MultiBase Cosmos

Notes to version 5.6

BASE100

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

Table of contents

1. Incompatibilities

It is mandatory to recompile the application if it was compiled with previous versions of Cosmos.

It is mandatory to modify the programs in the application if it uses the prnpag32.dll dll. Some functions have changed the type of some parameters from "smallint" to "integer".

This is the list of functions that have changed the type of parameters or return values:

rootControl(HPage as integer) return **integer**
nextControl(HPage as integer, idm as **integer**) return **integer**
controlByName(HPage as integer, usrid as char) return **integer**
controlByUsrIdm(HPage as integer, usridm as **integer**) return **integer**
getControlName(HPage as integer, idm as **integer**) return char
getControlUsrIdm(HPage as integer, idm as **integer**) return **integer**
childControl(HPage as integer, idm as **integer**) return **integer**
getWidth(HPage as integer, idm as **integer**) return integer
getHeight(HPage as integer, idm as **integer**) return integer
parentControl(HPage as integer, idm as **integer**) return **integer**
setControlText(HPage as integer, idm as **integer**, text as char)
count(HPage as integer, idm as **integer**) return integer
addBand(HPage as integer, idmGroup as **integer**, idmBand as **integer**) return boolean
remainingSpace(HPage as integer, idm as **integer**) return integer
setPropStr(HPage as integer, idm as **integer**, prop as char, text as char)
setPropInt(HPage as integer, idm as **integer**, prop as char, valueprop as integer)
getPropStr(HPage as integer, idm as **integer**, prop as char, var text as char, len as smallint)
getPropInt(HPage as integer, idm as **integer**, prop as char) return integer
moveControl(HPage as integer, idm as **integer**, x as smallint, y as smallint) return boolean
changeSizeControl(HPage as integer, idm as **integer**, xSize as smallint, ySize smallint) return boolean
getXPos(HPage as integer, idm as **integer**) return integer
getYPos(HPage as integer, idm as **integer**) return integer
setXPos(HPage as integer, idm as **integer**, xPos as smallint)
setYPos(HPage as integer, idm as **integer**, yPos as smallint)
setBoxRoundCornerRadio(HPage as integer, idm as **integer**, radio as smallint)

2. Implementations

2.1 Cosmos

2.1.1 Cosmos Runtime

- New dll coshttpdll. This dll allows to establish connections to web server with the http protocol.
- Multi-language. The Cosmos Runtime allows the change the language of the control's labels and control's comments. (**see Annex 4**).
- The Cosmos Runtime allows to run a function or a method in a Java class (**see Annex 5**).
- New event ListSpreadSheetColChange. This event is launched when the current cell is changed to another cell in the same row in a List Box (string or sql), and the cell navigation is enabled (SetListSpreadSheetNavigation method).
- New Methods of the SimpleControl class that allow to:
  - Accept and cancel editing List Box (editable list): ListInvokeAcceptEdit and ListInvokeCancelEdit (**see Annex 6.1**).
  - Backup and read the list attributes (List Box string and sql) at runtime: SetListStatusStr y GetListStatusStr methods (**see Annex 6.1**).
  - Get a string with the configuration of a grouped list (**see Annex 6.1**).
- New method in the Char class that allows to replace characters in a char object.
- New function in the TTXMLDLL dll that allows to create a new document from an XML file and a style file. New function that allows to change the name or the content in a node of the xml document.

2.1.2 IDE

- The information in the Find in Files tab of the Output window is shown by columns.
- New option in the popup menu (right mouse button) that allows to copy a line to the clipboard.
- Code insight.

  1. In this version you can change el color of the tokens in the code editor.

     The option to change the token's colors is in the Menu Tools, option Settings, tab Editor. You can change the color of: keyword (reserved words, default color: red), identifiers (i.e. name of controls, default color: black), numbers (default color: blue), comments (default color: green), delimiters (default color: black) and character strings (default color: blue).

     The information will be saved automatically in the Cosmos configuration file.

