



# MultiBase Cosmos

Notes to versions 7.2 and 7.2.1

**BASE100**

BASE 100, S.A.  
[www.base100.com](http://www.base100.com)

## Index

<b>1. IMPLEMENTATIONS, IMPROVEMENTS AND BUG FIXES OF VERSION 7.2.1</b>	<b>4</b>
1.1 IMPLEMENTATIONS AND IMPROVEMENTS	4
1.1.1 <i>JSON Class</i>	4
1.2 BUG FIXES	4
1.2.1 <i>Runtime</i>	4
<b>2. IMPLEMENTATIONS AND IMPROVEMENTS OF VERSION 7.2</b>	<b>5</b>
2.1 ARRAY CLASS	5
2.2 CHAR CLASS	5
2.3 FORMTABLE CLASS	5
2.4 JSON CLASS	6
2.5 CONTROLS	6
2.5.1 <i>Band Group</i>	6
2.5.2 <i>Bitmap</i>	6
2.5.3 <i>Drop Edit and Edit Field</i>	6
2.5.4 <i>Grid</i>	6
2.5.5 <i>List Box</i>	6
2.5.6 <i>Tab Control</i>	7
2.6 COSMOS WEBSERVER	7
2.6.1 <i>Log File</i>	7
2.7 COSMOS 7 VISUAL STYLE	8
2.8 EVENTS	8
2.8.1 <i>Form</i>	8
2.8.2 <i>Drag &amp; Drop</i>	8
2.8.3 <i>RClickHeader</i>	8
2.9 EXPORT TO EXCEL	8
2.10 ICONS	9
2.11 EDITABLE LISTS	9
2.12 AUTOMATIC EDITING MODE	9
2.13 DIGITAL SIGNATURE	9
2.14 PREVIEW IN REPORTS	9
2.15 IMPROVEMENTS	10
<b>3. NEW METHODS</b>	<b>11</b>
3.1 CHAR CLASS	11
3.1.1 <i>LoadFromFile</i>	11
3.2 MODULE CLASS	11
3.2.1 <i>LoadCustomColors</i>	11
3.2.2 <i>GetLastPreviewZoom</i>	11
3.2.3 <i>GetSystemTimeUTC</i>	11
3.2.4 <i>IsRunningInCosmosWebServer</i>	12
3.2.5 <i>SetLastPreviewZoom</i>	12
3.3 PRNDOCUMENT CLASS	12
3.3.1 <i>GetLastPreviewZoom</i>	12
3.4 SIMPLECONTROL CLASS	13

3.4.1	<i>GetDragColumn</i> .....	13
3.4.2	<i>GetDropColumn</i> .....	13
3.4.3	<i>Refresh</i> .....	13
3.4.4	<i>SetListAllRowsHeight</i> .....	13
<b>4.</b>	<b>NEW PROPERTIES AND EVENTS</b> .....	<b>14</b>
4.1	AUTO FIT PROPERTY .....	14
4.2	VARLENGTH PROPERTY.....	14
4.3	CHANGESIZE EVENT .....	14
<b>5.</b>	<b>ENVIRONMENT VARIABLES</b> .....	<b>15</b>
<b>6.</b>	<b>BUG FIXES OF VERSION 7.2</b> .....	<b>18</b>
6.1	RUNTIME.....	18
6.2	DEVELOPMENT ENVIRONMENT.....	19
6.3	COSMOS WEBSERVER.....	19
6.4	CTSQL .....	19
6.5	COSMOSDATA.....	19

© Copyright BASE 100, S.A. All rights reserved. No part of this document may be reproduced or transmitted by any means without prior written permission of the copyright owner. All products cited in this document are registered trademarks or registered trademarks of their respective owners.

[NT\_CO\_7.2-7.2.1\_v1\_en.docx]

## 1. Implementations, improvements and bug fixes of version 7.2.1

---

### 1.1 Implementations and improvements

#### 1.1.1 JSON Class

The possibility of handling data of decimal type has been implemented.

