

MultiBase Cosmos

Notas a la versión 7.2

NUEVA UTILIDAD INCLUIDA A PARTIR DE ESTA VERSIÓN

[CosSignFile](#). La dll CosSignFile (cossigfile.dll) permite firmar digitalmente archivos en formato PDF, XML y binario, así como la firma de ficheros de factura electrónica XML en formato FacturaE versiones 3.2.1 y 3.2.2.

BASE100

BASE 100, S.A.
www.base100.com

Índice

1. IMPLEMENTACIONES Y MEJORAS	4
1.1 CLASE ARRAY	4
1.2 CLASE CHAR.....	4
1.3 CLASE FORMTABLE.....	4
1.4 CLASE JSON.....	5
1.5 CONTROLES	5
1.5.1 <i>Band Group</i>	5
1.5.2 <i>Bitmap</i>	5
1.5.3 <i>Drop Edit y Edit Field</i>	5
1.5.4 <i>Grid</i>	5
1.5.5 <i>List Box</i>	5
1.5.6 <i>Tab Control</i>	6
1.6 COSMOS WEBSERVER.....	6
1.6.1 <i>Fichero de Log</i>	6
1.7 ESTILO VISUAL COSMOS 7	7
1.8 EVENTOS.....	7
1.8.1 <i>Form</i>	7
1.8.2 <i>Drag & Drop</i>	7
1.8.3 <i>RClickHeader</i>	7
1.9 EXPORTACIÓN A EXCEL.....	7
1.10 ICONOS.....	8
1.11 LISTAS EDITABLES	8
1.12 MODO DE EDICIÓN AUTOMÁTICO	8
1.13 FIRMA DIGITAL.....	8
1.14 VISTA PREVIA EN LOS LISTADOS	8
1.15 MEJORAS	9
2. NUEVOS MÉTODOS	10
2.1 CLASE CHAR.....	10
2.1.1 <i>LoadFromFile</i>	10
2.2 CLASE MODULE	10
2.2.1 <i>LoadCustomColors</i>	10
2.2.2 <i>GetLastPreviewZoom</i>	10
2.2.3 <i>GetSystemTimeUTC</i>	10
2.2.4 <i>IsRunningInCosmosWebServer</i>	11
2.2.5 <i>SetLastPreviewZoom</i>	11
2.3 CLASE PRNDOCUMENT.....	11
2.3.1 <i>GetLastPreviewZoom</i>	11
2.4 CLASE SIMPLECONTROL.....	12
2.4.1 <i>GetDragColumn</i>	12
2.4.2 <i>GetDropColumn</i>	12
2.4.3 <i>Refresh</i>	12
2.4.4 <i>SetListAllRowsHeight</i>	12
3. NUEVAS PROPIEDADES Y EVENTOS	13

3.1	PROPIEDAD AUTO FIT	13
3.2	PROPIEDAD VARLENGTH.....	13
3.3	EVENTO CHANGESIZE.....	13
4.	VARIABLES DE ENTORNO	14
5.	CORRECCIONES	17
5.1	RUNTIME.....	17
5.2	ENTORNO DE DESARROLLO	18
5.3	COSMOS WEBSERVER.....	18
5.4	CTSQL	18
5.5	COSMOSDATA.....	18

© 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.

[NTC072v1]

1. Implementaciones y mejoras

1.1 Clase Array

Se ha aumentado el tamaño máximo de un Array, que ahora podrá ser de 2.147.483.647 elementos.

El tipo de los parámetros de los métodos Resize de la clase Array y Size de la clase Container pasa de Smallint a Integer.



Esta modificación hace que los programas compilados con versiones anteriores no puedan ejecutarse con esta versión. Es necesario recompilar las aplicaciones.

1.2 Clase Char

Se ha eliminado el límite de 32.767 bytes en objetos de la clase Char, estableciéndose ahora en 2.147.483.647 bytes. Para ello, se han modificado algunas de las funciones de la clase Char con objeto de poder manejar cadenas de caracteres con una longitud mayor de 32.767 bytes.

Operadores modificados a partir de esta versión:

- Operator +
- Operator =
- Operator []

```
obj[ idx as integer, len as integer default 1 ]
```

Métodos modificados a partir de esta versión:

- Replace
- ReplaceAt
- Translate
- Locate
- Count
- NumWords
- GetWord
- StrRepeat

NOTA: La definición del tamaño inicial de un objeto Char seguirá siendo como máximo de 32.767 bytes. El tamaño del Char se podrá modificar en ejecución con los operadores de asignación y concatenación.