  2. New shortcut [Ctrl]+[Space] in the Code section of a class. Cosmos will show a list of objects, constants, variables, controls, methods and properties that starts with the letter(s) at the left of the cursor.

3. Ability to locate the definition of an item of the application of a simpler way. In this new version, pressing [Ctrl] + [t] over the item, the mouse cursor will be positioned in the place where the item is defined.

   Limitations of the points 2 and 3:

   I. Commands are not included.

   II. If the control parent is a tab control, the focus will be positioned in the control, but not in the tab page where the control is located.

2.2 DBMS (CTSQL)

From this version, the database manager will generate, optionally, a file of statistics where information on the execution of SQL statements is displayed in connection to the database:

- Prepare, Open, Execute, Fecth of every SQL statement.
- Time taken in the execution by every function of the SQL statements.

If the client application is a Cosmos application, the DBMS will take into account in the statistics all the SQL statements executed by the Cosmos Runtime in the predefined commands (EditUpdate, Add, etc) and the pure SQL statements executed with the methods of the SqlCursor class, SqlStatement class and SqlServer class.

To activate/deactivate the statistics have been implemented the following mechanisms:

1. SQL statement:

```
set statistics to 1 (activate the statistics)
set statistics to 0 (deactivate the statistics)
```

   The statistics file will be generated when “set statistics to 0” is executed or the client is disconnected.

2. CTSQLSTATISTICS environment variable:

   The possible values are: TRUE/FALSE or YES/NO.

   TRUE to activate the statistics and FALSE to deactivate the statistics.

   This environment variable can be defined in the ctsql.ini file or in the Database Connection section (cosmos.ini or project’s ini file). This variable can also be defined with the PutEnv method of the Module class and the SetValue method of the SqlServer class. The call to the methods must be done before the call to the Connect method.

   The statistics file will be generated after disconnecting from the database.

In case of using the environment variable and the SQL statement, the statistics file will be generated after the SQL statement (set statistics to 0).

The appearance of the statistics file is as follows:

Fecha	24/04/2015 16:36:25
BBDD	h:\ctl3606\mbdemo\almacen
IP Cliente	127.0.0.1

ID	Statement	Nº de Prepare	Nº de Open	Nº de Execute	Nº de Fetch	Tiempo Prepare (ms)	Tiempo Open (ms)	Tiempo Ejecución (ms)	Tiempo Fetch (ms)	Instrucción	Índice Usado
0	FNDBCHANGE	1	0	1	0	0,05	0,00	2,06	0,00	database almacen;	<ninguno>
1	FNSELECT	1	1	0	194	1,48	0,26	0,00	1,02	select * from articulos;	<ninguno>
2	FNSELECT	1	1	0	55	0,21	0,31	0,00	0,83	select * from articulos where existencias > 75;	<ninguno>
3	FNSELECT	1	1	0	164	0,19	0,30	0,00	1,31	select * from articulos where articulo > 30 ;	articulos primary

These are the data shown for each SQL statement:

ID	Internal identifier of the SQL statement.
Statement	Internal code of the SQL statement.
Nº de Prepare:	Number of times that the SQL statement has been prepared.
Nº de Open	Number of times that the internal cursor has been opened, in case of query statements.
Nº de Execute	Number of times that the SQL statement has been executed.
Nº de Fetch	Shows the number of times that the Fetch instruction has been executed in a SQL Statement (in case of query statements).
Tiempo Prepare (ms)	Time (in millisconds) that the DBMS took in execute the 'n' Prepare instructions.
Tiempo Open(ms)	Time (in millisconds) that the DBMS took in execute the 'n' Open instructions.
Tiempo Ejecución (ms)	Time (in millisconds) that the DBMS took in execute the 'n' Execute instructions.
Tiempo Fetch (ms)	Time (in millisconds) that the DBMS took in execute the 'n' Fetch instructions.
Instrucción	SQL Statement.
Índice usado	Name of the database index used in the SQL Statement. This column also will be displayed with a background color. These colors are: • Green. The cell's background color is green when: The SQL statement needs an index and the DBMS has selected one index. The SQL statement don't needs an index, i.e: "database stock".

