

Date and Time Functions

Introduction

If you are using these functions in conjunction with either the `Now()` or `Now_()` functions, be aware that the time zone returned is the one configured on the machine running the agent. All Cloud Agents are in UTC. Private Agents will vary based on the specific machine each agent is running on. If you are using a conversion function take this into account if your project is designed to run both in the cloud and on a Private Agent.

Time Zone Codes

The time zone parameters are one of the standard time zone codes, either a full name (TZ code) such as "America/Los_Angeles", a UTC offset such as "UTC-8:00", or an abbreviation such as "PST" (three-letter code). Time zone codes are case-sensitive. Full name TZ codes are recommended, if a time zone is being used, as "AST" can refer to either "Atlantic" or "Arabic" standard time.

- TZ codes (an unofficial list) can be found at https://en.wikipedia.org/wiki/List_of_tz_database_time_zones.
- Three-letter codes are listed at https://en.wikipedia.org/wiki/List_of_time_zone_abbreviations. These are deprecated by the IANA, though currently supported.
- Additional information can be found at <https://www.iana.org/time-zones>.

Converting UNIX Times to dates

If a UNIX time is a number that includes millisecond precision, you can use code similar to this to convert a UNIX time to a date:

```
// If you have a UNIX timestamp with milliseconds such as
createdate = "1478119530707";
// You can convert it to a date by
// truncating the last three digits and using FormatDate:
date = double(Left(createdate, 10));
FormatDate(date, "yyyy-mm-dd");
```

ConvertTimeZone

Declaration

```
string ConvertTimeZone(date d, string fromTZ, string toTZ[, bool
is_european_format, bool ignoreDST])

string ConvertTimeZone(string d, string fromTZ, string toTZ[, bool
is_european_format, bool ignoreDST])
```

Syntax

```
ConvertTimeZone(<d>, <fromTZ>, <toTZ>[, <is_european_format>, <ignoreDST>])
```

Required Parameters

- **date**: A date, either as a date object or as a date string
- **fromTZ**: The time zone to be converted from, expressed as a [time zone code](#)
- **toTZ**: The time zone to be converted to, expressed as a [time zone code](#)

Optional Parameters

- **is_european_format**: This optional flag determines if the European date format is to be used. When `true`, the format is day and then month; when `false`, the format is month and then day.
- **ignoreDST**: By default, daylight savings is accounted for when converting between the four major US time zones. Set this optional flag to `true` to override this action.

On This Page

- [Introduction](#)
- [ConvertTimeZone](#)
- [CVTDate](#)
- [DateAdd](#)
- [DayOfMonth](#)
- [DayOfWeek](#)
- [FormatDate](#)
- [GeneralDate](#)
- [GetUTCFormattedDate](#)
- [GetUTCFormattedDateTime](#)
- [LastDayOfMonth](#)
- [LongDate](#)
- [LongTime](#)
- [MediumDate](#)
- [MediumTime](#)
- [MonthOfYear](#)
- [Now](#)
- [Now_](#)
- [ShortDate](#)
- [ShortTime](#)

Search in This Topic

Related Functions

Related Topics

- [Cloud Studio](#)
- [Functions](#)
- [Scripts](#)

Last updated: Nov 20, 2019

Description

Take a date and returns it converted from one time zone to another time zone.

NOTE: If you are using `ConvertTimeZone` in conjunction with `Now()` or `Now_()`, be aware that the time zone returned is the one configured on the machine running the agent. All Cloud Agents are in UTC. Private Agents will vary based on the specific machine each agent is running on. If you are using a conversion function, such as `ConvertTimeZone`, take this into account if your project is designed to run both in the cloud and on a Private Agent.

