
IgniteFusion Reference Guide

IgniteFusion Server 2.3r

Copyright 2002, All Rights Reserved

IgniteFusion Reference Guide
Version 2.3r

IgniteFusion Reference Guide.....	1
1. Introduction.....	5
2. Definitions.....	6
Ignite Tag.....	7
Ignite File.....	7
Tag Based Script.....	7
Variable.....	7
3. Getting Help, Technical Support.....	9
4. Programming tags	10
ign:break	10
ign:call	11
ign:case	12
ign:catch.....	13
ign:class	14
ign:continue.....	16
ign:if.....	17
ign:else.....	18
ign:else_if.....	19
ign:if_exists.....	20
ign:if_not_exists.....	21
ign:exit	22
ign:for.....	23
ign:function	24
ign:goto	25
ign:include.....	26
ign:param	27
ign:label	28
ign:noprocess	29
ign:split	30
ign:return.....	31
ign:set.....	32
ign:switch	33
ign:throw.....	34
ign:try.....	35
ign:while	36
5. Directory Access	37
ign:dir.....	37
ign:dir_create	38
ign:dir_delete	39
6. File Access	40
ign:file_read	40
ign:file_write	41
ign:file_move	43
ign:file_upload.....	44
ign:file_delete	46
ign:file_copy	47
7. Graphics	48
ign:gfx_make	48
ign:gfx_line	50
ign:gfx_line_to.....	51
ign:gfx_move_to	52

IgniteFusion Reference Guide

Version 2.3r

ign:gfx_circle	53
ign:gfx_arc	54
ign:gfx_rect	56
ign:gfx_text	57
ign:gfx_display	58
ign:gfx_view	59
8. E-Mail Tags	60
ign:email	60
9. Database Tags	62
ign:db_open	62
ign:sql	63
ign:sql_run	64
ign:db_datasources	65
ign:db_fields	66
ign:db_sqldrivers	67
ign:db_tables	68
10. OCX	69
ign:object	69
11. Cookies	70
ign:cookie	70
12. Searching	71
ign:search	71
13. Internet Communication	73
ign:http_io	73
14. Global Objects	74
_Param	74
_Cookie	75
15. Global Functions	76
Asc()	76
Char()	78
FileExists()	79
Find()	80
FormatNum()	81
GetScriptURL()	82
GetScriptName()	83
GetScriptDir()	84
GetMainScriptName()	85
GetMainScriptDir()	86
GetServerName()	87
Mid()	88
GetServerVar()	89
Len()	90
GetWordCount()	91
GetWord()	92
Lower()	93
Replace()	94
Sleep()	95
Trim()	96
TrimLeft()	97
TrimRight()	98
Upper()	99
GetAsString()	100
GetAsInt()	101
GetAsFloat()	102
IsNum()	103

IgniteFusion Reference Guide

Version 2.3r

IsAlpha()	104
IsSpace()	105
IsDate()	106
16. INI File Functions	107
IniFileWrite()	107
IniFileRead()	108
17. Date Functions	109
GetDate()	109
FormatDateTime()	110
AddDays()	111
AddSeconds()	112
AddMinutes()	113
AddHours()	114
AddWeeks()	115
GetTime()	117
18. DataBase object functions	118
MoveNext()	118
MovePrev()	119
MoveFirst()	120
MoveLast()	121
Move()	122
Close()	123
FieldCount()	124
FieldName()	125
FieldType()	126
FieldTypeName()	127
FieldLen()	128
FieldScale()	129
IsFieldNull()	130
FieldData()	131
FieldDataByName()	132
FieldIndexByName()	133
19. TIFF image functions	134
TiffImageCount()	134
TiffAppend()	135
TiffExtract()	136
20. Number Functions	137
Abs()	137
DecimalFormat()	138
Sin()	139
Cos()	140
Tan()	141
Sqrt()	142
Pi()	143
Log()	144
Log10()	145
OctalFormat()	146
HexFormat()	147
Power()	148
Round()	149
Trunc()	150
Rand()	151
21. Bit Functions	152
BitOR(x,y)	152
BitAnd(x,y)	153

IgniteFusion Reference Guide

Version 2.3r

BitXOr(x,y).....	154
BitShift(x,y)	155
BitNot(x).....	156
22. String Functions	157
CharAt().....	158
Len()	159
Compare()	160
CompareNoCase().....	161
Find()	162
FindNoCase().....	163
ReverseFind().....	164
ReverseFindNoCase()	165
FindOneOf().....	166
FindWord()	167
GetWord().....	168
GetWordCount()	169
InsertChars()	170
RemoveChars()	171
Upper().....	172
Lower()	173
Left().....	174
Right().....	175
TrimRight().....	176
TrimLeft()	177
Trim().....	178
Mid().....	179
RepeatString().....	180
Replace().....	181
Reverse().....	182
UUEncode().....	183
UUDecode().....	184
HTMLEncode()	185
DBEncode()	186
23. IGNITE SQL Reference.....	187
Adding Records	187
Deleting Records.....	188
Retrieving Data	189
ASCII Chart	190
Ignite Tag Index	191

1. Introduction

IgniteFusion Server is full programming language, by extending HTML, provides a powerful and easy set of tag commands enabling the creation of websites capable of generating and processing information on the fly.

This guide provides the developer with specific reference information on ***IgniteFusion*** tags used in developing ***IgniteFusion script files***.

2. Definitions

This section gives some basic definitions associated with the use of *IgniteFusion*.

IgniteFusion Reference Guide

Version 2.3r

Ignite Tag

An **Ignite** tag is an HTML-like tag that while delivering more power and functionality, have the same "look and feel" as HTML tags. Like HTML, Ignite tags are surrounded by angle brackets and are composed of the tag name followed by tag attributes and values associated with those attributes.

<ign:tag attribute1="value1" attribute2="value2">

Values assigned to Ignite tag attributes should appear in double quotes. The only exception to this involves values assigned to attributes under the **ign:set** tag. (Please go to reference below for further details on the **ign:set** tag and on syntax for other specific tags.)

Note: Attributes of tags that have a double quote are used as strings, however attributes that are enclosed in single quotes are evaluated before being used.

As with HTML tags, many **Ignite** tags occur in pairs – a tag pair being composed of the tag and an associated end tag.

Ignite IF tag: **<ign:IF>**
Ignite end IF tag: **</ign:IF>**

The **Ignite** tag is the basic unit used in developing websites with the **Ignite Server**. **Ignite** tags are executed by the **Ignite Server**.

Ignite File

Just as an HTML file is a file with a .htm or .html extension, an **Ignite** file is any file with a **.ign** extension. Ignite files usually contain both **Ignite** and HTML tags and are used to implement the setup, maintenance and content management functions for a website.

Tag Based Script

A tag based script is an **Ignite** file that constitutes a script or set of instructions for performing website functions.

Variable

A **variable** in an **Ignite** file functions as a place holder or storage area for data or information that is not "hard coded" into the file. Information may be supplied to a variable from a number of sources. For example, the information may be taken from a database, may be the product of a calculation, may be entered by the user or may be taken from information entered in the **submit** field of an online form.

Variables may be created by using the **ign:set** tag, as in the following examples:

<ign:set MyVariable="whatever">

Variables can be passed in to the pages through the URL. This can be done by setting the variable to the NAME attribute of an HTML **<INPUT>** tag.

<INPUT TYPE=text NAME="MyVariable" VALUE="#MyVariable#">

IgniteFusion Reference Guide

Version 2.3r

The contents of a **variable** may be displayed by placing pound signs (#) before and after the **variable** name.

#MyVariable#

3. Getting Help, Technical Support

developers and users are directed to the **Customer Support** section of the **Ignite** website

www.IgniteFusion.com for access to FAQs and further information on *Ignite*.

4. Programming tags

ign:break

Description:

The **<ign:BREAK>** tag is used to jump out of a loop statement, such as a ign:FOR loop.

Syntax:

<ign:BREAK>

Example:

```
<ign:FOR FROM=5 TO=10 INDEX="Index1">  
  <ign:IF Index1 == 3>  
    <ign:BREAK>  
  </ign:IF>  
  #Index1#<br>  
</ign:FOR>
```

ign:call

Description:

The **<ign:CALL>** tag is used to call a user defined function.

Syntax:

simple function call

```
<ign:CALL MyFunction( )>
```

function call with return value

```
<ign:CALL retval = MyFunction( )>
```

with arguments

```
<ign:CALL MyFunction( aa,bb,cc )>
```

Example:

```
<ign:CALL MyFunc( num, rvalue ) >
```

```
<ign:FUNCTION MyFunc( aa, var rvalue ) >
```

```
    <ign:IF aa == 0>
```

```
        <ign:RETURN>
```

```
    <ign:ELSE>
```

```
        <ign:set rvalue = rvalue + aa>
```

```
        <ign:CALL MyFunc( aa -1, rvalue ) >
```

```
    </ign:IF>
```

```
</ign:FUNCTION>
```

```
<ign:CALL num = MyNum() >
```

```
<ign:FUNCTION MyNum() >
```

```
    <ign:RETURN 42>
```

```
</ign:FUNCTION>
```

See Also:

<ign:FUNCTION>, **<ign:RETURN>**

ign:case

Description:

The **<ign:CASE>** tag is used together with the **<ign:SWITCH>** to check the condition for executing code. Multiple cases are separated by a comma. A case without a condition is used as the default case.

Syntax:

```
<ign:SWITCH expression>
  <ign:CASE expression1, expression2, expression3>
    ...code...
  <ign:CASE >
    for all default cases not found
```

</ign:SWITCH>

Example:

```
<ign:switch "SUN">
  <ign:case "FRI">
    Friday
  <ign:case "SAT" ,"SUN">
    Weekend
  <ign:case>
    A weekday
</ign:switch>
```

Output:

Sunday

See Also:

<ign:SWITCH>

ign:catch

Description:

The **<ign:CATCH>** tag is used together with **<ign:TRY>** tag for trapping and processing errors. When an error occurs the system jumps to the position of the catch statement for the respective error type. The system then executes the code inside the catch block.

Syntax:

```
<ign:CATCH TYPE="ErrType" >
```

Standard Error types:

```
"Application"  
"File"  
"Database"  
"Security"
```

Example:

```
<ign:function OnError(>  
    <b>OnError</b><br>  
    Error:#GetError()#</b><br>  
    #GetErrorNum()#<br>  
    #GetErrorType()#<br>  
</ign:function>  
  
<ign:try>  
    <ign:SQL_RUN name="get_info">  
        #get_info.DBFIELD#  
    </ign:SQL_RUN >  
  
    <ign:catch Type="database">  
        oops i found a database error  
    </ign:catch>  
    <ign:catch>  
        oops i found an error  
    </ign:catch>  
</ign:try>
```

See Also:

<ign:TRY>, **<ign:THROW>**

ign:class

Description:

The **<ign:CLASS>** tag is used to define a set of functions and variables that are related. After defining a class an object variable can be created by setting a variable to the class object using the “**new**” keyword.