### 1.2 Bug fixes

#### 1.2.1 Runtime

- If the Modified method of the FromTable class was used in a form, when leaving a field and entering in the next edition field, when writing a value in the last one the value was selected, being able to cause the overwriting of the text previously entered.
- The implementation made in version 7.2 of Cosmos that allowed to use icons with transparency (alpha channel) has caused that, in some cases, the PNG files are not displayed correctly. This option has been deactivated in the case of PNG files, and the operation being the same as in versions prior to 7.2.

## 2. Implementations and improvements of version 7.2

### 2.1 Array class

The maximum number of elements of an Array has been increased, which can now be 2,147,483,647 elements.

The data type of the parameters in the Resize method of the Array class and the Size method of the Container class changes from Smallint to Integer.



***This modification means that programs compiled with previous versions can not be executed with this version. It is necessary to recompile the applications.***

### 2.2 Char class

The limit of 32,767 bytes has been eliminated in objects of the Char class, now being set to 2,147,483,647 bytes. To do this, some of the functions of the Char class have been modified in order to be able to handle character strings with a length greater than 32,767 bytes.

Operators modified from this version:

- Operator +
- Operator =
- Operator [ ]

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

Methods modified from this version:

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

NOTE: The definition of the initial size of a Char object will remain a maximum of 32,767 bytes. The size of the Char object can be modified at run time with the assignment and concatenation operators.



***This change in the parameter data type or return of the functions supposes that the binaries are incompatible.***

***In addition, you should check the type of return variables, function's parameters and methods redefinitions (see below methods).***

### 2.3 FormTable Class

The queries have been optimized for the case of the tables of a FormTable associated with a Grid control. For this, the environment variable LOOKUPOUTERJOIN has been implemented.

If the environment variable is defined, instead of creating a cursor for each Lookup, a single phrase is created with Outer Joins to the Lookup tables.

## 2.4 Json Class

The algorithm of the SaveToFile method and that of the Converter to Char has been optimized.

## 2.5 Controls

### 2.5.1 Band Group

The possibility of changing the size of the child controls of the bands of a running Band Group control before adding the band has been implemented.

NOTE: If you enlarge the size of some child control of any of the bands (Box control), you should also enlarge the band size if the width of the band is smaller, since it does not expand automatically.

### 2.5.2 Bitmap

The **Auto Fit** property has been implemented in the Bitmap controls so that the image associated with the control is not deformed when the size of the running control is changed.

This property has been implemented for controls of the Form class and controls for the Page class.

### 2.5.3 Drop Edit and Edit Field

It has been implemented the possibility that Cosmos *remembers* the position of the cursor in the editing controls so that, when accessing the control again, the cursor is placed in the same position (**EDITFIELDNOMARKONGETFOCUS**). When returning the focus to the control the text of the field will not be selected.

### 2.5.4 Grid

From this version it is possible to copy the text of the Drop List and Check type controls (ALLOWCOPYCONTROLDATA). In the case of Check type controls, the values that will be copied will be 0 or 1 (0 if the check is not checked and 1 if it is checked).

It has been implemented the possibility that the On Click event, in the Grid controls, will not be launched when clicking on the header of the Grid. For this, the environment variable NOSENDCLICKONGRIDCLICKHEADER must be defined.

The AlternateBackColor method of the SimpleControl class can be invoked from the Grid type controls, in addition to the controls of type List Box.

We have optimized the queries that are made from Cosmos when a Grid control has a table associated with it and some of the fields in the Formtable are of type Lookup (LOOKUPOUTERJOIN).

### 2.5.5 List Box

It will be possible to define the height in pixels of the rows of a list. The defined size will be applied to all the rows in that list. For this, the SetListAllRowsHeight method of the SimpleControl class must be executed.

It has been implemented the possibility of moving in the list controls from left to right with the mouse wheel. To do this, hold down the **[Ctrl] + mouse wheel**.

If a cell in a list has a style assigned, its background color will be the one defined in the style, regardless of whether the row it belongs to has the focus.

From this version it is allowed to totalize by columns of type DATE and DATETIME.

### 2.5.6 Tab Control

The **RClickHeader** event has been implemented.

