém como resposta a situação atual do MDF-e informado.

É aconselhável alimentar o componente com os dados do MDF-e a ser consultado, lendo o XML do mesmo através do LoadFromFile (por exemplo), desta forma se o XML estiver assinado, receberá as TAGs referentes ao protocolo de autorização caso este tenha sido autorizado.

ACBrMDFe1.ConsultarMDFeNaoEnc(ACNPJ: String): Boolean;

Através dessa função podemos consultar os MDF-e enviados e não encerrados.

Essa função possui um parâmetro chamado **ACNPJ** onde devemos informar o CNPJ do emitente do MDF-e.

Se existem um ou mais será retornado uma lista contendo a chave e o numero de protocolo de autorização.

ACBrMDFe1.EnviaEventoMDFe(idLote : Integer): Boolean;

Através dessa função podemos enviar para SEFAZ qualquer tipo de evento disponível para o MDF-e, por exemplo: Cancelamento, Encerramento e Inclusão de Condutor.

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>-procEventoMDFe.xml. Neste arquivo temos a solicitação e o retorno da SEFAZ com o protocolo de autorização e o status que o evento foi registrado.

ACBrMDFe1.Manifestos.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**.

ACBrMDFe1.Manifestos.Add: Manifesto;

Usado para adicionar um novo Manifesto para ser alimentado com os dados pertinentes ao manifesto, a principio podemos adicionar dezenas de Manifestos, mas a SEFAZ só aceita lotes com no máximo 1 Manifesto. O Manifesto adicionado será o último da lista.

ACBrMDFe1.Manifestos.Insert(Index: Integer): Manifesto;

Usado para inserir em uma determinada posição um novo Manifesto para ser alimentado com os dados pertinentes ao manifesto.

**ACBrMDFe1.Manifestos.LoadFromFile(CaminhoArquivo: String;
AGerarMDFe: 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.

AGerarMDFe é opcional e 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.

**ACBrMDFe1.Manifestos.LoadFromStream(Stream: TStringStream;
AGerarMDFe: 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.

AGerarMDFe é opcional e 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.

ACBrMDFe1.Manifestos.LoadFromString(AString: String; AGerarMDFe: 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.

AGerarMDFe é opcional e 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.

ACBrMDFe1.Manifestos.SaveToFile(PathArquivo: String = ""): Boolean;

Essa função salva em disco os Manifestos carregados no componente no formato XML.

A função possui o parâmetro:

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

ACBrMDFe1.Manifestos.GetNamePath: String;

Esta função no momento retorna apenas uma string contendo “**Manifesto**”.

b.Procedimetos

ACBrMDFe1.SetStatus(const stNewStatus : TStatusACBrMDFe);

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: stMDFeIdle, stMDFeStatusServico, stMDFeRecepcao, stMDFeRetRecepcao, stMDFeConsulta, stMDFeRecibo, stMDFeEmail, stMDFeEvento, stMDFeEnvioWebService

ACBrMDFe1.ImprimirEvento;

Imprimir um evento carregado no componente.

ACBrMDFe1.ImprimirEventoPDF;

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

ACBrMDFe1.EnviaEmailEvento(const sSmtpHost, sSmtpPort, sSmtpUser, sSmtpPasswd, sFrom, sTo, sAssunto: String; sMensagem : TStrings; SSL : Boolean; EnviaPDF: Boolean = true; sCC: TStrings = nil; Anexos:TStrings=nil; PedirConfirma: Boolean = False; AguardarEnvio: Boolean = False; NomeRemetente: String = "; TLS : Boolean = True);

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 DAMDFE.

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 enviado usando o TLS – Transport Layer Security – protocolo criptográfico usado no envio de e-mail.

ACBrMDFe1.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; StreamMDFe : 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 enviado usando o TLS – Transport Layer Security – protocolo criptográfico usado no envio de e-mail.

StreamMDFe 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.

ACBrMDFe1.Manifestos.GerarMDFe;

Gera o XML do MDF-e, o componente deve estar alimentado com os dados pertinentes ao manifesto.

ACBrMDFe1.Manifestos.Assinar;

Gera, assina e salva em disco um MDF-e, o componente deve estar alimentado com os dados pertinentes ao manifesto.

ACBrMDFe1.Manifestos.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.

ACBrMDFe1.Manifestos.Imprimir;

Imprime o DAMDFE de todos os MDF-e previamente carregados no componente.

ACBrMDFe1.Manifestos.ImprimirPDF;

Gera e salva a imagem do DAMDFE de todos os MDF-e previamente carregados no componente no formato PDF.

ACBrMDFe1.Manifestos.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) do MDF-e ao condutor do veículo carregado no componente, opcionalmente podendo anexar também o PDF do DAMDFE.

Como o componente comporta uma lista de Manifestos devemos indicar no parâmetro de **Items** qual é o Manifesto 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 DAMDFE.

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 enviado 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 do MDF-e.

Arquivos XML do MDF-e:

Pedido de Consulta do Status do Serviço	<AAAAMMDDHHMMSS>-ped-sta.xml	
Status do Serviço	<AAAAMMDDTHHMMSS>-sta.xml	
Envio de Lote de MDF-e	<numLote>-env-lot.xml	
Recibo	<numLote>-rec.xml	
Pedido do Resultado do Processamento do Lote	<numRecibo>-ped-rec.xml	
Resultado do Processamento do Lote	<numRecibo>-pro-rec.xml	
MDF-e	<chave>-mdfe.xml	
Pedido de Consulta Situação Atual	<chave>-ped-sit.xml	
Situação Atual da MDF-e	<chave>-sit.xml	
Pedido de Registro de Evento	<chave>-ped-eve.xml	
Registro de Evento	<chave>-eve.xml	
Compartilhamento de Registro de Evento	<ID do Evento>-procEventoMDFe.xml	
Documento Fiscal Eletrônico	<chave>-MDFeDFe.xml	

Arquivos PDF do MDF-e:

DAMDFE	<chave>-MDFe.pdf	
DAEvento	<ID do Evento>-procEventoMDFe.pdf	

<ID do Evento> = <Tipo de Evento + Chave + Numero Sequencial do Evento>

Arquivos XML do MDF-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.