

Manual de Utilização das DLLs de Leitura e Geração de Arquivos

Rev. 1.02

Visão Geral

A solução para ECFs baseados no protocolo FiscNET para a leitura dos dados do ECF e posterior geração dos arquivos ASCII no formato do ATO COTEPE ICMS 17/04 passa pela utilização de duas DLLs:

- LEITURA.DLL – DLL de Leitura da Memória Fiscal e das Memórias de Fita-Detalhe do ECF para Emissores de Cupom Fiscal que utilizem o protocolo FiscNET. O arquivo gerado será no formato binário.
- ATO17.DLL – DLL que extrai as informações, a partir dos arquivos binários gerados pela DLL de Leitura, formatadas pelas especificações do ATO COTEPE ICMS 17/04.

DLL de Leitura das Memórias do ECF

A DLL de leitura das memórias do ECF tem a função de ler, a partir da porta de comunicação do ECF, os dados binários compactados da Memória Fiscal e da Memória de Fita-Detalhe do ECF, que serão gravados em um único arquivo em disco no computador. Pelo fato das memórias estarem contidas em um único arquivo, este recebe normalmente a extensão TDM (abreviatura para "ToDas as Memórias").

Devido às limitações de velocidade impostas pela comunicação serial, o processo de leitura pode ser bastante demorado, dependendo da quantidade de informações gravadas nas respectivas memórias.

Nenhuma leitura poderá ser extraída a partir do arquivo binário antes que este seja completamente lido a partir do ECF.

Com o objetivo de diminuir o tempo necessário para efetuar a leitura periódica das memórias do ECF, a DLL de leitura possui uma função de atualização do arquivo binário que lerá do ECF apenas as informações ainda não gravadas no respectivo arquivo, isto é, apenas os registros que foram acrescentadas à Memória Fiscal ou à Memória de Fita-Detalhe desde a última leitura completa do arquivo binário serão lidas e acrescentadas ao arquivo, mantendo-o atualizado.

A leitura binária poderá ter seu processo de leitura ou atualização interrompido pelo aplicativo (ou pelo operador). Neste caso as informações já lidas e gravadas no arquivo binário não precisarão ser novamente lidas caso a leitura seja retomada em outro momento. Isto permite que leituras muito longas, característica da primeira leitura binária do ECF, possam ser

fracionadas em períodos de tempo mais curtos e mais adequados à rotina do estabelecimento comercial.

Cada arquivo binário lido poderá conter os dados da Memória Fiscal do ECF e de várias Memórias de Fita-Detalhe do mesmo ECF, podendo ser associado na forma de um arquivo para cada ECF.

Interface da DLL de Leitura

Abaixo as funções disponíveis na DLL de leitura:

void DLLReadDefineBaudRate (long InBaudRate);

Função que define o baud-rate que deverá ser utilizado pela DLL para efetuar a leitura dos dados do ECF. Caso o baud-rate pretendido seja diferente de 115.200 bps, esta função deverá ser chamada antes do início de qualquer leitura.

Parâmetro

<InBaudRate> – especifica o baud-rate, em bits por segundo, que deverá ser utilizado pela DLL na comunicação com o ECF.

Observação 1: Caso o baud-rate do ECF seja o padrão de 115.200 bps, não há necessidade de efetuar a chamada desta função.

Observação 2: O baud-rate deverá ser especificado antes da chamada da função de leitura das memórias.

void DLLReadDefineTrataErro (int (*pTratadora) (long InErro, char *szDescricao, char bErroNaoFatal));

Especifica a função do aplicativo que será chamada pela DLL de leitura para indicar a ocorrência de erros durante a leitura do arquivo. Caso o aplicativo implemente, a função tratadora deverá ser definida antes do início da comunicação.

Parâmetro

<pTratadora> – endereço da função no software aplicativo, que tratará os erros gerados pela DLL durante a leitura dos dados binários.

Parâmetros recebidos pela função tratadora de erros:

<InErro> – Código do erro, de acordo com a Tabela de Erros da DLL de Leitura a seguir.

<szDescricao> – Texto com uma descrição sucinta do erro.

<bErroNaoFatal> – Parâmetro que indica se o erro que ocorreu é passível de retentativa. Por exemplo, se ocorrer uma falha de comunicação, pose-se

tentar repetir a operação, mas caso o arquivo informado seja inválido, a operação não poderá ser repetida.

