



# MultiBase Cosmos

Notes to version 7.8

**BASE100**

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

## Index

<b>1. IMPLEMENTATIONS .....</b>	<b>3</b>
1.1 APIS .....	3
1.1.1 COSSIGNFILE .....	3
1.1.2 TTCHARTDLL.....	3
1.2 DEVELOPMENT ENVIRONMENT.....	3
1.2.1 Visual editor .....	3
1.2.2 Modules editor.....	3
1.3 RUNTIME.....	3
1.3.1 JSON Class.....	3
1.3.2 Module Class.....	3
1.3.3 Numeric systems converters .....	4
1.3.4 Image export.....	4
<b>2. NEW METHODS.....</b>	<b>5</b>
2.1 BINARY CLASS.....	5
2.1.1 LoadFromNumber method.....	5
2.1.2 SHA256 method .....	5
2.2 CHAR CLASS .....	5
2.2.1 GetIntegerFromCharHex method.....	5
2.2.2 GetSmallintFromCharHex method .....	6
2.3 CONTROL CLASS .....	6
2.3.1 ExportImageEx method.....	6
2.4 JSON CLASS .....	6
2.4.1 GetChildList method.....	6
2.5 MODULE CLASS.....	7
2.5.1 GetClipboardTextUTF8 method .....	7
2.5.2 LoadJVM method .....	7
2.6 NUMERIC CLASS.....	7
2.6.1 ToCharHex method .....	7
2.7 TTCHARTDLL.....	8
2.7.1 ChartExportImage function.....	8
2.8 API COSSIGNFILE.....	8
2.8.1 CosSignFileSignXAdES_Ex.....	8
<b>3. BUG FIXES .....</b>	<b>10</b>
3.1 RUNTIME.....	10
3.2 REPOSITORY .....	10
3.3 CODE EDITOR .....	10
3.4 APIS.....	10

## 1. Implementations

---

### 1.1 APIS

In this version, two APIs have been implemented: [COSQRDLL](#) and [COSCOMPRESSDLL](#):

- The COSQRDLL DLL allows you to generate an image file of a QR code.
- The COSCOMPRESSDLL DLL allows you to create a compressed file from the files and folders that you specify. It also allows you to unzip files.

#### 1.1.1 COSSIGNFILE

The necessary modifications have been made so that Cosmos can sign XML files in XAdEs format.

In addition, a new function has been implemented, `CosSignFileSignXAdES_Ex`, which allows adding a series of extra parameters to the signature.

The version of API `CosSignfile.dll` that ships with Cosmos version 7.8 is version 1.1.0, which uses the AutoFirma version 1.8 libraries.

#### 1.1.2 TTCHARTDLL

A function has been implemented in the API that allows saving the chart image in an image file (BMP, JPG, etc.).

The `ChartExportImage` function has been implemented, which receives as a parameter the handle of the chart window, the output file and the image type (BMP, JPG).

## 1.2 Development environment

### 1.2.1 Visual editor

The ability to swap the position between two controls, including `taborder`, has been implemented.

### 1.2.2 Modules editor

The ability to select / deselect multiple modules (includes and libraries) from an alphabetically ordered list has been implemented.

## 1.3 Runtime

### 1.3.1 JSON Class

The possibility of traversing a JSON without having to know the name of the properties has been implemented. To do this, the `GetChildList` method of the JSON class has been implemented, which returns a string with the name of the children of a JSON separated by pipes.

### 1.3.2 Module Class

The `GetClipboardTextUTF8` method of the Module class has been implemented, which allows to get text from the clipboard in utf-8.

The LoadJVM method has been implemented, which allows loading a Java virtual machine with a CLASSPATH. The implementation of this method allows the use of several APIs that use Java classes in the same *cosrun.exe* process.

### 1.3.3 Numeric systems converters

A series of methods have been implemented that allow converting a number from one number system to another (decimal, hexadecimal and binary).

- The methods GetSmallinFromCharHex and GetIntegerFromCharHex have been implemented in the Char class. These methods allow obtaining the Smallint or Integer number from a string containing its hexadecimal representation.
- The LoadFromNumber method has been implemented in the Binary class. This method receives as a parameter an object of the Smallint or Integer classes and stores it in the Binary object.
  - If the parameter it receives is of type Smallint, it will occupy 2 bytes.
  - If the parameter it receives is of type Integer, it will occupy 4 bytes.
  - If the parameter it receives is of type decimal it will not store anything.
  - It will be stored in the processor's own format (Little-endian on compatible x86 Intel processors).
- The ToCharHex method has been implemented in the Numeric class, which allows obtaining the hexadecimal representation of a number.

