make.tol de Xls.Reader

Las funciones de esta librería Xls.Reader permiten construir lectores de ficheros Excel Xls, la versión clásica de Microsoft, a través de una conexión Odbc de Windows de Microsoft. La funcionalidad de este fichero principal make.tol es sólo probar las diferentes funcionalidades de esta librería que se implementan dentro del fichero xls.tol. Esta librería asume que existe una conexión Odbc de nombre conocido y que se guarda en una de las variables de control llamada XlsOdb y que esta conexión Odbc apunta a un fichero Excel del tipo Xls.

Las conexiones Odbc apuntan a un fichero o base de datos concreto, por lo que para no tener que crear una conexión Odbc para cada fichero Excel que se quiera leer, la conexión Odbc ha de apuntar siempre a un fichero Excel especial que hace de buffer temporal de lectura y cuya ruta se puede cambiar a través de la variable de control, XlsBuf. En cualquier caso, para las principales funcionalidades de lectura de esta librería Xls.Reader, existen 2 versiones: a) la que lee usuando la conexión Odbc y el fichero buffer Excel que indiquen las variables de control XlsOdb y XlsBuf, respectivamente y b) otras funciones diferentes en las que el nombre de la conexión Odbc y la ruta del fichero Excel buffer se pasan como parámetros de entrada.

La librería Xls.Reader proporciona funcionalidades para: a) asignar la variable de control Odbc o usar el valor por defecto, b) asignar la variable de control del fichero Excel buffer o usar el fichero por defecto, c) verificar que es un fichero Excel del tipo Xls, d) para copiar el fichero Excel que se quiere leer sobre el buffer de lectura (que es el fichero al que apunta la conexión Odbc) y e) para leer un rango de celdas de una de las hojas del fichero Excel. La funcionalidad de lectura Excel Xls de esta librería, a través de Odbc, usa internamente la versión de Sql (Structured Query Language) limitada del Excel de Microsoft.

Para las pruebas de la funcionalidad de esta libraría Xls.Reader de lectura de Excel Xls a través de Odbc se han empleado: a) cuatro ficheros Excel de prueba, con diferentes volumentes y tipo de datos, que se guardan en el directorio dat y b) un fichero Excel buffer que se guarda en el directorio tmp. Pero pueden leerse otros ficheros Excel, emplearse otro buffer de lectura y ubicarlos en los directorios que se deseen. Con estos casos de prueba se ha comprobado el funcionamiento de las funciones de esta librería para las versiones de Tol 1.1.1, 1.1.5, 1.1.6 y 2.0.1.

Árbol de ficheros

- Xls.Reader funciones de lectura de hojas Excel Xls a través de una conexión Odbc
  - ← make.tol programa de test de las funciones de lectura de Excel con Odbc
  - ← make.bat mandato de ejecución del programa de test de lectura Excel
  - tol directorios de fichero con código fuente Time Oriented Languaje
    - cmm funciones comunes del programa de lectura con Odbc de Excel Xls
      - ← xls.tol librería de funciones para leer Excel Xls a través de Odbc
    - ← inc.tol para la inclusión de ficheros en lenguaje de programación Tol
  - → lectura.excel.xls.odbc.png imagen de la lectura Tol de un fichero Excel a través de

Odbc

→ xls_reader.pdf documento de funciones de lectura con Odbc de ficheros Excel Xls

Declaraciones

Inclusiones

- `Set allInc`
  Inclusion de las funciones comunes.

Pruebas

- `Real timIni`
  Para controlar tiempos.
- `Real tst001`
  Primera prueba.
- `Real tst002`
  Segunda prueba.
- `Real tst003`
  Tercera prueba.
- `Real tst004`
  Cuarta prueba.