- Orange. The cell's background color is orange if the SQL statement needs an index for the optimization and is not using any index. I.e. "select * from items where stock > 75", and there is not an index with the field "items" as first field.

3. Improvements

- GetUrlFileEx. The function has been modified to accept the sftp, scp and https protocols.
- CallWebService. The method has been modified in order to accept up to 32.767 bytes in the "header-Send" parameter.
- New method CallWebServiceEx. This method allows to connect to a web service. The difference between this method and the method CallWebService is that, instead of a string with the information that the method must send to the web service, the new method receives a file path as parameter. This file has the information that the method must send to the web service, so the limitation of 32.767 bytes in the headerSend parameter is removed.
- The limitation of 32.767 lines in the source code of a debugged module has been removed.
- Code editor. The limitation of 32.767 lines in the source code of a edited module has been removed.
- The limitation of 32.767 elements in a Preview print has been removed.
- The Interactive-Sql tool (Csql.exe) allows to run a single selected statement.
- Grouped lists (ShowMultiColumnGroupDlg and ShowListAsMultiColumnGroup methods). New environment variable SHOWAGGSINCOMPACTMODE. This environment variable allows to show the group total header when the list is in compact mode without detail and the group node is closed.

4. Multi-language

From this version the Cosmos Runtime can change the language of the control's labels and control's comments at runtime.

The labels and the comments are stored in a text file with the translated text.

The structure of the file is as follows:

```
Label=Translated_label
```

Example:

```
Control de Acceso=Access Control
Password:=Password:
Codigo Cliente=Customer Code
Empresa=Company
Salir=Exit
```

If the control's label don't match any of the labels in the file, the original label will be shown.

You can tell the Cosmos Runtime will use the translation file in two ways:

1. Environment variable MULTILANGUAGEFILE.

```
[Environment]
MULTILANGUAGEFILE=c:\cosmos\project\Almafac\english.txt
```

2. A new parameter in cosrun.exe.

-multilanguagefile	This paremeter designates the absolute path or relative path (to the project) of the translation file.

In order to make it easier to detect labels that have not been translated, we have implemented the possibility that the Cosmos Runtime generates a log file that shows the labels and comments without correspondence in the translation file during the execution of the application. The path of the debug file will be indicated with an environment variable (MULTILANGUAGEDEBUGFILE) or with a new Cosrun.exe parameter: "-multilanguagedebugfile".

It is not mandatory to declare the MULTILANGUAGEDEBUGFILE nor the parameter, but it is advisable during the application's translation phase.

Both environment variables are declared in the Environment section of the application's INI file.

This implementation only affects to the labels of the controls assigned in design time. If the application changes a control label at runtime, this label prevails over the translation file's label.

5. Calling a Java method/function from Cosmos

From this version, the Cosmos Runtime can run Java methods from a Cosmos module.

Cosmos needs:

1. Installed Java Virtual Machine.

2. Dll cosjavadll.dll.

   This dll is an intermediate between Cosmos Runtime and Java (The dll links with jvm.dll).

3. Dll to establish the connection between Cosmos and Java.

   Cosmos uses the JNI framework to connect the Cosmos Runtime and Java.

   It is mandatory that the PATH environment variable included the path of the jvm.dll and msvcr100dll files from Java (32-bit).

4. Java method declared in the Cosmos module.

   Syntax:

   ```
   access javaclass package java_method(parameters) [RETURN class]
   ```

access	This field is required. This parameter tells to the Cosmos Runtime the kind of access of the function: public, private or protected.
javaclass	Reserved word. This field tells to the Cosmos Runtime that the function declared is a java method.
package	Quoted string that tells the package and the class where the Java method is defined.
Java_method	Quoted string that tells the java method name.
parameters	List of valid COOL identifiers, separated by commas. The "JavaObject" keyword is used to assign the parameters of the Java method that must map objects.
RETURN	Optional word. Tells that the function returns a value.
class	Type of the return value.

