

MultiBase Cosmos

Notas a la versión 4.2, release 2

BASE100

BASE 100, S.A.

www.base100.com

Índice

1	NUEVAS FUNCIONALIDADES	3
1.1	IMPLEMENTACIONES	3
1.2	MÉTODOS DE LA CLASE PRNDOCUMENT	3
1.3	MÉTODOS DE LA CLASE SIMPLECONTROL	4
1.4	MÉTODOS DE LA CLASE FORMTABLE	5
1.5	MÉTODOS DE LA CLASE FORM	5
2	MEJORAS	7
3	CORRECCIONES	8

© Copyright BASE 100, S.A. Todos los derechos reservados. Ninguna parte de este documento puede ser reproducida ni transmitida por medio alguno sin permiso previo por escrito del titular del copyright. Todos los productos citados en este documento son marcas registradas o marcas comerciales registradas de sus respectivos propietarios.

[NT_CO_4.2.2_v1.docx]

1 Nuevas funcionalidades

1.1 Implementaciones

Posibilidad de uso de ToolTips multilinea. Para indicar un salto de línea se empleará la cadena de caracteres “\n”.

Posibilidad de buscar columnas de una tabla en el repositorio del proyecto en el entorno de desarrollo.

Se ha mejorado el posicionamiento dentro de las listas al teclear sobre ellas. Al pulsar teclas sin lapso de tiempo entre ellas, la búsqueda se realiza de manera incremental, mientras que si se deja un lapso de tiempo entre las pulsaciones se irá situando sobre el elemento que comience por el carácter representado por la última tecla marcada.

Se ha creado una nueva variable de entorno: REMOVELOGFILEONEXIT. Esta variable permite que el motor “CTS QL monitor” elimine el fichero de *log* al finalizar la sesión. Si su valor es TRUE borrará el fichero, mientras que si es FALSE no lo hará. Su valor por defecto es FALSE. Esta variable de entorno debe indicarse en el fichero de entorno “ctsql.ini”.

Asignación de iconos a las columnas de un *grid* en tiempo de ejecución. Para ello se han implementado dos métodos: SetGridColumnIcoFile y SetGridColumnIco ([ver Métodos de la clase SimpleControl](#)).

Posibilidad de crear *lookup* dinámicos para objetos FormTable (ver [Métodos de la clase FormTable](#)).

Posibilidad de crear *lookup* dinámicos para objetos Form (ver [Métodos de la clase Form](#)).

Posibilidad de ocultar los botones de impresión de manera dinámica en la ventana de previsualización de listados y de hacer *zoom* sobre la página (ver [Métodos de la clase PrnDocument](#)).

Se ha implementado la posibilidad de crear y clonar algunos controles ([ver Métodos de la clase SimpleControl](#)).

Controles que se pueden crear:

RADIO	TEXT	SPIN
BITMAP	SLIDER	BOX
PANEL	BUTTON	PERCENTAGE BOX
EDIT	CHECK	

Controles que se pueden clonar:

TEXT	SPIN	BUTTON
PERCENTAGE BOX	RADIO	CHECK
SLIDER	PANEL	BOX

Code Insight. Cuando se abre un paréntesis o unas comillas en el entorno de desarrollo, éstos se cerrarán automáticamente.

1.2 Métodos de la Clase PrnDocument

PreviewEx

Este método permite ocultar los botones “Print” y “Direct Print” de la ventana de preview, así como de indicar el porcentaje de zoom de manera dinámica.

```
PreviewEx(VAR aWindow as Window, maximize as Boolean default FALSE,
hidePrintBtn as Boolean default FALSE,hidePrintDirectbtn as Boolean default
FALSE ,zoom as Smallint default 100)
```

Parámetros:

aWindow	Objeto de tipo Window cuya ventana será padre de la ventana de presentación preliminar del documento. Si no se indica este parámetro, su valor por defecto es NULL. La ventana padre será la del último Form modal abierto.
maximize	Si es TRUE muestra la ventana maximizada.
hidePrintBtn	Booleano donde se indica si se quiere ocultar o no el botón <i>Print</i> del <i>Preview</i> . Si se pasa TRUE se oculta el botón, mientras que si se pasa FALSE se muestra.
hidePrintDirectbtn	Booleano donde se indica si se quiere ocultar o no el botón <i>DirectPrint</i> del <i>Preview</i> . Si se pasa TRUE se oculta el botón, mientras que si se pasa FALSE se muestra.
zoom	Entero donde se indica el <i>zoom</i> que se quiere hacer.

1.3 Métodos de la Clase SimpleControl

CreateControl

Permite crear controles y clonarlos dinámicamente.

```
CreateControl(newCtrName as Char ,likeCtr as
SimpleFormControl ,newCtrType as Char ,nItems as Smallint,
newCtrPage as Smallint)
```

Parámetros:

newCtrName	Nombre del control nuevo o clonado.
likeCtr	Indica el control del cual se quiere clonar el control nuevo. Si no tiene valor se creará un control nuevo. Si tiene valor el control que se genere será una réplica del que se indique en este parámetro.
newCtrType	Indica el tipo del nuevo control creado. Es un <i>char</i> . Se ignora cuando se indica un valor en el parámetro.
nItems	Indica el número de ítems que tendrá el control creado en el caso de que sea un <i>Radio Button</i> o un <i>Spin</i> . Este parámetro se ignora si se ha indicado un valor en el parámetro "likeCtr".
newCtrPage	Indica la página del control padre donde se ubicará el nuevo control.