Este cambio en el tipo de dato de parámetros o de retorno de las funciones supone que los binarios son incompatibles.

Además, se deberá chequear el tipo de las variables de retorno, parámetros de las funciones y redefinición de los métodos indicados anteriormente.

1.3 Clase FormTable

Se han optimizado las consultas para el caso de las tablas de un FormTable asociado a un Grid. Para ello, se ha implementado la variable de entorno LOOKUPOUTERJOIN.

Si la variable de entorno está definida, en lugar de crear un cursor por cada Lookup se crea una sola frase con Outer Joins a las tablas de Lookup.

1.4 Clase Json

Se ha optimizado el algoritmo del método SaveToFile y el del Conversor a Char.

1.5 Controles

1.5.1 Band Group

Posibilidad de cambiar el tamaño de los controles hijos de las bandas de un control Band Group en ejecución antes de añadir la banda.

NOTA: Si se amplía el tamaño de algún control hijo de alguna de las bandas (control Box) habrá que ampliar también el tamaño de la banda si el ancho de ésta fuese menor, ya que éste no se amplía automáticamente.

1.5.2 Bitmap

Se ha implementado la propiedad **Auto Fit** en los controles Bitmap para que la imagen asociada al control no se deforme cuando se cambie el tamaño del control en ejecución.

Esta propiedad se ha implementado tanto para controles de la Screen como en los controles de la clase Page.

1.5.3 Drop Edit y Edit Field

Se ha implementado la posibilidad de que Cosmos *recuerde* la posición del cursor en los controles de edición para que, al acceder de nuevo al control, el cursor se sitúe en la misma posición (**EDITFIELDNOMARKONGETFOCUS**). Al regresar el foco al control el texto del campo no estará seleccionado.

1.5.4 Grid

A partir de esta versión es posible copiar el texto de los controles de tipos Drop List y Check (ALLOWCOPYCONTROLDATA). En el caso de los controles de tipo Check, los valores que se copiarán serán 0 ó 1 (0 si no está marcado el check y 1 si está marcado).

Se ha implementado la posibilidad de que el evento On Click, en los controles Grid, no se lance al hacer clic en la cabecera del Grid. Para ello habrá que definir la variable de entorno NOSENDCLICKONGRIDCLICKHEADER.

El método AlternateBackColor de la clase SimpleControl se podrá invocar desde los controles de tipo Grid, además de en los controles de tipo List Box.

Se han optimizado las consultas que se realizan desde Cosmos cuando un control Grid tiene asociada una tabla y algunos de los campos del Formtable son de tipo Lookup (LOOKUPOUTERJOIN).

1.5.5 List Box

Se podrá definir la altura en píxeles de las filas de una lista. El tamaño que se defina se aplicará a todas las filas de esa lista. Método SetListAllRowsHeight de la clase SimpleControl.

Posibilidad de desplazarse en las listas de izquierda a derecha con la rueda de ratón. Para ello, mantenga pulsada la tecla **[Ctrl]+rueda del ratón**.

Si una celda de una lista tiene asignado un estilo, su color de fondo será el definido en el estilo, independientemente de que la fila a la que pertenece tenga el foco.

A partir de esta versión se permite totalizar por columnas de tipo DATE y DATETIME.

1.5.6 Tab Control

Se ha implementado el evento **RClickHeader**.

NOTA. Si el evento está siendo utilizado como evento genérico, habrá que tener en cuenta que, a partir de esta versión, el código también se ejecutará al pulsar con el botón derecho del ratón sobre la pestaña del Tab Control.

1.6 Cosmos WebServer

Se ha implementado la variable de entorno **CWSIGNOREMAIN** para controlar si se debe o no ejecutar la función *main* del módulo que implementa el servicio REST que se está ejecutando.

Se ha implementado el método **IsRunningInCosmosWebServer** de la clase Module para consultar si la aplicación Cosmos se está ejecutando desde Cosmos WebServer.

Por defecto, siempre que se produzca un error y no se haya activado ningún código de HTTP con el método SetExecStatus de la clase Module, Cosmos activará el error 500 de HTTP.

