



# MultiBase Cosmos

Notes to version 8.2

**BASE100**

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

## Index

<b>1. IMPLEMENTATIONS .....</b>	<b>3</b>
1.1 RUNTIME .....	3
1.2 DEVELOPMENT ENVIRONMENT .....	4
1.3 APIS .....	5
1.3.1 COSHTTPDLL .....	5
1.3.2 COSSMTPDLL.....	5
1.4 CSQL .....	5
1.5 COSREP.....	5
1.6 ODBC CLIENT.....	5
<b>2. NEW METHODS AND FUNCTIONS.....</b>	<b>6</b>
2.1 MODULE CLASS .....	6
2.2 CLASE SIMPLECONTROL.....	7
2.3 COSHTTPDLL.....	12
<b>3. ENVIRONMENT VARIABLES .....</b>	<b>15</b>
<b>4. FIXES.....</b>	<b>16</b>
4.1 RUNTIME .....	16
4.2 DEVELOPMENT ENVIRONMENT .....	16
4.3 CSQL .....	16
4.4 COSMOS WEBSERVER .....	17
4.5 CTSQL .....	17

## 1. Implementations

---

### 1.1 Runtime

UrlSendCommand method. The DIRNAMES command has been implemented, it returns a list with the name of the files and directories of the path it receives as a parameter.

Allow numerical values in hexadecimal and binary notation in the COOL language. Hexadecimal numbers will start with 0x, and binary numbers with 0b.

The DBUSER user password size has been expanded. Starting with Cosmos version 8.2, it supports up to 256 characters.

### Graphic controls

Possibility of drawing the icons of the disabled controls of a «Form» without shading.

#### Box control

- Rounded boxes in Box controls.
- SetBoxIcoFile method of the SimpleControl class, which allows assigning an icon file to a Box control and all its children.

#### Check control

Rounded boxes in Check controls.

#### Tab control

- Possibility of visually prioritizing the tabs of a Tab Control, like an accordion menu.
- SetTabIcoFile method of the SimpleControl class, which allows assigning an icon file to a Tab control.
- GetTabIcoFile method of the SimpleControl class, which allows obtaining the name of the icons file of a Tab control.

#### Edit control

- Possibility to assign an icon to the «Edit Field DateTimePicker» control.
- If the «ALLOWCOPYCONTROLDATA» environment variable is defined, copying the content of the “Edit Field password” controls is not allowed.

#### List Box control

Ability to select the background color of the selected row from a list. The “«LISTCURRENTROWBACKCOLOR» environment variable has been implemented.

#### RichText control

Ability to define the size of the print page margins in millimeters. The RichTextPrintEx method has been implemented.

## Visual style mode 7

- New property in the «Custom Colors» section Color\_Flap\_Tab. Allows you to assign a color to the tabs of the TabControl control.
- Ability to assign «Custom Controls» and «Custom Colors» sections per control, not just per application as in previous versions.

## 1.2 Development environment

The implementations carried out in the Cosmos visual editor are detailed below.

- Debug. New options:
  - *Debug Project*. It allows you to run the project in debug mode taking the main module as input (defined as «Main Module» in its properties).
  - *Show Selected Variable data*. In the «Watch» window of the Cosmos debug, a new button has been added to view and copy the content of the selected variable.
- *Module window*.
  - *Drag & Drop*. Starting with version 8.2, you can drag to copy variables and objects between different areas: object editor, «Form» and «Page» class variable editor, and table editor.
  - *Module context menu*. New options:
    - *Find String in the module*. Allows you to search for a string of characters within the selected module at the level of:
      - Module. Select the «Find String» context menu option in the main node of the module. Excludes libraries and includes.
      - The module libraries. Select the «Find String» context menu option in the library node.
      - The module includes. Select the «Find String» context menu option in the include node.
      - A module library. Select the «Find String» context menu option in the library in which you want to search.
      - An include of the module. Select the «Find String» context menu option in the include in which you want to perform the search.
    - *Open containing folder in the module*. Allows you to open the containing folder within the selected module at the level of:
      - Module. Select the context menu option «Open containing folder» in the main node of the module.
      - Library. Select the module library, display the context menu and choose the «Open containing folder» option.
      - Include. Select the module include, display the context menu and choose the «Open containing folder» option.
- *Graphic editor of a Form screen*. New features:
  - Search by variable name in the «Control Information» window.

