



WebTricks Custom T-SQL Functions

String Functions

Math Functions

**Bringing the power and ease of ColdFusion's
built-in functions to SQL Server**

Version 1.0

User/Developer Guide

Table of Contents

About WebTricks Custom T-SQL Functions	3
T-SQL	3
About Custom/User-Defined T-SQL Functions	3
WebTricks Functions	3
System Requirements.....	4
Compatibility	4
Terms of Use.....	4
About WebTricks.com.....	4
Using Custom Functions.....	4
Creating Functions	4
Calling Functions	4
String Functions	5
CJustify	5
DecimalFormat	6
DollarFormat.....	6
HTMLCodeFormat	7
HTMLEditFormat	7
LJustify	8
RemoveChars.....	9
RJustify	9
Trim.....	10
YesNoFormat.....	10
Math Functions	11
BitAnd.....	11
BitNot.....	12
BitOr	12
BitXor	13
DecrementValue	13
IncrementValue.....	14
System Functions	14
GetFunctionList	15

About WebTricks Custom T-SQL Functions

This document is to familiarize you with the details of the WebTricks Custom T-SQL functions.

T-SQL

T-SQL (or Transact SQL) is the “flavor” or SQL used by Microsoft SQL Server. T-SQL is an extended version of standard SQL that contains many function calls that will only work with Microsoft SQL Server.

About Custom/User-Defined T-SQL Functions

In addition to the functions that come with Microsoft SQL Server, custom – or user-defined – SQL functions can be created to extend functionality, much the same way custom tags and user-defined functions do in ColdFusion.

WebTricks Functions

While T-SQL has many built-in functions, it does not provide the same formatting capabilities of ColdFusion’s built-in functions with which many ColdFusion developers have become familiar. Unless you have an advanced knowledge of T-SQL, it may be difficult to replicate the desired functionality.

Familiar Function Names

The WebTricks Custom T-SQL functions not only replicate the functionality of ColdFusion, they also replicate the function name and usage of ColdFusion functions, making it easy to remember how to call the functions.

When To Use

While extremely useful, custom T-SQL functions do take more processing time than built-in function calls and can slow down your query processing if not used wisely. The best time to use custom T-SQL functions are as follows:

- The query results will be passed directly to an external resource (such as a Flash application or via a Web Server) and you cannot control formatting on the receiving end
- The query results need to be processed as is – without the use of additional ColdFusion formatting
- The function will aid in processing a trigger or stored procedure

Learning Tools

Many of the WebTricks Custom T-SQL functions contain advanced T-SQL code processing and they make an excellent learning tool when creating your own user-defined functions and stored procedures.

System Requirements

The custom functions will work on any Microsoft SQL Server 2000 database. The functions can be called directly in the database, or can be used from any programming language that makes a call to a Microsoft SQL Server 2000 database.

Compatibility

The functions have been tested with Microsoft SQL Server 2000 and were compared to ColdFusion functions on ColdFusion MX 7.

Terms of Use

You are permitted to modify or customize your copy of the functions to your liking; however you may not redistribute to other parties. You are not permitted to redistribute the functions unless explicitly authorized and licensed by WebTricks/Limited Reality LLC. You are not permitted to re-post or circulate the source code, or any derivative works, on the Internet, newsgroups, in emails or in any other manner.

About WebTricks.com

WebTricks.com is a tips and tricks site for web developers specializing in ColdFusion, JavaScript and SQL. It has been a popular ColdFusion resource since 1997. Please visit us at <http://www.webtricks.com>.

Using Custom Functions

Creating Functions

Each custom function contained in this pack has its own file named *function_name.sql*. The file can be opened within SQL Server's Query Analyzer and executed to create the function.

If at any time the script is modified and you wish to update existing function you can either DROP/delete the function or modify the script to use the ALTER statement as opposed to the CREATE statement.

Calling Functions

In T-SQL, custom functions must be called using a two-part name:

```
user_name.function_name()
```

Typically, "dbo" can be used as the *user_name* value when calling the function.