Syntax:

```
<ign:CLASS NAME="MyClass" EXTENDS = "MyParentClass">  
... code ...  
</ign:CLASS>
```

Example:

```
<ign:class Name="Numbers">  
  Hello1<br>  
  <ign:set m_data[1] = "Uno">  
  <ign:set m_data[2] = "Dos">  
  <ign:set m_data[3] = "Tres">  
  
  <ign:function GetName( n)>  
    <ign:set Name = m_data[n]>  
    <ign:return Name>  
  </ign:function>  
  <ign:function MaxNumber( )>  
    <ign:set len = m_data.Len()>  
    <ign:return len>  
  </ign:function>  
</ign:class>  
  
<ign:class Name="SpecialNumbers" extends="Numbers">  
  Hello2<br>  
  <ign:function GetThree()>  
    <ign:return GetName(3)>  
  </ign:function>  
</ign:class>  
  
<b>Report List</b><br>  
<ign:set Ob=new("SpecialNumbers")>  
  
<ign:set num = Ob.GetThree()>  
num: #num#<br>
```

Output:

```
Hello1  
Hello2  
num: Tres
```


ign:continue

Description:

The **<ign:CONTINUE>** tag is used to loop to next iteration of a loop statement, such as a ign:FOR loop.

Syntax:

<ign:CONTINUE>

Example:

```
<ign:FOR FROM=5 TO=10 INDEX="Index1">  
  <ign:IF Index1 == 3>  
    <ign:CONTINUE>  
  </ign:IF>  
  #Index1#<br>  
</ign:FOR>
```

ign:if

Description:

The **<ign:IF>** statement evaluates a condition and executes a block of code if the condition is true. Logical operators of AND, OR, and NOT can be used.

Comparison operators:

== – Equal to (double equal signs are used to compare values)

!= – Not Equal to (exclamation followed by an equal sign)

.LT. – Less than

.LE. – Less than or equal to

.GT. – Greater than

.GE. – Greater or equal to

Syntax:

<ign:IF *condition*

Perform this block of code...

</ign:IF>

Example:

```
<ign:IF (Temperature .LT. 65) OR ( Temperature .GT. 85)>
```

```
    The weather is nice
```

```
<ign:ELSE>
```

```
    #Temperature# degrees is unbearable
```

```
</ign:IF>
```

See Also:

<ign:ELSE>, **<ign:ELSE_IF>**, **<ign:IF_EXISTS>**

ign:else

Description:

The **ELSE** tag must be used in conjunction with **<ign:IF>**. The **<ign:ELSE>** statement is placed inside the **<ign:IF>** block, allowing for a second operation to be executed only if the **<ign:IF>** condition is false.

Syntax:

```
<ign:IF Condition>  
    If true Perform this block of code  
<ign:ELSE>  
    Otherwise Perform this block of code  
</ign:IF>
```

Example:

```
<ign:IF 100 == 50 * 2>  
    Correct  
<ign:ELSE>  
    Otherwise Perform this block of code  
</ign:IF>
```

See Also:

<ign:IF>, **<ign:IF_EXISTS>**, **<ign:FOR>**

ign:else_if

Description:

The **ELSE_IF** tag must be used with the **<ign:IF>** tag. The **<ign:ELSE_IF>** statement is placed inside the **<ign:IF>** block, allowing for a second operation to be tested only if the **<ign:IF>** condition is false.

Syntax:

```
<ign:IF Condition>  
    If true Perform this block of code  
<ign:ELSE_IF Condition2>  
    Perform this block of code  
<ign:ELSE>  
    Otherwise Perform this block of code  
</ign:IF>
```

See Also:

<ign:IF>, **<ign:ELSE>**, **<ign:IF_EXISTS>**

ign:if_exists

Description:

The **<ign:IF_EXISTS>** tag checks to see if a variable exists.

Syntax:

```
<ign:IF_EXISTS variable  
    Perform this block of code  
</ign:IF>
```

Example:

```
<ign:set MyVar = "Hello">  
  
<ign:IF_EXISTS MyVar>  
    MyVar contains:#MyVar#  
<ign:ELSE>  
    MyVar Does not Exist  
</ign:IF>
```

Output:

MyVar contains:Hello

See Also:

<ign:ELSE>, **<ign:IF>**

ign:if_not_exists

Description:

The **<ign:IF_NOT_EXISTS>** tag checks to see if a variable does not exist.

Syntax:

```
<ign:IF_NOT_EXISTS variable  
    Perform this block of code  
</ign:IF>
```

Example:

```
<ign:IF_NOT_EXISTS MyNemo>  
    MyNemo Does not Exist  
<ign:ELSE>  
    MyNemo contains:#MyNemo#  
</ign:IF>
```

Output:

MyNemo Does not Exist

See Also:

<ign:ELSE>, **<ign:IF>**

ign:exit

Description:

The <ign:exit> tag can be used to end a script. When the exit tag is found the system terminates execution of the script.

Syntax:

<ign:Exit>

Example:

```
<ign:param Name="MyParam" default="">
```

```
<ign:if MyParam == "">
```

```
    No data entered
```

```
    <ign:exit>
```

```
</ign:if>
```

Thank you for your input

ign:for

Description:

The **<ign:FOR>** allows developers to execute a loop of code. The code inside the FOR loop will be executed until the index reaches the TO value.

Syntax:

```
<ign:FOR FROM=StartNumber TO=EndNumber INDEX="IndexVariable" STEP=Increment>  
    Perform this block of code...  
</ign:FOR>
```

Attributes	Description	Required
FROM	Starting value	Y
TO	Stop value	Y
INDEX	Index variable to be incremented	Y
STEP	Amount to increment index by (1 is default)	N

Example:

```
<ign:FOR FROM=5 TO=10 INDEX="Index1">  
    #Index1#<br>  
</ign:FOR>
```

Output:

```
5  
6  
7  
8  
9  
10
```


ign:function

Description:

The **<ign:FUNCTION>** tag is used to create a user defined function. Any parameter name preceded with the VAR keyword can be modified in the function. All other variable will not be changed when the function is returned. The function will terminate when either a **<ign:RETURN>** or **</ign:FUNCTION>** is encountered.

Syntax:

```
<ign:FUNCTION MyFunction( aa, bb, VAR cc)>
```

```
</ign:FUNCTION >
```

Example:

```
<ign:FUNCTION MyFib( aa, var rvalue) >  
    <ign:IF aa == 0>  
        <ign:RETURN>  
    <ign:ELSE>  
        <ign:set rvalue = rvalue + aa>  
        <ign:CALL MyFib( aa -1, rvalue) >  
    </ign:IF>
```

```
</ign:FUNCTION>
```

```
<ign:set num=4>  
<ign:set rvalue=0>  
<ign:CALL MyFib( num, rvalue) >
```

Fib of:#num# is #rvalue#

Output:

Fib of:4 is 10

See Also:

<ign:CALL>, **<ign:RETURN>**

ign:goto

Description:

The **<ign:GOTO>** tag is used to jump to a given label whose position is defined by the **<ign:label>** tag.

Syntax:

```
<ign:GOTO NAME="MyLabel">
```

...This code is skipped....

```
<ign:LABEL NAME="MyLabel">
```

Example:

```
<ign:set X = 3>  
<ign:GOTO NAME="TestLabel">  
<ign:set X = 100>  
<ign:LABEL NAME="TestLabel">
```

X=#X#

Output:

X=3

See Also:

<ign:LABEL>

ign:include

Description:

The **<ign:INCLUDE>** tag is used to include other source files into the current script.

Syntax:

```
<ign:INCLUDE FILE="MyFile.txt">
```

Note: The file attribute can contain an absolute or relative path.

Relative paths that start with a backslash are relative to the webserver home directory (usually c:/Inetpub/wwwroot). If only a filename is given then the file is defined to be in the same directory as the current script file.

Example:

```
<ign:INCLUDE FILE="/MyFile.inc">
```

ign:param

Description:

The **<ign:PARAM>** tag declares a variable passed from a URL if it exists. It will use the default value if the variable does not exist in the URL. The “default” attribute is optional. However if no default is set and the variable does not exist in the URL, an error will be generated .

Syntax:

<ign:PARAM NAME="MyVar" Default="InitialValue" >

Attributes	Description	Required
NAME	Name of variable	Y
DEFAULT	Default value	N
ARRAY	Flag to indicate array values	N

Example:

```
<html>
  <head>
    <title>ign:param Sample</title>
  </head>
  <body>
    <ign:param name="PhoneNumber" default="none">

    <ign:if PhoneNumber=="none" or PhoneNumber==" ">
      <form action="">
        A required parameter has not been received.<br>
        Please enter a Phone Number:
        <input type="text" name="PhoneNumber" value="#PhoneNumber#">
        <input type="submit" value="Send">
      </form>
    <ign:else>
      Thank you. <br>
      Phone Number: #PhoneNumber#
    </ign:if>
  </body>
</html>
```

ign:label

Description:

The **<ign:LABEL>** tag is used to define a label that can be used as a target label in a goto statement

Syntax:

<ign:LABEL NAME="MyLabel">

Example:

```
<ign:set X = 3>  
<ign:GOTO NAME="TestLabel">  
<ign:set X = 100>  
<ign:LABEL NAME="TestLabel">
```

X=#X#

Output:

X=3

See Also:

<ign:GOTO>

ign:noprocess

Description:

The **<ign:NOPROCESS>** tag is used to stop the IgniteFusion script processor from interpreting code between the NPPROCESS tags.

Syntax:

```
<ign:NOPROCESS>  
....  
</ign:NOPROCESS>
```

See Also:

<ign:BREAK>

ign:split

Description:

The **<ign:SPLIT>** tag is used loop through a string word by word.

Syntax:

```
<ign:SPLIT LIST= DelimitedString DELIMITERS="SeparatorChars" INDEX="VariableName">  
    Day: #DayOfWeek#  
</ign:SPLIT>
```

Attributes	Description	Required
LIST	String to parse	Y
INDEX	Name of variable to hold word for each loop iteration	Y
DELIMITERS	String of character to split list up, defaults to comma	N

Example:

```
<ign:SPLIT LIST= "Mon;Tue Wed,Thu,Fri,Sat,Sun" DELIMITERS="; , " INDEX="DayOfWeek">  
    #DayOfWeek#  
</ign:SPLIT>
```

Output:

Mon Tue Wed Thu Fri Sat Sun

ign:return

Description:

The **<ign:RETURN>** tag is used to return from a user defined function. Return may also be used to return a value to the function caller.

Syntax:

<ign:RETURN *value***>**

Example:

```
<ign:function abc(>  
    <ign:return 100>  
</ign:function>
```

```
<ign:call x = abc(>
```

Returned value is #x#

Output:

Returned value is 100

ign:set

Description:

The **<ign:set>** tag is used to assign a value to a variable. If the variable does not exist then it is created by the tag, otherwise the existing value associated with the variable is overwritten. The **SET** command is also used to call functions and interact with objects.

Syntax:

<ign:set *variable = Expression***>**

Example:

```
<ign:set MyNum=(50*100) + 2>  
#MyNum#<br>
```

```
<ign:set MyText="ABC"&"DEF"&"GHI">  
#MyText#
```

Output:

```
102  
ABCDEFGHI
```

ign:switch

Description:

The **<ign:SWITCH>** tag is used together with the **<ign:CASE>** tag. Similar to the **ign:IF** and **ign:ELSE_IF** statement.