5. File java.options

   The parameters of the Java Virtual Machine are declared in this file.

   ```
   -Djava.class.path=c:\samples\chart\java\ejemplo1
   -Xms8m
   -Xmx24m
   ```

   If this file don't exists in the project's folder, the Cosmos Runtime will search a folder called "jars" inside the project's folder.

The “jars” folder must include all the “jar files” needed for the execution of the Java methods.

The Java Virtual Machine MUST be the 32-bit version.

5.1 Cosmos objects and Java objects type match in the function’s parameter definition

The Cosmos Runtime can run Java methods and functions if the type of the parameters are: String, short, Short, double, Double, int, Integer, boolean and Boolean.

The return data type must be a primitive data type or an object of String class.

The correspondence between the data type of the parameters in a Java function or method and a Cosmos function are as follows:

Java	Cosmos
String	char
short	smallint
Short	javaobject smallint
double	decimal
Double	javaobject decimal
int	integer
Integer	javaobject integer
boolean	boolean
Boolean	javaobject boolean

If the parameter of the Java function or method is an object of a basic class instead of a primitive type, the JavaObject word must be included in the definition of the parameter.

6. New methods

6.1 SimpleControl Class methods

- ListInvokeAcceptEdit. This method allows to accept the edition in a List Box Control, so the programmer can define a shortcut that allows to perform the action. The [Enter] key will perform the same action. The definition of a new accelerator don't disables the default shortcut.

  Syntax:

  ```
  ListInvokeAcceptEdit()
  ```

- ListInvokeCancelEdit. This methods allows to cancel the edition in a List Box Control, so the programmer can define a shortcut that allows to perform the action. The [Esc] key will perform the same action. The definition of a new accelerator don't disables the default shortcut.

  Syntax:

  ```
  ListInvokeCancelEdit()
  ```

- GetListMultiColumnGroupStr. This method returns a string with the configuration value of a grouped tree list created with the ShowListAsMultiColumnGroup or ShowMultiColumnGroupDlg methods.

  Syntax:

  ```
  GetListMultiColumnGroupStr() return Char
  ```

  Returns:

  A string with the same format as the first parameter of the ShowListAsMultiColumnGroup method.

- GetListStatusStr. This method returns a string with attributes of the columns of the List Box control (String or Sql), drop edit and drop list.

  These are the attributes returned for each column:

COLNUMBER	Column Id.
COLWIDTH	Column width in pixels.
COLVISIBLE	Column status (visible/invisible).
COLWIDTHCHARS	Column width in number of characters.
COLPOSITION	Column position.

  Syntax:

  ```
  GetListStatusStr() return char
  ```

  Returns:

  String of characters.

- SetListStatusStr. This method allows to modify the attributes of the columns in a control of type List Box (String or Sql), drop edit and drop list.

  The method receives as parameter a string of characters with the attributes for each column.

Syntax:

```
SetListStatusStr(statusStr as Char)
```

Parametsrs:

statusStr	Attributes and values for each column. Format : The column separator is a semicolon (';') and the attribute's separator is colon (':').

Example:

"COLNUMBER=1:COLWIDTH=57:COLVISIBLE=1:COLWIDTHCHARS=11:COLPOSITION=1;COLNUMBER=2:COLWIDTH=57:COLVISIBLE=1:COLWIDTHCHARS=11:COLPOSITION=2**;**COLNUMBER=3:COLWIDTH=57:COLVISIBLE=1:COLWIDTHCHARS=11:COLPOSITION=3**;**COLNUMBER=4:COLWIDTH=57:COLVISIBLE=0:COLWIDTHCHARS=11:COLPOSITION=4;COLNUMBER=5:COLWIDTH=57:COLVISIBLE=1:COLWIDTHCHARS=11:COLPOSITION=5;COLNUMBER=6:COLWIDTH=57:COLVISIBLE=1:COLWIDTHCHARS=11:COLPOSITION=6;COLNUMBER=7:COLWIDTH=57:COLVISIBLE=1:COLWIDTHCHARS=11:COLPOSITION=7;COLNUMBER=8:COLWIDTH=57:COLVISIBLE=1:COLWIDTHCHARS=11:COLPOSITION=8;"

6.2 Module Class methods

- GetMemoryStatus. This method gets the current phisical and virtual memory status.