NOTE. If the event is being used as a generic event, it must be taken into account that, from this version, the code will also be executed when the right mouse button is clicked on the Tab Control tab.

## 2.6 Cosmos WebServer

The environment variable **CWSIGNOREMAIN** has been implemented to control whether or not to execute the main function of the module that implements the REST service that is being executed.

The **IsRunningInCosmosWebServer** method of the Module class has been implemented to check if the Cosmos application is running from Cosmos WebServer.

By default, whenever an error occurs and no HTTP code has been activated with the SetExecStatus method of the Module class, Cosmos will trigger HTTP error 500.

### 2.6.1 Log File

The output of the errors generated by the Cosmos WebServer runtime during the processes can be dumped into a file if a file with the name **CWSLog.log** is created in the temporary directory.

#### Cosmos WebServer as service

If an error occurs during the execution of a Cosmos service and the ErrorLevel is greater than 0, the information about the error will be stored in the log file, as well as the method and module where the error occurred, and the call stack of the application.

If an error occurs during execution and the ErrorLevel equals 0, the error information will not be written to the log file.

If the Trace method is executed in the source code of the Cosmos service, the text associated with that method will be written to the log file.

#### Cosmos WebServer as application

If an error occurs during the execution of a Cosmos service and the ErrorLevel is greater than 0, information about the error will be stored in the log file, as well as the method and module where the error occurred, and the call stack from the application. This same information will be displayed on the screen through a MessageBox.

If an error occurs during execution and the ErrorLevel equals 0, the error information will not be written to the log file and will not be displayed on the screen.

If the Trace method is executed in the source code of the Cosmos service, a MessageBox with the text associated with the Trace method will be displayed on the screen and the text will not be written to the log file.

## 2.7 Cosmos 7 Visual Style

From this version you can change the visual style of a Cosmos application at runtime. The changes that can be made are the following:

- Activate or deactivate the Cosmos 7 style. The value of the COSMOSVISUALMODE variable can be modified with the PutEnv method of the Module class.
- Changing the color scheme of the application when the Cosmos 7 style is active. The **LoadCustomColors** method of the Module class has been implemented.

It will not be necessary to define the **[Custom Colors]** section in the configuration files used by Cosmos as it was in version 7.0. In this way you can have several color schemes for the same Cosmos application.

For either of the two changes to be effective, the new Refresh method of the SimpleControl class must be called (this method has been implemented in this version).

## 2.8 Events

### 2.8.1 Form

Its has been implemented The **ChangeSize** event of the Form class, which allows to know when the size of a window has been changed at run time.

### 2.8.2 Drag & Drop

The **GetDragColumn** and **GetDropColumn** methods have been implemented, which allow to know, respectively, in the List Box controls, the columns from which has been dragged (drag) and an the column where has been dropped (drop).

### 2.8.3 RClickHeader

The **RClickHeader** event has been implemented for controls of type Tab. This event will be launched when the user clicks with the right mouse button on a tab of the Tab control.

## 2.9 Export to Excel

In addition to the XML 2003 spreadsheet format that Cosmos supports when exporting to Excel, from this version it is possible to export to Excel in **XLSX** format.

To be generated with this new format, the name of the files that are created from any of the methods or utilities of Cosmos must have the extension **XLSX**.

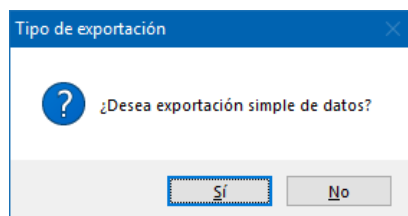
In the cases of the methods that receive the name of the file as a parameter they should be modified. These methods are the following:

- **UnloadTo** method of the SqlServer class.
- **ExportToExcel** method of the SimpleControl class.
- **SelectWindow** method of the SqlServer class.
- **SelectWindowAt** of the SqlServer class.

If the **Save As** option is available when the file is generated, it will be necessary to **choose Excel File Type (\*.xlsx)**. These cases are:



- Sql-Interactive.
- Preview of the listings (**Preview** method). If the **Save As** option is selected as **Excel File type (\*.xlsx)**, in addition to exporting the data to Excel in the new format, it will automatically export them in simple mode without displaying the following dialog element:



From the **output** window of the Cosmos development environment it is possible to automatically export the **errores.xlsx** (Build) and **findinfiles.xlsx** (Find in Files) files.