Syntax:

```
<ign:SWITCH expression  
    <ign:case expression2  
    <ign:case expression3, expression4  
    <ign:case  
</ign:SWITCH>
```

Example:

```
<ign:switch "SUN">  
    <ign:case "FRI">  
        Friday  
    <ign:case "SAT","SUN">  
        Weekend  
    <ign:case  
        A weekday  
</ign:switch>
```

Output:

Sunday

See Also:

<ign:CASE>

ign:throw

Description:

The **<ign:THROW>** tag is used to trigger an error event which will jump to the CATCH block defined within the TRY block

Syntax:

<ign:THROW TYPE=ErrorType MESSAGE=ErrorMessage>

Predefined error types:

"Application"
"File"
"Database"
"Security"
"User"

Example:

```
<ign:function OnError(>
    <b>OnError</b><br>
    Error:#GetError()#</b><br>
    #GetErrorNum()#<br>
    #GetErrorType()#<br>
</ign:function>

<ign:try>
    ...
    <ign:if my_error_condition>
        <ign:throw TYPE="User" Message="My Error">
    </ign:if>
    ...
    <ign:catch Type="database">
        oops I found a database error
    </ign:catch>
    <ign:catch>
        oops I found an error
    </ign:catch>
</ign:try>
```

See Also:

<ign:CATCH>, **<ign:TRY>**

ign:try

Description:

The **<ign:TRY>** tag is used to define an error handling block. All error encountered inside the TRY tag will jump to the CATCH block defined within the TRY block

Syntax:

```
<ign:TRY>
    ...
    <ign:CATCH>
        ...
    </ign:CATCH>
</ign:TRY>
```

Example:

```
<ign:function OnError()>
    <b>OnError</b><br>
    Error:#GetError()#</b><br>
    #GetErrorNum()#<br>
    #GetErrorType()#<br>
</ign:function>

<ign:try>
    <ign:SQL_RUN name="get_info">
        #get_info.DBFIELD#
    </ign:SQL_RUN >

    <ign:catch Type="database">
        oops i found a database error
    </ign:catch>
    <ign:catch>
        oops i found an error
    </ign:catch>
</ign:try>
```

See Also:

<ign:CATCH>, **<ign:THROW>**

ign:while

Description: The **<ign:WHILE>** tag allows developers to execute in a loop while a given condition is true.

Syntax:

<ign:WHILE *condition***>**

</ign: WHILE >

Example:

```
<ign:set num = 0>
<ign:WHILE (num .lt. 3) >
    <ign:set num = num + 1 >
    HELLO #num# <br>
</ign:WHILE>
```

Output:

```
HELLO 1
HELLO 2
HELLO 3
```

See Also:

<ign:FOR>, **<ign:BREAK>**

5. Directory Access

ign:dir

Description: The `<ign:DIR >` tag allows directory folder browsing.

Syntax:

```
<ign:DIR filter="c:\data\*.*)" index="MyFile">  
    #MyFile.isDir#,  
    #MyFile.isHidden#,  
    #MyFile.isSystem#,  
    #MyFile.isReadOnly#,  
    #MyFile.Name#<br>  
</ign:DIR >
```

ign:dir_create

Description: The **<ign:DIR_CREATE >** tag to creates a new directory folder.

Syntax:

<ign:DIR_CREATE NAME="*MyDir*" >

See Also:

<ign:DIR_DELETE>, **<ign:DIR>**

ign:dir_delete

Description: The **<ign:DIR_DELETE >** tag is used to delete an empty directory folder, the command will fail if the directory is not empty.

Syntax:

```
<ign:DIR_DELETE NAME="MyDir" >
```

See Also:

<ign:DIR_CREATE >, **<ign:DIR>**

6. File Access

ign:file_read

Description: The **< ign:FILE_READ >** tag reads the contents of a file located on the server, the content is then copied into the indicated variable.

Syntax:

<ign:FILE_READ FILE="ABC..TXT" VARIABLE="MyVar" ERROR="ErrString" ERRNO="ErrorNo" >

Attributes	Description	Required
FILE	File Name	Y
VARIABLE	Variable to copy file content to	Y
ERROR	Error string if and an error is encountered	N
ERRNO	Erro number returned if error is encountered	N

Note: The file attribute can contain an absolute or relative path.

Relative paths that start with a backslash are relative to the webserver home directory (usually c:/inetpub/wwwroot). If only a filename is given then the file must be in the same directory as the current script file.

Example:

```
<ign:FILE_READ FILE="test.dat" VARIABLE="MyData" ERROR="Err">
<ign:if Err != "">
    Problem Reading data: #Err#
    <ign:exit>
</ign:if>
```

The file content is: #MyData#

ign:file_write

Description:

The **<ign:FILE_WRITE >** tag writes the contents of a variable or data stream to a file.

There are two ways to use this tag to write to a file; by using the “data” attribute to assign a value, or by setting the “inside” flag to TRUE and inserting the data between the start and end tags.

Syntax:

Syntax of data defined by data attribute

```
<ign:FILE_WRITE FILE="ABC.TXT" DATA="Hello World" APPEND="Y"  
ERROR="MyErrString"  
ERRNO="MyErrorNo"  
>
```

syntax of data between start and end tags

```
<ign:FILE_WRITE FILE="ABC.TXT" INSIDE ="Y" ERROR="MyErrString" ERRNO="MyErrorNo" >  
Here is some data to insert into a file  
The data can consist of several lines including other tags  
</ign:FILE_WRITE >
```

Attributes	Description	Required
FILE	File Name	Y
DATA	Content or variable to write to file	N
APPEND	Flag to indicate overwrite or append to file (by default the file is overwritten)	N
INSIDE	Flag used to choose if data is defined in tag by data attribute or is sandwiched between start and end tag.	N
ERROR	Error string if and an error is encountered	N
ERRNO	Erro number returned if error is encountered	N

Note: The file attribute can contain an absolute or relative path.

Relative paths that start with a backslash are relative to the webserver home directory (usually c:/Inetpub/wwwroot). If only a filename is given then the file is defined to be in the same directory as the current script file.

Example:

IgniteFusion Reference Guide

Version 2.3r

```
<ign:FILE_WRITE FILE="ABC.TXT" DATA="Hello World" APPEND="Y"  
ERROR="MyErrString"  
ERRNO="MyErrorNo"  
>
```

```
<ign:FILE_WRITE FILE="ABC.TXT" INSIDE ="Y" ERROR="MyErrString" ERRNO="MyErrorNo" >  
Here is some data to insert into a file  
</ign:FILE_WRITE >
```

ign:file_move

Description:

The **<ign:FILE_MOVE>** tag allows files to be moved from one directory to another on the server.

Syntax:

<ign:FILE_MOVE FILE="C:\ABC\TEST.TXT" DEST_PATH="C:\DATA\ ">

Attributes	Description	Required
FILE	Source file name (including path)	Y
DEST_PATH	Destination path to copy file	Y
OVERWRITE	Y/N flag used to allow existing destination file to be overwritten (Default is "N")	N

ign:file_upload

Description: The <ign:FILE_UPLOAD> tag allows files to be uploaded to the server.

Syntax:

```
<ign:FILE_UPLOAD NAME="MyFile" DEST_PATH="C:\DATA\  
OVERWRITE="Y" UNIQUE="N">
```

Attributes	Description	Required
NAME	Variable name from calling html page	Y
DEST_PATH	Destination path to copy file	Y
OVERWRITE	Allow existing destination file to be overwritten	N
UNIQUE	Generates a random/unique destination file name	N

Notes: For uploading files, the HTML form tag must posted and encoded as a multipart data, such as;

```
<form action="xxx.IGN" method="post" enctype="multipart/form-data">
```

```
</form>
```

Returns the fields:

```
MyFile.DestPath()  
MyFile.DestName()  
MyFile.SrcName()  
MyFile.SrcPath()
```

Example:

```
<HTML>
<BODY>

<ign:param NAME="MyFile" DEFAULT="">

<ign:IF MyFile != "">
  <ign:FILE_UPLOAD NAME="MyFile"
    DEST_PATH="C:\DATA\" OVERWRITE="yes" UNIQUE="NO">
    <BR>
    Source Name:      #MyFile.SrcName()#<br>
    Source Path:      #MyFile.SrcPath()#<br>
    Data length:      #MyFile.Len()#<br>
    Destination Path: #MyFile.DestPath()#<br>
    Destination Name: #MyFile.DestName()#<br>
    <BR>
  </ign:IF>
  <ign:ELSE>
    <form action="" method="post" enctype="multipart/form-data">
      Select File <input type=file name="MyFile">
      <BR>
      <input type=submit>
    </form>
  </ign:IF>

</BODY>
</HTML>
```

ign:file_delete

Description:

The **<ign:FILE_DELETE>** tag deletes a file or files from the server.

Syntax:

<ign:FILE_DELETE FILE="*FileName*" >

Example:

<ign:FILE_DELETE FILE="C:\DATA\ABC.TXT" >

ign:file_copy

Description:

The **<ign:FILE_COPY>** tag copies a file on the server.

Syntax:

```
<ign:FILE_COPY NAME="C:\DATA\ABC.TXT" DEST_PATH="c:\xxx"  
OVERWRITE="Y"  
>
```


7. Graphics

ign:gfx_make

Description: The **<ign:GFX_MAKE>** tag is used to setup width and height values to create an image the end tag used to save all graphic commands and create the image file.

Syntax:

```
<ign:GFX_MAKE
  NAME="MyGraph"
  FILE="NewFileName"
  WIDTH=300 HEIGHT=300
  ORIGIN="BottomLeft"
  MAX_X="150.50"
  MAX_Y="150.50"
  MIN_X="-150.50"
  MIN_Y="-150.50"
  BGCOLOR="WHITE"
>
....
```

</ign:GFX_MAKE>

Attributes	Description	Required
NAME	Unique name used to identify the graphic	Y
FILE	File name of file to create (must be a GIF file)	Y
WIDTH	Image width in pixels (maximum 2048, default is 100)	N
HEIGHT	Image height in pixels (maximum 2048, default is 100)	N
ORIGIN	origin (TopLeft,TopRight,BottomLeft, BottomRight) default is BottomLeft	N
MAX_X	Maximum x value	N
MAX_Y	Maximum y value	N
MIN_X	Minimum x value	N
MIN_Y	Minimum y value	N
BGCOLOR	Background color (HTML color values)	N
BACKGROUND	Background Image (WARNING: does not support all gif types)	N

IgniteFusion Reference Guide

Version 2.3r

Example:

```
<ign:gfx_make name="mygraph" file="\tmp\temp.gif" width=100 height=100 bgcolor="white">  
  <ign:gfx_line x1=10 y1=0 x2=50 y2=50 color="cyan">  
    <ign:gfx_text x1=10 y1=10 size=1.5 value="My Graph Text" color="red">  
</ign:gfx_make>
```

Output:

ign:gfx_line

Description: The `<ign:GFX_LINE>` tag to draw a line.

Syntax:

```
<ign:GFX_LINE X1=10 Y1=10 X2=200 Y2=200 COLOR="#FF00FF" SCALE="Y">
```

Attributes	Description	Required
X1	Starting x position	Y
Y1	Starting y position	Y
X2	Ending x position	Y
Y2	Ending y position	Y
COLOR	Line color (HTML color values)	N
SCALE	Y/N flag to scale data to Min Max value (default Y)	N

Example:

```
<ign:gfx_make name="mygraph" file="\tmp\temp.gif" width=100 height=100 bgcolor="white">  
  <ign:gfx_line x1=10 y1=0 x2=50 y2=50 color="cyan">  
    <ign:gfx_text x1=10 y1=10 size=1.5 value="My Graph Text" color="red">  
</ign:gfx_make>
```

Output:

ign:gfx_line_to

Description: The **<ign:GFX_LINE_TO>** tag to draw a line from the current position to the given position.