Valor a ser retornado pela função tratadora de erros:

Se o valor recebido no parâmetro **<bErroNaoFatal>** for diferente de zero, a função tratadora poderá retornar o valor **-1** para indicar que a ação que gerou o erro deve ser repetida ou **ZERO** caso a leitura deva ser abortada.

Se o valor recebido no parâmetro **<bErroNaoFatal>** for **ZERO**, independentemente do valor retornado pela função tratadora, a leitura será abortada pela DLL.

TABELA DE ERROS DA DLL DE LEITURA			
Valor(hexa)	Nome do Erro	Descrição	Permite Retentativa?
0x0000	DLLErroOK	Não houve erro	Não
0x1001	DLLErroAbreArquivo	Arquivo binário encontra-se aberto. Esta ação poderá ser repetida, solicitando ao usuário que o arquivo seja fechado.	Sim
0x4001	DLLErroCargaDLL	DLL de comunicação do protocolo FiscNET (DLLG2.DLL) não pôde ser carregada.	Não
0x4002	DLLErroCanalCOM	Canal de comunicação indicado não pôde ser aberto.	Não
0x4003	DLLErroBinarioInvalido	Formato do arquivo binário indicado é inválido para esta versão da DLL.	Não
0x4004	DLLErroLeituraArquivo	Erro de leitura do arquivo binário indicado.	Não
0x4005	DLLErroEscritaArquivo	Erro de escrita no arquivo binário.	Não
0x4006	DLLErroBinarioECF	Número de fabricação do ECF gravado anteriormente no arquivo binário informado não confere com o número de fabricação informado no parâmetro <szSerieECF> da função <DLLReadLeMemorias>.	Não
0x4007	DLLErroECFInvalido	Número de fabricação indicado no parâmetro <szSerieECF> da função <DLLReadLeMemorias> não confere com o número de fabricação do ECF conectado na porta de comunicação indicada.	Não
0x4008	DLLErroFaltaRAM	Não há memória suficiente para a execução satisfatória da DLL de leitura.	Não
0x4009	DLLErroCriaArquivo	Não foi possível criar o arquivo com o nome especificado.	Não

0x400a	DLLErroArquivoIncompleto	Não foi possível gerar dados a partir do arquivo binário especificado pois este não encontra-se com a leitura completa,	Não
Demais	Erros de Comunicação	Erros de comunicação retornados pela DLL de comunicação DLLG2.DLL. Dependendo do contexto e do tipo de erro ocorrido o parâmetro <bErroNaoFatal> poderá assumir valor ZERO ou diferente de ZERO.	Depende da situação e do código de erro de comunicação

int DLLReadLeMemorias (char *szPortaSerial, char *szNomeArquivo, char *szSerieECF, char bAguardaConcluirLeitura);

Função que lerá através da porta <szPortaSerial> do PC as memórias do ECF com número de fabricação <szSerieECF> e gravará no arquivo indicado no parâmetro <szNomeArquivo>. Caso o arquivo indicado pelo parâmetro <szNomeArquivo> já exista, ele será atualizado com os dados lidos do ECF.

Esta função retornará imediatamente, apenas disparando o processo de leitura, caso o parâmetro <bAguardaConcluirLeitura> seja informado com o valor ZERO. Neste caso o final da execução da leitura poderá ser monitorado através da função <DLLReadStatus>.

Atenção: Caso o parâmetro <bAguardaConcluirLeitura> seja informado com valor diferente de ZERO, a DLL não poderá ser liberada da memória até que a comunicação seja concluída, mesmo que sem sucesso, caso contrário poderão ser geradas exceções de acesso à memória no aplicativo.

Caso o parâmetro <bAguardaConcluirLeitura> seja informado com qualquer valor diferente de ZERO, a função <DLLReadLeMemorias> retornará apenas após o final da leitura, seja ela concluída satisfatoriamente ou não.

Parâmetros

<szPortaSerial> – Indica o nome da porta de comunicação do PC por onde será realizada a leitura dos dados. Por exemplo, "COM1".