Examples

```
ConvertTimeZone(Now(), "UTC", "America/Los_Angeles", false, false);
// If Now() returns "2017-03-10 18:34:37"
// returns "2017-03-10 10:34:37"

ConvertTimeZone("02/06/2017 5:25:00", "America/Los_Angeles", "America/New_York");
// Returns "2017-02-06 08:25:00" (shifts the time by 3 hours)
```

[\[Back to Top\]](#)

CVTDate

Declaration

```
string CVTDate(date d, string inputFormat, string outputFormat)

string CVTDate(string d, string inputFormat, string outputFormat)
```

Syntax

```
CVTDate(<d>, <inputFormat>, <outputFormat>)
```

Required Parameters

- **d:** A date object or date string
- **inputFormat:** A [format string](#), specifying the format of the input date
- **outputFormat:** A [format string](#), specifying the format of the output date

Description

Converts a date object or date string in the input format to a date string in the output format.

NOTE: If you are using `CVTDate` in conjunction with `Now()` or `Now_()`, be aware that the time zone returned is the one configured on the machine running the agent. All Cloud Agents are in UTC. Private Agents will vary based on the specific machine each agent is running on. If you are using a conversion function, such as `CVTDate`, take this into account if your project is designed to run both in the cloud and on a Private Agent.

Examples

In these examples, it is assumed that all agents are running in the cloud, and times returned would be UTC. "myDate" is the date June 19, 1994.

Formula	Input	Output
<code>CVTDate(myDate, "mmddyy", "yyyymmdd")</code>	061994	19940619

<code>CVTDate(myDate, "mmddyy", "ShortDate")</code>	061994	6/19/94
<code>CVTDate(myDate, "mmddyy", "LongDate")</code>	061994	Sunday, June 19, 1994
<code>CVTDate(Now(), "GeneralDate", "dd-mmm-yy, HH:MM AP")</code>	<i>on 1/31/17 at 12:39 PM in MST</i>	31-Jan-17, 07:39 PM
<code>CVTDate(Now(), "GeneralDate", "dd-mmm-yy, HH:MM.SS AP")</code>	<i>on 1/31/17 at 12:39:13 PM in MST</i>	31-Jan-17, 07:39:13 PM
<code>CVTDate(Now_(), "GeneralDate", "ddmmyyyy, HH:MM:SS.zzz AP")</code>	<i>on 1/31/17 at 12:39:13.310 PM in MST</i>	31Jan2017, 07:39:13.310 PM
<code>CVTDate(Now(), "GeneralDate", "yyyy-mm-dd_HH_MM")</code>	<i>on 1/31/17 at 12:39:13.310 PM in MST</i>	2017_01_31_19_39_13
<code>CVTDate(Now(), "UTC", "yyyy-mm-ddTHH:MM:SS.zzzZ")</code>	<i>on 1/31/17 at 12:39:13.310 PM in MST</i>	2017-01-31T19:39:13.310Z
<code>CVTDate(myDate, "mmddyy", "%Y%m%d%a")</code>	061994	19940619Sun
<code>CVTDate(myDate, "mmddyy", "%Y%m %d %a")</code>	061994	1994 06 19 Sun
<code>CVTDate(myDate, "mmddyy", "%Y%m %d %A")</code>	061994	1994 06 19 Sunday

Format Strings

The format of both the input date string and the output date string are specified in the same way by using either characters for placeholders or one of the four predefined date formats (`GeneralDate`, `LongDate`, `MediumDate`, or `ShortDate`).

For inputs, the year, month, day, hour, minute, second, and AM/PM are read from the date string at the position where `y`, `m`, `d`, `H`, `M`, `S`, `SSS`, and `AP` appear in the input format string.

For outputs, the format string is copied to the data string with the year, month, day, hour, minute, and second replacing the characters of `y`, `m`, `d`, `H`, `M`, `S`, `SSS`, and `AP`.

Alternatively, the substitutions used in the ANSI C method `strftime` can be used. For example, the string `"%Y-%m-%d"` would result in the [ISO 8601](#) date format (`yyyy-mm-dd`). See the [external documentation on `strftime`](#) for a detailed description.

