

De DBF para o Interbase/Firebird num piscar de olhos

Autor: Caio Oliveira – Set/2002

Introdução

Muitos usuários do Interbase ou e Firebird, como eu, estão vindo de sistema legados desenvolvidos em Clipper e, a maior dificuldade inicialmente é, como fazer para colocar o mais rapidamente possível, toda a minha estrutura de dados no padrão DBF dentro do meu novo banco de dados IB/FB. Esse artigo trata exatamente dessa questão; a migração da sua base de dados DBF para o IB ou FB.

Os programas apresentados abaixo foram usados e estarão sendo usados por mim, no decorrer desse ano e talvez mais adiante em casos reais; onde estaremos migrando todos os clientes que usam nosso legado para o nosso novo sistema. E, agora talvez, por muitos colegas.

O código fonte dos programas estão disponíveis no site da CFLP em Downloads/Migração: DBFtoINTERBASE

Divirtam-se !

Começando o trabalho

Vamos sub dividir o processo em 03 fases distintas; são elas:

- 1a.Parte - Gerando os scripts SQL do IB/FB à partir da estrutura atual formato DBF.
- 2a.Parte - Usando o IBConsole para rodar os SCRIPTS gerado.
- 3a.Parte - Transportando os dados das tabelas .DBF para o banco de dados IB/FB.

1a.Parte:Gerando os scripts SQL do IB/FB à partir da estrutura atual formato .DBF

Para isso criamos um programa escrito em Clipper 5.2e que lê a estrutura da tabela .DBF e converte essa estrutura mantendo as características de cada campo no formato Interbase/Firebird. O programa executável pronto é fornecido juntamente com o código fonte de todos os programas desse artigo; portanto, se você não quiser saber como isso foi feito pode pular o tópico seguinte e ir direto para "executando o CRIASQL".

Tipos de dados .DBF x Tipos de dados Interbase/Firebird

A estrutura de dados de uma tabela .DBF é um tanto quanto diferente de uma tabela IB/FB e, portanto, o programa CRIASQL teve de tratar desses pormenores. Acho importante conhecer algumas dessas diferenças básicas agora de forma que serão de suma importância para que você entenda à parte do código que trata do processo de transporte de dados das suas tabelas .DBF para o IB/FB.

No .DBF nós temos basicamente os seguintes campos de dados possíveis: Caracter, Numeric, Logical, Date e Memo; enquanto que no IB/FB vamos ter que lidar com novos tipos de dados e com a falta do Logical (Boolean). Mas, isso não vai ser problema.

Segue abaixo uma tabela com as conversões de tipos de dados que são realizadas pelo programa CRIASQL.

Campo da Estrutura .DBF

DATE

CHARACTER

NUMERIC (c/decimal 0 e Tamanho <=5)

NUMERIC (c/decimal 0 e Tamanho >5)

NUMERIC (c/decimal > 0)

LOGICAL

MEMO

DATAHORA (campo específico que usamos para arquivar a data e hora da inclusão do registro)

DATAHALT (campo específico que usamos para arquivar a data e hora da alteração no registro).

Campo Convertido na Estrutura IB/FB

DATE

CHAR(n)

SMALLINT

INTEGER

NUMERIC(n,d)

CHAR(1) DEFAULT 'N'

BLOB SUB-TYPE TEXT SEGMENT SYZE 80

data_hoje (Domain; script fornecido)

data_hoje (Domain; script fornecido)

O tipo CHAR do IB/FB vc. vai querer usar !

No CRIASQL nós optamos por não usar CHAR portanto você pode alterar isso no script pronto ou mais tarde quando lhe convier; ou até fazer uma implementação no código fonte do CRIASQL.

Nosso objetivo aqui não é o de definir a melhor forma de estrutura de tabelas e portanto, vamos adiante. No site da CFLP há vários artigos que falam sobre o assunto entre outras informações importantes.

DOMAINS

O CRIASQL usou alguns DOMAINS (domínios) para personalizar o formato de alguns campos de dados. Os DOMAINS são usados no IB/FB para organizar os formatos dos campos de dados.