Syntax:

<ign:GFX_LINE_TO X1=10 Y1=10 COLOR="#FF00FF" SCALE="Y">

Attributes	Description	Required
X1	Ending x position	Y
Y1	Ending y position	Y
COLOR	Line color (HTML color values)	N
SCALE	Y/N flag to scale data to Min Max value (default Y)	N

Example:

Output:

ign:gfx_move_to

Description: The <ign:GFX_MOVE_TO> tag to set the current draw position to the given position.

Syntax:

<ign:GFX_MOVE_TO X1=10 Y1=10 COLOR="#FF00FF" SCALE="Y">

Attributes	Description	Required
X1	Ending x position	Y
Y1	Ending y position	Y
COLOR	Set the default color (HTML color values)	N
SCALE	Y/N flag to scale data to Min Max value (default Y)	N

Example:

Output:

ign:gfx_circle

Description: The <ign:GFX_CIRCLE> tag to draw a circle.

Syntax:

```
<ign:GFX_CIRCLE  
  X1=xpos Y1=ypos  
  RADIUS=radius  
  COLOR="Color"  
  FILLED="Yes/No"  
  SCALE="Y"  
>
```

Attributes	Description	Required
X1	Center x position	Y
Y1	Center y position	Y
RADIUS	Length of radius	Y
COLOR	circle color (HTML color values)	N
FILLED	Yes/No Flag indicating to fill or not	N
SCALE	Y/N flag to scale data to Min Max value (default Y)	N

Example:

Output:

ign:gfx_arc

Description: The **<ign:GFX_ARC>** tag to draw an arc (a partial circle).

Syntax:

```
<ign:GFX_ARC
  X1=150
  Y1=150
  RADIUS=50
  FROM=0
  TO=180
  COLOR="AQUA"
  FILLED="Yes/No"
  SCALE="Y"
>
```

Attributes	Description	Required
X1	Center x position	Y
Y1	Center y position	Y
RADIUS	Length of radius	Y
FROM	Starting position in degrees. Degrees range from 0 to 360 clockwise starting on the left x-axis	Y
TO	Ending position in degrees	Y
COLOR	circle color (HTML color values)	N
FILLED	Yes/No Flag indicating to fill or not	N
SCALE	Y/N flag to scale data to Min Max value (default Y)	N

Example:

```
<ign:GFX_ARC
X1=50
Y1=50
RADIUS=50
FROM=0
TO=310
COLOR="AQUA"
FILLED="Yes"
>
```

Output:

ign:gfx_rect

Description: The <ign:GFX_RECT> tag is used to draw a rectangle.

Syntax:

```
<ign:GFX_RECT  
    X1=10 Y1=140  
    X2=50 Y2=200  
    COLOR="Color"  
    FILLED="Yes/No"  
    SCALE="Y"  
>
```

Attributes	Description	Required
X1	Top left x position	Y
Y1	Top left y position	Y
X2	Bottom right x position	Y
Y2	Bottom right y position	Y
COLOR	Line color (HTML color values)	N
FILLED	Yes/No Flag indicating to fill or not	N
SCALE	Y/N flag to scale data to Min Max value (default Y)	N

Example:

Output:

ign:gfx_text

Description: The <ign:GFX_TEXT> tag to draw text.

Syntax:

```
<ign:GFX_TEXT
    X1=xpos Y1=ypos
    VALUE="Text"
    COLOR="Color"
    SIZE=nSize
    ANGLE=nDegrees
    SCALE="Y"
>
```

Attributes	Description	Required
X1	Starting x position	Y
Y1	Starting y position	Y
VALUE	Text to display	Y
COLOR	Text color (HTML color values)	N
SIZE	Text size	N
ANGLE	Angle to write text line	N
SCALE	Y/N flag to scale data to Min Max value (default Y)	N

Example:

```
<IGN:GFX_MAKE NAME="MYGRAPH" FILE="\test.gif"
    WIDTH=400 HEIGHT=400 bgcolor="WHITE"
    max_x=600 max_y=600
    origin="BottomLeft"
>
    <ign:for from=45 to=360 step=45 index="idx">
        <IGN:GFX_TEXT X1=300 Y1=300 VALUE=" hello" size=2 angle=idx color="blue">
    </ign:for>

</IGN:GFX_MAKE >

```

Output:

ign:gfx_display

Description: The `<ign:GFX_DISPLAY>` tag to display an image.

Syntax:

`<ign:GFX_DISPLAY NAME="MyGraph" USEMAP="MyMap">`

Attributes	Description	Required
NAME	Unique name used by gfx_make to identify the graphic	Y
USEMAP	HTML style coordinates map	N

ign:gfx_view

Description: The **<ign:GFX_VIEW>** tag to set image viewport and reset image scaling.

Syntax:

<ign:GFX_VIEW X1=100 Y1=100 X2=200 Y2=200 SCALE="Y">

Attributes	Description	Required
X1	Top left x position	Y
Y1	Top left y position	Y
X2	Bottom right x position of new view	Y
Y2	Bottom right y position of new view	Y
SCALE	Y/N flag to scale X1,Y1,X2,Y2 (default Y)	N

8. E-Mail Tags

ign:email

Description: The <ign:EMAIL> tag is used to send email using internet SMTP protocol.

Syntax:

```
<ign:EMAIL
  TO="info@mycompany.com"
  FROM="userx@mycompany.com"
  SERVER="email.mycompany.com"
  USERNAME="userx"
  PASSWORD="secret"
  SUBJECT="My Test"
  MESSAGE="Hello This is a test message"
  ATTACHMENT= "c:\data\test.txt"
>
```

```
<ign:EMAIL
  TO="info@mycompany.com"
  FROM="userx@mycompany.com"
  SERVER="emailmycompany.com"
  USERNAME="userx"
  PASSWORD="secret"
  SUBJECT="My Test"
  INSIDE="Y"
  ATTACHMENT= "c:\data\test.txt"
>
Hello This is a test message
</ign:EMAIL>
```

Attributes	Description	Required
TO	Email address of recipient	Y
FROM	Email address of sender	Y
SERVER	Email server	Y
USERNAME	Login username for email server	N
PASSWORD	Login password for email server	N
MESSAGE	Email message	N
INSIDE	Flag to Indicate message embedded inside tags	N
SUBJECT	Subject title for email	N
ATTACHMENT	Filename of attachment	N
CC	Email address to send copy to	N
PORT	SMTP port number	N

9. Database Tags

ign:db_open

Description: This tag defines a database connection to be held open and re-used in future SQL calls. Use this command to optimize database access.

Syntax:

```
<ign:db_open  
    HANDLE="H11"  
    DSN=mydsn  
    CONNECT=" odbc; uid=User1;PWD=secret"  
>
```

Example:

```
<ign:db_open HANDLE="T1" DSN="PrimeDB">  
  
<ign:sql HANDLE="T1" Name="GetInfo">  
    SELECT TOP 10 * FROM MyTable  
</ign:sql>
```

See also:

```
<ign:sql>
```

ign:sql

Description: The **<ign:SQL>** tag defines a database query by setting the data source and the SQL statement to be used.

Syntax:

```
<ign:SQL DSN="MyDSN" NAME="MyQuery">  
    SELECT * FROM MyTable WHERE num > 40  
</ign:SQL>
```

Note: be sure to use single quotation marks for strings in the select statements

Attributes	Description	Required
NAME	Unique name you assign to the query	Y
DSN	ODBC Data Source Name (DSN)	N
CONNECT	Connection string (for userid, password, etc)	N
HANDLE	Name of previously opened handle (see ign:db_open)	N
KEEPOPEN	Keeps the query alive	N

See Also: **<ign:SQL_RUN>**

ign:sql_run

Description: The `<ign:SQL_RUN>` tag loops through all records retrieved by the **SQL** statement previously defined in the **ign:SQL** statement.

Syntax:

```
<ign:SQL_RUN Name="MyQuery">  
    FieldName value = #MyQuery.FieldName#  
</ign:SQL_RUN>
```

Attributes	Description	Required
NAME	Name of previously defined query	Y

See Also: `<ign:SQL>`

ign:db_datasources

Description: The **<ign:DB_DATASOURCES>** tag loops through all the ODBC data sources available.

Syntax:

```
<ign:DB_DATASOURCES INDEX="MyDataSrc" >  
    #MyDataSrc.Name#  
    #MyDataSrc.Descr#  
</ign:DB_DATASOURCES>
```

ign:db_fields

Description: The < **ign:DB_FIELDS** > tag loops through all the fields in a data base table.

Syntax:

```
<ign:DB_FIELDS
    TABLE="MainTable"
    DSN="MainODBC"
    INDEX="MyField"
    CONNECT="USR=george;PWD=secret"
>
    #MyField.Name#
    #MyField.FieldLen#
    #MyField.TypeId#
    #MyField.Type#
</ign:DB_FIELDS>
```

ign:db_sqldrivers

Description: The <ign:DB_SQLDRIVERS> tag loops through all the ODBC drivers and returns information.

Syntax:

```
<ign:DB_SQLDRIVERS INDEX="MyDataSrc">  
    #MyDataSrc.Name#  
    #MyDataSrc.Descr#  
</ign:DB_SQLDRIVERS>
```

ign:db_tables

Description: The **<ign:DB_TABLES>** tag loops through all the ODBC tables in a database.