1.6.1 Fichero de Log

La salida de los errores generados por el runtime de Cosmos WebServer durante los procesos podrá ser a fichero si en el directorio temporal se crea un fichero con el nombre **CWSLog.log**.

Cosmos WebServer como servicio

Si se produce un error durante la ejecución de un servicio Cosmos y el ErrorLevel es mayor de 0, en el fichero de log se volcará información referente al error, así como el método y el módulo donde se ha producido el error, y la pila de llamadas de la aplicación.

Si se produce un error durante la ejecución y el ErrorLevel es igual a 0, no se escribirá la información del error en el fichero de log.

Si en el código fuente del servicio Cosmos se ejecuta el método Trace, en el fichero de log se escribirá el texto asociado a dicho método.

Cosmos WebServer como aplicación interactiva

Si se produce un error durante la ejecución de un servicio Cosmos y el ErrorLevel es mayor de 0, en el fichero de log se volcará información referente al error, así como el método y el módulo donde se ha producido el error, y la pila de llamadas de la aplicación. Esta misma información se mostrará en pantalla mediante un MessageBox.

Si se produce un error durante la ejecución y el ErrorLevel es igual a 0, no se escribirá la información del error en el fichero de log y no se mostrará por pantalla.

Si en el código fuente del servicio Cosmos se ejecuta el método Trace, se mostrará por pantalla un MessageBox con el texto asociado al método Trace y no escribirá el texto en el fichero de log.

1.7 Estilo Visual Cosmos 7

A partir de esta versión se podrá cambiar en ejecución el estilo visual de una aplicación Cosmos. Los cambios que se pueden hacer son los siguientes:

- Activar o desactivar el estilo Cosmos 7. El valor de la variable COSMOSVISUALMODE se podrá modificar con el método PutEnv de la clase Module.
- Cambiar el esquema de colores de la aplicación cuando el estilo Cosmos 7 esté activo. Se ha implementado el método **LoadCustomColors** de la clase Module.

No será necesario definir la sección **[Custom Colors]** en los ficheros de configuración utilizados por Cosmos como sucedía en la versión 7.0. De esta manera se podrán tener varios esquemas de colores para una misma aplicación Cosmos.

Para que cualquiera de los dos cambios sean efectivos habrá que llamar al nuevo método **Refresh** de la clase SimpleControl (este método se ha implementado en esta versión).

1.8 Eventos

1.8.1 Form

Se ha implementado el evento **ChangeSize** de la clase Form que permite conocer cuándo se ha cambiado, en tiempo de ejecución, el tamaño de una ventana.

1.8.2 Drag & Drop

Se han implementado los métodos **GetDragColumn** y **GetDropColumn** que permiten conocer respectivamente, en los controles List Box, las columnas desde la que se ha arrastrado (drag) y sobre la que se ha soltado (drop) un elemento.

1.8.3 RClickHeader

Se ha implementado el evento **RClickHeader** para los controles de tipo Tab. Este evento se lanzará cuando el usuario pulse con el botón derecho del ratón sobre una pestaña del control.

1.9 Exportación a Excel

Además del formato de hoja de cálculo XML 2003 que soporta Cosmos al exportar a Excel, a partir de esta versión es posible exportar a Excel en formato **XLSX**.

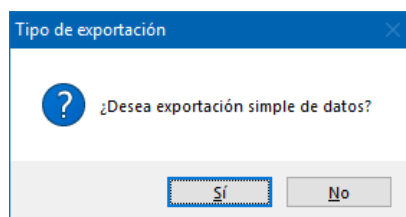
Para que se generen con este nuevo formato, el nombre de los ficheros que se crean desde cualquiera de los métodos o utilidades de Cosmos deberán tener la extensión **XLSX**.

En los casos de los métodos que reciben el nombre del fichero como parámetro habrá que modificarlo. Estos métodos son los siguientes:

- Método **UnloadTo** de la clase SqlServer.
- Método **ExportToExcel** de la clase SimpleControl.
- Método **SelectWindow** de la clase SqlServer.
- Método **SelectWindowAt** de la clase SqlServer.