Ex.: Ao invés de toda vez declararmos um campo de valor monetário como NUMERIC(14,2) podemos criar um DOMAIN chamado "valormonetario":

```
CREATE DOMAIN valormonetario NUMERIC(14,2) DEFAULT 0.00;
```

Como fizemos com os nossos campos DATAHORA e DATAHALT, criamos o DOMAIN 'data_hoje':

```
CREATE DOMAIN data_hoje TIMESTAMP DEFAULT 'today'
```

Executando o CRIASQL

O CRIASQL é o nome dado ao programa que gera o script SQL à partir da estrutura de uma tabela .DBF. Esse programa possui a seguinte sintaxe:

```
CRIASQL <arquivo_mascara> TO <nomescript.sql>
```

onde: <arquivo_mascara> = nome do arquivo ou máscara (MEU.DBF ou *.DBF) do(s) arquivo(s) cuja estrutura vai ser gerada.

<nomescript.sql> nome do script que vai ser gerado (Ex.: meuscript.sql)

Ex.: Acione o "prompt do MSDOS"; posicione o cursor no diretório onde se encontra a sua base de dados .DBF e digite o comando abaixo para gerar um SCRIPT contendo a estrutura IB/FB prontinha.

```
CRIASQL *.DBF TO seuscript.sql
```

2a.Parte:Usando o IBConsole para rodar o SCRIPT gerado pelo CRIASQL

Com o script SQL em mãos sua nova estrutura IB/FB contendo todas as tabelas e campos de dados (na íntegra) está à alguns cliques do mouse.

Antes de rodar o script gerado pelo CRIASQL

Antes de usar o seu "seuscript.sql" é necessário criar o banco de dados e, em seguida rodar o script contendo os DOMAINS utilizados pelo CRIASQL na sua estrutura (disponível no DBFtoINTERBASE.ZIP).

- 1.Acione o IBConsole (aproveite e atualize o seu: IBConsole 344 até quando escreví esse artigo). Vc. pode usar outro utilitário se preferir.
- 2.Conecte com o seu banco de dados IB/FB
- 3.Conecte com o seu .gdb (se não criou .GDB ainda vc. deve fazê-lo agora)
- 4.Acesse a opção Interactive SQL
- 5.Carregue e execute o script "domains.sql"
- 6.Carregue e execute o script "errosconversao.sql"
- 7.Carregue o script "meuscript.sql" e execute-o.
- 8.sua estrutura de dados IB/FB está pronta para receber dados.

Dê um clique em "Tables" para ver as tabelas. Gostou ?. Até aqui foi fácil, não?. Vamos agora para a 3a.parte e também, a mais delicada do processo "transporte de dados do velho .DBF para o IB/FB".

3a.Parte: Transportando os dados das tabelas .DBF para o banco de dados IB/FB.

Agora vamos trabalhar um pouquinho com o Delphi. O Programa de transporte de dados foi escrito em Delphi 5 e acredito que deve rodar bem nas novas versões do produto, pois foram usados os componentes básicos dessa maravilhosa e poderosa ferramenta.