## 2.10 Icons

Possibility of using icons with transparencies (alpha channel) in Cosmos. It has been implemented for ICO and PNG files.

It is necessary to define the variable USEALPHACHANNELICONS with value YES or TRUE.

## 2.11 Editable Lists

In addition to using the **[Return or Enter]** key to accept the modifications made in a cell, from this version you can also use the cursor movement keys (environment variable **LISTEDITACCEPTVALUEONARROWKEYS**) or simply remove the focus from the List Box control or from the edited cell (environment variable **LISTEDITACCEPTVALUEONLOSINGFOCUS**).

## 2.12 Automatic editing mode

From this version it is possible to add new records if the **Allow Add** property is checked.

## 2.13 Digital signature

The CosSignFile dll (cossigfile.dll) allows to digitally sign files in PDF, XML and binary format, as well as the signature of XML electronic invoice files in FacturaE format versions 3.2.1 and 3.2.2.

For more information, see the technical notes of the [CosSignFile](#) dll.

## 2.14 Preview in reports

Export to Excel in **XLSX** format.

The ability to save the zoom values used in a preview has been implemented, for which the environment variable **REMEMBERPREVIEWZOOM** and the methods **GetLastPreviewZoom** of the classes PrnDocument and Module, and **SetLastPreviewZoom** of the Module class have been defined.

## 2.15 Improvements

- When defining the environment variable ALLOWCOPYCONTROLDATA with value TRUE or YES, the user will be shown a message indicating that the text has been copied.
- You can modify the value of the ZOOMFORM environment variable dynamically using the PutEnv method of the Module class. This value can be modified as many times as desired. Form objects that are already loaded into memory will not be affected by that change.
- GetEditSel method of the SimpleControl class. From this version, when the variable associated with the control is of type Date or Integer, the first and last selected element can be obtained. In previous versions, it was only possible if the associated variable was of Char type.
- The LoadFrom method of the SqlServer class will not check the rows it will not load. If the first row of the file is a header, for example, not checking the row will not have to modify the file before loading it.
- The possibility of storing in the database the rich text of an Edit Field control has been implemented.  
NOTE: The limit of the record length of a table in CTSQL is 32,727 bytes.
- A new method has been implemented in Cosmos that returns the UTC time.
- The VarLength property has been implemented for Variable controls in listings of the Page class and in listings with the prnpag32.dll.

## 3. New methods

---

### 3.1 Char class

#### 3.1.1 LoadFromFile

This method allows loading the contents of an ASCII file into a variable of type Char.

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

Parameter:

File	Absolute path of the file.
------	----------------------------

### 3.2 Module class

#### 3.2.1 LoadCustomColors

It allows to load, from a file, a palette of colors to personalize the controls of the application of Cosmos with visual style Cosmos 7.

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

Parameter:

filePath	Path of the file with the definition of the properties of the color scheme with which Cosmos controls will be displayed in the applications.
----------	--

NOTE: To refresh/repaint the window it will be necessary to execute the Refresh method.

#### 3.2.2 GetLastPreviewZoom

Returns the zoom percentage with which the user viewed the last report in the preview window (Preview and PreviewEx methods).

Regardless of the percentage of zoom that has been assigned with the **SetLastPreviewZoom** method, if the user modifies it in the Preview window, the value that will return this method from that moment will be the one indicated by the user when the window was closed.

```
GetLastPreviewZoom() return Smallint
```

This method does not require that the environment variable **REMEMBERPREVIEWZOOM** be defined.

#### 3.2.3 GetSystemTimeUTC

Returns the UTC time.

```
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)
```

Parameters:

year	Parameter by reference in which the current year is returned.
------	---

Month	Parameter by reference in which the current month is returned.
dayOfWeek	Parameter by reference in which the current day of the week is returned.
day	Parameter by reference in which the current day is returned.
hour	Parameter by reference in which the current hour is returned.
minute	Parameter by reference in which the minutes of the current time is returned.
second	Parameter by reference in which the seconds of the current time is returned.
milliseconds	Parameter by reference in which the milliseconds of the current time is returned.