  Syntax:

  ```
  GetMemoryStatus(infoType as Integer, var memoryIfo as Decimal)
  ```

infoType	This parameter indicates the memory state to be consulted. The values are: 1. A number between 0 and 100 that tells the percent of phisical memory used (0 tells that there is not used memory and 100 tells that there is not free memory). 2. The amount of real phisical memory, in bytes. 3. The amount of phisical memory, in bytes. 4. The amount of total bytes that can be stored in the pagination file. 5. The amount of bytes available in the pagination file. 6. The maximum amount of memory the current process can commit, in bytes. 7. The size of the user-mode portion of the virtual address space of the calling process, in bytes.
memoryInfo	Parameter by reference where the return value is stored.

  Returns:

  TRUE if the method could be executed without errors. FALSE if the first parameter is not between 1 and 7.

6.3 Char Class methods

- Translate. This method allows to replace characters in a Char class object. The characters listed in the first parameter will be replaced in the object by the corresponding characters listed in the second parameter. The method modifies the value of the Char object.

  Syntax:

  ```
  Translate(stringToReplace as Char, replacementString as Char)
  ```

stringToReplace	List of characters that will be replaced in the char object.
replacementString	List of characers that are to replace those listed in the first parameter.

7. Events

- ListSpreadSheetColChange. This event is launched when the focus is changed from one cell to another cell in the same row of the List Box control (string or Sql), and navigation by their cells is enabled (method SetListSpreadSheetNavigation).

8. APIs. New functions

8.1 TTXMLDLL API functions

- TTXmlApplyXSLTAndSave. This function allows to create a new document from an XML file with data and a style file. The new document can be a different file than an XML.

  Syntax:

  ```
  TTXmlApplyXSLTAndSave(xmlFileName as char, xsltFileName as char, outXmlFile as char, var bytes as integer) return integer
  ```

  Parameters:

xmlFileName	XML document with data.
xsltFileName	Style document.
outXmlFile	New document created from xmlFileName and xsltFileName.
bytes	Parameter by reference. Returns the number of bytes of the new document.

  Returns:

  Return code:

  0: Successful execution.

  -1: Failed to apply the style file to de document.

  -2: Failed loading or parsing the style file.

  -3: Failed loading or parsing the XML file.

  -4: Failed creating the processing context.

- TTXmlUpdateNodeContent. This function allows to modify the content of an XML node.

  Syntax:

  ```
  TTXmlUpdateNodeContent(Doc as integer, node as integer, Content as Char, charset as integer)
  ```

  Parameters:

doc	XML document identifier (code returned by TTXmlNewDoc and TTXmlOpenDoc).
node	Node identifier.
content	New node content. If the parameter value is NULL, the content is deleted.
charset	Character set of the “content” parameter.

- TTXmlUpdateNodeName. This function allows to change the name of an XML node.

  Syntax:

  ```
  TTXmlUpdateNodeName(Doc as integer, node as integer, name as Char, charset
  as integer)
  ```

  Parameters:

doc	XML document identifier (code returned by TTXmlNewDoc and TTXmlOpenDoc)
node	Node identifier.
name	New node name.
charset	Character set of the “name” parameter.

9. New APIs

9.1 Coshttpdll.dll functions

This dll allows to connect to web servers with the http protocol.

9.1.1 CosHttpNewRequest

This function creates a new request to the server.

Syntax:

```
CosHttpNewRequest() return integer
```

Returns:

Request ID.

9.1.2 CosHttpRequestSetUrl

This function sets the server's URL to connect and send request.