SetGridColumnIco

Este método permite asignar un icono a una columna de un control *grid*.

```
SetGridColumnIco(index as Integer ,icon as Smallint
default 0) return Boolean
```

Parámetros:

index	Indica el número de la columna sobre la que se aplicará el método.
icon	Indica el número de icono que se asignará.

Retorna:

TRUE en caso de realizar la operación y FALSE en caso contrario.

SetGridColumnIcoFile

Asigna un fichero de iconos a un control *grid*.

```
SetGridColumnIcoFile(icoFile as Char) return Boolean
```

Parámetros:

icoFile Indica el nombre del fichero de iconos.

Retorna:

TRUE en caso de realizar la operación y FALSE en caso contrario.

1.4 Métodos de la Clase FormTable

LookupColumn

Este método permite crear un *lookup* en ejecución. La variable *lookup* podrá ser un campo del objeto FormTable o una pseudocolumna de la clase FormTable. Muestra el contenido de la columna que se indica en el parámetro *collookup* en la variable *object*.

Los *lookup* se resuelven teniendo en cuenta el orden en el que se encuentran definidos los objetos en el FormTable.

```
LookupColumn(object as Char, joinname as Char, tablereferenced  
as Char, fkcols as Char, refcols as Char, collookup as Char) return Boolean
```

Parámetros:

object Esta variable debe ser del mismo tipo que la variable de la columna sobre la que se quiera hacer el lookup, y tiene que estar definida dentro de un objeto de la clase FormTable.

joinname Nombre del *join* que se va a crear.

tablereferenced Nombre de la tabla a la que se hace referencia.

fkcols Nombre de las columnas por las que se desea hacer el enlace. Estos campos corresponden a la tabla activa del Form. Si hay más de una columna, éstas irán separadas por comas.

refcols Nombre de las columnas a las que se va a hacer referencia.

collookup Nombre de la columna de la tabla referenciada que se quiere mostrar.

Retorna:

TRUE en caso de haber realizado la operación correctamente y FALSE en caso de producirse algún error.

1.5 Métodos de la Clase Form

SetLookup

Permite crear *lookups* dinámicos de objetos definidos en un Form.

```
Setlookup(obj1 as Char, obj2 as Char, tabref as Char, tocolumn as  
Char, lookupcolumn as Char) return Boolean
```

Parámetros:

obj1	<p>Lista de objetos del Form que, al modificar su valor, hará que se recojan los valores en los objetos del Form que se indiquen.</p> <p>Esta lista de parámetros debe ir separada por comas.</p> <p>Este objeto podrá formar parte de varios <i>lookup</i>.</p> <p>El número de objetos indicado en este parámetro ha de ser igual al número de columnas especificadas en el parámetro "tocolumn" de este método.</p>
obj2	<p>Objeto del Form donde se recoge el valor.</p>
tabref	<p>Nombre de la tabla de la que queremos obtener el valor.</p> <p>En este parámetro sólo se podrá indicar una única tabla.</p>
tocolumn	<p>Lista de columnas de comparación de la tabla de referencia.</p>
lookupcolumn	<p>Columna de la tabla de la que se va a obtener el valor.</p> <p>El número de elementos de este parámetro sólo puede ser uno.</p>

Retorna:

TRUE si los parámetros son correctos y ha podido realizar el lookup. En caso contrario retorna FALSE.

2 Mejoras

Se ha actualizado la documentación relativa al servicio “monser” (ver [Puesta en marcha y configuración del servicio Monser](#)).

Se ha optimizado la opción de búsqueda de cadenas y ficheros en el entorno de desarrollo.

3 Correcciones

Code Insight

Error de protección general cuando se escribía “.” detrás del nombre de un objeto definido en el módulo, en un *include* o en una librería, y el módulo no tenía código.

CORREGIDO

Dropedit

Cuando un control *dropedit* tenía dos columnas, la primera vez se podía pinchar sobre cualquiera de ellas para seleccionar un valor. Si se intentaba repetir la operación sólo se podía hacer sobre la primera columna.

CORREGIDO

Automatic Sort en los grid

Perdía la *where* cuando se pulsaba en la cabecera de una columna por la que se quería ordenar.

CORREGIDO

Entorno de desarrollo

No clonaba correctamente un clase *Form* cuyo bloque de código fuera superior a 32 Kb.

CORREGIDO (se amplía el buffer a 2 Mbytes)

Cosmos

Si una caja con *scroll* estaba dentro de otra caja también con *scroll*, el desplazamiento no se realizaba correctamente y no se veía el campo que tenía el foco.

CORREGIDO

Cosrun

Si existía un control habilitado sobre una caja deshabilitada, el control habilitado no recibía los eventos del ratón, ya que Cosmos pensaba que este control era hijo.

CORREGIDO

Active X

Error de ejecución al intentar hacer un *GetProperty* de un elemento *VT_DATE* de una *ActiveX*.

CORREGIDO

Cuando intentábamos cambiar el tamaño de un *control list box* con *scroll* horizontal utilizando el método *Size*, el control se redimensionaba, pero la barra de *scroll* no.

CORREGIDO

Al instalar el Run Time de Cosmos, cuando se intentaba ejecutar el *PageEdit* mostraba un error indicando “cant main Windows”.

CORREGIDO

Método *LoadSelect*. Si la *SELECT* tenía una cláusula *ORDER BY* en mayúsculas, mostraba el error: “Parámetro incorrecto recibido por el servidor de CTSQL”. Cargaba la lista, pero devolvía un 0 en lugar del número de filas cargadas.

CORREGIDO