Syntax:

```
<ign:DB_TABLES
  INDEX="MyTable"
  DSN="MainODBC"
>
  #MyTable.Name#
  #MyTable.Type#
  #MyTable.Attribs#
  #MyTable.Owner#
</ign:DB_TABLES>
```

10. OCX

ign:object

Description: The <ign:OBJECT> tag allows the Server to host **ActiveX** and **OCX** Objects

Syntax:

```
<ign:OBJECT NAME="MyObject" CLASSID="MyCtl.Engine1">
```

```
<ign:set MyObject.Property = "Red">
```

```
<ign:set ret = MyObject.Property >
```

```
<ign:set ret = MyObject.Method( "Field1", "Field2")>
```

Attributes	Description	Required
NAME	Name of to defined object	Y
CLASSID	Class ID of object	Y

11. Cookies

ign:cookie

Description: The **<ign:COOKIE>** tag creates and deletes cookies which can be read by the client browser. To read the cookie use the syntax **#Cookie.Name#**

Syntax:

```
<ign:COOKIE  
NAME="MyCookie"  
VALUE="my information"  
EXPIRES="Sep-2-00"  
>
```

12. Searching

ign:search

Description: The **<ign:SEARCH>** tag performs a text search on files on the server

Syntax:

```
<ign:SEARCH
  DIR="c:/inetpub/wwwroot/"
  TEXT="abc"
  INDEX="idx"
  CASE="YES"
  SUBDIR="no"
  REGEX="y"
  MAXHITS=1
>
```

Found:

FileName:	#idx. HitFileName #
Character Offset:	#idx. Offset #
Hit Count:	#idx. HitCount #
BeforeSnippet:	#idx. BeforeSnippet #
Snippet:	#idx. Snippet #
AfterSnippet:	#idx. AfterSnippet #

</ign:SEARCH>

Attributes	Description	Required
DIR	Directory of file to search	Y
TEXT	Text to search for	Y
INDEX	Name of variable for receiving the result object	Y
SUBDIR	Yes/No flag to indicate if subdirectories should be searched	N
CASE	flag for case sensitive search	N
REGEX	flag for Regular Expression search	N
MAXHITS	Maximum hits per page (0 – retrieve all)	N

Example:

```
<ign:SEARCH
DIR="c:/Inetpub/wwwroot/"
TEXT="abc"
INDEX="idx"
CASE="YES"
SUBDIR="Y"
REGEX="y"
MAXHITS=1
>
```

Files Found:

```
FileName: #idx.HitFileName#<br>
Character Offset: #idx.Offset#<br>
Hit Count: #idx.HitCount#<br>
```

```
<ign:set XX=idx.BeforeSnippet>
<ign:set BSnip=XX.HTMLEncode()>
```

```
<ign:set XX=idx.Snippet>
<ign:set Snip=XX.HTMLEncode()>
```

```
<ign:set XX=idx.AfterSnippet>
<ign:set ASnip=XX.HTMLEncode()>
```

```
<a href="#idx.HitFileName#"> Snippet: #BSnip# <font color=red><b>#Snip#</b></font> #ASnip# </a>
<br><br>
```

```
</ign:SEARCH>
```

Output:

```
Found: FileName: c:/Inetpub/wwwroot/test.ign
Character Offset: 1153
Hit Count: 1
Snippet: <ign:function abc ()> <ign:return 10
```

13. Internet Communication

ign:http_io

Description: The **<ign:HTTP_IO>** tag requests that the **Ignite** server get data from other Internet websites and place the data into a variable.

Syntax:

```
<ign:HTTP_IO  
  URL="http://www.abc.com/test.ign?name=xx"  
  VARIABLE="MyVar"  
  PROXYIP="myprox.com"  
  PROXYPORT="8080"  
  ERROR="MyErrString"  
  ERRNO="MyErrorNo"  
>
```

Attributes	Description	Required
URL	Target URL to get data from	Y
VARIABLE	variable to receive contents	Y
PROXYIP	IP address of proxy server	N
PROXYPORT	Port number for the proxy server	N
ERROR	variable to receive error message if any	N
ERRORNO	variable to receive error number	N

14. Global Objects

_Param

Description: This object contains information about the Parameters passed to the current script

```
_Param.Count()  
_Param.Data()  
_Param.Name()
```

```
<ign:for from=1 index="idx" to=_Param.Count() >  
  _Param.Name(idx)#<br>  
  _Param.Data(idx)#<br>  
  ----<br>  
</ign:for>
```

_Cookie

Description: This object contains information about the Cookie passed to the current script

```
_Cookie.Count()  
_Cookie.Data()  
_Cookie.Name()
```

```
<ign:for from=1 to=_Cookie.Count() index="idx">  
  Name:##_Cookie.Name(idx)#<br>  
  Data:##_Cookie.Data(idx)#<br>  
  ----<br>  
</ign:for>
```

15. Global Functions

Asc()

Description:

returns an ASCII number when given an letter

Syntax:

```
<ign:set MyNum = Asc( "A")>
```

Example:

```
<b>ASCII Chart</b>

<table border=1 cellpadding=0 cellspacing=0>
<tr>

<ign:set NumCols=4>
<ign:set NumRows=128/NumCols>

<ign:set SpecialChars = "NUL,SOH,STX,ETX,EOT,ENQ,ACK,BEL,"
                        & "BS,HT,LF,VT,FF,CR,SO,SI,DLE,DC1,"
                        & "DC2,DC3,DC4,NAK,SYM,ETB,CAN,EM,SUB,"
                        & "ESC,FS,GS,RS,US,SP">

<ign:for from=1 to=NumRows index="idx">

  <ign:if idx==1>
    <!-- Print Table Header -->
    <tr>
      <ign:for from=1 to=NumCols index="col">
        <td>Decimal</td><td>Octal</td><td>Hex</td><td>Character</td>
        <td bgcolor=cyan>&nbsp;</td>
      </ign:for>
    </tr>
  </ign:if>

  <tr>
    <ign:set AsciiNum=idx-1>

    <!-- Print Decimal,Octal,Hex and Ascii Data ---->
    <ign:for from=1 to=NumCols index="col">
      <td>#DecimalFormat(AsciiNum,"000")#</td>
      <td>#OctalFormat(AsciiNum,3)#</td>
      <td>#HexFormat(AsciiNum,3)#</td>

      <td>
        <ign:if AsciiNum .le. 32>
          #SpecialChars.GetWord(AsciiNum+1)#
        <ign:else>
          #Char(AsciiNum)#
        </ign:if>
      </td>
      <td bgcolor=cyan>&nbsp;</td>
      <ign:set AsciiNum=AsciiNum+NumRows>
    </ign:for>
  </tr>

</ign:for>
</tr>
</table>
```

Char()

Description: returns a character when given the ASCII number for the character

Syntax:

```
<ign:set MyCh = Char( 13)>
```

FileExists()

Description: Checks the existence of a file. If the file exists the function returns a TRUE (**1**) otherwise FALSE (**0**).

Syntax:

```
<ign:set tmp= FileExists("c:\temp.dat")>
```

Example:

```
<ign:set tmp= FileExists("c:\temp.dat")>  
<ign:IF tmp == TRUE>  
    Yes The file exist  
<ign:ELSE>  
    The file does not exist  
</ign:IF>
```


Find()

Description: Finds the character position (zero based) of characters in a string, if not found the function returns -1

Syntax:

```
<ign:set position= Find("This is a test", "te", 2)>
```

Arguments:

Value – a string value

substring – a string to find

first character to search from – optional parameter used to indicate first character to search from

FormatNum()

Description: This function takes a numeric string and formats it for output.

Syntax:

```
<ign:set tmp= FormatNum("143.9943", "4.2", "$")>
```

Arguments:

- Value** – a string value
- Width** – The whole number represents the minimum number of digits to display. The value after the decimal point represents the number of digits after the decimal point.
- Prefix** – a prefix to the output

GetScriptURL()

Description: Returns the URL (Uniform Resource Locator) of the current running script

Syntax:

```
<ign:set ScriptURL = GetScriptURL()>
```

Example:

```
<ign:set ScriptURL = GetScriptURL()>  
This file's url is:#ScriptURL#
```

GetScriptName()

Description: Returns the full file name of the current script or include file

Syntax:

```
<ign:set ScriptName = GetScriptName()>
```

Example:

```
<ign:set ScriptName = GetScriptName()>  
This file's name is:#ScriptName#
```

GetScriptDir()

Description:

Returns the directory path where the current script (or include file) resides

Syntax:

```
<ign:set ScriptDir = GetScriptDir()>
```

Example:

```
<ign:set ScriptDir = GetScriptDir()>  
This file directory is:#ScriptDir#
```

GetMainScriptName()

Description: Returns the filename of the main script

Syntax:

```
<ign:set MainScriptName = GetMainScriptName()>
```

Example:

```
<ign:set MainScriptName = GetMainScriptName()>  
This main script file name is :#MainScriptName#
```

GetMainScriptDir()

Description: Returns the directory path of the main script

Syntax:

```
<ign:set MainScriptDir = GetMainScriptDir()>
```

Example:

```
<ign:set MainScriptDir = GetMainScriptDir()>  
This file directory is:#MainScriptDir#
```

GetServerName()

Description: Returns the URL of the server

Syntax:

```
<ign:set MaintUrlr = GetServerName ()>
```

Example:

```
<ign:set MainURL = GetServerName ()>
```

This server url is:# MainURL #

Mid()

Description: Returns a number of the characters starting at a given zero based offset

Syntax:

```
<ign:set StringValue= Mid("House", nStart, nLen)>
```

zero based, string, start, length

Arguments:

Value – a string value

Offset – starting offset (zero based)

Num characters – optional parameter used to indicate number of character to extract, if this parameter is not passes the function will return up to the end of the string

GetServerVar()

Description: Returns the given server environment variable

Syntax:

```
<ign:set StringValue= GetServerVar("REMOTE_ADDR")>
```

Some useful variables are:

HTTP_USER_AGENT
REMOTE_ADDR
REMOTE_HOST
HTTP_FROM
HTTP_REFERER

Len()

Description: Returns the length or number of the characters in a string

Syntax:

```
<ign:set ValueLength=Len( value)>
```

Example:

```
<ign:set Name = "DEF">
```

```
<ign:set ret = Len( Name )>
```

The number of characters in '#Name#' is #ret#

Output:

The number of characters in 'DEF' is 3

GetWordCount()

Description: Returns a word count from a string

Syntax:

```
<ign:set WCount=GetWordCount("first,second,third", ",")>
```

GetWord()

Description: This function splits up a string and extracts and returns the n'th (1-based) word from the string

Syntax:

GetWord(string1, Num [,Delim])

Example:

```
<ign:set Name = "one/two/three">  
<ign:set ret = GetWord( Name , 2, "/" )>
```

The second word in "#Name#" is "#ret#"

Output:

The second word in "one/two/three" is "two"

Lower()

Description: Returns an lower case string

Syntax:

```
<ign:set myString=Lower("Now is The Time")>
```

Replace()

Description: Returns a string after replacing all occurrences of one substring with another substring

Syntax:

```
<ign:set myString=Replace( OriginalString, stringFrom, stringTo)>
```

Example:

```
<ign:set myString=Replace("pet dog","dog","cat")>
```

Sleep()

Description: This function performs a wait for a given number of milliseconds before processing continues

Syntax:

<ign:call Sleep(*nMilliseconds*)>

Trim()

Description: Returns the characters in a string after removing any blanks or white space (tabs, spaces carriage returns, etc) on the left and right

Syntax:

```
<ign:set myString=Trim(" House ")>
```

TrimLeft()

Description: Returns the characters in a string after removing any blanks or white space (tabs, spaces carriage returns, etc) on the left

Syntax:

```
<ign:set myString=TrimLeft("  House")>
```

TrimRight()

Description: Returns the characters in a string after removing any blanks or white space (tabs, spaces carriage returns, etc) on the right

Syntax:

```
<ign:set myString=TrimRight("House ")>
```

Upper()

Description: Returns an upper case string

Syntax:

```
<ign:set myString=Upper("abcdef")>
```

GetAsString()

Description: Forces a variable to a String type

Syntax:

```
<ign:set MyString = GetAsString( x)>
```

GetAsInt()

Description: Forces a variable to a Integer type

Syntax:

```
<ign:set MyNum = GetAsInt( x)>
```

GetAsFloat()

Description: Forces a variable to a Floating point type

Syntax:

```
<ign:set MyFloat = GetAsFloat( x)>
```

IsNum()

Description: Returns TRUE if a variable is a number (0-9)

Syntax:

```
<ign:set X = "123">  
<ign:IF IsNum( X)>  
    X is a Number  
<ign:ELSE>  
    X is not a Number  
</ign:IF>
```

Output:

X is a Number

IsAlpha()

Description: Returns TRUE if variable is Alphabetical (a-z,A-Z)

Syntax:

```
<ign:set X = "123">  
<ign:IF IsAlpha( X)>  
    X is Alphabetical  
<ign:ELSE>  
    X is not Alphabetical  
</ign:IF>
```

Output:

X is not Alphabetical

IsSpace()

Description: Returns TRUE if a variable consists of whitespace characters (spaces, tabs, carriage returns)

Syntax:

```
<ign:set X = " 123">  
<ign:IF IsSpace( X)>  
    X is white Space  
<ign:ELSE>  
    X is not white Space  
</ign:IF>
```

Output:

X is not white Space

IsDate()

Description: Checks for a valid date string , returns TRUE (1) on success otherwise returns FALSE (0).