### 1.3.4 Image export

A new method of the Control class, ExportImageEx, has been implemented. This method, unlike the ExportImage method, will export the image associated with the control without it having to be in the foreground.

## 2. New methods

---

### 2.1 Binary class

#### 2.1.1 LoadFromNumber method

This method allows you to load a Smallint or an Integer into a Binary object, similar to the LoadFromChar method.

Syntax:

```
LoadFromNumber(intval as Numeric)
```

Parameters:

intval	Smallint or Integer class object.
--------	-----------------------------------

If the object is a Smallint it will occupy 2 bytes. If it is an Integer it will occupy 4 bytes.

If the object it receives as a parameter is a decimal, it will not store anything.

The format in which it will be stored will be that of the processor (Little-endian on compatible Intel x86 processors, that is, the least significant byte is stored in the lowest memory address and the most significant byte in the highest).

#### 2.1.2 SHA256 method

This method allows generating a SHA256 hash of the data contained in the Binary object. The hash will return it in a 64-character alphanumeric string that will represent 256 bits (32 bytes) in hexadecimal notation.

Syntax:

```
SHA256(VAR string as Char)
```

Parameters:

String	Variable of type char passed by reference where the SHA256 hash will be returned.
--------	---

### 2.2 Char class

#### 2.2.1 GetIntegerFromCharHex method

This method returns an Integer from its hexadecimal representation stored in a Char object.

Syntax:

```
GetIntegerFromCharHex() return integer
```

### 2.2.2 GetSmallintFromCharHex method

This method returns a Smallint from its hexadecimal representation stored in a Char object.

Syntax:

```
GetSmallintFromCharHex() return Smallint
```

## 2.3 Control class

### 2.3.1 ExportImageEx method

This method saves the control to an image file.

Unlike the ExportImage method, this method will export the image associated with the control without the control being in the foreground.

Syntax:

```
ExportImageEx(filePath as Char ,imageType as Char default NULL ,properties as Char default NULL)
```

Parameters:

filePath	Path where the image will be saved.
imageType	Image format. Possible values: BMP and JPG.
Properties	This parameter can have the following values (separated by a semicolon “;”): <ul style="list-style-type: none"><li>• MONOCHROME: The image is generated in black and white.</li><li>• QUALITY=n: The quality factor n can be a value from 0 to 100. This property is for the JPEG format.</li><li>• ZOOM=n: The image is saved with the magnification indicated in n.</li></ul>

This method cannot be used on UserControls.

## 2.4 Json class

### 2.4.1 GetChildList method

This method returns a string with all the elements of a node of the JSON object. The names of the elements will be separated by the character «|». This method can be used to traverse the structure of a JSON object without first knowing the name of its properties.

Syntax:

```
GetChildList() return char
```

## 2.5 Module class

### 2.5.1 GetClipboardTextUTF8 method

This method allows you to get the clipboard content in utf-8 format.

Syntax:

```
GetClipboardTextUTF8() return char
```

### 2.5.2 LoadJVM method

This method allows you to load a Java virtual machine with a specific CLASSPATH.

The loaded virtual machine will be the one indicated in the COSMOSUSEJAVAVERSION and COSMOSUSELASTJAVAVERSION variables. If these variables are not defined, the virtual machine indicated in the system's PATH variable will be loaded.

Syntax:

```
LoadJVM(classpath as Char ,loadJarsFolder as Boolean ,loadJavaClasspathFile as Boolean) return integer
```

Parameters:

Classpath	Full path of the jar files.
loadJarsFoder	Boolean value that indicates whether the jar files located in the jars directory of the project should be added to the classpath.
loadJavaClasspathFile	Boolean value that indicates whether the jar files indicated in the java.options file should be added to the classpath.

Returns:

- 0. If there is no error.
- -1. If there is already a Java virtual machine loaded for the running process.
- -2. Java virtual machine could not be loaded.

## 2.6 Numeric class

### 2.6.1 ToCharHex method

This method allows you to obtain the hexadecimal representation of a number. The value is returned in a Char object.

Syntax:

```
ToCharHex() return Char
```

If it is executed on a Smallint, it returns a String of 4 characters, 2 for each byte, with the hexa-decimal representation of the number.

If it is executed on an Integer, it returns a String of 8 characters, 2 for each byte, with the hexa-decimal representation of the number.

If executed on a Decimal, it returns a String of at least 8 characters, 2 for each byte, with the hexadecimal representation of the number. If the number has decimals, it will return the decimal part and the integer part separated by a point.

If the number is negative, it returns the hexadecimal string as 2's complement.