<szNomeArquivo> – Indica o nome do arquivo onde deverão ser gravados os dados binários lidos do ECF. Caso o arquivo já exista, este será atualizados com as informações lidas do ECF.

< szSerieECF > – Indica o número de fabricação do ECF para o qual se deseja realizar a leitura.

<bAguardaConcluirLeitura > – Caso seja informado o valor ZERO, a função <DLLReadLeMemorias> retorna imediatamente, iniciando a leitura através de novas linhas de execução (threads) do sistema operacional. Caso seja informado com valor diferente de ZERO, a função <DLLReadLeMemorias> retornará apenas após o final da leitura.

void DLLReadCancelaLeitura (void);

Função que cancelará o processo de leitura em execução e gravará os dados lidos até o momento do cancelamento no arquivo indicado.

void DLLVersao (char *szVersao);

Função que retorna o nome da atual versão da DLL.

Parâmetro

<szVersao > - Ponteiro para o texto finalizado com ZERO, que conterá a atual versão da DLL. O tamanho da área reservada ao texto deverá ser de, no mínimo, 10 bytes.

int DLLReadStatus (DLLReadStatusData *pStatusData);

Função que retorna o status atual do processo de leitura.

Parâmetros

<pStatusData > - Estrutura de dados que conterá a situação atual da leitura ou da atualização dos dados do ECF. A estrutura contém os seguintes campos:

ESTRUTURA DE DADOS DLLReadStatusData		
<i>Tipo</i>	<i>Campo</i>	<i>Tamanho</i>
inteiro	nTamanhoStatus	4 bytes
Tamanho, em bytes, desta estrutura. Deverá ser preenchida pelo aplicativo antes da chamada à função DLLReadStatus().		
<i>Tipo</i>	<i>Campo</i>	<i>Tamanho</i>
inteiro	nSituacaoAtual	4 bytes
Status atual da leitura. Valor preenchido pela função DLLReadStatus() de acordo com a Tabela de Status da DLL de leitura. Equivale também ao valor retornado pela função DLLReadStatus().		
<i>Tipo</i>	<i>Campo</i>	<i>Tamanho</i>
inteiro	nCodigoErro	4 bytes
Caso esteja interrompida por erro, indica o código do erro. Neste caso a descrição da atual situação da leitura no campo <szSituacaoAtual> desta estrutura estará indicando a descrição completa do erro.		
<i>Tipo</i>	<i>Campo</i>	<i>Tamanho</i>
inteiro longo	lnTamanhoLeitura	4 bytes
Tamanho total da etapa de leitura em andamento.		
<i>Tipo</i>	<i>Campo</i>	<i>Tamanho</i>
inteiro longo	lnProgressoLeitura	4 bytes
Progresso atual da etapa de leitura em andamento, variando de ZERO até <lnTamanhoLeitura>.		
<i>Tipo</i>	<i>Campo</i>	<i>Tamanho</i>
texto	szSituacaoAtual	128 bytes
Nome ou descrição da atual etapa de leitura em andamento. Caso o campo <nCodigoErro> contenha algum valor diferente de ZERO, este campo conterá uma descrição do erro que ocasionou o encerramento da leitura.		

Retorno

Status atual de execução da leitura (também retornado no campo <nSituacaoAtual> da estrutura DLLReadStatusData), de acordo com a tabela abaixo:

TABELA DE STATUS DA DLL DE LEITURA	
Valor(hexa)	Descrição
0x0000	Indica que a DLL ainda não iniciou nenhum processo de leitura.
0x0010	Em processo de carga da DLL de comunicação.
0x0020	Obtendo informações do ECF
0x0030	Verificando ou criando o arquivo binário.
0x0040	Obtendo informações da Memória Fiscal.
0x0050	Obtendo informações da Memória de Fita-Detalhe do ECF.
0x0060	Lendo dados da Memória Fiscal.
0x0070	Lendo dados da Memória de Fita-Detalhe.
0x0080	Encerrando o processo de leitura.
0x0100	Processo de leitura ou atualização concluído com sucesso.
0x0200	Processo de leitura ou atualização concluído com erro.
0x0210	Processando notificação de erro.
0x0300	Processando cancelamento da leitura solicitada pelo aplicativo.
0x0400	Processo de leitura ou atualização foi cancelado.