Si al generar el fichero está disponible la opción **Save As** habrá que elegir como **Tipo** de fichero **Excel File (*.xlsx)**. Estos casos son:

- Sql-Interactive.
- Vista previa de los listados (**Preview**). Si en la opción **Save As** se selecciona como tipo **Excel File (*.xlsx)**, además de exportar los datos a Excel en el nuevo formato, los exportará automáticamente en modo simple sin mostrar el siguiente elemento de diálogo:



Desde la ventana de **output** del entorno de desarrollo de Cosmos es posible exportar automáticamente los ficheros **errores.xlsx** (Build) y **findinfiles.xlsx** (Find in Files).

1.10 Iconos

Posibilidad de utilizar iconos con transparencias (alpha channel) en Cosmos. Se ha implementado para ficheros ICO y PNG.

Es necesario definir la variable USEALPHACHANNELICONS con valor YES o TRUE.

1.11 Listas editables

Además de utilizar la tecla **[Return o Intro]** para aceptar las modificaciones realizadas en una celda, a partir de esta versión se podrán emplear también las teclas de movimiento del cursor (variable de entorno **LISTEDITACCEPTVALUEONARROWKEYS**) o simplemente retirar el foco del control List Box o de la celda editada (variable de entorno **LISTEDITACCEPTVALUEONLOSINGFOCUS**).

1.12 Modo de edición automático

A partir de esta versión existe la posibilidad de añadir nuevos registros si está marcada la propiedad **Allow Add**.

1.13 Firma digital

La dll CosSignFile (cossigfile.dll) permite firmar digitalmente archivos en formato PDF, XML y binario, así como la firma de ficheros de factura electrónica XML en formato FacturaE versiones 3.2.1 y 3.2.2.

Para más información consulte las notas técnicas de la dll [CosSignFile](#).

1.14 Vista previa en los listados

Exportación a Excel en formato **XLSX**.

Se ha implementado la posibilidad de guardar los valores del zoom utilizados en una vista previa, para lo cual se han definido la variable de entorno **REMEMBERPREVIEWZOOM** y los métodos **GetLastPreviewZoom** de las clases PrnDocument y Module, y **SetLastPreviewZoom** de la clase Module.

1.15 Mejoras

- Al definir la variable de entorno ALLOWCOPYCONTROLDATA con valor TRUE o YES se mostrará al usuario un mensaje indicando que el texto se ha copiado.
- Se podrá modificar el valor de la variable de entorno ZOOMFORM de forma dinámica utilizando el método PutEnv de la clase Module. Este valor podrá ser modificado todas las veces que se desee. Los objetos Form que ya estén cargados en memoria no se verán afectados por dicho cambio.
- Método GetEditSel de la clase SimpleControl. A partir de esta versión, cuando la variable asociada al control sea de tipo Date o Integer, se podrá obtener el primer y último elemento seleccionado. En versiones anteriores solo era posible si la variable asociada era de tipo Char.
- El método LoadFrom de la clase SqlServer no chequeará las filas que no va a cargar. Si la primera fila del fichero es una cabecera, por ejemplo, al no chequear la fila no habrá que modificar el fichero antes de cargarlo.
- Se ha implementado la posibilidad de almacenar en la base de datos el texto enriquecido de un control Edit Field.
NOTA: El límite de la longitud de registro de un tabla en CTSQL es de 32.727 bytes.
- Se ha implementado un nuevo método en Cosmos que retorna la hora UTC.
- Se ha implementado la propiedad VarLength para controles Variable en listados de la clase Page y en listados con la prnpag32.dll.

2. Nuevos métodos

2.1 Clase Char

2.1.1 LoadFromFile

Este método permite cargar el contenido de un fichero ASCII en una variable de tipo Char.

```
LoadFromFile(file as Char)
```

Parámetro

File	Ruta completa del fichero.
------	----------------------------

2.2 Clase Module

2.2.1 LoadCustomColors

Permite cargar, desde un fichero, una paleta de colores para personalizar los controles de la aplicación de Cosmos con estilo visual Cosmos 7.

```
LoadCustomColors(filePath as Char)
```

Parámetro:

filePath	Ruta del fichero con la definición de las propiedades del esquema de colores con las que se visualizarán los controles de Cosmos en las aplicaciones.
----------	---

NOTA: Para refrescar/repintar la ventana será necesario ejecutar el método Refresh.

2.2.2 GetLastPreviewZoom

Retorna el porcentaje de zoom con el que el usuario visualizó el último listado en la ventana de vista previa (métodos Preview y PreviewEx).