These codes can appear as required in the format string to specify how to read the input or to write the output:

Format	Input or Output
YYYY	4-digit year
yy	(last) 2-digit year
??yy	2- or 4-digit year
mmm	Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec
mm	2-digit month (01-12)
?m	1- or 2-digit month (1-12)
ddd	Julian date (001-366)

dd	2-digit day (01-31)
?d	1- or 2-digit day (1-31)
HH	Hour (00-23) or (01-12) if AM/PM is specified using AP
MM	Minute (00-59)
SS	Second (00-59)
zzz	Millisecond (.001-.999)
AP	AM or PM
GeneralDate	6/19/94 5:34:23 PM
LongDate	Sunday, June 19, 1994
MediumDate	19-Jun-1994
ShortDate	6/19/94

[\[Back to Top\]](#)

DateAdd

Declaration

```
string DateAdd(string datePart, int number, date d)
string DateAdd(string datePart, int number, string d)
```

Syntax

```
DateAdd(<datePart>, <number>, <d>)
```

Required Parameters

- **datePart**: A part code describing the part of the date to apply the addition to
- **number**: An integer number to be added to the date part
- **d**: The date object or a date string to be operated on

Description

Returns a date string after adding a number to a specified part of a date object.

These codes are used to describe the date parts:

Date Part	Part Code
Year	YYYY
Month	mm
Day	dd
Hour	hh
Minute	mi
Second	ss
Millisecond	zzz

Examples

```
DateAdd("yyyy", 1, Now());
// Adding one year to a date
// If Now() returns 2017-03-10 18:46:41
// returns "2018-03-10 18:46:41"

DateAdd("dd", 1, "2017-03-10");
// Adding one day to a date
// Returns "2017-03-11"
```

[\[Back to Top\]](#)

DayOfMonth

Declaration

```
int DayOfMonth(date d)

int DayOfMonth(string d)
```

Syntax

```
DayOfMonth(<d>)
```

Required Parameters

- **d**: A date object or date string

Description

Returns the day of the month (1-31) of a date object or date string.

Examples

```
DayOfMonth(Now());
// Returns 25 if today is December 25
```

[\[Back to Top\]](#)

DayOfWeek

Declaration

```
int DayOfWeek(date d)

int DayOfWeek(string d)
```

Syntax

```
DayOfWeek(<d>)
```

Required Parameters

- **d**: A date object or date string

Description

Returns the day of the week for a date object or date string, with 0 for Sunday, 1 for Monday, on through 6 for Saturday.

This definition is independent of locale. For the weekday name, call `FormatDate()` instead.

Examples

```
DayOfWeek(Now());
// Returns 0, if today is Sunday

Now() + (6 - DayOfWeek(Now()))*24*60*60;
// Returns the date object of the last day of the current week,
// assuming Saturday (day 6) is the last day of the week
```

[\[Back to Top\]](#)

FormatDate

Declaration

```
string FormatDate(date d, string format)

string FormatDate(string d, string format)
```

Syntax

```
FormatDate(<d>, <format>)
```

Required Parameters

- **d**: A date object or date string
- **format**: A [format string](#), specifying the format of the output date

Description

Converts a date object to a string according to a [format string](#). This is similar to the `CVTDate()` function and uses the same format strings.

Examples

```
FormatDate(Now(), "%w");
// Similar to DayOfWeek(Now()), except the return value
// is a string value of today's weekday name, such as "Wed"

FormatDate("2017-12-07", "%A");
// Returns the weekday name of the date string, such as "Thursday"

LPadChar(String(Long(FormatDate(d, "yyyy")) - 1900), "0", 3) + FormatDate(
d, "ddd");
// Returns, for a date d, a string formatted as a JD Edwards Date
("CYYDDD")
// If d = "2000-12-31" (a leap year), returns "100366"
// See https://docs.oracle.com/cd/E26228_01/doc.93/e21961/julian_date_conv.
htm#WEAWX259
```

[\[Back to Top\]](#)

GeneralDate

Declaration

```
string GeneralDate(date d)

string GeneralDate(string d)
```

Syntax

```
GeneralDate(<d>)
```

Required Parameters

- **d**: A date object or date string

Description

Returns a string in the [general date format](#) for a date object or date string.