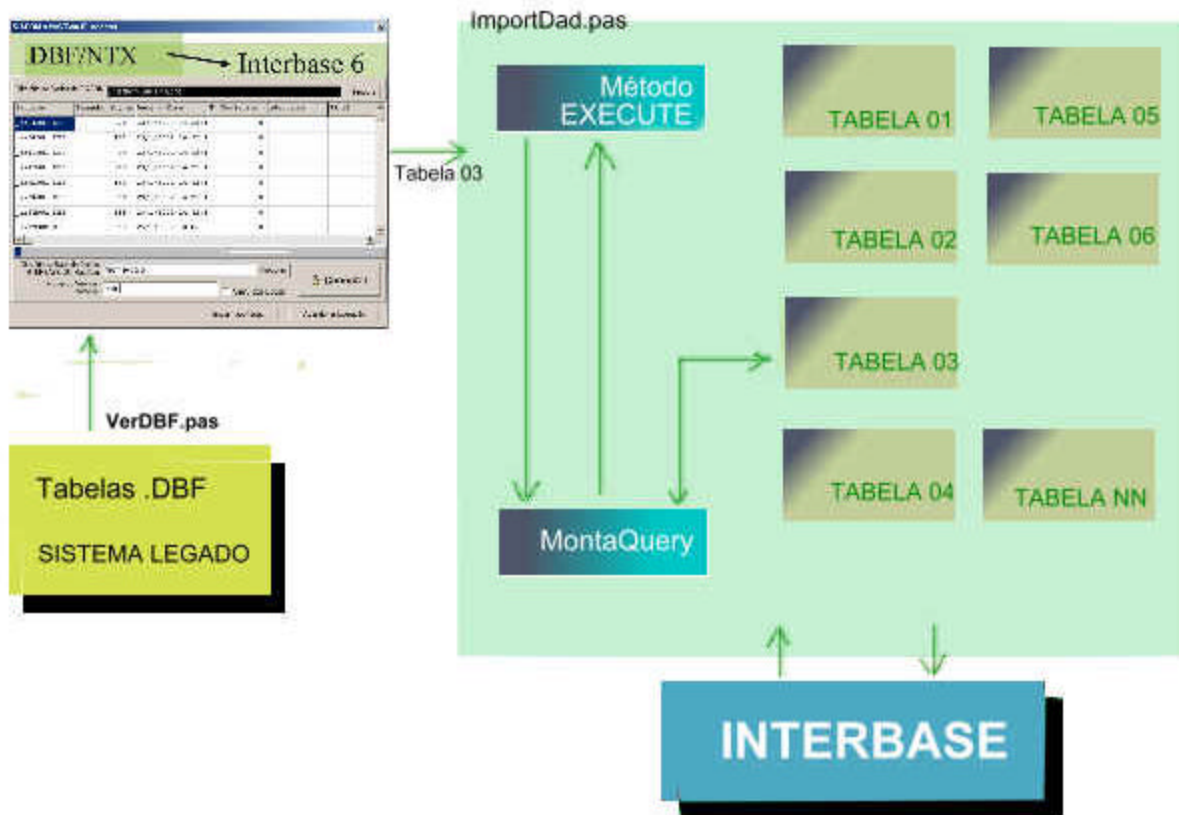


Fig.01 - esquema do aplicativo de conversão.

Requisitos para compilar esse programa

- ⑩ Delphi 5 (SP1)- Professional Edition ou Enterprise Edition
- ⑩ Componentes IBX (nativo); usei a IBX 5.03
- ⑩ Código Fonte do DBFtoINTERBASE (disponível no site da CFLP)
Um pouco de paciência
- ⑩ O utilitário IBConsole ou outro que vc. preferir: IBExpert, etc....
- ⑩ e é claro!, um servidor de banco de dados Interbase/Firebird.

Explicando o Código do programa

Units	Descrição
SIACOMADP.pas SIACOM.pas	Unit principal do projeto. Formulário principal; controle do programa. (o nome SIACOM em homenagem ao meu sistema legado). Os botões "ver/executar script" a programação ai não foi completada, is ser usada para acionar e executar no BD os scripts para criação dos arquivos (tabelas) eu usei o IBConsole para isso.
DirList.pas	Formuário para usado para seleção do diretório da(s) base(s) de dado(s). (acionado pelos botões "procurar" do formulário principal).
DataMod.pas	Módulo de dados da aplicação (componentes de acesso à dados ficam aqui).
VerDBF.pas	Unit (Thread) responsável pela leitura das informações iniciais dos arquivos de dados .DBF; devolvendo essas informações para o GRID do formulário principal.

Verscript.pas

Seria responsável pela script visualização do(s) SCRIPT(s) à serem executados no BD, mas, não foi utilizada nesse projeto. Vc. pode implementar esse recurso se quiser (tá "meia boca!"). Unidade responsável pela importação dos dados do seus .DBF para o IB/FB. Essa é a única parte do programa que vc. vai precisar "mexer", pois, nessa unit vc. programa a importação de cada tabela do seu antigo .DBF.

ImportDad.pas

Eu deixei três exemplos prontos, para serem "copiados" ou "modificados" será necessário mudar apenas os nomes das tabelas e campos, os exemplos, mostrados nessa unit, convertem todos os tipos de dados; então acredito que não haverá nenhuma dificuldade.

Formulário Principal

Nesse formulário estão dispostos todos os controles da aplicação; desde o acesso à base .DBF até a conexão com o banco de dados IB/FB.

A conexão ao banco de dados pode ser realizada Local ou Remota.

Leitura de informações dos .DBF

Para ler as informações dos arquivos .DBF digite o nome do diretório no campo "Diretório de dados dos DBF" e pressione <Enter> ou acione o botão "procurar" ao lado do campo para localizar o diretório.

Após ter informado o diretório o sistema "lê" as informações de cada .DBF nesse diretório e, em seguida mostra as informações: nome da tabela, data e hora da última alteração e número de registros existentes em cada tabela. Agora você deve executar a conexão com o IB/FB.