For example, if the PC has the summer time in Spain, which is UTC + 2, in the case that the local time is 17:00:00, this function should return at 15:00:00.

### 3.2.4 IsRunningInCosmosWebServer

This method lets you know if the Cosmos application is running from Cosmos WebServer.

```
IsRunningInCosmosWebServer() return boolean
```

Returns:

TRUE. If the application is running from Cosmos WebServer.

FALSE. If the application is running from Cosrun.

### 3.2.5 SetLastPreviewZoom

Assigns a zoom percentage to the Cosmos reports in the preview window (Preview or PreviewEx methods).

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

Parameter:

Zoom                      Zoom percentage.

For the operation of this method to be correct, the environment variable **REMEMBERPREVIEWZOOM** must be defined and its value must be TRUE or YES.

NOTE: The joint use of the methods **GetLastPreviewZoom** and **SetLastPreviewZoom** and the environment variable **REMEMBERPREVIEWZOOM** allows, for example, to store the percentage of Preview in the user's configuration when the application finishes and recover it when it is executed again.

## 3.3 PrnDocument class

### 3.3.1 GetLastPreviewZoom

Returns the zoom percentage with which the user viewed the last report in the preview window (Preview and PreviewEx methods).

```
GetLastPreviewZoom() return Smallint
```

It is not necessary to define the environment variable **REMEMBERPREVIEWZOOM**.

### 3.4 SimpleControl class

#### 3.4.1 GetDragColumn

In the drag & drop mechanism, this method allows to obtain the column of the List Box control from which the element was dragged.

```
GetDragColumn() return Smallint
```

NOTE: This information can also be obtained with the GetCurrentItem method.

#### 3.4.2 GetDropColumn

In the drag & drop mechanism, this method allows obtaining the column of the List Box control over which the element was dropped.

```
GetDropColumn() return Smallint
```

#### 3.4.3 Refresh

It allows to repaint the Form control.

```
Refresh()
```

For example, after calling the **LoadCustomColors** method or modifying the value of the **COSMOSVISUALMODE** variable it will be necessary to execute this method (Frame.Refresh).

#### 3.4.4 SetListAllRowsHeight

It allows to define the height in pixels of the rows of a List Box control. The height will be the same for all rows.

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

Parameter:

Height	Number that indicates the height of the row in pixels.
--------	--

## 4. New properties and events

---

### 4.1 Auto Fit property

In the Bitmap controls, the image fits to the size of the control without deforming.

If the Form has the Resizing property checked, the Bitmap control the **Auto Fit** property, and the user modifies the size of the running window, the image will not be deformed, but will be adapted to the size of the control.

It has also been implemented in the Page class and in the templates (\* .pag files) that use the prnpag32 dll.

### 4.2 VarLength property

The VarLength property has been implemented for Variable controls in reports of the Page class and in reports with the prnpag32.dll dll.

a) In Page class.

When you execute the GetProperty method of the Control class on a control of type "variable" of Page, it returns the length with which the control has been defined.

Returns the value on a variable of type Smallint.

b) In prnpag32.dll

When executing the getPropInt function of the prnpag32.dll, passing it as a parameter the identifier of a "variable" control of a PAG file, it returns the length with which the control has been defined.

### 4.3 ChangeSize event

Event of the Form class that is thrown when the size of the window of a Form in execution is modified.

## 5. Environment variables

---

### **CWSIGNOREMAIN**

If a program defines a *main* function, it is always executed before the call to the function that is invoked from the Cosmos REST service.

The environment variable CWSIGNOREMAIN has been implemented to control whether the *main* function must be ignored or executed before the call to the function.

If the variable is not defined, or defined with a value of FALSE or NO, the *main* function will be executed.

If it is defined with value TRUE or YES, the *main* function will not be executed.

This environment variable can be defined in the Environment section of the project configuration file or in the Cosmos.ini.