- *Select Parent* from the context menu. Allows you to select the parent control of the control.
- *Status bar*. Show name of the variable associated with the control.
- *Code editor*. New features:
  - Enable functions by key combination of deleting entire line in progress («CONTROL-SHIFT-D»), duplicating line («CONTROL-D») and text selection. To select text, place the cursor in front of the first letter of the word to be selected, then move after the last letter of the text you want to select and press «SHIFT + Click».
  - Change the font size («CONTROL+Mouse Wheel»).
- *Code insight*. Pressing the «enter» key after the «begin» keyword will add the «end» keyword to the next line.

## 1.3 APIS

### 1.3.1 COSHTTPDLL

Implementation of the PATCH verb of an HTTP request and implementation of the possibility of sending multipart requests.

### 1.3.2 COSSMTPDLL

Starting with version 8.2 of Cosmos, the date in the RFC 2822 format is sent in the email header, including time zone.

## 1.4 CSQL

- Possibility of consulting information from tables, columns and indexes in a simple way. Instructions such as: SHOW TABLES, SHOW COLUMNS FROM <table>, SHOW INDEXES FROM <table>, SHOW REFERENCES FROM <table> y SHOW REFERENCES TO <table>.

This statement queries data in the catalog tables of the CTSQL database. Therefore, it can only be used with connections to a CTSQL database and with the Informix or Oracle gateway, but not with ODBC connections to other DBMSs.

- Sentence execution results window. In it you can consult: errors, number of records in the query, number of records modified, deleted and inserted.

## 1.5 COSREP

Execute command silently. A new parameter has been implemented for this command “-logfile <log file path>”. The result of the operations is recorded in the file indicated.

## 1.6 ODBC Client

The process of mass insertion of data into ODBC connections has been optimized.

## 2. New methods and functions

---

### 2.1 Module class

#### **AddCustomColors**

It allows loading, at run time, the definition of a new «Custom Colors» section, which can be used later by the «SetCustomColors» method. In this way, it will be possible for the application controls to be displayed with a color scheme different from that defined with the «LoadCustomColors» method.

Each call to the «AddCustomColors» method will load a new definition at runtime. The number of calls is unlimited.

Syntax:

```
AddCustomColors(filePath as Char) return integer
```

Parameters:

filePath                      Path of the file that contains the new «Custom Colors» section.

Returns:

Integer number with the identifier that can be used later by the «SetCustomColors» method.

The identifier returned by the «AddCustomColors» method can be used in any module of the application. It is possible to execute the «AddCustomColors» method multiple times with different «Custom Colors» definition files, since each time it is executed it returns a different identifier, and use different "Custom Colors" definitions for different controls.

#### **AddCustomControls**

It allows loading, at run time, the definition of a new «Custom Controls» section, which can later be used by the «SetCustomControls» method. In this way, it will be possible for the application controls to be displayed with a property scheme different from that defined with the «LoadCustomControls» method.

Syntax:

```
AddCustomControls(filePath as Char) return integer
```

Parameters:

filePath                      Path of the file that contains the new «Custom Controls» section.

Returns:

Integer number with the identifier that can be used later in the «SetCustomControls» method.

The identifier returned by the «AddCustomControls» method can be used in any module of the application. It is possible to execute the «AddCustomControls» method multiple times with different «Custom Controls» definition files, since each time it is executed it returns a different identifier, and use different «Custom Controls» definitions for different controls.

### **SetGlobalEditDateTimePickerIcon**

Allows you to assign an icon to all «Edit DateTimePicker» type application controls that have not been assigned an icon.

Syntax:

```
SetGlobalEditDateTimePickerIcon(iconFile as Char ,icon as Smallint default 0)
```

Parameters:

iconFile	Icon file label.
Icon	Icon identifier.

## **2.2 Clase SimpleControl**

### **SetCustomColors**

This method assigns to a control and optionally its children a «Custom Colors» template previously loaded with the «AddCustomColors» method of the Module class.