Set allInc

```
////////////////////////////////////////////////////////////////////////////
Set  allInc = Include("tol/inc.tol");
////////////////////////////////////////////////////////////////////////////
PutDescription("Inclusion de las funciones comunes.", allInc);
////////////////////////////////////////////////////////////////////////////
```

Real timIni

```
////////////////////////////////////////////////////////////////////////////
Real timIni = Copy(Time);
////////////////////////////////////////////////////////////////////////////
PutDescription("Para controlar tiempos.", timIni);
////////////////////////////////////////////////////////////////////////////
```

Real tst001

```
////////////////////////////////////////////////////////////////////////////
Real tst001 = If(FALSE, // Poner true para realizar este test
{
  Set  View(SetOfReal(XlsIsExcelXls(XlsBuf)), ""); // Ha de ser cierto

  Set  carTab = XlsReadSheet("dat/car.xls", "Hoja1", "A", "I");
  Set  View(carTab, "");

  Card(carTab)
},FALSE);
////////////////////////////////////////////////////////////////////////////
PutDescription("Primera prueba.", tst001);
```

```
////////////////////////////////////////////////////////////////////////////
```

Real tst002

```
////////////////////////////////////////////////////////////////////////////
Real tst002 = If(FALSE, // Poner true para realizar este test
{
  // Este Excel tiene 1 sola hoja,
  // por lo que se puede leer sin pasarle su nombre.
  Set  gasTab = XlsReadSheet("dat/gas.xls", "", "A", "H");
  Set  View(gasTab, "");

  Card(gasTab)
},FALSE);
////////////////////////////////////////////////////////////////////////////
PutDescription("Segunda prueba.", tst002);
////////////////////////////////////////////////////////////////////////////
```

Real tst003

```
////////////////////////////////////////////////////////////////////////////
Real tst003 = If(FALSE, // Poner true para realizar este test
{
  Set  truTab = XlsReadSheet("dat/tru.xls", "", "A", "H");
  Set  View(truTab, "");

  Card(truTab)
},FALSE);
////////////////////////////////////////////////////////////////////////////
PutDescription("Tercera prueba.", tst003);
////////////////////////////////////////////////////////////////////////////
```

Real tst004

```
////////////////////////////////////////////////////////////////////////////
Real tst004 = If(FALSE, // Poner true para realizar este test
{
  // De las 2 hojas de nombre complejo y sencillo, lee sencillo
  Set  tinTab = XlsReadSheet("dat/tin.xls", "sencillo", "A", "C");
  Set  View(tinTab, "");

  Card(tinTab)
},FALSE);
////////////////////////////////////////////////////////////////////////////
PutDescription("Cuarta prueba.", tst004);
////////////////////////////////////////////////////////////////////////////
```

xls.tol de Xls.Reader

Leer ficheros Excel apoyandose en una conexion Odbc. Esta conexion ha de existir y, para no tener que crear una conexion Odbc para cada fichero Excel que hay que leer, ha de apuntar a un fichero Excel especial que hace de buffer temporal de lectura.

Declaraciones

Variables de control

- `Text XlsBuf`
  Fichero auxiliar para leer Excel Xls.
- `Text XlsOdb`
  Conexion ODBC por defecto para Excel Xls.

Constantes

- `Text Xls2By`
  2 bytes del inicio de un Xls.

Funciones

- `Set XlsSetCtr(Text newBuf, Text newOdb)`
  Memoriza el camino al fichero buffer Excel por defecto y a la conexion Odbc que se emplearan a partir de la llamada a esta funcion, esto es, actualiza los parametros de control. Retorna los valores anteriores por si se desean restaurar posteriormente.
- `Real XlsIsExcelXls(Text inpPth)`
  Retorna cierto si inpPth tiene la extension de un fichero Excel Xls y sus 2 primeros bytes coinciden con los de un fichero Excel Xls.
- `Real XlsCopy(Text filInp, Text filOut)`
  Retorna cierto si pueda copiar el fichero filInp sobre filOut. No se copia si el origen no existe o no es un Excel Xls.
- `Set XlsReadSheetOdbc(Text xlsPth, Text shtNam, Text iniPos, Text endPos, Text xlsOdb, Text xlsBuf)`
  Retorna una tabla, como conjunto de filas de conjunto de celdas, resultado de leer del fichero Excel Xls de ruta xlsPth, la hoja shtNam, desde la posicion iniPos a la posicion endPos. Esta lectura Odbc mediante Sql asume que la primera fila del fichero Excel son los nombres de los campos. Las posiciones iniciales y finales pueden ser del tipo: - A y K: lee de la columna A a la columna K todas las filas hasta el final o - B4 y Z9: lee el rectangulo de celdas derminado por esas 2 esquinas. Esta funcion recibe como parametros la conexion Odbc xlsObc y la ruta al fichero buffer Excel en el parametro xlsBuf. Retorna el conjunto vacio en caso de error.
- `Set XlsReadSheet(Text xlsPth, Text shtNam, Text iniPos, Text endPos)`