### **DESTROYICONBEFORESETICON**

If it is defined with value TRUE or YES, using the SetIcon method of the Form class frees the resources of the existing icon.

This variable can be defined in the project configuration file, in the Cosmos file or with the PutEnv method of the Module class. Its value can be modified, during the execution of the application, using the PutEnv method.

### **EDITFIELDNOMARKONGETFOCUS**

If it is defined with value TRUE or YES allows Cosmos to remember, in the controls Drop Edit and Edit Field, the position in the focus remained when leaving the field so that, when returning to that field, its position is the same but without having the selected text.

This variable can be defined in the project configuration file, in the Cosmos.ini file or with the PutEnv method of the Module class. Its value can be modified, during the execution of the application, using the PutEnv method.

### **LISTEDITACCEPTVALUEONLOSINGFOCUS**

In versions prior to 7.2, when editing a cell (within editable lists), if you did not press [Return or Enter] and the control lost focus, the ListCancelEdit event was launched and the changes made were lost.

As of version 7.2, the environment variable LISTEDITACCEPTVALUEONLOSINGFOCUS allows you to avoid this loss of information by throwing the ListAcceptEdit event when the edited cell loses focus. For this, the value of the variable must be TRUE or YES.

This variable can be defined in the [Environment] section of the project configuration file, in the Cosmos file and with the PutEnv method of the Module class.

The value can be modified during the execution of the application using the PutEnv method.

Possible values are: TRUE or YES and FALSE or NO.

NOTE: The [Esc] key always throws the ListCancelEdit event, regardless of the value assigned to the variable.

See the variable **LISTEDITACCEPTVALUEONARROWKEYS** below.

### 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

This environment variable allows that instead of generating a Cursor for each Lookup of a FormTable, Cosmos generates a single phrase Sql with Outer Joins to the Lookup tables.

This variable can be defined in the project configuration file, in the Cosmos file, with the PutEnv method of the Module class or the SetOption method of the Module, Form or FormTable classes. Its value can be modified during the execution of the application using the PutEnv method.

***This variable is only functional in the Grid controls.***

### NOSENDCLICKONGRIDCLICKHEADER

In version 7.0 of Cosmos, the On Click event was implemented in the client area of the Grid type controls. This caused that when launching the On Click event it was not possible to determine if it had been pressed in the header or in the client area of the control.

From version 7.2, when defining this variable with value TRUE or YES, clicking on the header of a Grid will only launch the On ClickHeader event.

This variable can be defined in the project configuration file, in the Cosmos file or with the PutEnv method of the Module class. Its value can be modified during the execution of the application using the PutEnv method.

Its possible values are: TRUE or YES and FALSE or NO.

### REMEMBERPREVIEWZOOM

This environment variable allows the preview of the reports (Preview and PreviewEx methods) to be displayed with the same percentage as the last preview made. For this, the variable must be defined with the value TRUE or YES.



***If the variable is defined, Cosmos will ignore the zoom parameter of the PreviewEx method of the PrnDocument class.***

This variable can be defined in the project configuration file, in the Cosmos file or with the PutEnv method of the Module class. Its value can be modified during the execution of the application using the PutEnv method.

Its possible values are: TRUE or YES and FALSE or NO.



NOTE: To obtain the zoom percentage value, the GetLastPreviewZoom method of the PrnDocument class is used.

#### **USEALPHACHANNELICONS**

This environment variable allows you to use icons with transparencies (alpha channel) in Cosmos. For ICO and PNG files.

It is necessary to define the variable USEALPHACHANNELICONS with value YES or TRUE.

This environment variable can be defined in the Environment section of the project configuration file or in the Cosmos.ini.