Syntax:

```
SetCustomColors(colorsId as Integer ,changeChilds as Boolean)
```

Parameters:

colorsId	Identifier of the «Custom Colors» template previously created with the «AddCustomColors» method. It corresponds to the return value of «AddCustomColors».
changeChilds	Boolean that indicates whether the child controls will be assigned the properties/attributes of the new «Custom Colors» template.

### **SetCustomControls**

This method assigns to a control and optionally its children a «Custom Controls» template previously loaded with the «AddCustomControls» method of the Module Class.

Syntax:

```
SetCustomControls(controlsId as Integer ,changeChilds as Boolean)
```

Parameters:

controlsId	Identifier of the «Custom Controls» template previously created with the «AddCustomControls» method. It corresponds to the return value of «AddCustomControls».
changeChilds	Boolean that indicates whether the child controls will be assigned the properties/attributes of the new «Custom Controls» template.

### SetDateTimePickerMethod

It allows you to tell Cosmos that when you press the button of a control with the «DateTimePicker» property, instead of displaying the Windows calendar, it invokes a Cosmos function that will be responsible for implementing this functionality.

Only for controls of type Edit Field DateTimePicker.

Syntax:

```
SetDateTimePickerMethod(method as Char)
```

Parameter:

Method	Name the function you want to invoke.
--------	---------------------------------------

The called function must have the following characteristics:

- Receive as a parameter the «Edit Field» control of type «DateTimePicker» that invokes it.
- Return a string with the new value of the «Edit Field».

Example:

```
ctrEditDateTimePicker.SetDateTimePickerMethod("MiMetodoDateTime");
public function MiMetodoDateTime(mictr as SimpleFormControl) return
char
objects begin
  f as frmCalendar
  xPos yPos as integer
  xSize ySize as integer
end
begin
  mictr.GetSize(xSize, ySize);
  mictr.GetPos(xPos, yPos);
  xPos += 7;
  yPos += ySize + 32;
  f.Frame().Move(xPos, yPos);
  f.selectedDay = mictr.Text;
  f.Run;
  return f.selectedDay;
end
```



### SetTabControlHierarchy

It allows you to visually prioritize the tabs of a «TabControl», like an accordion menu.

Only for «TabControl» with vertical tabs.

Syntax:

```
SetTabControlHierarchy(hierarchy as Char ,allOpen as Boolean) return boolean
```

Parameters:

Hierarchy	Character string that describes the position of each tab within the tab hierarchy. The definition of each tab will be delimited by the character «;», and each level will be separated by the character «.». The tab identifier is the tab number within the control.  Container tabs can be created by indicating a label. The identifier of these new tabs will be their label. These new tabs do not accept the «On click» event.  If its value is «NULL», the TabControl tabs will be displayed with the same appearance as they have in layout.
allOpen	Boolean that indicates whether the tabs will be displayed folded or unfolded.

Returns:

TRUE If the execution of the method has been correct.

FALSE If the execution of the method has been incorrect (the control is not a vertical «TabControl» control).

Example.

In the design of the TAB control, the following tabs have been defined:

1. Maestros
2. Artículos
3. Clientes
4. Proveedores
5. Provincias
6. Formas de Pago
7. Albaranes
8. Líneas Albarán

To show the following design running:

```
Maestro
  Artículos
  Cliente
  Proveedores
  Auxiliares (está pestaña no está en diseño)
  Provincias
  Formas de Pago
  Albaranes (está pestaña no está en diseño)
  Albaranes
  Líneas de Albaran
```

The value of the Hierarchy parameter will be:

```
str = "1;1.2;1.3;1.4;Auxiliares.5;Auxiliares.6;Albaranes.7;Albaranes.8";
```

The definition of tabs will be:

- 1; Definition of tab 1.
- 1.2; Definition of tab 2, as a child of tab 1.
- 1.3; Definition of tab 3, as a child of tab 1.
- 1.4; Definition of tab 4, as a child of tab 1.
- Auxiliares.5; Definition of tab 5, as a child of the «Auxiliares» tab.
- Auxiliares.6; Definition of tab 6, as a child of the pestaña «Auxiliares» tab.
- Albaranes.7; Definition of tab 7, as a child of the «Albaranes» tab.
- Albaranes.8; Definition of tab 6, as a child of the «Albaranes» tab.