Retorna una tabla, como conjunto de filas de conjunto de celdas, resultado de leer del fichero Excel Xls de ruta xlsPth, la hoja shtNam, desde la posicion iniPos a la posicion endPos. Esta lectura Odbc mediante Sql asume que la primera fila del fichero Excel son los nombres de los campos. Las posiciones iniciales y finales pueden ser del tipo: - A y K: lee de la columna A a la columna K todas las filas hasta el final o - B4 y Z9: lee el rectangulo de celdas derminado por esas 2 esquinas. Esta funcion asume la conexion Odbc (XlsOdb) y la ruta al fichero buffer Excel por defecto (XlsBuf).

Variables de control

Text XlsBuf

```
////////////////////////////////////////////////////////////////////////////
Text XlsBuf = "tmp/excelbuffer.xls"; //
////////////////////////////////////////////////////////////////////////////
PutDescription(
"Fichero auxiliar para leer Excel Xls.",
XlsBuf);
////////////////////////////////////////////////////////////////////////////
```

Text XlsOdb

```
////////////////////////////////////////////////////////////////////////////
Text XlsOdb = "ExcelBuffer";
////////////////////////////////////////////////////////////////////////////
PutDescription(
"Conexion ODBC por defecto para Excel Xls.",
XlsOdb);
////////////////////////////////////////////////////////////////////////////
```

Constantes

Text Xls2By

```
////////////////////////////////////////////////////////////////////////////
Text Xls2By = Char(208)+Char(207);
////////////////////////////////////////////////////////////////////////////
PutDescription(
"2 bytes del inicio de un Xls.",
XlsOdb);
////////////////////////////////////////////////////////////////////////////
```

Set XlsSetCtr()

```
////////////////////////////////////////////////////////////////////////////
Set  XlsSetCtr(Text newBuf, // Camino del fichero Excel buffer por defecto
               Text newOdb) // Identificador Odbc por defecto
////////////////////////////////////////////////////////////////////////////
{
  Set  oldVal = SetOfSet(Copy(XlsBuf), Copy(XlsOdb)); // Valores actuales
  Text(XlsBuf:=Copy(newBuf)); // Memoriza el nuevo buffer
  Text(XlsOdb:=Copy(newOdb)); // Memoriza la nueva conexion Odbc
  oldVal // Retorna los antiguos valores
};
////////////////////////////////////////////////////////////////////////////
```

```
PutDescription(
"Memoriza el camino al fichero buffer Excel por defecto y a la conexion Odbc
que se emplearan a partir de la llamada a esta funcion, esto es, actualiza
los parametros de control.
Retorna los valores anteriores por si se desean restaurar posteriormente.",
XlsSetCtr);
////////////////////////////////////////////////////////////////////////////
```

Real XlsIsExcelXls()

```
////////////////////////////////////////////////////////////////////////////
Real XlsIsExcelXls(Text inpPth) // Input path, the Xls Excel file path
////////////////////////////////////////////////////////////////////////////
{
  If(!TextEndAt(ToLower(inpPth), ".xls"), FALSE, // Comprobar extension
  {
    Text ini2Ch = Sub(ReadFile(inpPth),1,2);
    ini2Ch == Xls2By                              // Comprobar el inicio
  })
};
////////////////////////////////////////////////////////////////////////////
PutDescription(
"Retorna cierto si inpPth tiene la extension de un fichero Excel Xls y
sus 2 primeros bytes coinciden con los de un fichero Excel Xls.",
XlsIsExcelXls);
////////////////////////////////////////////////////////////////////////////
```