## 6. Bug fixes of version 7.2

---

### 6.1 Runtime

- In the List Box controls, when the list was not of the Sql type, the runtime allowed vertical scrolling down even though there was no data to show.
- In the Grid controls, when one of its columns was a Check control and the control narrowed, sometimes the control could not be marked / unchecked with the mouse.
- Invoke method of the ActiveX class. If the parameters of the ActiveX method were passed by reference and the type was not of the Variant class, it did not return any value.
- If in a report the %TOTALPAGES% constant was indicated and the list went directly to the printer, Cosmos did not substitute the constant for its value and the text %TOTALPAGES% appeared in the report.
- The Json class did not allow assigning a decimal to a property with the Set method.
- ExportToExcel method. In version 7.0 of Cosmos, the visual styles of the list controls were not exported, although they would have been indicated with the AllowExportListCellStyles method.
- AllowColumnHeaderFilter method. There was an error in the execution of the program and the application stopped working if the size of the data exceeded 255 characters.
- ShowListAsMultiColumnGroup method. Calling this method before the list was loaded caused a general protection error.
- If the environment variable SHOWGRIDTOOLTIP was defined and in the Grid lines there was a Button control, when positioned above the button a tooltip with illegible characters was shown.
- Export to Excel. The export of the date type fields did not work correctly. They were always exported in DD / MM / YYYY format and did not respect the mask.
- When the automatic edit mode is active, pressing the [Down arrow] key launches the **AcceptEdit** command of the Form class. In version 7.0 of Cosmos, in addition, the On ChangeRow event was always thrown, whether the AcceptEdit method of the FormTable class was executed or not. This caused that the edition of the table could not be canceled if the AcceptEdit method was not executed. As of version 7.2, the event on ChangeRow will be launched only if the **AcceptEdit** method is executed.
- In automatic editing mode, if a Grid did not have the Allow Add property checked, a record could be registered when the mouse clicked on the first row of the Grid after making a query in which no results had been obtained. This happened because the changing row table event was launched, the focus was placed on the first Grid record and the table was edited.
- Optimization of queries with Sql Server. In a connection with Sql Server, after the first definition of a **for update cursor**, read and write locks were made that were not necessary.
- If the Sql statement generated automatically from a Cosmos maintenance (FormTable) was greater than 8,192 bytes, in connections with Sql Server it returned the following message: "Sql statement too long".
- Sql Server using ODBC. It did not detect that a Cursor was for update if in the query it appeared FOR UPDATE in uppercase.
- SetSuggestionBox method. The list of suggestions that this method showed was sometimes not shown in the window that contained the editing control that invoked it.
- The SetIcon method did not release resources. When a Form was opened several times, there came a time when the icons were not displayed correctly (see variable DESTROYICONBEFORESETICON).
- Control Check. If a control had the ReadOnly property checked, it was not possible to click on it, but you could change its value with the space bar.

- Control Drop Edit. If the control had the ReadOnly property checked, it was allowed to select a value from the list, as in previous versions, but it did not modify the value of the edit field.

## 6.2 Development environment

- When generating the maintenance of tables with many fields using the Wizard of the development environment, a general protection error occurred.
- Struct class. When deleting all the elements of a structure and trying to create a new one of Char type, pressing the button produced a general protection error and the process was aborted.

## 6.3 Cosmos WebServer

- The printing of reports using the prnpag32 dll was not performed correctly on some occasions.
- Sometimes the string returned by a REST request from Cosmos WebServer was truncated. The character that caused this behavior is the masculine ordinal numeral (character º, ASCII 167). The same behavior was caused by the feminine ordinal numeral (character º, ASCII 166).
- When executing a Cosmos REST service many times (more than 10,000), the global variables lost their value.
- The GetEnv and GetIni methods did not return the value of the variables defined in the configuration file when they were executed from a REST service.
- Cosmos WebServer did not reset the status return code of the HTTP request after each execution. If no return code was assigned from the Cosmos program, Cosmos WebServer should return code 200. The problem was that if an error code was assigned in the execution of a service and the next execution was not assigned the return code, instead of returning 200 (its default value), it returned the code assigned in the previous execution.

## 6.4 CTSQL

- When establishing several connections to the same database, sometimes the following message was produced: "Unable to open the system catalog (systables)". This error only occurred in the Windows sql engine.

## 6.5 CosmosData

- The modifications made to the definition of the calculated fields were not saved.
- In the presentation of the report, the fields of type Smallint and Integer were not formatted correctly.
- When executing the report and generating a list, the cell where the result of the aggregate function was shown was not formatted correctly. It did not show decimals.