Examples

```
GeneralDate(Now());
// If Now() is 11:59:23 am on September 16, 2000,
// returns "09/16/2000 11:59:23 AM"

GeneralDate("2017-12-07");
// Returns "12/07/2017 12:00:00 AM"
```

[\[Back to Top\]](#)

GetUTCFormattedDate

Declaration

```
string GetUTCFormattedDate(date d, string time_zone_id[, bool
is_european_format])

string GetUTCFormattedDate(string d, string time_zone_id[, bool
is_european_format])
```

Syntax

```
GetUTCFormattedDate(<d>, <time_zone_id>[, <is_european_format>])
```

Required Parameters

- **d**: A date object or date string
- **time_zone_id**: The time zone to be converted from, expressed as a [time zone code](#)

Optional Parameters

- **is_european_format**: This optional flag determines if the European date format is to be used. When `true`, the format is day and then month; when `false`, the format is month and then day.

Description

Returns a date string without time information. Converts a date object or date string to a string according to a time zone code.

WARNING: If passed a date with a time, the `GetUTCFormattedDate()` function truncates the time **before** converting to UTC. This means that timestamps that are after midnight UTC will be returned as occurring on the date before, as shown in the example above for the "America/Los_Angeles" time zone.

NOTE: If you are using `GetUTCFormattedDate` in conjunction with `Now()` or `Now_()`, be aware that the time zone returned is the one configured on the machine running the agent. All Cloud Agents are in UTC. Private Agents will vary based on the specific machine each agent is running on. If you are using a conversion function, such as `GetUTCFormattedDate`, take this into account if your project is designed to run both in the cloud and on a Private Agent.

Examples

```
// If Now() is 2017-12-09 18:46:41, then:

GetUTCFormattedDate(Now(), "UTC", false);
// Returns "2017-12-09"

GetUTCFormattedDate(Now(), "America/Los_Angeles", false);
// Returns "2017-12-09"
```

[\[Back to Top\]](#)

GetUTCFormattedDateTime

Declaration

```
string GetUTCFormattedDateTime(date d, string time_zone_id[, bool
is_european_format])

string GetUTCFormattedDateTime(string d, string time_zone_id[, bool
is_european_format])
```

Syntax

```
GetUTCFormattedDateTime(<d>, <time_zone_id>[, <is_european_format>])
```

Required Parameters

- **d:** A date object or date string
- **time_zone_id:** The time zone to be converted from, expressed as a [time zone code](#)

Optional Parameters

- **is_european_format:** This optional flag determines if the European date format is to be used. When `true`, the format is day and then month; when `false`, the format is month and then day.

Description

Returns a date string with time information. Converts a date object or date string to a string according to a time zone code.

NOTE: If you are using `GetUTCFormattedDateTime` in conjunction with `Now()` or `Now_()`, be aware that the time zone returned is the one configured on the machine running the agent. All Cloud Agents are in UTC. Private Agents will vary based on the specific machine each agent is running on. If you are using a conversion function, such as `GetUTCFormattedDateTime`, take this into account if your project is designed to run both in the cloud and on a Private Agent.

Examples

```
// If Now() is 2017-12-09 18:46:21, then:

GetUTCFormattedDateTime(Now(), "UTC", false);
// Returns "2017-12-09T18:46:21Z"

GetUTCFormattedDateTime(Now(), "America/Los_Angeles", false);
// Returns "2017-12-10T02:46:21Z"
```

[\[Back to Top\]](#)

LastDayOfMonth

Declaration

```
date LastDayOfMonth(date d)

date LastDayOfMonth(string d)
```

Syntax

```
LastDayOfMonth(<d>)
```

Required Parameters

- **d**: A date object or date string

Description

Returns a date object representing the last day of the month for a date object or date string.

Examples

```
LastDayOfMonth(Now());
// If today is in February of a leap year,
// returns the date object of February 29
```

[\[Back to Top\]](#)

LongDate

Declaration

```
string LongDate(date d)

string LongDate(string d)
```

Syntax

```
LongDate(<d>)
```

Required Parameters

- **d**: A date object or date string

Description

Returns a string in the [long date format](#) for a date object or date string.