Syntax:

```
CosHttpRequestSetUrl(requestID as integer, url as char) return integer
```

Parameters:

requestID	Request ID returned by CosHttpNewRequest.
url	Web server URL.

Returns:

0 Successful execution.

-1 The request ID don't exists.

9.1.3 CosHttpRequestSetMethod

This function sets the request method.

Implemented request methods: HEAD, GET, POST PUT, DELETE, OPTIONS.

Syntax:

```
CosHttpRequestSetMethod(requestID as integer, method as char) return integer
```

Parameters:

requestID	Request ID.

Returns:

0 Successful execution.

-1 The request ID don't exists.

9.1.4 CoshttpRequestAddHeaderStr

This function allows to add a parameter to the http request header.

This function must be called for each new header parameter.

Syntax:

```
CosHttpRequestAddHeaderStr(requestID as integer, headerStr as char) return
integer
```

Parameters:

requestID	Request ID.
headerStr	Header parameter.

Returns:

0 Successful execution.

-1 The request ID don't exists.

9.1.5 CosHttpRequestSetBody

This function sets the request http body. This parameter can be a quoted string with the body content or a file path that stores the body content.

Syntax:

```
CosHttpRequestSetBody(requestID as integer, bodyStr as char, fromFile as
boolean) return integer
```

Parameters:

requestID	Request ID.
bodyStr	Request http body. The value of this parameter is a quoted string. If the value of the "fromFile" parameter is FALSE, this parameter will set the http body string. If the value of the "fromFile" parameter is TRUE, this parameter will set the path of a file with the body content.
fromFile	This parameter sets whether the request body is in a file or on a quoted string.

Returns:

0 Successful execution.

-1 The request ID don't exists.

9.1.6 CosHttpSetResponseFile

This function sets the path of the file where the server response is stored.

Syntax:

```
CosHttpSetResponseFile(requestID as integer, responseFile as char) return
integer
```

Parameters:

requestID	Request ID.
responseFile	Absolute path to the response file.

Returns:

0 Successful execution.

-1 The request ID don't exists.

9.1.7 CosHttpSetReponseHeaderFile

This function sets the file name where the header of the server's response will be stored.

Syntax:

```
CosHttpSetResponseHeaderFile(requestID as integer, responseHeaderFile as
char) return integer
```

Parameters:

requestID	Request ID.
responseHeaderFile	Absolute path to the header response file .

Returns:

0 Successful execution.

-1 The request ID don't exists.

9.1.8 CosHttpSendRequest

This function executes the request specified in the CosHttpRequestSetMethod function. The server will respond and the files specified in CosHttpSetResponseFile and CosHttpSetResponseHeaderFile will be created.

Syntax:

```
CosHttpSendRequest(requestID as integer, onlyHeaders as boolean) return in-
teger
```

Parameters:

requestID	Request ID.
onlyHeaders	Boolean parameter that indicates whether the response file includes header and body or only header. Possible values: TRUE and FALSE. If TRUE, the server will not include the body in the response, and the file specified in the CosHttpSetResponseFile function will not be created. The only file that will be created is that specified in the CosHttpSetResponseHeaderFile function. This case can only be used in the GET method. Not all web servers implement the functionality to return only the headers, so it may return an error message.

Returns:

Error code. If the process has been executed successfully, it returns 0.

Error codes and description:

Code	Description	Code	Description
0	No error	46	Obsolete code
1	Unsupported protocol	47	Number of redirects hit maximum amount
2	Failed initialization	48	An unknown option was passed in to libcurl
3	URL using bad/illegal format or missing URL	49	Malformed telnet option
4	A requested feature, protocol or option was not found built-in in this libcurl due to a build-time decision.	50	Obsolete code
5	Couldn't resolve proxy name	51	SSL peer certificate or SSH remote key was not OK
6	Couldn't resolve host name	52	Server returned nothing (no headers, no data)
7	Couldn't connect to server	53	SSL crypto engine not found
8	FTP: weird server reply	54	Can not set SSL crypto engine as default
9	Access denied to remote resource.	55	Failed sending data to the peer
10	FTP: The server failed to connect to data port Fallo al conectar al puerto	56	Failure when receiving data from the peer
11	Password desconocida. FTP: unknown PASS reply".	57	Obsolete code
12	Time out	58	Problem with the local SSL certificate
13	FTP: unknown PASV reply	59	Couldn't use specified SSL cipher
14	Formato 227 respuesta desconocida. FTP: unknown 227 response format	60	Peer certificate cannot be authenticated with given CA certificates
15	FTP: can't figure out the host in the PASV response	61	Unrecognized or bad HTTP Content or Transfer-Encoding
16	Error in the HTTP2 framing layer	62	Invalid LDAP URL
17	FTP: couldn't set file type	63	Maximum file size exceeded
18	Transferred a partial file	64	Requested SSL level failed
19	FTP: couldn't retrieve (RETR failed) the specified file	65	Send failed since rewinding of the data stream failed
20	Obsolete code	66	Failed to initialise SSL crypto engine
21	Quote command returned error	67	Login denied
22	HTTP response code said error	68	TFTP: File Not Found
23	Failed writing received data to disk/application	69	TFTP: Access Violation
24	Obsolete code	70	Disk full or allocation exceeded
25	Upload failed (at start/before it took off)	71	TFTP: Illegal operation
26	Failed to open/read local data from file/application	72	TFTP: Unknown transfer ID
27	Out of memory	73	Remote file already exists
28	Timeout was reached	74	TFTP: No such user

Code	Description	Code	Description
29	Obsolete code	75	Conversion failed
30	FTP: command PORT failed	76	Caller must register CURLOPT_CONV_ call-back options
31	FTP: command REST failed	77	Problem with the SSL CA cert (path? access rights?)
32	Obsolete code	78	Remote file not found
33	Requested range was not delivered by the server	79	Error in the SSH layer
34	Internal problem setting up the POST	80	Failed to shut down the SSL connection
35	SSL connect error	81	Socket not ready for send/recv
36	Couldn't resume download	82	Failed to load CRL file (path? access rights?, format?)
37	Couldn't read a file.	83	Issuer check against peer certificate failed
38	LDAP: cannot bind	84	FTP: The server did not accept the PRET command.
39	LDAP: search failed	85	RTSP CSeq mismatch or invalid CSeq
40	Obsolete code	86	RTSP session error
41	A required function in the library was not found	87	Unable to parse FTP file list
42	Operation was aborted by an application callback	88	Chunk callback failed
43	A libcurl function was given a bad argumen	89	The max connection limit is reached
44	Obsolete code	90	SSL public key does not match pinned public key
45	Failed binding local connection end		

9.1.9 CosHttpGetErrorStr

This function returns the response's error string corresponding to the execution of the CosHttpSendRequest function.

Syntax:

```
CosHttpGetErrorStr (requestID as integer) return Char
```

Parameters:

requestID Request ID.

Returns:

A character string with the error message.

9.1.10 CosHttpGetReturnCode

This function returns the status code of the http protocol.

Syntax:

```
CosHttpGetReturnCode(requestID as integer) return integer
```

Parameters:

requestID	Request ID.

Returns:

Http status code.

9.1.11 CosHttpFreeRequest

This function frees the request and all the memory used during the request process. It is mandatory to use after the end of the conversation with the server.

Syntax:

```
CosHttpFreeRequest(requestID as integer) return integer
```

Parameters:

requestID	Request ID.

Returns:

0 Successful execution.

-1 The request ID don't exists.

9.1.12 CosHttpUseSSL

This functions sets if the DLL will use a secure connection.

Syntax:

```
CosHttpUseSSL(requestID as integer, useSSL as boolean) return integer
```

Parameters:

requestID	Request ID.
useSSL	Booleano parameter. Available values: TRUE (uses secure connection) or FALSE (don't uses secure connection), default FALSE.