Independientemente del porcentaje de zoom que se haya asignado con el método **SetLastPreviewZoom**, si el usuario lo modifica en la ventana del Preview, el valor que retornará este método a partir de ese momento será el indicado por el usuario cuando se cerró la ventana.

```
GetLastPreviewZoom() return Smallint
```

Este método no necesita que esté definida la variable de entorno **REMEMBERPREVIEWZOOM**.

2.2.3 GetSystemTimeUTC

Retorna la hora UTC.

```
GetSystemTimeUTC(VAR year as Integer ,VAR month as Integer ,VAR dayOfWeek as Integer ,VAR day as Integer ,VAR hour as Integer ,VAR minute as Integer ,VAR second as Integer ,VAR milliseconds as Integer)
```

Parámetros:

year	Parámetro por referencia en el que retorna el año actual.
------	---

Month	Parámetro por referencia en el que retorna el mes actual.
dayOfWeek	Parámetro por referencia en el que retorna el día de la semana actual.
day	Parámetro por referencia en el que retorna el día actual.
hour	Parámetro por referencia en el que retorna la hora actual.
minute	Parámetro por referencia en el que retorna los minutos de la hora actual.
second	Parámetro por referencia en el que retorna los segundos de la hora actual.
milliseconds	Parámetro por referencia en el que retorna los milisegundos de la hora actual.

Por ejemplo, si el PC tiene la hora de verano de España, que es UTC+2, en el caso de que la hora local sea las 17:00:00, esta función debería retornar las 15:00:00.

2.2.4 IsRunningInCosmosWebServer

Este método permite saber si la aplicación Cosmos se está ejecutando desde Cosmos WebServer.

```
IsRunningInCosmosWebServer() return boolean
```

Retorna:

TRUE. Si la aplicación se está ejecutando desde Cosmos WebServer.

FALSE. Si la aplicación se está ejecutando desde Cosrun.

2.2.5 SetLastPreviewZoom

Asigna un porcentaje de zoom para visualizar los listados de Cosmos en la ventana de vista previa (métodos Preview o PreviewEx).

```
SetLastPreviewZoom(zoom as Smallint)
```

Parámetro:

Zoom	Porcentaje de zoom.
------	---------------------

Para que el funcionamiento de este método sea correcto la variable de entorno **REMEMBERPREVIEWZOOM** tiene que estar definida y su valor tiene que ser TRUE o YES.

NOTA: El uso conjunto de los métodos **GetLastPreviewZoom** y **SetLastPreviewZoom** y la variable de entorno **REMEMBERPREVIEWZOOM** permite, por ejemplo, almacenar el porcentaje de Preview en la configuración del usuario cuando finalice la aplicación y recuperarlo cuando ésta se vuelva a ejecutar.

2.3 Clase PrnDocument

2.3.1 GetLastPreviewZoom

Devuelve el porcentaje de zoom con el que el usuario visualizó el último listado en la ventana de vista previa (métodos Preview y PreviewEx).

```
GetLastPreviewZoom() return Smallint
```

No es necesario definir la variable de entorno **REMEMBERPREVIEWZOOM**.

2.4 Clase SimpleControl

2.4.1 GetDragColumn

En el mecanismo de drag&drop este método permite obtener la columna del control List Box desde la que se arrastró (*drag*) el elemento.

```
GetDragColumn() return Smallint
```

NOTA: Esta información se puede obtener igualmente con el método GetCurrentItem.

2.4.2 GetDropColumn

En el mecanismo drag&drop este método permite obtener la columna del control List Box sobre la que se soltó (*drop*) el elemento.

```
GetDropColumn() return Smallint
```

2.4.3 Refresh

Permite repintar el control de la *clase* Form.

```
Refresh()
```

Por ejemplo, después de llamar al método **LoadCustomColors** o modificar el valor de la variable COSMOSVISUALMODE será necesario ejecutar este método (Frame.Refresh).

2.4.4 SetListAllRowsHeight

Permite definir la altura en píxeles de las filas de un control List Box. La altura será la misma para todas las filas.

```
SetListAllRowsHeight(height as Smallint)
```

Parámetro:

Height	Número que indica la altura en píxeles de la fila.
--------	--

3. Nuevas propiedades y eventos

3.1 Propiedad Auto Fit

En los controles Bitmap la imagen se adapta al tamaño del control sin deformarse.

