

NOTAS SOBRE EL USO DEL CONTROLADOR **DBFCDX**

Introducción

Estas notas, obtenidas de los manuales de Clipper y de la experiencia con el uso de este controlador, solo pretende ser una guía de introducción para todos aquellos que nunca han usado este tipo de ficheros y quieren empezar a usarlo ahora. Además de las funciones que incluyo para el uso de ficheros .cdx existen más que vienen explicadas en el manual y la guía norton de clipper.

Como se podrá ver, el uso de éste sistema de ficheros índice tiene unas posibilidades casi ilimitadas (siempre estará la capacidad del sistema) para trabajar con bases de datos .dbf.

Jose A. Suárez Moreno
Grupo Olivares 2000

Descripción del Controlador DBFCDX

DBFCDX es el controlador del Clipper compatible con FoxPro 2. Éste controlador dispone de una serie de ventajas:

- Compatibilidad con el formato FoxPro 2
- Índices compactos
- Índices compuestos
- Índices condicionales
- Ficheros memo más pequeños y eficientes que los del formato DBFNXTX
- Mecanismo de almacenamiento de datos no tabulares.

Éste controlador permite crear y mantener ficheros de índice y memo (.cdx y .fpt).

- Crear ficheros índice más pequeños. Los datos clave se almacenan en un formato comprimido que reduce bastante el tamaño del fichero índice.

- Crear ficheros de índice compuestos que contienen varios índices (tags), lo que hace posible que la apertura de varios índices tengan un mismo manejador de ficheros. Un mismo fichero .cdx puede contener un número ilimitado de claves de índice en formato comprimido.

- Crear índices condicionales (FOR/WHILE/REST/NEXT).
- Crear ficheros memo con un tamaño de hasta 4,2 Gb.
- Crear ficheros memo con un tamaño mínimo del bloque de 1 byte.

- Reutilizar el espacio de los ficheros memo.
- Almacenar todos los tipos de datos de CA-Clipper (menos bloque de código) en ficheros memo.
- Utilizar las extensiones de Clipper (funciones BLOB) para la gestión de ficheros y campos de ficheros memo.

Índices compactos

El controlador DBFCDX crea índices compactos. En éstos ficheros los datos clave se guardan en formato comprimido, permitiendo una considerable reducción del tamaño del fichero índice.

Índices compuestos

Un índice compuesto (.cdx) es un fichero de índice, llamado contenedor de órdenes, que contiene varios índices, denominados tags. Los índices compuestos permiten que su aplicación pueda acceder a índices distintos usando un solo manejador de fichero. Por ésta razón es posible sobrepasar el límite de 15 ficheros de índice de Clipper. El controlador DBFCDX permite la apertura de tantos índices por área de trabajo como manejadores de fichero y memoria haya disponible.

Cuando se abre un índice compuesto, todos los tags del índice se actualizan automáticamente mientras se modifican los registros. Un tag de un índice es esencialmente igual que un índice sencillo y dispone de las mismas funciones. Por defecto, el primer tag (en orden de creación) de un índice compuesto es el índice activo.

Se debe tener en cuenta que hay algunas cosas a tener en cuenta al usar DBFCDX. Por ejemplo:

```
SET ORDER TO // orden nulo, deja el .cdx abierto para actualizar
```

```
SET ORDER TO TAG <cNombreOrden> [ IN <xcNombreContenedorOrdenes> ]
```

Índices condicionales

Con el controlador DBFCDX se pueden crear índices con una cláusula FOR incorporada. Estos índices condicionales disponen de una condición formada por cualquier expresión, incluyendo una función definida por el usuario. Cuando se actualiza la base de datos, solo se añaden al índice los registros que cumplen la condición, y se eliminan (del tag) automáticamente los registros que cumplían la condición anteriormente y ya no la cumplen.

INSTALACIÓN DE LOS FICHEROS DEL CONTROLADOR DBFCDX

El controlador DBFCDX viene en dos ficheros: DBFCDX.LIB y _DBFCDX.LIB. Se pueden encontrar en el subdirectorio \LIB del clipper 5.3.