Syntax:

```
<ign:set y = IsDate( DateString, DateFormat)>
```

Example:

```
<ign:set y = IsDate( "02/28/02", "%MM/%DD/%YY")>  
y=#y#<br>
```

16. INI File Functions

IniFileWrite()

Description: Write to an MS Windows style ini file

Syntax:

```
<ign:set xx = IniFileWrite("C:\Inetpub\wwwroot\test1.ini", "MyMainKey", "MySubKey", "Hello World")>
```

Example:

```
<ign:set IsOkay = IniFileWrite("C:\data\test1.ini", "MyMainKey", "MySubKey", "Hello World")>
<ign:if IsOkay == TRUE>
    ini File updated successfully
<ign:else>
    Problem writing to ini File
</ign:if>
```

IniFileRead()

Description: Read from an MS Windows style ini file

Syntax:

```
<ign:set xx = IniFileRead("C:\Inetpub\wwwroot\test1.ini", "MyMainKey", "MySubKey")>  
<ign:set xx = IniFileRead("C:\Inetpub\wwwroot\test1.ini", "MyMainKey", "MySubKey2", "MyDefaultValue")>
```

Example:

```
<ign:set val = IniFileRead("C:\data\test1.ini", "MyMainKey", "MySubKey")>  
Value Found=#val#<br>
```

```
<ign:set val = IniFileRead("C:\data\test1.ini", "MyMainKey", "UnknownSubKey", "MyDefaultValue")>  
Value Found=#val#<br>
```

17. Date Functions

GetDate()

Description: Get the current date and time, or parse the input parameters into a date value.

Syntax:

```
<ign:set MyDate = GetDate()>  
<ign:set MyDate = GetDate("10/21/01 11:31:05", "%MM/%DD/%YY %h:%m:%s")>  
<ign:set MyDate = GetDate(12,30,2001)>
```

Example:

```
<ign:set MyDate = GetDate()>  
Current date is :#MyDate#<br>  
  
<ign:set MyDate = GetDate("10/21/01 11:31:05", "%MM/%DD/%YY %h:%m:%s")>  
The parsed date is :#MyDate#<br>  
  
<ign:set MyDate = GetDate(12,30,2001)>  
The parsed date is :#MyDate#<br>
```

Output:

```
Current date is :12/19/2001 18:11:20  
The parsed date is :10/21/2001 11:31:05  
The parsed date is :12/30/2001 00:00:00
```

FormatDateTime()

Description:

Syntax:

```
<ign:set Str =MyDate.FormatDateTime("%M/%D/%Y")>
```

Example:

```
<ign:set MyDate = GetDate(12,30,2001)>
```

The parsed date is :#MyDate#


```
<ign:set Str =MyDate.FormatDateTime("%M/%D/%Y")>
```

Str = #Str#

Output:

The parsed date is :12/30/2001 00:00:00

Str = 12/30/01

AddDays()

Description:

Syntax:

```
<ign:set MyDate = MyDate.AddDays( nDays)>
```

Example:

```
<ign:set MyDate = GetDate(12,01,2001)>  
The parsed date is :#MyDate#<br>
```

```
<ign:set StrOut = MyDate.AddDays(-5)>  
StrOut = #StrOut#<br>
```

Output:

```
The parsed date is :12/01/2001 00:00:00  
StrOut = 11/26/2001 00:00:00
```


AddSeconds()

Description:

Syntax:

```
<ign:set MyDate = MyDate.AddSeconds( dVal)>
```

Example:

AddMinutes()

Description:

Syntax:

```
<ign:set MyDate =MyDate.AddMinutes( dVal)>
```

Example:

AddHours()

Description:

Syntax:

```
<ign:set MyDate = MyDate.AddHours( dVal)>
```

Example:

AddWeeks()

Description:

Syntax:

```
<ign:set MyDate = MyDate.AddWeeks( dVal)>
```

Example:

IgniteFusion Reference Guide
Version 2.3r

MONTH1 "%M" 12
MONTH2 "%MM" 12
MONTH3 Month name 3 chars "%MMM" Dec
MONTH4 Month name full "%MMMM" December
DAY1 "%D" 15
DAY2 2 digit day "%DD" 15
DAY3 WeekDay 3 chars "%DDD" Sat
DAY4 WeekDay FullName "%DDDD" Saturday
YEAR1 2 digit year without century "%Y" 01
YEAR2 2 digit year without century "%YY" 01
YEAR4 4 digit year with century "%YYYY" 2001
LOCALE_DATE "%A" Sat Dec 15, 2001
LOCALE_TIME "%B" 21:39:04
UNIX_DATETIME UNIX Style date-time "%C" Dec 15 22:51:45 2001
DAYOFYEAR "%j" 349
PERCENT "%%" %
HOUR1 digit Hour (01-12) "%h" 09
HOUR2 2 digit Hour (01-12) "%hh" 09
HOUR3 digit Hour (01-23) "%H" 21
HOUR4 2 digit Hour (01-23) "%HH" 21
MIN1 digit minute(00-59) "%m" 39
MIN2 2 digit minute(00-59) "%mm" 39
SEC1 digit second(00-59) "%s" 04
SEC2 2 digit second(00-59) "%ss" 04
AM1 A.M. or P.M. "%x" PM
AM2 A.M. or P.M. "%xx" PM

GetTime()

Description: Returns the current time in a given format

Syntax:

```
<ign:set SecondValue=GetTime("%H")>
```

%a - Abbreviated weekday name

%A - Full weekday name

%b - Abbreviated month name

%B - Full month name

%c - Date and time representation appropriate for locale

%d - Day of month as decimal number (01 – 31)

%H - Hour in 24-hour format (00 – 23)

%I - Hour in 12-hour format (01 – 12)

%j - Day of year as decimal number (001 – 366)

%m - Month as decimal number (01 – 12)

%M - Minute as decimal number (00 – 59)

%p - Current locale's A.M./P.M. indicator for 12-hour clock

%S - Second as decimal number (00 – 59)

%U - Week of year as decimal number, with Sunday as first day of week (00 – 53)

%w - Weekday as decimal number (0 – 6; Sunday is 0)

%W - Week of year as decimal number, with Monday as first day of week (00 – 53)

%x - Date representation for current locale

%X - Time representation for current locale

%y - Year without century, as decimal number (00 – 99)

%Y - Year with century, as decimal number

%z, %Z - Time-zone name or abbreviation; no characters if time zone is unknown

18. DataBase object functions

MoveNext()

Description: move to next database record

Syntax:

```
<ign:set ok =MyDB.MoveNext( )>
```

Example:

MovePrev()

Description: move to previous database record

Syntax:

```
<ign:set ok =MyDB.MovePrev( )>
```

Example:

MoveFirst()

Description: move to the first database record

Syntax:

```
<ign:set ok =MyDB.MoveFirst( )>
```

Example:

MoveLast()

Description: move to the last database record

Syntax:

```
<ign:set ok =MyDB.MoveLast( )>
```

Example:

Move()

Description: move to the n'th database record

Syntax:

```
<ign:set ok =MyDB.Move( nPos )>
```

Example:

Close()

Description: close record set

Syntax:

```
<ign:set ok =MyDB.Close( )>
```

Example:

```
<ign:set X = GetInfo.MoveFirst()>  
<ign:while X>  
    #GetInfo.CORPNAME#<br>  
    <ign:set X = GetInfo.MoveNext()>  
</ign:while>  
<ign:set X = GetInfo.Close()>
```

FieldCount()

Description: returns the number of fields retrieved from select statement

Syntax:

```
<ign:set nCount = testdb.FieldCount(>
```

Example:

```
<ign:SQL DSN="TestTypes" Name="testdb">  
    SELECT * FROM Table1  
</ign:SQL>  
  
<ign:set nCount = testdb.FieldCount(>  
#nCount#<br>  
  
<ign:FOR INDEX="idx" FROM=1 TO=nCount>  
    <ign:set FName = testdb.FieldName(idx)>  
    <ign:set FType = testdb.FieldTypeName(idx)>  
    <ign:set FData = testdb.FieldData(idx)>  
    FieldData: #FData#<br>  
    Name:#FName#<br>  
    TypeName:#FType#<br>  
</ign:FOR>
```

FieldName()

Description: returns the name of a field given a field number (1 based array of retrieved field names)

Syntax:

```
<ign:set name = testdb.FieldName( nFieldNum)>
```

Example:

FieldType()

Description: returns the SQL data type of a field given a field number (1 based array of retrieved fields)

Syntax:

```
<ign:set type = testdb.FieldType( nFieldNum)>
```

Example:

FieldTypeNames()

Description: returns the name of a SQL data type of a field given a field number (1 based array of retrieved fields)

Syntax:

```
<ign:set typeName = testdb.FieldTypeNames( nFieldNum)>
```

Example:

FieldLen()

Description: returns the SQL data field length of a field given a field number (1 based array of retrieved fields)

Syntax:

```
<ign:set len = testdb.FieldLen( nFieldNum)>
```

Example:

FieldScale()

Description: returns the SQL data field Scale (number of digits to the right of the decimal point in a number) of a field given a field number (1 based array of retrieved fields)

Syntax:

```
<ign:set num = testdb.FieldScale( nFieldNum)>
```

Example:

IsFieldNull()

Description: returns the SQL indicator if field is null (1 - true, 0 - false) of a field given a field number (1 based array of retrieved fields)

Syntax:

```
<ign:set isnull = testdb.IsFieldNull( nFieldNum)>
```

Example:

FieldData()

Description: returns the field data from a field given a field number (1 based array of retrieved fields)

Syntax:

```
<ign:set FData = testdb.FieldData( nFieldNum)>
```

Example:

FieldDataByName()

Description: returns the field data from a field given a field name

Syntax:

```
<ign:set FData = testdb.FieldDataByName( "MyField")>
```

Example:

FieldIndexByName()

Description: returns the field number (1 based) given a field name

Syntax:

```
<ign:set index = testdb.FieldIndexByName( "MyField")>
```

Example:

```
<ign:SQL DSN="TestTypes" Name="testdb">
```

```
    SELECT * FROM Table1
```

```
</ign:SQL>
```

```
<ign:set nCount = testdb.FieldCount()>
```

```
#nCount#<br>
```

```
<ign:FOR INDEX="idx" FROM=1 TO=nCount>
```

```
    <ign:set FName = testdb.FieldName(idx)>
```

```
    <ign:set FType = testdb.FieldTypeName(idx)>
```

```
    <ign:set FData = testdb.FieldData(idx)>
```

```
    FieldData: #FData#<br>
```

```
    Name: #FName#<br>
```

```
    Type: #FType#<br>
```

```
</ign:FOR>
```