Returns:

0 Successful execution.

-1 The request ID don't exists.

9.1.13 CosHttpIncludeHeaderInResponse

This function tells to the DLL if the header of the server's reponse is included in the same file that the body of the server's response.

Syntax:

```
CosHttpIncludeHeaderInResponse(requestID as integer, includeHeader as boolean) return integer
```

Parameters:

requestID	Request ID.

includeHeader	Boolean parameter. Available values: TRUE (header and body are included in the same file that is seted in the CosHttpSetResponseFile function), and FALSE, the default value, that indicates that the header will be stored in the file specified in the CosHttpSetResponseHeaderFile function.

Returns:

0 Successful execution.

-1 The request ID don't exists.

9.1.14 CosHttpSetAuthUser

This function sets the login user name.

Syntax:

```
CosHttpSetAuthUser(requestID as integer, user as char) return integer
```

Parameters:

requestID	Request ID.
user	User name.

Returns:

0 Successful execution.

-1 The request ID don't exists.

Ver método CosHttpSetAuthMethod.

9.1.15 CosHttpSetAuthPasswd

This function sets the password corresponding to the login user name specified in the CosHttpSetAuthUser function.

Syntax:

```
CosHttpSetAuthPasswd(requestID as integer, passwd as char) return integer
```

Parameters:

requestID	Request ID.
passwd	Password.

Returns:

0 Successful execution.

-1 The request ID don't exists.

See CosHttpSetAuthMethod function.

9.1.16 CosHttpSetAuthMethod

This function sets the authentication metod used to connect to the server.

Syntax:

```
CosHttpSetAuthMethod(requestID as integer, authMethod as integer) return in-
teger
```

Parameters:

requestID	Identificador de la petición.
authMethod	Authentication type. Available values:
	0 No authentication.
	1 Basic.
	2 Digest.
	3 GSS-Negotiate.
	4 NTLM.

Returns:

0 Successful execution.

-1 The request ID don't exists.

9.1.17 CosHttpSetTimeout

This function sets a connection timeout.

Syntax:

```
CosHttpSetTimeout(requestID as integer, secondsTimeout as integer) return
integer
```

Parameters:

requestID	Request ID.
secondsTimeout	Timeout in seconds.

returns:

0 Successful execution.

-1 The request ID don't exists.

10. Bug fixes

10.1 Runtime

- The LookupColumn method of the FormTable class showed no lookup values when the lookup column was the result of a table join and the LOOKUPDELAYED environment variable was TRUE.
- ReplaceAt method of Char class. If the replace string is bigger than the complete string, didn't frees memory.
- Cosrun error in a Terminal-Server sesión in Windows 2012 and Windows 8.
- Grouped list in compact mode. The mouse cursor didn't works if the nodes are closed.

10.2 Cosmos

- SelectWindow method. The OK button didn't responds to mouse clicks.

10.3 IDE

- Double click in the code editor. If the word has the "_" character, the whole word isn't selected.
- Code insight. General protection error when the user writes the token "Self." in the code section of a Page class.
- Code insight. Char object defined in the "variable" section of a Page class. The IDE did not shows the object's methods.
- Code insight. When you define an object of the Decimal, Smallint or Integer class, if a Numeric class method is selected, Cosmos doesn't shows the Numeric class methods.
- Code insight. If the user writes the "this" token, the methods and properties window is not shown.
- Code insight. The methods and properties window was closed if not exists an element that matches the search criteria.
- Code Insight. The Class conversor were not shown.
- IDE, "Find string in Files" option. Error when the search string is "a" and the "Match Whole Word" and "Match Case" are selected.

10.4 CTSQL

- LTRIM and RTRIM functions don't return the correct size.
- Buf fixed in the version 3.6 0.36 of the CTSQL engine when the user name before the table name in a query was longer than 8 characters. This error is only present when the client is the Multibase's JDBC driver.

10.5 MONITOR

- Locked files. Sometimes, the names of table's files had not legible characters.