Examples

```
LongDate(Now());  
// If Now() is 11:59:23 am on September 16, 2000,  
// returns "Saturday, September 16, 2000"
```

[\[Back to Top\]](#)

LongTime

Declaration

```
string LongTime(date d)  
  
string LongTime(string d)
```

Syntax

```
LongTime(<d>)
```

Required Parameters

- **d**: A date object or date string

Description

Returns a string in the long time format for a date object or date string.

Examples

```
LongTime(Now());  
// If Now() is 11:59:23 am on September 16, 2000  
// returns "11:59:23 AM"
```

[\[Back to Top\]](#)

MediumDate

Declaration

```
string MediumDate(date d)  
  
string MediumDate(string d)
```

Syntax

```
MediumDate(<d>)
```

Required Parameters

- **d**: A date object or date string

Description

Returns a string in the [medium date format](#) for a date object or date string.

Examples

```
MediumDate(Now());  
// If Now() is 11:59:23 am on September 16, 2000  
// returns "16-Sep-00"
```

[\[Back to Top\]](#)

MediumTime

Declaration

```
string MediumTime(date d)  
  
string MediumTime(string d)
```

Syntax

```
MediumTime(<d>)
```

Required Parameters

- **d**: A date object or date string

Description

Returns a string in the medium time format for a date object or date string.

Examples

```
MediumTime(Now());  
// If Now() is 11:59:23 am on September 16, 2000  
// returns "11:59 AM"
```

[\[Back to Top\]](#)

MonthOfYear

Declaration

```
int MonthOfYear(date d)  
  
int MonthOfYear(string d)
```

Syntax

```
MonthOfYear(<d>)
```

Required Parameters

- **d**: A date object or date string

Description

Returns the month (1-12) for a date object or date string.

Examples

```
MonthOfYear(Now());
// If Now() is 11:59:23 am on September 16, 2000
// returns "9"
```

[\[Back to Top\]](#)

Now

Declaration

```
date Now()
```

Syntax

```
Now()
```

Description

Returns a date object representing the date and time values at the moment the function was run. The fraction of the second is truncated.



NOTE: Be aware that the time zone of the date returned is the one configured on the machine running the agent. All Cloud Agents are in UTC. Private Agents will vary based on the specific machine each agent is running on. If you are using a conversion functions, take this into account if your project is designed to run both in the cloud and on a Private Agent.

Examples

```
Now();
// If Now() is 11:59:23 am on September 16, 2000,
// returns a date object representing 09/16/2000 11:59:23 AM
```

[\[Back to Top\]](#)

Now_

Declaration

```
date Now_()
```

Syntax

```
Now_()
```

Description

Returns a date object representing the date and time values at the moment the function was run. The time value includes the fraction of second (milliseconds).

i NOTE: Be aware that the time zone of the date returned is the one configured on the machine running the agent. All Cloud Agents are in UTC. Private Agents will vary based on the specific machine each agent is running on. If you are using a conversion functions, take this into account if your project is designed to run both in the cloud and on a Private Agent.

Examples

```
Now_();
// If Now_() is 11:59:23.123 am on September 16, 2000
// returns a date object representing 09/16/2000 11:59:23.123 AM
```

[\[Back to Top\]](#)

ShortDate

Declaration

```
string ShortDate(date d)
string ShortDate(string d)
```

Syntax

```
ShortDate(<d>)
```

Required Parameters

- **d:** A date object or date string

Description

Returns a string in the [short date format](#) for a date object or date string.

Examples

```
ShortDate(Now());
// If Now() is 11:59:23 am on September 16, 2017
// returns "9/16/17"
```

[\[Back to Top\]](#)

ShortTime

Declaration

```
string ShortTime(date d)
string ShortTime(string d)
```

Syntax

```
ShortTime(<d>)
```

Required Parameters

- **d**: A date object or date string

Description

Returns a string in the short time format for a date object or date string.

Examples

```
ShortTime(Now());  
// If Now() is 11:59:23 am on September 16, 2017  
// returns "11:59"
```

[\[Back to Top\]](#)