Conexão com o IB/FB

Para conectar com o banco de dados IB/FB digite o nome do diretório/nome do arquivo .GDB no campo "Base de dados Interbase/Firebird" ou, se o banco de dados estiver localizado na máquina local, você pode usar também o botão "procurar".

Se o servidor local "marque o checkbox 'Servidor Local'" e em seguida dê um clique no botão "Conectar".

Se o servidor for remoto, digite o nome do servidor no campo "Nome do Servidor" e em seguida dê um clique no botão "conectar".

Se conectado a descrição do botão passa à ser "desconectar". ATENÇÃO!!!. não execute a importação ainda, pois, não vai funcionar se você não fez a adaptação de suas tabelas na Unit "ImportDad.pas". Veja a seguir um exemplo de como fazer.

Unidade de Transporte de dados .DBF to INTERBASE/Firebird (ImportDAD.pas)

Essa Unit é responsável pelo processo de "conversão" dos dados do .DBF para o banco de dados IB/FB. É nela que deverão ser codificados os processos de conversão de cada tabela do seu legado em .DBF.

Alguém pode estar perguntando se não existe uma forma automática de fazer isso, eu não conheço, embora acho que com mais tempo seja possível desenvolver à partir daqui uma solução mais generalizada de conversão de arquivos do DBF para o Interbase; mas, acredito também que se o objetivo é importação de uma base de dados específica (como foi o meu caso) com esse procedimento o resultado é 100% seguro.

Abaixo veremos um dos procedimentos de exemplo que está disponível no código, e verão então como é fácil fazer as alterações; o tempo que se leva para fazer essas alterações, vai depender da quantidade de arquivos que se tem que importar.

Importação da tabela exemplo SCRTDOC.DBF

O método *execute* da Thread "ImportDad" é chamado para cada tabela da lista do GRID no formulário principal. Esse método por sua vez executa os seguintes procedimentos:

- 1.Faz um COUNT na tabela no IB/FB para ver se tem algum registro nessa tabela
- 2.Se houverem registros interrompe o processo de importação dessa tabela e finaliza a thread.
- 3.Senão "seta" o GENERATOR da tabela para o valor "0" continua...
- 4.Inicia o LOOP de importação de cada registro da tabela .DBF
- 5.Chama a procedure "MontaQuery" (que monta o SQL de inclusão do registro no IB/FB).

6. Verifica o Texto da SQL Query: se contiver a palavra "NAO" interrompe a importação da tabela. O sistema somente importa as tabelas que estiverem sido previamente configuradas por vc.. A procedure montaquery checa o nome da tabela e faz a chamada ao procedimento dessa tabela especificamente.
7. Verifica o Texto da SQL Query: se tiver em branco ou contiver a palavra ERRO interrompe a importação do registro atual. Isso pode ocorrer se houver algum problema na leitura do registro na tabela; exemplo, campo MEMO estragado ou registro "baleado" o que não é muito raro num .DBF :). Cada erro que ocorrer aqui o sistema grava um registro na tabela "errosconversao" do IB/FB. O script para a criação dessa tabela está disponível no diretório "SCRIPTS" juntamente com os outros programas do pacote DBFtoInterbase.ZIP.
8. Senão (tudo certo), executa a importação do registro; em seguida volta ao procedimento 5 para cada registro da tabela.
9. Ao importar todos os registros da tabela o thread é finalizado.

Um dos exemplos de conversão que está disponível no código (bem simples) é o da tabela abaixo:

Estrutura do .DBF (SCRTDOC.DBF)				Estrutura do Interbase (SCRTDOC)	
Campo	Tipo	Tam	Dec	Nome do Campo	Características
DOC_CODI	C	2	0	ID_SCRTDOC	Integer
DOC_DESC	C	35	0	DOC_CODI	CHAR(2) NOT NULL
				DOC_DESC	CHAR(20)

INDICES:

DOC_CODI -> iCodigo
DOC_DESC -> iDescricao

INDICES:

ID_SCRTDOC primary key

Note que a tabela do Interbase contém um novo campo de nome ID_SCRTDOC; criamos esse campo para funcionar como a nova chave primária da nossa tabela; já que no .DBF não usávamos esse recurso. Aplicamos esse método para todas as tabelas, aliás, o programa CRIASQL é quem faz isso, o qual nós usamos para gerar os scripts com as estruturas das tabelas.