19. TIFF image functions

TiffImageCount()

Description: Returns the number of images in a TIFF file

Syntax:

```
<ign:set nCount = TiffImageCount("FileName")>
```

Example:

```
<ign:set nCount = TiffImageCount("c:\bmp\aaa.tiff")>  
TiffImageCount: #nCount #<br>
```

TiffAppend()

Description: appends a TIFF image to TIFF file

Syntax:

```
<ign:set xx = TiffAppend("OutputFile", "InputFile")>
```

Example:

```
<ign:set xx = TiffAppend("c:\bmp\aaa.tif", "c:\bmp\bbb.tif")>  
Append returned #xx#<br>
```


TiffExtract()

Description: extracts a single TIFF image from a TIFF file

Syntax:

```
<ign:set xx = TiffExtract("NewFile", "OriginalFile", nImageNum)>
```

Example:

```
<ign:set xx = TiffExtract("c:\bmp\new.tif", "c:\bmp\aaa.tif", 2)>  
TiffExtract #xx#<br>
```

20. Number Functions

Abs()

Description: Returns the absolute value of a number

Syntax:

```
<ign:set x = Abs( num)>
```

DecimalFormat()

Description: Formats a number for output

Syntax:

```
<Ign:set xx = aa.DecimalFormat("PositiveMask ; NegativeMask ; ZeroMask")>
```

Mask definition

999 - for number place holders

0 – for fix length and/or zero padding

Example:

```
<Ign:set aa = -1024.0>
```

```
<Ign:set xx = aa.DecimalFormat(" $ 9,990.00;($ 9,990.00); Zero")>
```

```
Decimal Format=#xx#<br>
```

Output:

```
Decimal Format=($ 924.00)
```

Sin()

Description: Returns the Sine of a number

Syntax:

```
<ign:set x = Sin( num)>
```

Cos()

Description: Returns the Cosine of a number

Syntax:

```
<ign:set x = Cos( num)>
```

Tan()

Description: Returns the Tangent of a number

Syntax:

```
<ign:set x = Tan( num)>
```

Sqrt()

Description: Returns the Square root of a number

Syntax:

```
<ign:set x = Sqrt( num)>
```

Pi()

Description: Returns the number Pi (3.14159)

Syntax:

```
<ign:set x = Pi( )>
```


Log()

Description: Returns the natural Log (base e) of a number

Syntax:

```
<ign:set x = Log( num)>
```

Log10()

Description: Returns the Log base 10 of a number

Syntax:

```
<ign:set x = Log10( num)>
```

OctalFormat()

Description: Formats a decimal number into a Octal (base eight) number

Syntax:

```
<ign:set str = num.OctalFormat( nPlaces)>
```

Example:

```
<ign:set aa=127>  
<ign:set oo = aa.OctalFormat(4)>  
val = #oo#<br>
```

Output:

```
val = 0177
```

HexFormat ()

Description: Formats a decimal number into a Hexidecimal (base sixteen) number

Syntax:

```
<ign:set str = num.HexFormat( nPlaces)>
```

Example:

```
<ign:set aa=127>  
<ign:set xx = aa.HexFormat(4)>  
val = #xx#<br>
```

Output:

```
val = 007F
```

Power()

Description: Power(x, y) returns x raised to the power of y

Syntax:

```
<ign:set x = Power( base, exponent)>
```

Round()

Description: Round(x) returns value rounded to the nearest integer

Syntax:

<ign:set n = Round(x)>

Trunc()

Description: Returns the whole part of a number, everything after the decimal point is truncated

Syntax:

```
<ign:set x = Trunc( num)>
```

Rand()

Description: returns a random value between 0 and 1

Syntax:

```
<ign:set x = Rand( )>
```


21. Bit Functions

BitOR(x,y)

Description: Returns the a bitwise x OR y

Syntax:

```
<ign:set val = BitXOr( x,y)>
```

BitAnd(x,y)

Description: Returns the a bitwise x AND y

Syntax:

```
<ign:set val = BitAnd( x,y)>
```

BitXOr(x,y)

Description: Returns the a bitwise x Exclusive OR y

Syntax:

```
<ign:set val = BitXOr( x,y)>
```

BitShift(x,y)

Description: Returns the a bitwise shift

Syntax:

```
<ign:set val = BitShift( x, nPosition)>
```

BitNot(x)

Description: Returns the a bitwise NOT

Syntax:

```
<ign:set val = BitNot( x)>
```

22. String Functions

CharAt()

Description: Returns the character at a given index position (indexes are 1 based)

Syntax:

<ign:set *pos* = **CharAt(** *nPos* **)>**

Len()

Description: Returns the number of characters in a string

Syntax:

```
string1.Len()
```

Example:

```
<ign:set Name = "DEF">  
<ign:set ret = Name.Len()>  
The number of characters in '#Name#' is #ret#<br>
```

Output:

The number of characters in 'DEF' is 3

Compare()

Description: Performs a case-sensitive comparison of two strings. The values returned are:

- 1 if string1 comes before string2 when in alphabetical order
- 0 if string1 is equal to string2
- 1 if string1 comes after string2 when in alphabetical order

Syntax:

```
string1.Compare( string2)
```

Example:

```
<ign:set Name = "DEF">
<ign:set ret = Name.Compare( "ABC")>

<ign:if ret == -1>
    "#Name#" is less than "ABC"
<ign:else_if ret == 0>
    "#Name#" is equal to "ABC"
<ign:else_if ret == 1>
    "#Name#" is greater than "ABC"
</ign:if>
```

Output:

"DEF" is greater than "ABC"

CompareNoCase()

Description: Performs a case-insensitive comparison of two strings. The values returned are:

- 1 if string1 comes before string2 when in alphabetical order
- 0 if string1 is equal to string2
- 1 if string1 comes after string2 when in alphabetical order

Syntax:

```
string1.CompareNoCase( string2)
```

Example:

```
<ign:set Name = "abc">  
<ign:set ret = Name.CompareNoCase( "ABC")>  
  
<ign:if ret == -1>  
    "#Name#" is less than "ABC"  
<ign:else_if ret == 0>  
    "#Name#" is equal to "ABC"  
<ign:else_if ret == 1>  
    "#Name#" is greater than "ABC"  
</ign:if>
```

Output:

"abc" is equal to "ABC"

Find()

Description: Returns the position of the first occurrence of a substring (string2) in the main string (string1) starting at a given position.
Returns 0 if substring is not found in the string. Find is case-sensitive.

Syntax:

```
string1.Find( string2 [, nStart])
```

Example:

```
<ign:set Name = "ABCDEFGH">  
<ign:set ret = Name.Find( "DEF")>  
  
<ign:if ret == 0>  
    "#Name#" does not contain the substring "DEF"  
<ign:else>  
    "#Name#" contains the substring "DEF" starting at the #ret#'th character  
</ign:if>
```

Output:

"ABCDEFGH" contains the substring "DEF" starting at the 4'th character

FindNoCase()

Description: Returns the position of the first occurrence of a substring (string2) in the main string (string1) starting at a given position.
Returns 0 if substring is not found in the string. FindNoCase is case-insensitive.

Syntax:

```
string1.FindNoCase( string2 [, nStart])
```

Example:

```
<ign:set Name = "ABCDEFGH">  
<ign:set ret = Name.FindNoCase( "def")>  
  
<ign:if ret == 0>  
    "#Name#" does not contain the substring "def"  
<ign:else>  
    "#Name#" contains the substring "def" starting at the #ret#'th character  
</ign:if>
```

Output:

"ABCDEFGH" contains the substring "def" starting at the 4'th character

ReverseFind()

Description: Returns the position of the last occurrence of a substring (strTarget) in the main string (string1) starting at a given position from the end.
Returns 0 if substring is not found in the string. ReverseFind is case-sensitive.

Syntax:

```
string1.ReverseFind( strTarget [,nEnd])
```

Example:

```
<ign:set Name = "ABCDEFGHI">
<ign:set ret = Name.ReverseFind( "DEFG")>

<ign:if ret == 0>
    "#Name#" does not contain the substring "DEFG"
<ign:else>
    "#Name#" contains the substring "DEFG" starting at the #ret#'th character
</ign:if>
```

Output:

"ABCDEFGHI" contains the substring "DEFG" starting at the 4'th character

ReverseFindNoCase()

Description: Returns the position of the last occurrence of a substring (string2) in the main string (string1) starting at a given position from the end. Returns 0 if substring is not found in the string. ReverseFindNoCase is case-insensitive.

Syntax:

```
string1.ReverseFindNoCase( strTarget [,nEnd])
```

Example:

```
<ign:set Name = "abcdefghi">
<ign:set ret = Name.ReverseFindNoCase( "DEF")>

<ign:if ret == 0>
    "#Name#" does not contain the substring "DEF"
<ign:else>
    "#Name#" contains the substring "DEF" starting at the #ret#'th character
</ign:if>
```

Output:

"abcdefghi" contains the substring "DEF" starting at the 4'th character

FindOneOf()

Description: Returns the position of the first occurrence of any character in the string set (stringset). Returns 0 if none of the characters are found. The search is case-sensitive.

Syntax:

```
string1.FindOneOf( stringset [,nStart])
```

Example:

```
<ign:set CharSet = ";,.">
<ign:set Name = "Hi, he said">
<ign:set ret = Name.FindOneOf( CharSet)>

<ign:if ret == 0>
    "#Name#" does not contain any characters in "#CharSet#"
<ign:else>
    "#Name#" contains character int the set at the #ret#'th position
</ign:if>
```

Output:

"Hi, he said" contains character int the set at the 3'th position

FindWord()

Description: Returns the word position in number of words for a given word

Syntax:

Example:

```
<ign:set Name = "one/two/three">  
<ign:set ret = FindWord( Name, "three", "/" )>  
Word num is: #ret#
```

Output:

Word num is: 3

GetWord()

Description: This function splits up a string and extracts and returns the n'th (1-based) word from the string

Syntax:

```
string1.GetWord( Num [,Delim])
```

Example:

```
<ign:set Name = "one/two/three">  
<ign:set ret = Name.GetWord( 2, "/" )>
```

The second word in "#Name#" is "#ret#"

Output:

The second word in "one/two/three" is "two"

GetWordCount()

Description: Returns a word count from a string

Syntax:

```
string1.GetWord( [Delim])
```

Example:

```
<ign:set Name = "one/two/three">  
<ign:set ret = Name.GetWordCount( "/")>
```

InsertChars()

Description: Returns a string with string2 inserted at a given character position (nFirstCharPos)

Syntax:

```
string1.InsertChars(string2, nFirstCharPos)
```

Example:

```
<ign:set Name = "one//three">  
<ign:set pos = Name.Find( "/" )>  
<ign:set ret = Name.InsertChars( "two", pos)>
```

The "#Name#" is now changed to "#ret#" by inserting at position #pos#

Output:

The "one//three" is now changed to "onetwo//three" by inserting at position 4

RemoveChars()

Description: Returns a string with characters removed from the given position

Syntax:

```
string1.RemoveChars( nFirstCharPos [,nNumChars]))
```

Example:

```
<ign:set Name = "one/two/three">  
<ign:set pos = Name.Find( "two")>  
<ign:set ret = Name.RemoveChars( pos, 3)>
```

The "#Name#" is now changed to "#ret#" by removing 3 characters starting at #pos#

Output:

The "one/two/three" is now changed to "one//three" by removing 3 characters starting at 5

Upper()

Description: Returns the string in upper case.

