



MultiBase Cosmos

Notas a la versión 7.4

BASE100

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

Índice

1. IMPLEMENTACIONES Y MEJORAS	3
1.1 COSMOS WEBSERVER.....	3
1.1.1 <i>Multihilo</i>	3
1.1.2 <i>Protocolo HTTPS</i>	3
1.1.3 <i>Cabeceras</i>	3
1.2 IMPRESIÓN DE LISTADOS EN FORMATO PDF	3
1.3 MULTI-IDOMA.....	3
1.4 CONEXIONES ODBC CONTRA MICROSOFT SQL SERVER	3
1.5 LLAMADA A FUNCIONES JAVA	4
1.6 PROPIEDAD AUTOFIT	4
1.7 CONTROLES	4
1.8 EXPORTACIÓN A EXCEL.....	4
1.9 VISTA PRELIMINAR DE IMPRESIÓN	4
1.10 EDITOR VISUAL.....	4
1.11 CSQL	5
1.12 MWLISTEN.....	5
2. NUEVOS MÉTODOS	6
2.1 CLASE JSON.....	6
2.1.1 <i>SetNull</i>	6
2.2 CLASE MODULE	6
2.2.1 <i>GetMultilanguageTranslation</i>	6
2.2.2 <i>IsLicenseRegistered</i>	6
2.2.3 <i>SetMultilanguageFile</i>	6
2.3 CLASE SIMPLECONTROL.....	7
2.3.1 <i>GetListSelectionType</i>	7
2.3.2 <i>SetListColumnEditButton</i>	7
2.3.3 <i>SetListColumnText</i>	7
2.3.4 <i>SetListSelectionType</i>	7
2.4 CLASE PRNDOCUMENT.....	8
2.4.1 <i>HideExportPDFButton</i>	8
2.4.2 <i>OpenPrinterPDF</i>	8
2.5 API PRNPAG32.DLL.....	8
2.5.1 <i>OpenPrinterPDF</i>	8
3. EVENTOS	9
4. VARIABLES DE ENTORNO	10
4.1 COSMOS WEBSERVER.....	11
4.1.1 <i>Multihilo</i>	11
4.1.2 <i>Configuración protocolo HTTPS</i>	11
5. CORRECCIONES	13
5.1 RUNTIME.....	13

1. Implementaciones y mejoras

Esta versión genera incompatibilidades con versiones anteriores.

1.1 Cosmos WebServer

1.1.1 Multihilo

Se ha implementado la posibilidad de que Cosmos WebServer sea multihilo.

Para ello, se ha incorporado una nueva variable de entorno, denominada [NUMBEROFTHREADS](#), que es la encargada de activar o desactivar esta funcionalidad.

1.1.2 Protocolo HTTPS

Se ha implementado la posibilidad de que Cosmos WebServer pueda configurarse con protocolo HTTPS. Ver más adelante en este mismo documento.

1.1.3 Cabeceras

Se ha implementado la posibilidad de que un servicio Cosmos Webserver pueda obtener el valor de las cabeceras de una petición.

En el fichero de configuración de los servicios se ha añadido un nuevo parámetro: **headerParameter**.

1.2 Impresión de listados en formato PDF

Se ha implementado la posibilidad de imprimir listados en formato PDF. Esto se podrá realizar por cualquiera de los siguientes procedimientos:

- Mediante el nuevo método OpenPrinterPDF de la clase PrnDocument y de la dll PrnPag32.dll.
- Utilizando el botón “PDF” de la ventana de vista preliminar de impresión.
- Mediante el método ExportDocumentToFile.

1.3 Multi-idioma

Se ha implementado la posibilidad de poder cambiar en tiempo de ejecución el fichero de multi-idioma, así como de poder cargar un fichero de multi-idioma y no modificar automáticamente los literales de los formularios con los textos traducidos.

También se ha añadido la posibilidad de consultar la traducción de un literal dentro de un fichero multi-idioma.

1.4 Conexiones ODBC contra Microsoft Sql Server

Se ha implementado la posibilidad de utilizar la variable LOOKUPOUTERJOIN en conexiones ODBC contra Microsoft Sql Server.

