

## STK (Start Kit DARUMA)

### Como Tratar os Retornos da DarumaFramework.dll

Primeiramente vamos entender como funciona a DarumaFramework.dll

A DarumaFramework.dll é uma dll integradora de alto nível, que foi desenvolvida para você programador, no intuito de proporcionar-lhe velocidade na programação para impressoras e terminais Daruma e além disso, uma comunicação com o hardware em altíssimo nível, evitando dessa maneira a escovação de bit's propriamente dita.

Todos os métodos que disponibilizamos nela possuem retornos, que indicam se a execução do método ocorreu com sucesso ou não, e neste caso, também nos indica o motivo de uma execução mal sucedida. Além de erros podemos receber avisos. Vamos ver neste STK como tratar estes retornos e deixar seu aplicativo com um alto nível de interação com o usuário.

Este Start Kit divide-se em 4 Partes:

1. Configurando o arquivo DarumaFramework.xml
2. Solicitando os Retornos
3. Analisando os Retornos
4. Exemplo em trecho de código em Linguagem C

#### 1. Configurando o arquivo DarumaFramework.xml

Fazer o tratamento de retornos normalmente dá um trabalho, ainda mais com as exigências do PAF-ECF para quem está em processo de homologação. Pensando nisso, dentre as configurações da DLL foi desenvolvido uma chave exclusiva para facilitar no tratamento de erros.

Ao subir a DLL na memória pela primeira vez, é criado junto ao seu executável um arquivo XML (DarumaFramework.xml), com a configuração e registros utilizados pela DLL.

Abra ele com um editor de sua preferência como por exemplo, o notepad++ ou o próprio Wordpad do Windows para que seja melhor a visualização. Nele você verá uma estrutura com tags informando o Produto e seus respectivos valores.

**ex:**

"<NOME DA CHAVE>"

Há um bloco exclusivo para impressora fiscal, este seria o <ECF>... </ECF> contendo todas as chaves de configuração para a Impressora Fiscal. Dentro deste bloco, utilizaremos a chave <RetornarAvisoErro>1</RetornarAvisoErro>, caso o valor desta chave seja "0", os retornos de método serão limitados a "0"(sem comunicação) ou "1"(método enviado), sendo sempre obrigatoriamente necessário analisar os códigos de erro e aviso.

Quando seu valor é configurado para "1" retornará o inteiro "-12" no Retorno do Método caso o método executado tenha algum aviso ou erro.

Assim podemos reduzir o tratamento de erros, validando apenas se o Retorno do Método é < 1, caso sim mostre o retorno conforme a tabela de retornos do Help Eletrônico das funções. Se o retorno for igual a -12 significa que temos aviso ou erro na execução do mesmo.

#### 2. Solicitando os Retornos

Para receber os códigos de erro e aviso referentes ao último comando executado, devemos utilizar o método **rStatusUltimoCmdInt\_ECF\_Daruma**, este método possui 2 variáveis inteiras indicadas por referência para receber o número de erro e de aviso.

### 3. Analisando os Retornos

Para analisar os códigos recebidos no passo anterior, não precisamos mais realizar vários testes de comparação de valores, podemos utilizar os comandos: **eInterpretarErro\_ECF\_Daruma** e **eInterpretarAviso\_ECF\_Daruma**.

Ambos os métodos possuem 2 parâmetros cada, um inteiro que usamos para indicar o código que foi recebido, e uma variável string.

Esta string, já contém o número e descrição do código, não precisando ser montado uma gigantesca estrutura de IF's ou Switch Case's.

### 4. Exemplo em trecho de código em Linguagem C

Agora que vimos os passos necessários para o tratamento de retornos, vamos ver um exemplo em um trecho de código em linguagem C:

```
void retornosExtendidos(int iRetornoMetodo)
{
    int iErro;
    int iAviso;
    char clInterpreta[201];memset(clInterpreta, 0, sizeof((string) clInterpreta));

    if(iRetornoMetodo < 1)
    {
        rStatusUltimoCmdInt_ECF_Daruma(&iErro,&iAviso);
        eInterpretarErro_ECF_Daruma(iErro,clInterpreta);
        printf("Erro ECF: %s\n", clInterpreta);
        eInterpretarAviso_ECF_Daruma(iAviso,clInterpreta);
        printf("Aviso ECF: %s\n\n", clInterpreta);
        printf("-----\n");
    }
    else
    {
        printf("Comando Executado com Sucesso!!!, Sem Erros e Sem Avisos\n\n");
        printf("-----\n");
    }
}
```

Se tiver alguma dúvida e/ou dificuldade, entre em contato com a nossa equipe de suporte ao desenvolvedor.

#### Telefone:

Suporte ao Desenvolvedor 0800 770 3320

#### E-mails:

suporte@daruma.com.br , desenvolvedores.suporte@daruma.com.br ,  
desenvolvedores.daruma@daruma.com.br , daruma.desenvolvedores@daruma.com.br ,  
suporte.ddc@daruma.com.br , ddc.suporte@daruma.com.br , suporte.desenvolvedores@daruma.com.br ,  
suporte.alexandre@daruma.com.br , claudenir@daruma.com.br

#### Skypes:

suporte\_daruma, desenvolvedores\_suporte\_daruma, suporte\_desenvolvedores\_daruma,  
desenvolvedores\_daruma, daruma.desenvolvedores, suporte\_ddc\_daruma, ddc\_suporte\_daruma,  
daruma\_suporte\_alexandre, claudenir\_andrade