



MultiBase Cosmos

Notas a la versión 7.6

BASE100

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

Índice

1. IMPLEMENTACIONES	3
1.1 APIS	3
1.2 RUNTIME.....	3
1.2.1 <i>Aspecto Visual. Cosmos 7</i>	3
1.2.2 <i>Controles de pantalla. Definir fuentes</i>	5
1.2.3 <i>Control List Box</i>	6
1.2.4 <i>Control Grid. Eventos</i>	6
1.2.5 <i>Control Edit Field</i>	6
1.2.6 <i>Visor de informes</i>	6
1.2.7 <i>Método SetSuggestionBox</i>	7
1.3 EDITOR VISUAL.....	7
1.3.1 <i>Editor de código</i>	7
1.3.2 <i>Editor del proyecto</i>	7
1.3.3 <i>Paleta de iconos</i>	7
2. NUEVOS MÉTODOS	8
2.1 CLASE MODULE	8
2.1.1 <i>LoadCustomControls</i>	8
2.1.2 <i>LoadCustomFonts</i>	8
2.2 CLASE PRNDOCUMENT.....	8
2.2.1 <i>SetPreviewBackgroundColor</i>	8
2.3 CLASE SIMPLECONTROL.....	8
2.3.1 <i>IgnoreCustomFonts</i>	8
2.3.2 <i>IgnoreCustomColors</i>	9
2.3.3 <i>SetEditDateTimePickerIcon</i>	9
2.3.4 <i>SetSuggestionBoxFont</i>	9
3. VARIABLES DE ENTORNO	10
4. CORRECCIONES	12
4.1 RUNTIME.....	12

1. Implementaciones

1.1 APIS

La DLL [COSSMTPDLL](#) permite el envío de correos electrónicos desde Cosmos.

Los correos electrónicos se podrán enviar tanto en formato texto como HTML, pudiendo adjuntar archivos en ambos formatos y, en el caso del HTML, incluir también imágenes en el cuerpo del mensaje.

1.2 Runtime

1.2.1 Aspecto Visual. Cosmos 7

En esta versión se han añadido nuevas propiedades en la sección [Custom Colors](#) y se ha implementado una nueva sección, denominada [Custom Controls](#).

Sección [Custom Colors](#)

En esta sección se podrán definir las siguientes características:

- El texto y el color de fondo de los controles Panel y Bar de manera independiente.
Para ello se han implementado las propiedades: `Color_Text_Panel`, `Color_Back_Panel` y `Color_Back_Bar`.
NOTA: Si no se define alguna de estas variables se tomará por defecto la definida para el control Box.
- El color del borde de los controles tipo Box, Radio, Check, Panel y Bar.
Las nuevas propiedades son: `Color_Border_Box`, `Color_Border_Radio`, `Color_Border_Check`, `Color_Border_Panel`, `Color_Border_Bary` `Color_Border_Text`.
- El color de fondo y del texto de un control de edición cuando se selecciona texto.
Las nuevas propiedades son: `Color_Selected_Text_Edit`, `Color_Selected_Back_Edit`, `Color_Selected_Text_DropEdit`, `Color_Selected_Back_DropEdit`, `Color_Selected_Text_DropList` y `Color_Selected_Back_DropList`.
- El color del borde y el color de texto de un control de edición.
Las nuevas propiedades son: `Color_Border_Edit`, `Color_Border_DropEdit`, `Color_Border_DropList`, `Color_Text_DropEdit`, `Color_Text_DropEdit`, `Color_Back_DropList`, `Color_Text_DropList`, `Color_Text_Edit` y `Color_Back_Edit`.
- Transparencia en el color de fondo de los controles.
Si a la propiedad "background" se le asigna la palabra clave "TRANSPARENT" en lugar de indicar un valor RGB, los controles tendrán un fondo transparente, es decir, se mostrarán sobre el color que tenga definido el control padre.

Para forzar a Cosmos a que el aspecto visual de las pantallas de la aplicación sea el indicado en el fichero de configuración y no el indicado en el fuente, se han implementado las variables de entorno [VISUALMODEFORZEFORCOLOR](#) y [VISUALMODEFORZEBACKCOLOR](#). Si no se quisiera mostrar algún control con este aspecto tendríamos que aplicarle el nuevo método [IgnoreCustomColors](#) de la clase SimpleControl.