Para ello, habrá que emplear la variable LOOKUPOUTERJOIN conjuntamente con la nueva variable de entorno [ODBCSQLSERVER](#), que será la encargada de indicar a Cosmos que la conexión ODBC es contra MS Sql Server.

1.5 Llamada a funciones Java

Se ha implementado la posibilidad de indicar a Cosmos qué versión de máquina virtual debe utilizar para ejecutar las funciones Java. Para ello se han creado las variables de entorno: [COSMOSUSELASTJAVAVERSION](#) y [COSMOSUSEJAVAVERSION](#).

La versión de la máquina virtual de Java debe ser la 1.8 o inferior.

1.6 Propiedad AutoFit

Se ha cambiado el identificador interno de esta propiedad para que cuando se edite el componente gráfico (template) de la clase Page, o un fichero PAG, la propiedad no se marque automáticamente.

Debido a esta modificación será necesario editar los ficheros PAG y volver a marcar la propiedad.

Esta modificación provoca también que los programas Cosmos deban ser recompilados.

1.7 Controles

Edit Field

Se ha implementado la posibilidad de poder asignar un valor a un control Edit Field de tipo Date/Time picker.

Listas

Se ha implementado la posibilidad de modificar el valor de una celda, así como también la de seleccionar dinámicamente el tipo de selección que se desea realizar en una lista (control List Box).

En las listas editables se ha implementado la posibilidad de añadir un botón a los campos de tipo Edit String y Edit Numeric. Al hacer clic en ese botón se lanzará el evento ListEditButtonClick.

Drop Edit y Drop List

Se ha implementado la posibilidad de añadir dinámicamente iconos a estos controles. Para ello se utilizará la propiedad Icon del control.

1.8 Exportación a Excel

A partir de esta versión, cuando al método ExportToExcel no se le indique como parámetro la ruta del fichero Excel, ofrecerá por defecto posibilidad de guardarlo en formato.xlsx.

1.9 Vista preliminar de impresión

En la ventana de vista preliminar de impresión de Cosmos se ha añadido un nuevo botón para poder imprimir los informes a PDF y un nuevo campo de edición para indicar el número de la pagina del listado que se desea visualizar.

El botón "PDF" se podrá ocultar utilizando el método HideExportPDFButton.

1.10 Editor Visual

La lista de proyectos abiertos recientemente mostrará su ruta completa y no solo el nombre como en versiones anteriores.

Paleta de iconos. A partir de esta versión se podrá redimensionar la ventana, posibilitando así la visualización completa del fichero de iconos.

1.11 CSQL

Se ha implementado la instrucción `attachconnection` para permitir la conexión a una base de datos utilizando el identificador definido en la sección `connections`, evitando de esta forma tener que seleccionarla desde el menú “`database/select connection`”.

```
attachconnection <conexion>
```

1.12 Mwlisten

Se ha ampliado la información que se muestra en el fichero de log. A partir de esta versión se registra la fecha y hora de la entrada del registro en el log.

Además, mientras está esperando a recibir una nueva conexión muestra el mensaje “esperando conexión...”

```
Fri Oct 04 10:28:59 2019 Esperando conexión ...  
Fri Oct 04 10:49:33 2019 Conexión establecida con el cliente 127.0.0.1 en la fecha 'Fri Oct 04  
10:49:33 2019'  
Fri Oct 04 10:49:33 2019 ctsql localizado en c:\MultiWay\lib\ctsql.  
Fri Oct 04 10:49:33 2019 Servidor listo. Puerto local 28000.  
Fri Oct 04 10:49:33 2019 Esperando conexión ...
```

2. Nuevos métodos

2.1 Clase Json

2.1.1 SetNull

Se ha implementado el método SetNull de la clase JSON que permite asignar un valor nulo a una propiedad o a un JSON.

```
SetNull (name as Char) return Boolean
```

Parámetro:

name	Nombre del propiedad.
------	-----------------------

2.2 Clase Module

2.2.1 GetMultilanguageTranslation

Permite obtener la etiqueta traducida del fichero multi-idioma a partir de la etiqueta sin traducir.

```
GetMultilanguageTranslation(text as Char) return Char
```

Parámetros:

Text	Texto definido en el fichero de multi-idioma (MULTILANGUAGEFILE).
------	---

Retorna:

El texto correspondiente a la traducción.

