

Contents

- [Language & syntax differences](#)
- [Date comparison](#)
- [Date parsing](#)
- [Boolean values converted into a string](#)

Language & syntax differences

Here all deviations from CFML Standard Syntax as well as a different runtime behavior are described.

Date comparison

The CFML Standard determines, that a string comparison in "if" clauses always has to be checked, whether the two compared strings could be date values, (e.g. `<cfif "01/01/2000" EQ "01.01.00">`). Railo only checks if both strings are dates if on at least one side of a comparison there is a date value. This checking is done at compile time. (e.g. `<cfif query.lastAccess EQ "01.01.00">`). The reason for this is that only in rare cases the two strings contain date values. To check these strings for date values is very time consuming and only done if desired.

If you nevertheless want to check whether the two strings contain date values, you should use the function `parseDateTime` to translate a variable or a string into a date value. (e.g. `<cfif parseDateTime(strDate) EQ parseDateTime("01.01.00")>`)

Date parsing

Railo does not support implicit parsing of a date value in the locale format (e.g. `<cfif now() EQ "Wednesday, January 30, 2002 7:02:12 AM PST">`). In order to achieve this you have to use the function `parseDateTime` (`<cfif now() EQ parseDateTime("Wednesday, January 30, 2002 7:02:12 AM PST")>`).

Again the reason for this is, that implicit parsing of strings to locale date formats is very time consuming.

Boolean values converted into a string

If you convert a boolean value into a string Railo generates the following values:

- true into "true"
- false into "false"

In the CFML standard a boolean will be converted into the above values only in compile time. At runtime boolean values are converted as follows:

- true into "Yes"
- false into "No"

Example:

```
#true#
```

Output Railo and CFML Standard: *true*

```
#1 EQ 1#
```

Output Railo: *true*

Output CFML Standard: *Yes*