Like ColdFusion, arguments passed into a function must be done within the function call's parenthesis. Since T-SQL is a typed language, the same rules that apply to parameter values in regular SQL queries apply to function arguments:

```
dbo.function_name('string_value_date', date_value, numeric_value, 'string_value')
```

String Functions

The following functions are contained within the String Functions pack:

- CJustify
- DecimalFormat
- DollarFormat
- HTMLCodeFormat
- HTMLEditFormat
- LJustify
- RemoveChars
- RJustify
- Trim
- YesNoFormat

CJustify

Purpose

Used to return a string with the string value centered in a string of supplied length

Usage

CJustify(*string_value*, *numeric_value*)

Returns

String Value

The string value centered in a string of the specified length

If the specified length is less than or equal to the length of the specified string, the string will be returned

If the specified length is greater than the length of the specified string, the returned string will be padded with spaces on the left and the right

If the number of padded spaces is odd, the extra space will be added to the right

Example

String	Length	ColdFusion	SQL
ColdFusion	9	ColdFusion	ColdFusion
ColdFusion	10	ColdFusion	ColdFusion
ColdFusion	11	ColdFusion	ColdFusion
ColdFusion	12	ColdFusion	ColdFusion
ColdFusion	13	ColdFusion	ColdFusion
ColdFusion	14	ColdFusion	ColdFusion
ColdFusion	15	ColdFusion	ColdFusion

DecimalFormat

Purpose

Used to return a numeric value formatted with 2 decimal places

Usage

DecimalFormat(*float_value*)

Returns

Numeric Value

The value formatted with 2 decimal places

Example

Numeric Value	ColdFusion	SQL
0	0.00	0.00
1.5	1.50	1.50
-12	-12.00	-12.00
13.251	13.25	13.25
13.256	13.26	13.26
3.14159265359	3.14	3.14

DollarFormat

Purpose

Used to return a numeric value formatted as US currency, with 2 decimal places

Usage

DollarFormat(*float_value*)

Returns

String Value

The value formatted with 2 decimal places, preceded by \$

Example

Numeric Value	ColdFusion	SQL
0	\$0.00	\$0.00
1.5	\$1.50	\$1.50
-12	(\$12.00)	(\$12.00)
13.251	\$13.25	\$13.25
13.256	\$13.26	\$13.26

Numeric Value	ColdFusion	SQL
3.14159265359	\$3.14	\$3.14

HTMLCodeFormat

Purpose

Used to return an HTML-escaped string, enclosed in PRE tags

Usage

HTMLCodeFormat(*string_value*)

Returns

String Value

The HTML-escaped string, enclosed in PRE tags

Example

String	ColdFusion	SQL
This is some code. WebTricks.com Come visit us on the web!	<pre>This is some code.
 WebTricks.com

 Come visit us on the web!</pre>	<pre>This is some code.
 WebTricks.com

 Come visit us on the web!</pre>

HTMLEditFormat

Purpose

Used to return an HTML-escaped string

Usage

HTMLEditFormat(*string_value*)

Returns

String Value

The HTML-escaped string

Example

String	ColdFusion	SQL
This is some code. WebTricks.com Come visit us on the web!	This is some code. WebTricks.com Come visit us on the web!	This is some code. WebTricks.com Come visit us on the web!

LJustify

Purpose

Used to return a string with the string value left justified in a string of supplied length

Usage

LJustify(*string_value*, *numeric_value*)

Returns

String Value

The string value left justified in a string of the specified length

If the specified length is less than or equal to the length of the specified string, the string will be returned

If the specified length is greater than the length of the specified string, the returned string will be padded with spaces on the right

Example

String	Length	ColdFusion	SQL
ColdFusion	9	ColdFusion	ColdFusion
ColdFusion	10	ColdFusion	ColdFusion
ColdFusion	11	ColdFusion	ColdFusion
ColdFusion	12	ColdFusion	ColdFusion
ColdFusion	13	ColdFusion	ColdFusion
ColdFusion	14	ColdFusion	ColdFusion
ColdFusion	15	ColdFusion	ColdFusion

RemoveChars

Purpose

Used to remove characters from a string

Usage

`RemoveChars(string_value, numeric_value_start, numeric_value_count)`

Returns

String Value

The string value with the count of characters removed from the start position
If no characters are found, null is returned

Differences from ColdFusion

Will return null value as opposed to error if invalid start and count parameters are passed in

Example

String	Start	Count	ColdFusion	SQL
ColdFusion	1	2	ldFusion	ldFusion
ColdFusion	2	7	Con	Con
ColdFusion	3	3	Cousion	Cousion
ColdFusion	5	3	Coldion	Coldion
ColdFusion	7	1	ColdFuion	ColdFuion

RJustify

Purpose

Used to return a string with the string value right justified in a string of supplied length

Usage

`RJustify(string_value, numeric_value)`

Returns

String Value

The string value right justified in a string of the specified length
If the specified length is less than or equal to the length of the specified string, the string will be returned
If the specified length is greater than the length of the specified string, the returned string will be padded with spaces on the left