Sección [Custom Controls](#)

Esta nueva sección permite modificar en ejecución algunas características de los controles de los *Form* de Cosmos. Estas características son: estética de las etiquetas, bordes del control, forma de las esquinas (redondeadas o cuadradas) y su relieve (normal, bajo y sobre relieve).

El aspecto del control se modificará cuando se defina alguna propiedad de las que se detallan a continuación, y afectará a todos los controles de la aplicación.

Esta sección se podrá definir en el fichero de configuración de Cosmos, del proyecto o en un fichero externo que se cargará cuando se ejecute el nuevo método [LoadCustomControls](#) de la clase Module. Además, será necesario tener definida la variable COSMOSVISUALMODE con valor 7 y no utilizar el método [IgnoreCustomColors](#) para el control.

A qué tipo de controles afecta.

EDIT, DROPLIST, DROPEdit, BOX, RADIO, CHECK, TEXT.

Qué características se podrán modificar.

- **BORDERS.** Indica qué bordes del control se desean dibujar. Si no se define esta variable se dibujarán los cuatro bordes. Sus posibles valores son: TOP, BOTTOM, LEFT y RIGHT.

Esta propiedad se podrá definir para los controles de tipo EDIT, DROPEdit, DROPLIST, BOX, RADIO BUTTON, CHECK y TEXT.

Ejemplo: EDIT="BORDERS:BOTTOM,LEFT"

- **CORNERRADIUS.** Indica el radio en píxeles de la circunferencia del borde de los controles. Si no se define esta variable los bordes se dibujarán cuadrados. Su valor es un número entero.

Sólo se aplica si el frame del control tiene activa la propiedad ETCHED.

Esta propiedad se podrá definir para los controles de tipo BOX, CHECK y RADIO BUTTON.

Ejemplo: BOX="CORNERRADIUS:15"

- **DRAWHEADER.** Al definir esta propiedad, el área donde se muestra la etiqueta tendrá un color de fondo distinto al resto del control. Este color será el indicado en la propiedad BORDER de la sección [Custom Colors](#) del control. El color por defecto es el RGB(160,160,160).

Esta propiedad se podrá definir para los controles de tipo BOX y RADIO BUTTON.

Ejemplo: BOX="DRAWHEADER"

- **LABELFORECOLOR.** Indica el color con el que se dibujará el texto de la etiqueta del control.

Esta propiedad podrá definirse para los controles de tipo BOX, CHECK y RADIO BUTTON.

Si no se define, la etiqueta se dibujará con el color indicado en la propiedad *Foreground* del *font* del control.

Valores posibles. Definición del color en formato RGB usando uno de estos métodos:

- a) La expresión Rgb(rojo, verde, azul).
- b) Con el valor en hexadecimal que representa el color, usando una expresión en el siguiente formato: #RRGGBB, siendo RR, GG y BB el valor hexadecimal correspondiente a rojo, verde y azul respectivamente.

Ejemplo:

BOX="LABELFORECOLOR:Rgb(128,255,255)"

RADIO:"LABELFORECOLOR:#80FFFF"

- **LABELALIGNMENT.** Indica la alineación del texto de la etiqueta del control. Si no se define esta variable, la alineación de la label será a la izquierda. Sus valores posibles son: LEFT, CENTER y RIGHT.

Esta propiedad podrá definirse para los controles de tipo BOX y RADIO BUTTON.

Ejemplo: BOX="LABELALIGNMENT:RIGHT"

- **BORDERSTYLE.** Indica el tipo de borde del control que se dibujará para los controles del tipo indicado. Si no se define esta variable, el tipo de borde del control con el que se dibujará el control será el indicado en tiempo de diseño. Sus valores posibles son: UP, DOWN y ETCHED.

Esta propiedad puede definirse para los controles de tipo BOX, RADIO BUTTON, CHECK y TEXT.

Ejemplo: BOX="FRAME:ETCHED"

Propiedades por control.

EDIT	BORDERS.
DROPEDIT	BORDERS.
DROPLIST	BORDERS.
BOX	BORDERS, CORNERRADIUS, LABELALIGNMENT, DRAWHEADER, LABELFORECOLOR y BORDERSTYLE.
RADIO	BORDERS, CORNERRADIUS, LABELALIGNMENT, DRAWHEADER, LABELFORECOLOR y BORDERSTYLE.
CHECK	BORDERS, CORNERRADIUS y BORDERSTYLE.
TEXT	BORDERS, CORNERRADIUS y BORDERSTYLE.

