



MultiBase Cosmos

Notes to version 7.8.3

BASE100

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

Index

1. IMPLEMENTATIONS.....	3
1.1 COSMOS WEBSERVER. CHARACTER ENCODING.....	3
1.2 CTSQL	3
1.3 EDIT FIELD CONTROL	4
1.4 SKIN	4
1.5 LOG4J.....	4
2. NEW METHOD	5
3. ENVIRONMENT VARIABLES.....	6
4. BUG FIXES	7
4.1 RUNTIME.....	7
4.2 CTSQL	7

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

NTCO783 V1

1. Implementations

1.1 Cosmos Webserver. Character encoding.

Two new properties have been implemented: `parametersCharset` and `returnCharset`.

Why is this implementation necessary?

In version 7.8 and earlier, Cosmos converted the character string sent in the request parameter to ANSI with the local code page. This caused that, if there was no correspondence between the characters of the request and those of the local ANSI encoding of the computer where Cosmos Webserver is running, the conversion and graphic representation of these would not be correct.

How it will work from versions after 7.8

By default it will work as in previous versions.

If it is necessary to work with a character set other than the ANSI of the local machine, the `parametersCharset` and `returnCharset` properties must be defined in the Cosmos Web service configuration xml file. By defining this property, Cosmos will convert to the charset indicated in `parametersCharset`, and the character strings sent in the response will use the character set indicated in the `returnCharset` property, defaulting to ANSI with the local code page.

Properties `parametersCharset` and `returnCharset`

Possible values are: UTF-8 or a number indicating the ANSI code page.

They are defined in the service configuration xml file at the method level.

`parametersCharset`. Character encoding of the parameter received by the Cosmos function that implements the request. If not defined, it defaults to ANSI with the local code page.

`returnCharset`. Character encoding of the return value of the Cosmos function that implements the request. If not defined, the default value is ANSI with local code page.

Example.

If a JSON with UTF-8 encoding with Cyrillic text is sent in the body of the request and the Cosmos service wants to work in ANSI, the value that must be indicated in the `parametersCharset` property must be «1251», which is the corresponding to ANSI Cyrillic encoding. If you want to work in UTF-8, the value to be indicated in the `parametersCharset` property must be «UTF-8».

If the response is to be sent in UTF-8, the value of the `returnCharset` property must be «UTF-8». The Cosmos function that implements the request must be responsible for converting to the character set indicated in `returnCharset`.

1.2 CTSQL

The version of CTSQL that is distributed with this version of Cosmos is 3.6 0.53.

The possibility of passing a number to the COUNT clause has been implemented, so that *select count(1)* behaves like *select count(*)*.

1.3 Edit Field control

As of this version it is not allowed to copy the content of an EditField control if it has the «password» property activated.

1.4 Skin

A new graphic element has been created, SKIN_LIST_ITEM_HEADER_PART_TEXT_DISABLED, which allows indicating the color of the Label when the control is disabled.

1.5 Log4j

In this version of Cosmos the version of log4j has been updated to version 2.17. This version of log4j fixes the vulnerability issue detected in December 2021.

2. New method

No new methods have been implemented.

3. Environment variables

ALLOWSELECTREADONLYDROPEdit

This variable allows a DropEdit control with the ReadOnly property active to select an item from the list. However, it will not allow you to modify the content of the edit control.

This environment variable allows you to emulate the behavior of a DropList control in a DropEdit control.

NOEXPORTCOLUMNWIDTHZERO

This variable allows the columns of a list defined with width 0 not to be exported to the Excel file.

4. Bug Fixes

4.1 Runtime

- Cosmos Webserver. In some circumstances, an error occurred in the execution of the application and it stopped working.
- GetWhere method. An error would occur and the application would crash if there was any variable of the Form-Table class that contained the word «limit». This error occurs since version 7.8 of Cosmos.
- TOTALIZECOLUMN. It did not correctly count the number of rows when they had non-numeric values.
- COSMOSVISUALMODE. If an editing control is defined in CustomControls with «bottom» borders, hiding the running control with the Invisible=TRUE property did not refresh the screen and the bottom line of the control was drawn.
- Active X. In some circumstances the GetProperty method did not return the value correctly.
- Skin. If the control was disabled the control's label was painted twice.
- Sometimes the message "UnknownMessage in SendCtrMessage\n or Form is not open." was displayed.
- Remote desktop. A random runtime error would sometimes occur when Cosmos was running on a remote desktop.
- SetIcon method of the Form class. It does not release resources, and may cause a drawing error in the application forms.
- SetBoxRoundCornerRadio method of the SimpleControl class. As of version 7.6.1 this method did not work correctly.

4.2 CTSQL

- The *select count(column)* sentence returns an incorrect value in the following circumstances:
 - all values in the column were null.
 - non-null repeated values and at least one null value.