Syntax:

```
string1.Upper()
```

Example:

```
<ign:set Name = "One/Two/Three">  
<ign:set ret = Name.Upper(>
```

The string "#Name#" is now "#ret#"

Output:

The string "One/Two/Three" is now "ONE/TWO/THREE"

Lower()

Description: Returns the string in lower case.

Syntax:

```
string1.Lower()
```

Example:

```
<ign:set Name = "One/Two/Three">  
<ign:set ret = Name.Lower()>
```

The string "#Name#" is now "#ret#"

Output:

The string "One/Two/Three" is now "one/two/three"

Left()

Description: Returns the Leftmost n characters of the string.

Syntax:

```
string1.Left( nLen)
```

Example:

```
<ign:set Name = "One Two Three">
```

```
<ign:set ret = Name.Left(3)>
```

The Left 3 characters of "#Name#" are "#ret#"

Output:

The Left 3 characters of "One Two Three" are "One"

Right()

Description: Returns the rightmost n characters of the string.

Syntax:

```
string1.Right( nLen)
```

Example:

```
<ign:set Name = "One Two Three">  
<ign:set ret = Name.Right(3)>
```

The Right 3 characters of "#Name#" are "#ret#"

Output:

The Right 3 characters of "One Two Three" are "ree"

TrimRight()

Description: Trims the Rightmost blank characters of the string.

Syntax:

```
string1.TrimRight()
```

Example:

```
<ign:set Name = "  One  ">  
<ign:set ret = Name.TrimRight(>
```

The Right blank characters of "#Name#" are trimmed "#ret#"

Output:

The Right blank characters of " One " are trimmed " One"

TrimLeft()

Description: Trims the Leftmost blank characters of the string.

Syntax:

```
string1.TrimLeft()
```

Example:

```
<ign:set Name = "  One  ">  
<ign:set ret = Name.TrimLeft(>
```

The Left blank characters of "#Name#" are trimmed "#ret#"

Output:

The Left blank characters of " One " are trimmed "One "

Trim()

Description: Trims the right and left blank characters of the string.

Syntax:

```
string1.Trim()
```

Example:

```
<ign:set Name = "  One  ">  
<ign:set ret = Name.Trim()>
```

The blank characters of "#Name#" are trimmed "#ret#"

Output:

The blank characters of " One " are trimmed "One"

Mid()

Description: Returns a number of characters (nLen) from the string beginning at a given starting position (nStartPos).

Syntax:

```
string1.Mid( nStartPos [,nLen])
```

Example:

```
<ign:set Name = "One/Two/Three">  
<ign:set ret = Name.Mid(5,3)>
```

The string extracted from "#Name#" is "#ret#"

Output:

The string extracted from "One/Two/Three" is "Two"

RepeatString()

Description: Returns a string appended itself nCount times.

Syntax:

```
string1.RepeatString( nCount)
```

Example:

```
<ign:set Name = "AB ">  
<ign:set ret = Name.RepeatString(3)>
```

The string "#Name#" is repeated three times "#ret#"

Output:

The string "AB " is repeated three times "AB AB AB "

Replace()

Description: Returns a string with all occurrences of stringFrom replaced with stringTo. This function is case-sensitive.

Syntax:

```
string1.Replace( stringFrom, stringTo)
```

Example:

```
<ign:set Name = "ABABabAB">  
<ign:set ret = Name.Replace("A","C")>
```

The string before "#Name#" and after "#ret#" is

Output:

The string before "ABABabAB" and after "CBCBabCB" is

Reverse()

Description: Returns a string with the characters in reversed order

Syntax:

```
string1.Reverse()
```

Example:

```
<ign:set Name = "DCBA">  
<ign:set ret = Name.Reverse()>
```

The string "#Name#" reversed is "#ret#"

Output:

The string "DCBA" reversed is "ABCD"

UUEncode()

Description: Returns a URL-Encoded string which can be used to pass strings with blanks and other characters in a URL. Spaces are replaced with %20, and non-alphanumeric characters are replaced with hexadecimal escape sequences.

Syntax:

```
string1.UUEncode()
```

Example:

```
<ign:set Name = "This, is a test!">
```

```
<ign:set ret = Name.UUEncode()>
```

```
<a href="http://www.abc.com?#Name#">To Pass a URL Click Here<a>
```

```
The string "#Name#" encoded is "#ret#"
```

Output:

[To Pass a URL Click Here](http://www.abc.com?#Name#) The string "This, is a test!" encoded is "This%2c%20is%20a%20test%21"

UUDecode()

Description: Decodes a URL-Encoded string

Syntax:

```
string1.UUDecode()
```

Example:

```
<ign:set Name = "This%2c%20is%20a%20test%21">  
<ign:set ret = Name.UUDecode()>
```

The string "#Name#" decoded is "#ret#"

Output:

The string "This%2c%20is%20a%20test%21" decoded is "This, is a test!"

HTMLEncode()

Description: Prepares a string for HTML output by replacing angle brackets with HTML codes so that they are not interpreted as html tags.

Syntax:

```
string1.HTMLEncode()
```

Example:

```
<ign:set XX="<test>">  
<ign:set XX=XX.HTMLEncode()>
```

#XX#

DBEncode()

Description: Prepares a string for storing to database by replacing single quote with two single quotes

Syntax:

string1.DBEncode()

Example:

23. IGNITE SQL Reference

Adding Records

To add a record to an existing database use the **INSERT INTO** statement. (*)

Example:

```
<ign:SQL DATASOURCE="mydb" NAME="myquest">  
    INSERT INTO mytable ( FirstName, LastName) VALUES ('Donald','duck')  
</ign:SQL>  
  
<ign:SQL_RUN Name="myquest">  
</ign:SQL_RUN>
```

. (*) Note that single quotes must be used in all SQL statements

Deleting Records

To delete a record to an existing database use the **DELETE FROM** statement. (*)

Example:

```
<ign:SQL DATASOURCE="mydb" NAME="myquest">  
    DELETE FROM mytable WHERE FirstName='Donald'  
</ign:SQL>
```

```
<ign:SQL_RUN Name="myquest">  
</ign:SQL_RUN>
```

. (*) Note that single quotes must be used in all SQL statements

Retrieving Data

To retrieve records from an existing database use the **SELECT * FROM** statement. (*)

Example:

```
<ign:SQL DATASOURCE="mydb" NAME="myquest">  
    SELECT * FROM mytable WHERE FirstName='Donald'  
</ign:SQL>
```

```
<ign:SQL_RUN Name="myquest">  
    Last Name is #myquest.LastName#  
</ign:SQL_RUN>
```

. (*) Note that single quotes must be used in all SQL statements

ASCII Chart

ASCII Chart

Decima l	Octa l	Hex	Characte r	Decima l	Octa l	Hex	Characte r	Decima l	Octa l	Hex	Characte r	Decima l	Octa l	Hex	Characte r
000	000	000	NUL	032	040	020	SP	064	100	040	@	096	140	060	^
001	001	001	SOH	033	041	021	!	065	101	041	A	097	141	061	a
002	002	002	STX	034	042	022	"	066	102	042	B	098	142	062	b
003	003	003	ETX	035	043	023	#	067	103	043	C	099	143	063	c
004	004	004	EOT	036	044	024	\$	068	104	044	D	100	144	064	d
005	005	005	ENQ	037	045	025	%	069	105	045	E	101	145	065	e
006	006	006	ACK	038	046	026	&	070	106	046	F	102	146	066	f
007	007	007	BEL	039	047	027	'	071	107	047	G	103	147	067	g
008	010	008	BS	040	050	028	(072	110	048	H	104	150	068	h
009	011	009	HT	041	051	029)	073	111	049	I	105	151	069	i
010	012	00A	LF	042	052	02A	*	074	112	04A	J	106	152	06A	j
011	013	00B	VT	043	053	02B	+	075	113	04B	K	107	153	06B	k
012	014	00C	FF	044	054	02C	,	076	114	04C	L	108	154	06C	l
013	015	00D	CR	045	055	02D	-	077	115	04D	M	109	155	06D	m
014	016	00E	SO	046	056	02E	.	078	116	04E	N	110	156	06E	n
015	017	00F	SI	047	057	02F	/	079	117	04F	O	111	157	06F	o
016	020	010	DLE	048	060	030	0	080	120	050	P	112	160	070	p
017	021	011	DC1	049	061	031	1	081	121	051	Q	113	161	071	q
018	022	012	DC2	050	062	032	2	082	122	052	R	114	162	072	r
019	023	013	DC3	051	063	033	3	083	123	053	S	115	163	073	s
020	024	014	DC4	052	064	034	4	084	124	054	T	116	164	074	t
021	025	015	NAK	053	065	035	5	085	125	055	U	117	165	075	u
022	026	016	SYM	054	066	036	6	086	126	056	V	118	166	076	v
023	027	017	ETB	055	067	037	7	087	127	057	W	119	167	077	w
024	030	018	CAN	056	070	038	8	088	130	058	X	120	170	078	x
025	031	019	EM	057	071	039	9	089	131	059	Y	121	171	079	y
026	032	01A	SUB	058	072	03A	:	090	132	05A	Z	122	172	07A	z
027	033	01B	ESC	059	073	03B	;	091	133	05B	[123	173	07B	{
028	034	01C	FS	060	074	03C	<	092	134	05C	\	124	174	07C	
029	035	01D	GS	061	075	03D	=	093	135	05D]	125	175	07D	}
030	036	01E	RS	062	076	03E	>	094	136	05E	^	126	176	07E	~
031	037	01F	US	063	077	03F	?	095	137	05F	_	127	177	07F	

IgniteFusion Reference Guide
Version 2.3r

Ignite Tag Index

IGN

BREAK	10
CALL	11
CASE	12
CATCH	13
CLASS	14
CONTINUE	16
COOKIE	70
DB_DATASOURCES	65
DB_FIELDS	66
DB_OPEN	62
DB_SQLDRIVERS	67
DB_TABLES	68
DIR	37
DIR_CREATE	38
DIR_DELETE	39
ELSE	18
ELSE_IF	19
EMAIL	60
EXIT	22
FILE_COPY	47
FILE_DELETE	46
FILE_MOVE	43
FILE_READ	40
FILE_UPLOAD	44
FILE_WRITE	41
FOR	23
FUNCTION	24
GFX_ARC	54
GFX_CIRCLE	53
GFX_DISPLAY	58
GFX_LINE	50
GFX_LINE_TO	51
GFX_MAKE	48
GFX_MOVE_TO	52
GFX_RECT	56
GFX_TEXT	57
GFX_VIEW	59
GOTO	25
HTTP_IO	73
IF	17
IF_EXISTS	20
IF_NOT_EXIST	21
INCLUDE	26
LABEL	28
NOPROCESS	29
OBJECT	69
PARAM	27
RETURN	31
SEARCH	71
SET	32

IgniteFusion Reference Guide
Version 2.3r

SPLIT.....	30
SQL.....	63
SQL_RUN	64
SWITCH.....	33
THROW.....	34
TRY	35
WHILE	36