DLL de Geração dos Arquivos

A DLL de Geração dos Arquivos (ATO17.DLL) tem a função de conversão do arquivo binário lido pela DLL de Leitura para um arquivo de texto formatado de acordo com as especificações do ATO COTEPE ICMS 17/04.

Para o caso de atendimento à Portaria CAT-52/07 do Estado de São Paulo, deverá ser gerada uma leitura do tipo TDM (ToDas as Memórias) para cada dia de movimento do ECF.

Interface da DLL de Geração dos Arquivos

Abaixo as funções disponíveis na DLL de Geração dos Arquivos:

void DLLVersao (char *szVersao);

Função que retorna o nome da atual versão da DLL.

Parâmetro

<szVersao> - Ponteiro para o texto finalizado com ZERO, que conterá a atual versão da DLL. O tamanho da área reservada ao texto deverá ser de, no mínimo, 10 bytes.

int DLLATO17GeraArquivo (char *szArquivoBinario, char *szArquivoTexto, char *szPeriodoIni, char *szPeriodoFIM, char TipoPeriodo, char *szUsuario, char *szTipoLeitura);

Função que executa a conversão do arquivo binário lido pela DLL de leitura em arquivo de texto de acordo com o layout estabelecido pelo ATO COTEPE ICMS 17/04.

Parâmetros

<szArquivoBinario> – Nome do arquivo binário criado pela DLL de Leitura a partir de onde serão lidos os dados.

<szArquivoTexto> – Nome do arquivo de texto que conterá os dados gerados no formato do ATO COTEPE ICMS 17/04.

<szPeriodoIni> – Texto que deve indicar o início do período a partir do qual será realizada a geração do arquivo.

<szPeriodoFIM> – Texto que deve indicar o final do período a partir do qual será realizada a geração do arquivo.

<TipoPeriodo> – Caracter que Indica como deve ser interpretados os parâmetros <szPeriodoIni> e <szPeriodoFIM>. Eles serão interpretados de acordo com a tabela abaixo:

Valor (Caracter)	Significado
'D'	Os parâmetros <szPeriodoIni> e <szPeriodoFIM> serão interpretados como um intervalo de datas de emissão dos documentos. Na leitura de MFD serão consideradas as datas de encerramento dos documentos e na leitura da MF serão consideradas as datas de emissão das Reduções Z.
'M'	Os parâmetros <szPeriodoIni> e <szPeriodoFIM> serão interpretados como um intervalo de datas de referência do movimento dos documentos, independentemente da data de sua emissão, isto permite a emissão de movimentos completos de determinados dias, mesmo que a Redução Z referente a um determinado dia tenha sido emitida em data posterior à data do movimento. Na leitura de MFD serão consideradas as datas de referência do movimento dos documentos e na leitura da MF serão consideradas as datas dos movimentos das Reduções Z.
'C'	Os parâmetros <szPeriodoIni> e <szPeriodoFIM> serão interpretados como um intervalo de datas de COOs dos documentos. Na leitura de MFD serão consideradas valores do COO dos documentos e na leitura da MF serão considerados o número das Reduções Z, indicados pelo respectivo Contador de Reduções Z. Caso seja solicitada uma leitura da MFD, deverá também ser informado o parâmetro <szUsuario>.

<szUsuario> – Caso o tipo de período indicado no parâmetro <TipoPeriodo> seja por intervalo de COOs, o texto do parâmetro <szUsuario> deverá indicar o número do proprietário da Memória Fiscal a que se refere o respectivo intervalo de COOs.

<szTipoLeitura> – Texto contendo o tipo de leitura prevista no ATO COTEPE ICMS 17/04 que deverá ser gerada, de acordo com a tabela abaixo:

Valor (Texto)	Significado
"TDM"	Indica que deverá ser gerada uma leitura de Todas as Memórias, de acordo com o AO COTEPE ICMS 17/04. Este é também o tipo de leitura que atende às determinações da Portaria CAT-52 do Estado de São Paulo.
"MF"	A leitura conterá apenas os dados presentes na Memória Fiscal. Tipo de leitura prevista no ATO COTEPE ICMS 17/04.
"MFD"	A leitura conterá apenas os dados presentes na Memória de Fita-Detalhe. Tipo de leitura prevista no ATO COTEPE ICMS 17/04.