Si el Form tiene marcada la propiedad Resizing, el control Bitmap la propiedad **Auto Fit** y el usuario modifica el tamaño de la ventana en ejecución, la imagen no se deformará, sino que se adaptará al tamaño del control.

También se ha implementado en la clase Page y en las plantillas (ficheros *.pag) que utilizan la dll prnpag32.

3.2 Propiedad VarLength

Se ha implementado la propiedad VarLength para controles Variable en listados de la clase Page y en listados con la prnpag32.dll

a) En clase Page.

Al ejecutar el método GetProperty de la clase Control sobre un control de tipo "variable" de Page, retorna la longitud con la que se ha definido el control.

Retorna el valor sobre una variable de tipo Smallint.

b) En prnpag32.dll

Al ejecutar la función getPropInt de la prnpag32.dll pasándole como parámetro el identificador de un control "variable" de un fichero PAG, retorna la longitud con la que se ha definido el control.

3.3 Evento ChangeSize

Evento de la clase Form que se lanza cuando se modifica el tamaño de la ventana de un Form en ejecución.

4. Variables de entorno

CWSIGNOREMAIN

Si un programa define una función *main*, ésta se ejecuta siempre antes de la llamada a la función que se invoca desde el servicio REST de Cosmos.

Se ha implementado la variable de entorno CWSIGNOREMAIN para controlar si se debe ignorar o se debe ejecutar la función *main* antes de la llamada a la función.

Si no se define la variable, o se define con valor FALSE o NO, se ejecutará la función *main*.

Si se define con valor TRUE o YES no se ejecutará la función *main*.

Esta variable de entorno se puede definir en la sección Environment del fichero de configuración del proyecto o en el Cosmos.ini.

DESTROYICONBEFORESETICON

Si está definida con valor TRUE o YES, al utilizar el método SetIcon de la clase Form se liberan los recursos del icono existente.

Esta variable se puede definir en el fichero de configuración del proyecto, en el de Cosmos o con el método PutEnv de la clase Module. Su valor se podrá modificar, durante la ejecución de la aplicación, utilizando el método PutEnv.

EDITFIELDNOMARKONGETFOCUS

Si está definida con valor TRUE o YES permite que Cosmos recuerde, en los controles Drop Edit y Edit Field, la posición en la quedó el foco al salir del campo para que, al volver a ese campo, su posición sea la misma pero sin tener el texto seleccionado.

Esta variable se puede definir en el fichero de configuración del proyecto, en el de Cosmos o con el método PutEnv de la clase Module. Su valor se podrá modificar, durante la ejecución de la aplicación, utilizando el método PutEnv.

LISTEDITACCEPTVALUEONLOSINGFOCUS

En versiones anteriores a la 7.2, al editar una celda (dentro de listas editables), si no se pulsaba [Return o Intro] y el control perdía el foco, se lanzaba el evento ListCancelEdit y se perdían los cambios realizados.

A partir de la versión 7.2, la variable de entorno LISTEDITACCEPTVALUEONLOSINGFOCUS permite evitar esta pérdida de información al lanzar el evento ListAcceptEdit cuando la celda editada pierde el foco. Para ello, el valor de la variable deberá ser TRUE o YES.

Esta variable puede ser definida en la sección [Environment] del fichero de configuración del proyecto, en el de Cosmos y con el método PutEnv de la clase Module.

El valor se podrá modificar durante la ejecución de la aplicación utilizando el método PutEnv.

Los valores posible son: TRUE o YES y FALSE o NO.

NOTA: La tecla [Esc] siempre lanza el evento ListCancelEdit, con independencia del valor asignado a la variable.

Ver la variable **LISTEDITACCEPTVALUEONARROWKEYS** a continuación.

LISTEDITACCEPTVALUEONARROWKEYS

En versiones anteriores a la 7.2, al editar una celda (dentro de listas editables), si no se pulsaba [Return o Intro] y el control perdía el foco, se lanzaba el evento ListCancelEdit y se perdían los cambios realizados.

A partir de la versión 7.2, la variable de entorno LISTEDITACCEPTVALUEONARROWKEYS permite evitar esta pérdida de información al lanzar el evento ListAcceptEdit cuando el usuario pulsa cualquiera de las teclas de movimiento del cursor para salir de la celda. Para ello, el valor de la variable deberá ser TRUE o YES.