ENLAZADO DEL CONTROLADOR DBFCDX

Para anlar el controlador indique al enlazador, en el área de librerías, que incluya los dos ficheros especificados más arriba.

USO DEL CONTROLADOR DBFCDX

Para usar éste controlador en un programa Clipper:

1.- Incluir el comando REQUEST DBFCDX al principio de su aplicación o del primer .prg que abra una base de datos DBFCDX.

2.- Use la cláusula VIA "DBFCDX" del mandato USE o bien especifique "DBFCDX" en el argumento *<cControlador>* si abre la base de datos con la función DBUseArea() o bien indique RDDSetDefault("DBFCDX") al principio de la aplicación para establecer el controlador por defecto en DBFCDX.

En este ejemplo se ilustra su uso:

REQUEST DBFCDX

.
.
.

USE Clientes INDEX Clientes NEW VIA "DBFCDX"

o bien

**REQUEST DBFCDX
RDDSetDefault("DBFCDX")**

.
.
.

USE Clientes INDEX Clientes NEW

CREACION DE INDICES CON DBFCDX

Para crear índices con DBFCDX utilizaremos el comando INDEX ON ... con una ligera variación: los TAGS.

**USE Clientes NEW VIA "DBFCDX"
INDEX ON Codigo TAG Codigo TO Clientes
INDEX ON Nombre TAG Nombre TO Clientes**

ABRIR UNA BASE DE DATOS CON INDICES CDX

Para abrir una base de datos con índices .cdx procederemos con el comando USE como se indica:

USE Clientes INDEX Clientes NEW VIA "DBFCDX"

ESTABLECER EN INDICE ACTUAL

Para establecer el índice actual con ficheros .cdx procederemos con una de éstas formas:

**USE Clientes INDEX Clientes NEW VIA "DBFCDX"
SET ORDER TO TAG Nombre IN Clientes**

o bien:

**USE Clientes INDEX Clientes NEW VIA "DBFCDX"
Clientes->(OrdSetFocus("Nombre" [,<Nombre del contenedor de ordenes>]))**

o también:

**USE Clientes INDEX Clientes NEW VIA "DBFCDX"
Clientes->(OrdSetFocus(2[,<Nombre del contenedor de ordenes>]))**

AVERIGUAR EL INDICE ACTIVO

Para saber que índice está activo en estos momentos actuaremos de la siguiente forma:

**USE Clientes INDEX Clientes NEW VIA "DBFCDX"
Clientes->(OrdSetFocus(2))**

cIndiceActivo := Clientes->(OrdSetFocus()) && devuelve "Nombre"

o bien

**USE Clientes INDEX Clientes NEW VIA "DBFCDX"
Clientes->(OrdSetFocus(2))**

nIndiceActivo := Clientes->(OrdNumber()) && devuelve 2

ESTABLECER SCOPES CON DBFCDX

Para establecer los límites de ámbito de control para los valores de las claves se utilizará la función ORDScope().

Para establecer que en nuestro fichero de Clientes nos saque solo los registros que tengan el código mayor o igual a 20 y sean inferior a 25 actuaremos del siguiente modo:

```
#define TOPSCOPE      0
#define BOTTOMSCOPE 1
USE Clientes INDEX Clientes NEW VIA "DBFCDX"
Clientes->( OrdSetFocus( 1 ) )           && Establecer el orden
Clientes->( DBGoTop() )
```

```
Clientes->( OrdScope( TOPSCOPE, "0020" ) )
Clientes->( OrdScope( BOTTOMSCOPE, "0025" ) )
```

```
WHILE !( Cliente->( Eof() ) )
    .
    .
    .
END DO
```

Se pueden cambiar los límites de forma dinámica y seguir trabajando como si tal cosa:

```
Clientes->( OrdScope( BOTTOMSCOPE, "0050" ) )           &&Límite inferior en 50
Clientes->( DBGoTop() )
```

```
WHILE !( Cliente->( Eof() ) )
    .
    .
    .
END DO
```

Si en vez de especificar un valor se especifica NIL, se tomará el límite, superior o inferior, del TAG activo.