Real XlsCopy()

```
////////////////////////////////////////////////////////////////////////////
Real XlsCopy(Text filInp, // Input file path
             Text filOut) // Output file path
////////////////////////////////////////////////////////////////////////////
{
  Text dos(Text pth) { Char(34)+Replace(pth, "/", "\\")+Char(34) };

  If(!FileExist    (filInp), FALSE, // Si el origen no existe no se copia
  If(!XlsIsExcelXls(filInp), FALSE, // Si no es Excel Xls no se copia
      System("copy " + dos(filInp) + " " + dos(filOut))))
};
////////////////////////////////////////////////////////////////////////////
PutDescription(
"Retorna cierto si pueda copiar el fichero filInp sobre filOut.
No se copia si el origen no existe o no es un Excel Xls.",
XlsCopy);
////////////////////////////////////////////////////////////////////////////
```

Set XlsReadSheetOdbc()

```
////////////////////////////////////////////////////////////////////////////
Set XlsReadSheetOdbc(Text xlsPth, // Direccion del fichero Excel
                     Text shtNam, // Nombre de la hoja a leer (sin $)
                     Text iniPos, // Posicion inicial tipo A1 o A
                     Text endPos, // Posicion final tipo Z9 o Z
                     Text xlsOdb, // Nombre de la conexion Odbc
                     Text xlsBuf) // Rura del buffer Odbc
////////////////////////////////////////////////////////////////////////////
{
  If(!XlsCopy(xlsPth, xlsBuf), Empty, // No se puede copiar
  {
    Real xlsOpn = DBOpen(xlsOdb, "", ""); // Sin usuario ni password
    Text xlsSql = "select * from ["+shtNam+"$"+iniPos+":"+endPos+"];";
```

```
    Set  xlsTab = DBTable(xlsSql);
    Real DBClose(xlsOdb);
    xlsTab
  })
};
////////////////////////////////////////////////////////////////////////
PutDescription(
"Retorna una tabla, como conjunto de filas de conjunto de celdas, resultado
de leer del fichero Excel Xls de ruta xlsPth, la hoja shtNam,
desde la posicion iniPos a la posicion endPos.
Esta lectura Odbc mediante Sql asume que la primera fila del fichero Excel
son los nombres de los campos.
Las posiciones iniciales y finales pueden ser del tipo:
- A y K: lee de la columna A a la columna K todas las filas hasta el final o
- B4 y Z9: lee el rectangulo de celdas derminado por esas 2 esquinas.
Esta funcion recibe como parametros la conexion Odbc xlsObc y la ruta al
fichero buffer Excel en el parametro xlsBuf.
Retorna el conjunto vacio en caso de error.",
XlsReadSheetOdbc);
////////////////////////////////////////////////////////////////////////
```

Set XlsReadSheet()

```
////////////////////////////////////////////////////////////////////////
Set XlsReadSheet(Text xlsPth, // Direccion del fichero Excel
                 Text shtNam, // Nombre de la hoja a leer (sin $)
                 Text iniPos, // Posicion inicial tipo A1 o A
                 Text endPos) // Posicion final tipo Z9 o Z
////////////////////////////////////////////////////////////////////////
{ XlsReadSheetOdbc(xlsPth, shtNam, iniPos, endPos, XlsOdb, XlsBuf) };
////////////////////////////////////////////////////////////////////////
PutDescription(
"Retorna una tabla, como conjunto de filas de conjunto de celdas, resultado
de leer del fichero Excel Xls de ruta xlsPth, la hoja shtNam,
desde la posicion iniPos a la posicion endPos.
Esta lectura Odbc mediante Sql asume que la primera fila del fichero Excel
son los nombres de los campos.
Las posiciones iniciales y finales pueden ser del tipo:
- A y K: lee de la columna A a la columna K todas las filas hasta el final o
- B4 y Z9: lee el rectangulo de celdas derminado por esas 2 esquinas.
Esta funcion asume la conexion Odbc (XlsOdb) y la ruta al fichero buffer Excel
por defecto (XlsBuf).",
XlsReadSheet);
////////////////////////////////////////////////////////////////////////
```

inc.tol de Xls.Reader

Inclusion de ficheros comunes

Declaraciones

Inclusiones comunes

- `Set xlsInc`

  Funciones de lectura de Excel Xls.

Set xlsInc

```
////////////////////////////////////////////////////////////////////////////
Set  xlsInc = Include("cmm/xls.tol");
////////////////////////////////////////////////////////////////////////////
PutDescription("Funciones de lectura de Excel Xls.", xlsInc);
////////////////////////////////////////////////////////////////////////////
```