The second parameter «allOpen» of the method will only fold/unfold the child tabs of «Auxiliaries y Albaranes».

### SetBoxIcoFile

This method allows you to assign an icon file to a «Box» control and all its children. If the «Box» control or one of its children already has an icon file assigned, it will be changed to the one indicated in this method.

Syntax:

```
SetBoxIcoFile(icoFile as Char) return boolean
```

Parameters:

icoFile                      Icon file identifier.

Returns:

Boolean that indicates whether the icon file could be assigned.

### **GetTabIcoFile**

This method allows you to consult the name of the icon file associated with a «Tab» control.

Syntax:

```
GetTabIcoFile() return char
```

Returns:

A string with the name of the icon file, NULL if no icon file is assigned.

### **RichTextPrintEx**

Prints the content of the control to the printer allowing the size of the margins to be indicated in millimeters.

Syntax:

```
RichTextPrintEx(printerName as Char ,showPrintDlg as Boolean default FALSE  
,leftMargin as Smallint default 0 ,topMargin as Smallint default 0  
,rightMargin as Smallint default 0 ,bottomMargin as Smallint default 0)
```

Parameters:

PrinterName	Name of the printer you want to print to.
showPrintDlg	If TRUE, it will display the printer selection box.
leftMargin	Integer number indicating the left margin of the print page.
topMargin	Integer number indicating the top margin of the print page.
rightMargin	Integer indicating the right margin of the print page.
bottomMargin	Integer number indicating the bottom margin of the print page.

### **SetTabIcoFile**

This method allows you to assign an icon file to a «Tab» control. If the «Tab» control already has an icon file assigned, it will be changed to the one indicated in this method.

Syntax:

```
SetTabIcoFile(icoFile as Char) return boolean
```

Parameters:

icoFile	Icon file identifier.
---------	-----------------------

Returns:

Boolean that indicates whether the icon file could be assigned.

## 2.3 COSHTTPDLL.

The possibility of sending multipart requests has been implemented, that is, requests in which the body is made up of more than one part. Each part, in turn, will be made up of a header followed by a body.

Multipart requests, unlike single-body requests, include the «Content-Type» header with the value «multipart/form-data». However, in COSHTTPDLL it is not necessary to specify it explicitly, since it includes them automatically when creating a «multipart» request.

In COSHTTPDLL, a request will be multipart when the request body is specified by calling the `CosHttpAddBodyPart` function.

For this, the following functions have been implemented:

### **CosHttpAddBodyPart**

Allows you to add a part to the body of a «multipart» request. Returns the identifier of the added part.

When you execute this function, the request will automatically become «multipart» and you will not need to execute the «`CosHttpRequestSetBody`» function, since the call to this function takes precedence over the call to the «`CosHttpAddBodyPart`» function.

Syntax:

```
public dll "coshttdll.dll" CosHttpAddBodyPart(requestID as integer) return integer
```

Parameters:

requestID	Identificador de la petición.
-----------	-------------------------------

Returns the following values:

>=0	Identifier of the request body part.
-1	The request identifier does not exist.

### **CosHttpSetPropertyStrBodyPart**

Assigns the value to the property.

Syntax:

```
public dll "coshttdll.dll" CosHttpSetPropertyStrBodyPart(requestID as integer, bodyPartId as integer, propId as char, propValue as char) return integer
```

## Parameter:

requestID	Request identifier.
bodyPartId	Identifier of the request body part («bodypart»).
propId	Property name. The possible values are:

Name	Description
NAME	Assigns a value to the bodypart's «name» property.
FILENAME	Assigns a value to the bodypart's «filename» property.
MIMETYPE	Assigns a value to the «Content-Type» header of the bodypart.
MIMEENCODER	Assign value to the «Content-Transfer-Encoding» header of the body part. The possible values are: <ul style="list-style-type: none"> <li>• «8bit». The data is not modified, and the «Content-Transfer-Encoding: 8bit» header is automatically added.</li> <li>• «7bit». The data is not modified, but it is checked that each byte is a 7-bit value. If not, an error is returned. The «Content-Transfer-Encoding: 7bit» header is automatically added.</li> <li>• «binary». The content data is not modified, and the «Content-Transfer-Encoding: binary» header is automatically added.</li> <li>• «base64». The data is converted to base64, and the «Content-Transfer-Encoding: base64» header is automatically added.</li> <li>• «quoted-printable». Data is encoded into printable lines within quotes of a maximum of 76 characters. The «Content-Transfer-Encoding: quoted-printable» header is automatically added.</li> </ul>
DATA	Assigns an alphanumeric string as a value to the «bodypart».
FILEDATA	Assigns the content of a file to the «bodypart» as a value.

propValue	Property Value.
-----------	-----------------

The return values of the functions are as follows:

0	The process has completed successfully.
-1	The request identifier does not exist.
-2	Identifier of the «bodyPart» does not exist.
-3	Unknown property.
-4	The data type assigned to the property is incorrect.

### 3. Environment variables

---

#### **DRAWDISABLEDICONASENABLED**

This environment variable allows the icons of a Form's controls, which have the «disabled» property set to TRUE, to be drawn without shading.

It must be defined in the «Environment» section of the «Cosmos.ini» configuration file, in the project configuration file or with the «PutEnv» method of the «Module» class.

Its possible values are: TRUE and FALSE, with TRUE being its default value.

Once the Cosmos runtime has read the value of this variable, its value cannot be modified.

#### **LISTCURRENTROWBACKCOLOR**

This environment variable allows you to indicate the background color of the selected row in a «List Box» control.

## 4. Fixes

---

### 4.1 Runtime

- «Random» method. An execution error occurred in the «Random» method of the Module class when the second parameter was one unit smaller than the first parameter. Now returns null.
- The «SetGlobalEditDateTimePickerIcon» method did not work correctly if the COSMOSVISUALMODE environment variable was set to 7.
- If the variable «VISUALMODEFORZEBACKCOLOR» was defined with a value TRUE, buttons with the «NOLABEL» attribute were not displayed with the colors indicated in the properties «Color\_Text\_Button\_Disabled», «Color\_Text\_Button\_Hot», «Color\_Text\_Button\_Pressed» and «Color\_Text\_Button\_Focused» of the «Custom Colors» section.
- A memory error occurred if the SetListColumnText method received the value «NULL» as a parameter.
- ExportToExcel method. When exporting the contents of a List Box control to Excel, the contents of columns that have been styled with a charset were not exported correctly.
- ExportToPDFEx method. It did not export the «underline» and «strikeout» properties of the fonts.
- DROP LIST of type SQL. The selected property did not select the record if its value is 1.
- Json Class. If the property name contained the character «.» (dot) does not correctly generate the object.
- General protection error when closing a window with a GRID to which more than 100 icons had been added.
- COSSMTPDLL. Sometimes it does not allow sending emails. The «Cossmtpsendmail» function returned error 56, and the server returned error 554. Some servers require the date in «RFC 2822» format in the mail header, including time zone.

### 4.2 DEVELOPMENT ENVIRONMENT

- Code Insight. It did not show the methods of the FormTable class if the table and the module had the same name.
- Code Insight. If a variable is defined in the «Objects» section of a Cosmos function and you write a variable followed by a dot, its methods would not be displayed if the code analyzer had not been previously passed («F6» key).

### 4.3 CSQL

When pressing the «F1» key to get help from a SQL statement on the word «select», Interactive SQL would freeze.



#### 4.4 Cosmos WebServer

The process would run indefinitely, without sending the response, if the icon file used in a module of the project did not exist in the specified path.

#### 4.5 CTSQL

The «select» statement with the following condition «matches "[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]" order by» for a table column that was part of an index did not return all records that met the condition.

When executing the «AVG" function, the database manager returned the following message «division by zero» if the column passed to it as a parameter has null values. This error occurs since Cosmos version 7.8.3.