Si no se ha definido un fichero de multi-idioma retornará NULL.

Si no se encuentra traducido el texto pasado como parámetro, retornará la propia cadena.

2.2.2 IsLicenseRegistered

Permite saber si la licencia de Cosmos está registrada en el PC donde se está ejecutando.

```
IsLicenseRegistered() return Boolean
```

2.2.3 SetMultilanguageFile

Permite cambiar de fichero multi-idioma en ejecución.

```
SetMultilanguageFile (path as Char) return Boolean
```

Parámetros:

path	Ruta absoluta o relativa del fichero multi-idioma (MULTILANGUAGEFILE).
------	--

Retorna:

Si encuentra el fichero y éste se carga correctamente en memoria, retorna TRUE. A partir de ese momento los textos traducidos serán los incluidos en el nuevo fichero.

Si no encuentra el fichero o no puede cargarlo, retornará FALSE. Los textos traducidos serán los del anterior fichero multi-idioma en caso de que ya hubiera uno definido.

2.3 Clase SimpleControl

2.3.1 GetListSelectionType

Retorna un smallint con el tipo de selección de la lista: (0 - NONE, 1 - SINGLE, 2 - MULTISELECT).

```
GetListSelectionType() return Smallint
```

2.3.2 SetListColumnEditButton

Este método permite asignar un botón a todas las celdas de una columna de un control lista en edición (listas tipo String o Numeric con valores 1 y 5 del parámetro editStyle en el método SetListColumnEditType de la clase SimpleControl).

```
SetListColumnEditButton (col as Smallint ,hasButton as Boolean ,buttonStr as Char default null) return Boolean
```

Parámetros:

col	Número de columna a la que se desea añadir/eliminar el botón.
hasButton	Booleano que indica si se desea añadir/eliminar el botón a la columna indicada en el primer parámetro.
buttonStr	Cadena de caracteres que aparecerá como “tooltip” al pasar el ratón por encima del botón del campo de edición. Si se indica el valor “null” no mostrará ningún tooltip.

Al pulsar el botón asociado al campo de edición, el runtime lanzará el evento ListEditButtonClick.

2.3.3 SetListColumnText

Este método permite modificar el valor de una celda en un control List Box (string y árbol).

```
SetListColumnText(row as Integer ,col as Integer ,text as Char)
```

Parámetros:

Row	Identificador de la fila a la que se desea modificar el valor.
col	Número de columna a la que se desea modificar el valor.
text	Texto que se va asignar a la celda.

2.3.4 SetListSelectionType

Este método permite cambiar el tipo de selección de la lista.

```
SetListSelectionType(selectionType as Smallint)
```

Parámetros:

SelectionType	Número que indica el tipo de selección. Los valores posibles son: : 0 - NONE, 1 - SINGLE, 2 - MULTISELECT)
---------------	---

2.4 Clase PrnDocument

2.4.1 HideExportPDFButton

Permite ocultar el botón “PDF” en la ventana de vista preliminar de impresión.

```
HideExportPDFButton (hideButton as Boolean default FALSE)
```

Parámetros:

HideButton	Indica si se desea ocultar o no el botón que aparece en la ventana de vista preliminar de impresión. Los valores posibles son: TRUE y FALSE. El valor por defecto es FALSE (no se oculta el botón).
------------	---

2.4.2 OpenPrinterPDF

Permite imprimir los informes de Cosmos en formato PDF.

```
OpenPrinterPDF (fileName as Char)
```

Parámetros:

fileName	Ruta donde se va a generar el fichero.
----------	--

2.5 Api PrnPag32.dll

2.5.1 OpenPrinterPDF

Permite imprimir los informes de Cosmos en formato PDF.

```
openPrinterPDF (hPrinter as integer, fileName as Char)
```

Parámetros:

hPrinter	Manejador de la impresora.
fileName	Ruta donde se va a generar el fichero.