1.2.2 Controles de pantalla. Definir fuentes

A partir de esta versión se podrá definir la fuente de todos los controles en el fichero de configuración de Cosmos o del proyecto. A cada tipo de control se le podrá asignar una fuente.

¿Cómo funciona?

Para activar esta opción es necesario definir la variable de entorno [COSMOSUSECUSTOMFONTS](#). Los valores de las fuentes se definirán en una nueva sección, denominada [Custom Fonts](#). La fuente asignada a cada tipo de control se podrá cambiar si se carga otro fichero con el método [LoadCustomFonts](#). La llamada a este método tendrá que hacerse antes de que se cree el objeto Form.

Para respetar la fuente original del control se utilizará el nuevo método [IgnoreCustomFonts](#) de la clase SimpleControl. Cuando se necesite utilizar una fuente distinta en un control determinado se podrá emplear el método SetProperty de la clase Control.

Solo se modifican las propiedades de las fuentes que se indiquen en el fichero.

Sección [Custom Fonts](#)

En esta sección se definirán los valores de las fuentes de los controles. Para ello, se emplearán las siguientes variables de entorno:

- FONT_BUTTON. Permite definir las fuentes de los botones.
- FONT_TEXT. Permite definir la fuente de los controles TEXT.
- FONT_RADIO_BUTTON_TEXT. Permite definir la fuente de los controles RADIO_BUTTON.
- FONT_CHECK_TEXT. Permite definir la fuente de los controles CHECK.
- FONT_BOX. Permite definir la fuente de los controles BOX.

- FONT_TAB. Permite definir la fuente de los controles TABCONTROL.
- FONT_PANEL. Permite definir la fuente de los controles PANEL.
- FONT_BUTTONGROUP. Permite definir la fuente de los controles BUTTONGROUP.
- FONT_BOXGROUP. Permite definir la fuente de los controles BOXGROUP.
- FONT_GRID. Permite definir la fuente de los controles GRID.
- FONT_EDIT. Permite definir la fuente de los controles EditFields.
- FONT_DROPLIST. Permite definir la fuente de los controles DROPLIST.
- FONT_DROPEDIT. Permite definir la fuente de los controles DROPEDIT.
- FONT_MENU. Permite definir la fuente de los controles MENU.
- FONT_LISTBOX. Permite definir la fuente de los controles LISTBOX.

1.2.3 Control List Box

Filtros

Las modificaciones que se realicen en una lista filtrada con el método AllowColumnHeaderFilter quedarán reflejadas en la misma tras eliminar los filtros.

NOTA: Esta modificación no se aplica en las listas de tipo Sql.

Alternancia de colores en las listas

Solo se mostrará la alternancia de colores en las listas cuando sus filas tengan datos. Para ello se han implementado dos mecanismos:

- Método AlternateBackColor. El valor del RGB que recibe el método es negativo.
- Definir la variable de entorno [ALTERNATEBACKCOLORIGNOREEMPTY](#).

1.2.4 Control Grid. Eventos

Se ha implementado una nueva variable de entorno, [NOSENDCLICKONGRIDCLICKBODY](#), para evitar que se lance el evento onClick al hacer clic en el cuerpo del control Grid.

1.2.5 Control Edit Field

Se ha implementado el método [SetEditDateTimePickerIcon](#) de la clase SimpleControl que permite asignar un icono a los controles Edit Field de tipo DateTimePicker.

1.2.6 Visor de informes

A partir de esta versión se podrá cambiar el color de fondo de la ventana y visualizar más de una página del listado en el visor de informes de Cosmos. Se visualizarán páginas completas. El número de páginas mostradas dependerá del tamaño de éstas. Cuanto más pequeñas, más páginas.

¿Cómo funciona?

Para poder visualizar más de una página en el visor de listados de Cosmos habrá que marcar la casilla de verificación “Multipágina”, situada en la parte inferior izquierda de la ventana.

Al activar el modo Multipágina:

- Se habilitarán/mostrarán los siguientes controles:
 - Casilla de verificación “Mostrar nº de Página”. Al activar esta casilla se mostrará el número de página del listado en la parte superior izquierda de cada hoja.
 - Botones zoom (+/-). Aumenta o disminuye el tamaño de las páginas.