Esta variable se puede definir en el fichero de configuración del proyecto, en el de Cosmos o con el método PutEnv de la clase Module. Su valor se podrá modificar durante la ejecución de la aplicación utilizando el método PutEnv.

Ver la variable **LISTEDITACCEPTVALUEONLOSINGFOCUS** comentada anteriormente.

LOOKUPOUTERJOIN

Esta variable de entorno permite que en lugar de generar un Cursor por cada Lookup de un FormTable, Cosmos genere una sola frase Sql con Outer Joins a las tablas del Lookup.

Esta variable se puede definir en el fichero de configuración del proyecto, en el de Cosmos, con el método PutEnv de la clase Module o el método SetOption de la clases Module, Form o FormTable. Su valor se podrá modificar durante la ejecución de la aplicación utilizando el método PutEnv.

Esta variable solo es funcional en los controles Grid.

NOSENDCLICKONGRIDCLICKHEADER

En la versión 7.0 de Cosmos se implementó el evento On Click en el área cliente de los controles de tipo Grid. Esto provocaba que al lanzar el evento On Click no fuese posible determinar si se había pulsado en la cabecera o en el área cliente del control.

A partir de la versión 7.2, al definir esta variable con valor TRUE o YES al pulsar en la cabecera de un Grid solo se lanzará el evento On ClickHeader.

Esta variable se puede definir en el fichero de configuración del proyecto, en el de Cosmos o con el método PutEnv de la clase Module. Su valor se podrá modificar durante la ejecución de la aplicación utilizando el método PutEnv.

Sus valores posibles son: TRUE o YES y FALSE o NO.

REMEMBERPREVIEWZOOM

Esta variable de entorno permite que la vista previa de los listados (métodos Preview y PreviewEx) se muestre con el mismo porcentaje que la última previsualización efectuada. Para ello, la variable deberá estar definida con el valor TRUE o YES.



Si la variable está definida, Cosmos no hará caso del parámetro zoom del método PreviewEx de la clase PrnDocument.

Esta variable se puede definir en el fichero de configuración del proyecto, en el de Cosmos o con el método PutEnv de la clase Module. Su valor se podrá modificar durante la ejecución de la aplicación utilizando el método PutEnv.

Sus valores posibles son: TRUE o YES y FALSE o NO.

NOTA: Para obtener el valor del porcentaje de zoom se utiliza el método GetLastPreviewZoom de la clase PrnDocument.

USEALPHACHANNELICONS

Esta variable de entorno permite utilizar iconos con transparencias (alpha channel) en Cosmos. Para ficheros ICO y PNG.

Es necesario definir la variable USEALPHACHANNELICONS con valor YES o TRUE.

Esta variable de entorno se puede definir en la sección Environment del fichero de configuración del proyecto o en el Cosmos.ini.