3. Eventos

ListEditButtonClick

Este evento es lanzado por Cosmos cuando el usuario hace clic sobre el botón de un campo de edición de un control lista al que se le ha indicado que muestre un botón con el método SetListColumnEditButton.

```
On event ListEditButtonClick control lstStrings(e as event)
objects begin
  str as char
end
begin
  str = "Fila [" + e.GetItem() + "] columna [" + e.GetControl().GetCurrentItem() + "];
  str.Trace;
end
```

4. Variables de entorno

COSDUMPDEBUGMODE

Si se define esta variable de entorno, el runtime de Cosmos generará un fichero de log en el directorio c:\tmp cuando se produzca un error de memoria.

Se define en el Environment global del fichero de configuración de Cosmos.

COSMOSUSELASTJAVAVERSION

Esta variable de entorno indica al runtime de Cosmos si los métodos Java se ejecutarán con la versión más reciente de Java instalada en el PC.

Esta variable admite los valores TRUE/YES o FALSE/NO.

Si se define a TRUE/YES, Cosmos ejecutará los métodos Java utilizando la versión más reciente de la máquina virtual instalada en el PC.

Si se define a NO o no se define, ejecutará los métodos Java con la máquina virtual que se encuentre definida en el PATH del usuario o del sistema.

Esta variable de entorno se debe definir en la sección [Environment] del INI del proyecto o del Cosmos.ini

COSMOSUSEJAVAVERSION

Esta variable de entorno indica al runtime de Cosmos la versión de Java con la que se ejecutarán desde Cosmos los métodos Java.

Si la versión de la máquina virtual indicada no se encuentra instalada en la máquina, mostrará un error indicando que no se ha podido arrancar la máquina virtual.

Esta variable de entorno se debe definir en la sección [Environment] del INI del proyecto o del Cosmos.ini.

Por ejemplo: COSMOSUSEJAVAVERSION=1.7.0_55

MULTILANGUAGENOCHANGELABELS

Se ha implementado la posibilidad de que una aplicación Cosmos cargue un fichero de multi-idioma, pero que no modifique automáticamente las etiquetas de los controles de pantalla.

Si esta variable está definida a YES o a TRUE, y a su vez está definida la variable MULTILANGUAGEFILE, Cosmos cargará el fichero multi-idioma, pero no modificará las etiquetas de los controles de pantalla.

Esta variable se podrá definir en la sección Environment del fichero INI del proyecto o del Cosmos.ini.

ODBCSQLSERVER

Se ha implementado la posibilidad de que la variable de entorno LOOKUPOUTERJOIN se pueda utilizar en conexiones ODBC contra MS Sql Server.

La variable de entorno LOOKUPOUTERJOIN indica al runtime de Cosmos que en los FormTable en controles Grid obtenga los valores de lookup a otras tablas con una sola frase SQL utilizando “outer joins”, en lugar de utilizar una frase SQL por cada lookup. Esto permite que la obtención de datos de lookup resulte óptima, aumentando la velocidad en la carga de datos en el grid.

Al ser distinta la sintaxis de frases SQL con outer joins en MultiBase/CTSOL y en MS Sql Server, es necesario indicar al runtime qué sintaxis debe utilizar al generar la frase.

Si la variable ODBCSQLSERVER está definida, o está definida a YES o TRUE, el runtime generará la frase SQL con sintaxis compatible con MS Sql Server.

Si la variable de entorno ODBCSQLSERVER no está definida o está definida con un valor distinto a YES o TRUE, en el caso de que la conexión sea ODBC, el runtime no generará la sentencia Sql con la cláusula OUTER JOIN.

Esta variable se podrá definir en la sección Environment de la conexión ODBC dentro del fichero INI del proyecto o del COSMOS.INI.

CENTERFORMINPARENT

Permite indicar a Cosmos que las ventanas hijas de un Form que tengan como coordenadas (0,0) salgan centradas respecto al padre, en lugar de que la esquina superior izquierda de la ventana hija se sitúe en el centro de la ventana padre.