Código exemplo usado na conversão dos dados da tabela acima

```
DM.IBQuery1.SQL.clear;
Try
DM.IBQuery1.SQL.Add( 'INSERT INTO SCRTDOC ' );
DM.IBQuery1.SQL.Add( '( ID_SCRTDOC, DOC_CODI, DOC_DESC )' );
DM.IBQuery1.SQL.Add( 'VALUES ' );
DM.IBQuery1.SQL.Add( '( gen_id(GEN_SCRTDOC,1), :cod, :desc )' );
With DM.Table1 DO Begin
DM.IBQuery1.ParamByName('cod').AsString := FieldByName('doc_codi').AsString;
DM.IBQuery1.ParamByName('desc').AsString := FieldByName('doc_desc').AsString;
End;
Except
DM.IBQuery1.SQL.Clear;
DM.IBQuery1.SQL.text := 'ERRO';
End;
```

Obs.: Outro exemplo também disponível no código é o da tabela CEP001 utiliza a conversão de tipos de dados como: Memo, Boolean, Currency, etc.

Para a conversão de data use o código abaixo como exemplo:

```
DM.IBQuery1.ParamByName('data').AsDateTime := FieldByName('data').AsDateTime;
```

Erros de conversão

O script "errosconversao" que você rodou durante o processo de criação da sua estrutura de dados IB/FB (na 2a.parte dessa matéria), criou uma tabela chamada "errosdeconversao" que vai ser usada pelo

programa de transporte de dados para anotar qualquer problema que ele encontre durante a conversão de determinado registro.

No final do processo você vai poder ver se ocorreu alguma falha durante a conversão e que tipo de falha foi essa.

Qualidade das estruturas de dados gerados pelo CRIASQL

O CRIASQL desempenha o papel de converter o formato de uma estrutura .DBF numa estrutura de banco de dados INTERBASE/FIREBIRD e convenhamos, muitos dos recursos disponíveis aqui agora, como CONSTRAINTS, DOMAINS, PROPRIEDADES AVANÇADAS DE CAMPOS DE DADOS (NOT NULL, DEFAULT, etc) não existem no .DBF então, o processo de conversão como sabemos não é exatamente uma perfeição, mas sim uma forma rápida e eficiente de sair do .DBF para Interbase/Firebird.

A partir daqui estamos com a nossa base de dados dentro do IB ou FB e podemos agora fazer uma reavaliação da mesma logo que tenhamos melhor domínio dessa nossa nova ferramenta de "guardar dados"; agora com mais tranquilidade e segurança.

Arquivos de Índice

Acho que não posso deixar de falar sobre os arquivos de índice: como ficam agora os meus "velhos conhecidos" arquivos de índice; não posso viver sem eles!!!; não vou ter que reindexar a base!!! (ave maria!). 90% das chamadas de suporte nas aplicações feitas em clipper surgem a partir de problemas com os nossos velhos índices.


Se você usa o DBFNTX; então deve ter uma dezena deles no diretório de dados da sua aplicação, mas se usa o DBFCDX deve ter uma dezena (embora um pouquinho mais elegante!) também.

Bom, no IB/FB isso muda radicalmente (os índices continuam a existir), mas, agora quem cuida deles é o bando de dados e não a sua equipe. Além do fato de que será necessário analisar, que muitos desses índices que eram usados no seu programas Clipper não são mais necessários (de cara, considere apenas aqueles que você usa para buscas e ignore aqueles que você usava para apresentar informações; o SQL faz isso para você.).

Eu não vou aprofundar nesse assunto, mesmo porque, não sou nenhum expert em IB/FB (dou minhas cacetadas). E, existem muitos artigos à respeito de como definir e quando definir algum novo índice. Não deixe de ler o artigo do nosso amigo Anderson Haertel (AHR) "[Índices performance e estatísticas no Interbase](#)" muito bom!.

Resumo

Espero que esses programas sejam tão úteis como têm sido para mim e que eu tenha conseguido nesse artigo expor de forma clara e objetiva o processo de migração DBFtoINTERBASE o qual nos propusemos inicialmente. Pretendo com essa contribuição, colaborar para o crescimento da comunidade Interbase/Firebird no Brasil. Estou à disposição para qualquer informação adicional sobre esse artigo pelo e-mail abaixo:

Artigo Original Caio José H. L. De Oliveira (Colaborador da CFLP) news@caiosis.com.br	
	Comunidade Firebird de Língua Portuguesa Visite a Comunidade em: http://www.comunidade-firebird.org
A Comunidade Firebird de Língua Portuguesa foi autorizada pelo Autor do Original para divulgar este trabalho	