## 2.7 TTCHARTDLL

### 2.7.1 ChartExportImage function

This function allows you to save the chart in an image file.

Declaration:

```
public dll "ttchartdll.dll" ChartExportImage(hWindow as integer, outFile as char, fileFormat as char) return boolean
```

Parameters:

hWindow	Window identifier.
outFile	Image file path.
FileFormat	Image format. Possible values: BMP and JPG.

## 2.8 API COSSIGNFILE

### 2.8.1 CosSignFileSignXAdES\_Ex

This function will sign the XML file passed as a parameter using the XAdES format.

Declaration:

```
CosSignFileSignXAdES_Ex (  
    signID as integer,  
    inFileFullPath as char,  
    outFileFullPath as char,  
    keystorePath as char,  
    keystoreFormat as char,  
    keyStorePasswd as char,  
    alias as char,  
    aliasPasswd as char,  
    algorithm as char,  
    format as char,  
    mode as char,  
    extraParams as char)  
returnInteger
```

Parameters:

signID	Signature identifier returned in the call to the CosSignFileCreateSigner function.
inPDFPath	Absolute path of the XML file to be signed.

outPDFPath	Absolute path of the output XML file resulting from the signing process of the XML file indicated in the "inPDFPath" parameter.
keystorePath	Absolute path of the certificate store file where the certificate that you want to use to carry out the signing process is stored.
keystoreFormat	Character string that indicates the format of the certificate store. The possible values of this parameter can be obtained by invoking the CosSignFileGetKeystoreFormats function (jks, pkcs12, etc.).
keystorePasswd	Password for the certificate store specified in the "keystorePath" parameter.
alias	Alias of the electronic certificate that you want to use to sign the file.
aliasPasswd	Password of the electronic certificate specified in the "alias" parameter.
algorithm	The encryption algorithm to use. The list of supported algorithms can be consulted by executing the CosSignFileGetSignAlgorithms function (SHA1withRSA, SHA256withRSA, etc.).
format	Signature format. The list of supported signature formats can be consulted by executing the CosSignFileGetPAdESSignFormats function.
mode	Signature mode. The list of supported signature modes can be consulted by executing the CosSignFileGetSignModes function (implicit, explicit, etc.).
extraParams	List of additional parameters. Possible values are: policyIdentifier, policyIdentifierHash, policyIdentifierHashAlgorithm, policyDescription, and policyQualifier.

The format is: *key:value*

```
extraParams = "policyIdentifier":"X.XX.XXX.X.X.X.X.X.X.X"
+ ";"
+ "policyIdentifierHash":"V81VVNGDCPen6VELRD1Ja8HARFk="
+ ";"
+ "policyIdentifierHashAlgorithm":"SHA-256"
+ ";"
+ "policyDescription":"Electronic sign"
+ ";"
+ "policyQualifier":"http://webpage";
```

## 3. Bug fixes

---

### 3.1 Runtime

- GetClipboardText method. When executing this method, the operating system clipboard was disabled as long as the *cosrun.exe* process did not terminate.
- Print to PDF. The variable control was not printing vertically.
- Allowcolumnheaderfilter method. Lists of type Sql.
  - With more than one table, the filter was not applied if the name of the column that this method received as a parameter was present in more than one table.
  - If the statement contained the *limit* clause, when filtering by the header the statement that Cosmos generated was not correct and no record was returned.
  - If the list had many records, filtering by a column gave a general protection error and the application stopped working.
- Export to PDF. It did not correctly export a list that used the Font Bancaixa128Ch.
- LoadFromFile method of the Char class. If the file to be loaded did not exist or it was not possible to load it, assign "" (quotation marks) to the variable instead of assigning it NULL.
- GetWindowsVersion method. In the case of Windows 10, the value did not return correctly.
- Execution of Java functions / methods. The exception buffer was not being properly cleared.
- The ExportToExcel Method always returned null instead of a Boolean.

### 3.2 Repository

- When converting a repository to XML and importing it into CRF, the columns of type enumerated or boolean were not created correctly. In the CRF format file, although the values were stored correctly, it did not store the number of values. This caused the compiler to not work correctly.
- Exporting a repository from CRF to XML did not export the "VALUES" and "TAGS" when the type was marked as "Generic".

### 3.3 Code editor

- Code Insight.
  - JSON class. By typing the character "." after a method of the JSON class it did not show the methods and / or properties that the return class exports.
  - MENU class. By typing the character "." after the Option method of the Menu class, a general protection error occurred in the code section of the class and the execution of the "Cosmos Visual Editor" application stopped.

### 3.4 APIS.

- COSSINGFILE. It did not sign correctly in XAdEs format.