- Control Spin en el margen inferior izquierdo. Permite desplazarse por las distintas páginas del listado.
 - *Primero*. Si tomamos la ventana del visor del listado como una página, al pulsar este botón retrocederá una página (no se desplaza hasta la primera página).
Si en un listado de 24 páginas se están visualizando las 12 últimas, al pulsar este botón se mostrarán las 12 primeras.
 - *Previo y Siguiente*. Si dividimos la ventana en celdas y en cada celda se muestra una página del listado (fila, columna), con estos botones se permite el desplazamiento por las filas de las celdas.
Si en la ventana del visor se muestran las 12 primeras páginas del listado, 6 en cada fila, al pulsar el botón Siguiente la primera página que se muestra en la primera celda será la número 7, y en la primera celda de la segunda fila será la nº 13.
 - *Último*. Si tomamos la ventana del visor del listado como una página, al pulsar este botón se visualizará la página siguiente (no se desplaza hasta la última página).
Si en un listado de 24 páginas se muestran las 12 primeras, al pulsar este botón se mostrarán las 12 últimas.
- La combinación “[Ctrl]+rueda del ratón”, además de hacer zoom, permite aumentar o disminuir el número de páginas que se muestran en el visor.

NOTA: Sólo se mostrarán páginas completas.

1.2.7 Método SetSuggestionBox

Se podrá definir la fuente a utilizar en la lista que muestra este método. Para ello se ha implementado el método [SetSuggestionBoxFont](#).

1.3 Editor Visual

1.3.1 Editor de código

Posibilidad de cambiar el color de fondo y personalizar los colores de las palabras reservadas.

A partir de esta versión se mostrará el número cada línea de código en el margen izquierdo.

1.3.2 Editor del proyecto

Posibilidad de abrir el directorio donde está ubicado el fuente del proyecto. Para ello se ha añadido una opción de menú “Open Containing Folder”. Esta opción se encuentra en el menú contextual del proyecto.

1.3.3 Paleta de iconos

Se ha implementado la posibilidad de buscar un icono por su nombre.

2. Nuevos Métodos

2.1 Clase Module

2.1.1 LoadCustomControls

Permite cargar la configuración definida en la sección [Custom Controls](#) del fichero que recibe como parámetro.

Sintaxis:

```
LoadCustomControls(filePath as Char)
```

Parámetro:

FilePath	Ruta completa del fichero.
----------	----------------------------

2.1.2 LoadCustomFonts

Permite cargar la configuración definida en la sección [Custom Fonts](#) del fichero que recibe como parámetro.

Sintaxis:

```
LoadCustomFonts(filePath as Char)
```

Parámetro:

FilePath	Ruta completa del fichero.
----------	----------------------------

2.2 Clase PrnDocument

2.2.1 SetPreviewBackgroundColor

Permite cambiar al color de fondo de la ventana que se muestra en el visor de listados.

Sintaxis:

```
SetPreviewBackgroundColor (color as integer)
```

Parámetro:

Color	Entero correspondiente al color en formato RGB.
-------	---

2.3 Clase SimpleControl

2.3.1 IgnoreCustomFonts

Permite indicar que un control determinado ignorará las propiedades definidas en la sección [Custom Fonts](#) para los controles de su tipo.

Sintaxis:

```
IgnoreCustomFonts ()
```

2.3.2 IgnoreCustomColors

Permite indicar que un control determinado ignorará las propiedades definidas en las secciones [Custom Colors](#) y [Custom Controls](#) para los controles de su tipo.

Sintaxis:

```
IgnoreCustomColors(ignore as Boolean)
```

Parámetro:

Ignore	Booleano que indica si se ignoran o no las propiedades. Este valor se podrá modificar en la aplicación tantas veces como sea necesario.
--------	---

2.3.3 SetEditDateTimePickerIcon

Permite asignar un icono a los controles Edit Field de tipo datetimepicker.

Sintaxis:

```
SetEditDateTimePickerIcon(iconFile as char, icon as Smallint default 0)
```

Parámetros:

iconFile	Identificador del fichero de iconos.
Icon	Índice del icono en el fichero de iconos.

2.3.4 SetSuggestionBoxFont

Permite definir la fuente utilizada en la lista que se muestra en el método SetSuggestionBox.

Sintaxis:

```
SetSuggestionBoxFont(Font as char) return Boolean
```

Parámetro:

Font	Fuente a utilizar.
------	--------------------

3. Variables de entorno

ALTERNATEBACKCOLORIGNOREEMPTY

Al definir esta variable con valor TRUE/YES los colores indicados en el método AlternateBackColor solo se mostrarán en las filas del control List Box que tengan datos. El color de fondo de las filas vacías será el color por defecto.