Esta variable se deberá definir en el fichero de configuración del proyecto, sección [Environment], con valor TRUE.

4.1 Cosmos WebServer

4.1.1 Multihilo

NUMBEROFTHREADS

Se ha implementado la posibilidad de que la ejecución de Cosmos WebServer sea multihilo, es decir, que permita procesar más de una petición REST a la vez, sin tener que esperar a que finalice una para procesar la siguiente.

Para ello, será necesario definir la variable de entorno NUMBEROFTHREADS en la sección [Server] del fichero INI del servicio y que Cosmos WebServer sea iniciado como servicio.

Cosmos WebServer será monohilo si es iniciado en modo aplicación.

El valor de esta variable será el número de hilos que deberá ser capaz de manejar el servidor.

Si la variable no está definida, Cosmos Webserver será monohilo.

Si está definida con valor 1, Cosmos Webserver será monohilo.

Cuando Cosmos Webserver esté definido como multihilo, la conexión a la base de datos solo se podrá hacer en arquitectura cliente-servidor.

*No se podrán establecer conexiones locales a la base de datos. En conexiones locales retornará el mensaje:
"Imposible conectar con el servidor de SQL. (Unable to connect to local database from Cosmos WebServer in MULTI-THREAD MODE. The connection must be client-server)".*

4.1.2 Configuración protocolo HTTPS

Si se desea configurar que las conexiones en CWS sean por HTTPS, en el fichero de configuración del servicio se deben definir las siguientes variables:

SSL_PORT

Indica el número de puerto donde estará escuchando el servidor.

Esta variable es obligatoria.

SSL_KEYSTORE

Indica la ruta del fichero de almacén de certificados.

Se deberá utilizar el carácter "/" para separar nombres de directorios.

Esta variable es obligatoria.

SSL_KEYSTORE_PWD

Indica la contraseña del almacén de certificados.

Esta variable es obligatoria.

SSL_KEYSTORE_ALIAS

Indica el alias del certificado que se desea utilizar. Si no se indica, utilizará el primer certificado del almacén.

Esta variable es opcional.

SSL_KEYSTORE_ALIAS_PWD

Indica la contraseña del alias indicado en SSL_KEYSTORE_ALIAS. Si no se especifica, utilizará la contraseña indicada en SSL_KEYSTORE_PWD.

Esta variable es opcional.

Notas:

- Si no se definen las variables PORT ni SSL_PORT, la conexión será HTTP (no segura), por el puerto 8081.
- Si se define la variable PORT y no se define la variable SSL_PORT, la conexión será HTTP (no segura), por el puerto indicado en la variable PORT.
- Si no se define la variable PORT y se define la variable SSL_PORT, la conexión será HTTPS (conexión segura), pero no se permitirá hacer conexión HTTP (no segura).
- Si se definen las variables PORT y SSL_PORT, las conexiones serán HTTP (conexión no segura) y HTTPS (conexión segura), por los puertos indicados respectivamente. Los números de puerto indicados en estas dos variables deberán ser diferentes.