5. Correcciones

5.1 Runtime

- En los controles List Box, cuando la lista no era de tipo Sql el runtime permitía hacer scroll vertical hacia abajo aunque no hubiera datos para mostrar.
- En los controles Grid, cuando una de sus columnas era un control Check y aquella se estrechaba, a veces no se podía marcar/desmarcar el control con el ratón.
- Método Invoke de la clase ActiveX. Si los parámetros del método del ActiveX eran pasados por referencia y el tipo no era de la clase Variant, no retornaba ningún valor.
- Si en un listado se indicaba la constante %TOTALPAGES% y el listado iba directamente a impresora, Cosmos no sustituía la constante por su valor y en el listado aparecía el texto %TOTALPAGES%.
- La clase Json no permitía asignar un decimal a una propiedad con el método Set.
- Método ExportToExcel. En la versión 7.0 de Cosmos no se exportaban los estilos visuales de las listas, aunque se hubiera indicado que sí con el método AllowExportListCellStyles.
- Método AllowColumnHeaderFilter. Se producía un error en la ejecución del programa y la aplicación dejaba de funcionar si el tamaño del dato superaba los 255 caracteres.
- Método ShowListAsMultiColumnGroup. La llamada a este método antes de la que la lista estuviese cargada provocaba un error de protección general.
- Si se definía la variable de entorno SHOWGRIDTOOLTIP y en las líneas del Grid existía un control Button, al posicionarse encima del botón se mostraba un tooltip con caracteres ilegibles.
- Exportación a Excel. No funcionaba correctamente la exportación de los campos tipo fecha. Se exportaban siempre con formato DD/MM/YYYY y no respetaba la máscara.
- Cuando está activo el modo automático de edición, al pulsar la tecla [Flecha abajo] y se lanza el comando **AcceptEdit** de la clase Form. En la versión 7.0 de Cosmos, además, se lanzaba siempre el evento On ChangeRow, tanto si se ejecutaba el método AcceptEdit de la clase FormTable como si no. Esto provocaba que no se pudiera cancelar la edición de la tabla si el método AcceptEdit no llegaba a ejecutarse. A partir de la versión 7.2 se lanzará el evento on ChangeRow solo si se ejecuta el método **AcceptEdit**.
- En el modo automático de edición, si un Grid no tenía marcada la propiedad Allow Add, se podía dar de alta un registro al pulsar con el ratón en la primera fila del Grid después de realizar una consulta en la que no se habían obtenido resultados. Esto sucedía porque se lanzaba el evento de tabla changing row, situándose el foco en el primer registro del Grid y se editaba la tabla.
- Optimización de consultas contra Sql Server. En una conexión contra Sql Server, después de la primera definición de un **cursor for update** se realizaban bloqueos de lectura y escritura que no eran necesarios.
- Si la sentencia generada automáticamente desde un mantenimiento de Cosmos (FormTable) era mayor de 8.192 bytes, en conexiones contra Sql Server éste devolvía el siguiente mensaje: "Sql statement too long".
- Sql Server mediante ODBC. No detectaba que un Cursor era **for update** si en la query aparecía FOR UPDATE en mayúsculas.
- Método SetSuggestionBox. La lista de sugerencias que mostraba este método en algunas ocasiones no se pintaba en la ventana que contenía el control de edición que lo invocaba.
- El método SetIcon no liberaba recursos. Cuando una pantalla se abría varias veces, llegaba un momento en que los iconos no se mostraban correctamente (ver variable DESTROYICONBEFORESETICON).
- Control Check. Si un control tenía marcada la propiedad ReadOnly, no era posible hacer clic sobre él, pero sí se podía cambiar su valor con la barra espaciadora.

- Control Drop Edit. Si el control tenía marcada la propiedad **ReadOnly**, se permitía seleccionar un valor de la lista, como en versiones anteriores, pero no modificaba el valor del campo de edición.

5.2 Entorno de desarrollo

- Al generar el mantenimiento de tablas con muchos campos utilizando el Wizard del entorno de desarrollo se producía un error de protección general.
- Clase Struct. Al borrar todos los elementos de una estructura e intentar crear uno nuevo de tipo Char, al pulsar el botón se producía un error de protección general y se abortaba el proceso.

5.3 Cosmos WebServer

- La impresión de listados que utilizan la dll prnpag32 no se realizaba correctamente en algunas ocasiones.
- En algunas ocasiones el string retornado por una petición REST de Cosmos WebServer se truncaba. El carácter que provocaba este comportamiento es el numeral ordinal masculino (carácter º, ASCII 167). El mismo comportamiento era provocado por el numeral ordinal femenino (carácter ¢, ASCII 166).
- Al ejecutar muchas veces un servicio REST de Cosmos (más de 10.000), las variables globales perdían su valor.
- Los métodos GetEnv y GetIni no retornaban el valor de las variables definidas en el fichero de configuración cuando se ejecutaban desde un servicio REST.
- Cosmos WebServer no reseteaba el código de retorno de estado de la petición HTTP después de cada ejecución. Si no se asignaba ningún código de retorno desde el programa Cosmos, Cosmos WebServer debía retornar el código 200. El problema radicaba en que si se asignaba un código de error en la ejecución de un servicio y en la siguiente ejecución no se asignaba el código de retorno, en lugar de retornar 200 (su valor por defecto), retornaba el código asignado en la anterior ejecución.

5.4 CTSQL

- Al establecer varias conexiones a una misma base de datos, a veces se producía el siguiente mensaje: "Imposible abrir el catálogo del sistema (systables)". Este error solo se producía en el motor de Windows.

5.5 CosmosData

- No se guardaban las modificaciones realizadas en la definición de los campos calculados.
- En la presentación del listado no se formateaban correctamente los campos de tipo Smallint e Integer.
- Al ejecutar el listado y generar una lista no se formateaba correctamente la celda donde se mostraba el resultado de la función de agregado. No mostraba los decimales.