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

Esta variable se podrá definir en la sección Environment del fichero de configuración de Cosmos, del fichero de configuración del proyecto o con el método PutEnv de la Clase Module.

Su valor no podrá ser modificado en tiempo de ejecución.

COSMOSUSECUSTOMFONTS

Al definir esta variable con valor TRUE la fuente asignada a los controles gráficos será la indicada en la sección [Custom Fonts](#).

Esta variable se define en la sección Environment del fichero de configuración de Cosmos, del fichero de configuración del proyecto o con el método PutEnv de la clase Module.

El valor de esta variable no se podrá modificar en ejecución.

NOSENDCLICKONGRIDCLICKBODY

Al definir esta variable con valor TRUE/YES no se lanzará el evento OnClick al hacer clic en el cuerpo del Grid.

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

Esta variable se ha implementado para mantener compatibilidad con la versión 7.0 de Cosmos (y anteriores) en las que el evento OnClick no estaba implementado en el área cliente del control Grid.

Esta variable se define en la sección Environment del fichero configuración de Cosmos, del fichero de configuración del proyecto o con el método PutEnv de la Clase Module.

Su valor no podrá ser modificado en tiempo de ejecución.

VISUALMODEFORZEBACKCOLOR

Fuerza a que el color de fondo de los controles, aunque se modifique en ejecución, sea el definido en la sección [Custom Colors](#) del fichero de configuración.

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

Si el valor de esta variable es FALSE/NO, o si no se define, el comportamiento será igual que en versiones anteriores a la 7.6, es decir, prevalece el color que se haya indicado en el fuente y no el definido en el fichero de configuración.

Esta variable se define en la sección Environment del fichero configuración de Cosmos, del fichero de configuración del proyecto o con el método PutEnv de la Clase Module.

Su valor no podrá ser modificado en tiempo de ejecución.

VISUALMODEFORZEFORCOLOR

Fuerza a que el color del texto de los controles, aunque se modifique en ejecución, sea el definido en la sección [Custom Colors](#) del fichero de configuración.

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

Si el valor de esta variable es FALSE/NO, o si no se define, el comportamiento será igual que en versiones anteriores a la 7.6, es decir, prevalece el color que se haya indicado en el fuente y no el definido en el fichero de configuración.

Esta variable se define en la sección Environment del fichero configuración de Cosmos, del fichero de configuración del proyecto o con el método PutEnv de la Clase Module.

Su valor no podrá ser modificado en tiempo de ejecución.

4. Correcciones

4.1 Runtime

- Método ExportToExcel. No se exportaba correctamente el tipo de fuente a formato XLSX.
- Impresión a PDF. Si en el diseño de una página de impresión se añadía un control Box encima de otro, ambos con color de fondo, al imprimir a PDF el color de fondo de la caja superior no se pintaba.
- Control List Box.
 - El método ComputeListColumns no calculaba correctamente los totales después de ejecutar el método Sort.
 - Se producía un error de memoria cuando se ejecutaba el método GetColumnSum sobre una columna numérica (integer) en un control LISTBOX de tipo string.
- LOOKUPOUTERJOIN. Se producía un error al hacer clic en la cabecera de una columna de un Grid que formaba parte de un lookup si la variable estaba definida con valor TRUE.
- Control List Box. Las modificaciones hechas en una lista filtrada con el método AllowColumnHeaderFilter y editada no quedaban guardadas al eliminar el filtro.
- Control Grid. Al hacer clic en la cabecera del control no siempre se lanzaban los eventos OnClick y OnClickHeader.
- El método WriteBinaryEx no retornaba el número de bytes escritos.
- El método GETURLFILEEX no descargaba algunas imágenes.
- Control Grids. Al hacer clic en la cabecera de estos controles no siempre se lanzaba el evento OnClick ni el evento OnHeaderClick.
- Se producía un error de memoria y la aplicación dejaba de funcionar cuando se ejecutaba el método GetColumnSum sobre una columna numérica (integer) en un control LISTBOX de tipo String.
- Skin de listas. El botón del GAP no respetaba la estética definida en el skin cuando se mantenía pulsado el ratón sobre él.
- Cosmos Visual mode 7.
 - No dibujaba correctamente el borde de un botón si éste era Highlight. Lo dibujaba siempre gris.
 - No dibujaba correctamente el borde de los botones “sin borde” y “sin label”.