Example

String	Length	ColdFusion	SQL
ColdFusion	9	ColdFusion	ColdFusion
ColdFusion	10	ColdFusion	ColdFusion
ColdFusion	11	ColdFusion	ColdFusion
ColdFusion	12	ColdFusion	ColdFusion
ColdFusion	13	ColdFusion	ColdFusion
ColdFusion	14	ColdFusion	ColdFusion
ColdFusion	15	ColdFusion	ColdFusion

Trim

Purpose

Used to return the string trimmed of both leading and trailing spaces

Usage

Trim(*string_value*)

Returns

String Value

The trimmed value

Example

String	ColdFusion	SQL
ColdFusion	ColdFusion	ColdFusion
ColdFusion	ColdFusion	ColdFusion
ColdFusion	ColdFusion	ColdFusion

YesNoFormat

Purpose

Used to return a yes/no value to a supplied string

Usage

YesNoFormat(*string_value*)

Returns

String Value

"Yes" if numeric and not 0, "yes" or "true"

"No" if numeric and 0, "no", "false" or any non-boolean value

Example

Test Value	ColdFusion	SQL
0	No	No
1	Yes	Yes
true	Yes	Yes
false	No	No
yes	Yes	Yes
no	No	No
-1	Yes	Yes
-2	Yes	Yes
1.5	Yes	Yes

Math Functions

The following functions are contained within the Math Functions pack:

- BitAnd
- BitNot
- BitOr
- BitXor
- DecrementValue
- IncrementValue

BitAnd

Purpose

Used to perform a bitwise logical AND operation

Usage

`BitAnd(numeric_value1, numeric_value2)`

Returns

Integer Value

The bitwise AND of two long integers

Example

Value 1	Value 2	ColdFusion	SQL
1	2	0	0
255	255	255	255
0	255	0	0

Value 1	Value 2	ColdFusion	SQL
255	0	0	0
125	1	1	1

BitNot

Purpose

Used to perform a bitwise logical NOT operation

Usage

BitNot(*numeric_value*)

Returns

Integer Value

The bitwise NOT of a long integer

Example

Value	ColdFusion	SQL
1	-2	-2
255	-256	-256
0	-1	-1
255	-256	-256
125	-126	-126

BitOr

Purpose

Used to perform a bitwise logical OR operation

Usage

BitOr(*numeric_value1*, *numeric_value2*)

Returns

Integer Value

The bitwise OR of two long integers

Example

Value 1	Value 2	ColdFusion	SQL
1	2	3	3
255	255	255	255

Value 1	Value 2	ColdFusion	SQL
0	255	255	255
255	0	255	255
125	1	125	125

BitXor

Purpose

Used to perform a bitwise logical XOR operation

Usage

`BitXor(numeric_value1, numeric_value2)`

Returns

Integer Value

The bitwise XOR of two long integers

Example

Value 1	Value 2	ColdFusion	SQL
1	2	3	3
255	255	0	0
0	255	255	255
255	0	255	255
125	1	124	124

DecrementValue

Purpose

Used to subtract one from the integer value

Usage

`DecrementValue(numeric_value)`

Returns

Integer Value

The computed value

Example

Value	ColdFusion	SQL
9	8	8

Value	ColdFusion	SQL
9.75	8	8
10.5	9	9
11.25	10	10
12	11	11
12.75	11	11
13.5	12	12
14.25	13	13
15	14	14

IncrementValue

Purpose

Used to add one to the integer value

Usage

IncrementValue(*numeric_value*)

Returns

Integer Value

The computed value

Example

Value	ColdFusion	SQL
9	10	10
9.75	10	10
10.5	11	11
11.25	12	12
12	13	13
12.75	13	13
13.5	14	14
14.25	15	15
15	16	16

System Functions

The following functions are contained within the System Functions pack:

- GetFunctionList

GetFunctionList

Purpose

Used to return a list of the custom functions created in the database

Usage

GetFunctionList()

Returns

String Value

List of all custom functions in the database

Differences from ColdFusion

Returns a list as opposed to a structure

Returns custom SQL functions

Example

SQL

BitAnd, BitNot, BitOr, BitXor, CJustify, CreateDate, CreateDateTime, CreateODBCDate, CreateTime, DateCompare, DateFormat, DayOfWeek, DayOfWeekAsString, DayOfYear, DaysInMonth, DaysInYear, DecimalFormat, DecrementValue, DollarFormat, fDateFormatShortDate, FirstDayOfMonth, GetFunctionList, Hour, HTMLCodeFormat, HTMLEditFormat, IncrementValue, IsLeapYear, ListFirst, ListLast, ListLen, LJustify, Minute, MonthAsString, ParagraphFormat, ParseDateTime, ParseDatTime, Quarter, RemoveChars, RJustify, Second, TimeFormat, Trim, Week, YesNoFormat