5. Correcciones

5.1 Runtime

- Error de ejecución en cosrun cuando se asignaba un valor a un parámetro de función de tipo datetime que no había sido definido por referencia (cláusula VAR), tenía default NULL y el parámetro no se ha pasado en la llamada a la función.
- Si la variable de entorno LOOKUPOUTERJOIN estaba activa, en algunos casos daba error de “columna ambigua”.
- El método SqlDescribe no retornaba el nombre completo de un campo cuando la conexión se establecía contra una base de datos de SqlServer y la longitud del campo de una tabla superaba los 20 dígitos. Se ha ampliado el *buffer* a 200 bytes.
- Método GetListInfo. Si en una lista se añadía información a nivel de fila con el método SetListInfo y se aplicaba un filtro, se perdía el valor info y retornaba 0.
- En el método ExportToExcel, cuando recibía como parámetro un null, al mostrar el cuadro de diálogo para indicar la ruta y el nombre del fichero no aparecía en la lista la extensión XLSX.
- Método ShowListFilterBar. En algunas ocasiones no filtraba correctamente las listas SQL cargadas con el contenido de una VIEW.
- Cuando en una lista estaba marcada la propiedad TWO PANELS, al hacer clic en la cabecera y lanzar el evento On ClickHeader, si se ejecutaba el método GetItem devolvía un valor incorrecto (mayor que un smallint).
- Se ha corregido el error de ejecución que se producía en cosrun al ejecutar un proyecto con debugger estando situados dentro de una función con una lista de argumentos y teniendo activado el check “Watch Locals”.
- Si una de las columnas de un GRID era un botón, al exportar a Excel, en la columna de hoja Excel se mostraban caracteres extraños. Además, si la exportación se hacía a CSV desplazada las etiquetas.
- No se exportaba a Excel el signo del euro (€).
- Método ExportToExcel.
 - Cuando recibía un null como valor del parámetro y se abría el cuadro de diálogo para indicar el nombre del fichero, no aparecía la extensión “xlsx” que debería mostrarse desde la versión 7.2 de Cosmos.
 - En una lista Sql, si existía un campo que tuviese una fecha con un valor nulo, si la exportación se hacía a fichero XLS daba un error de formato y no se podía abrir el fichero generado. Si la exportación era a fichero XLSX, el fichero se generaba y se podía abrir, pero el valor del campo era 00/01/1900 en lugar de null.
 - En un control Grid, si se exportaban datos de tipo fecha con valores nulos a fichero Excel en formato xls, éste no se generaba correctamente.
 - Si una de las columnas de un Grid era un botón, al exportar a formato csv las columnas no se correspondían con los datos.
 - Si se exportaba en formato xls el contenido de un Grid el Font de la cabecera no era el correcto.
 - Al exportar a formato xlsx una lista de agrupados rellenaba con 0 las celdas de las cabeceras.
- Propiedad AutoFit. Al editar el componente gráfico de una clase Page, si éste tiene un control bitmap ya no se marcará por defecto la propiedad AutoFit.
- Listas. Filtro en la cabecera. Si el valor por el que se intentaba filtrar no existía, se producía un error de memoria y se abortaba la ejecución.

- Los métodos TotalizeColumn y ComputeListColumnTotals no funcionaban correctamente cuando en una lista de tipo árbol se filtraban los datos y estos métodos eran invocados dentro del código del evento ListAcceptEditColumnFilter.
- LISTAS SQL. El método TotalizeColumn no realizaba correctamente la suma de columnas con valores decimales en una lista de tipo Sql si las cantidades superaban la unidad de millar y a la columna se le había asignado la máscara «#,##0» para que no mostrara los decimales.
- LISTAS SQL. Si la lista se cargaba con una query con cláusula limit, ésta se cargaba correctamente, pero al pinchar en la cabecera para ordenarla, en lugar de mostrar sólo los registros de la instrucción select del método Loadselect, mostraba todos los registros de la tabla, obviando la cláusula limit. Esta versión respeta la cláusula limit, pero al pulsar en la cabecera para ordenar la lista Sql los datos que mostrará no serán los mismos, ya que se vuelve a ejecutar la sentencia.
- Método Execute de la clase SqlStatement. Se producía un error de memoria al ejecutar una instrucción insert desde este método cuando la tabla tenía definida una columna de longitud mayor a 4.096 bytes.
- Método AlternateBackColor. En los controles Grid no se dibujaba correctamente la línea cuando el control hijo era de tipo Drop Edit o Drop List.
- Conexión local. Al abrir una base de datos con el método Connect de la clase SqlServer, si ésta ya estaba abierta previamente con otro SqlServer en una conexión local, retornaba el error -1107. Este error solo se produce en las siguientes versiones y releases: 7.2, 7.2.1 y 7.2.2.
- La exportación a Excel en formato xlsx de una lista agrupada (método ShowListAsMultiColumnGroup) estaba rellenando a 0 las celdas de las cabeceras. En la exportación a xls, estas celdas salen en blanco.
- El método ExportDocumentToFile no exportaba a XLSX.
- El método AlternateBackColor no funcionaba correctamente si el control Grid contenía controles Drop Edit y/o Drop List.