缺省日期格式为 %yyyy-%mm-%dd。

### DateToString

返回给定日期的字符串表示。可以指定字符串的格式（可选）。

* **输入**：date (date)，[format (string)]
* **输出**：result (string)
* **示例**。以下示例会将 mylink.mydate 列中包含的日期输出为字符串。如果 mylink.mydate 包含日期 2009 年 8 月 18 日，那么输出字符串为“2009-08-18”：

DateToString(mylink.mydate)

以下示例会将 mylink.mydate 列中包含的日期输出为 dd:mm:yyyy 格式的字符串。如果 mylink.mydate 包含日期 2009 年 8 月 18 日，那么输出字符串为“18:08:2009”：

DateToString(mylink.mydate, "%dd:%mm:%yyyy")

### DateToDecimal

将给定的日期返回为压缩十进制值。如果您的目标十进制数指定了小数位，那么部分日期会在小数点后显示。您可以指定格式字符串来指定如何以十进制数形式存储日期（可选）。缺省格式字符串为“%yyyy%mm%dd”，因此，日期 2009-08-25 将存储为十进制数 20090825。格式字符串只能指定包含数字的格式。例如，不能指定诸如“%yyyy-%mm-%dd”之类的格式字符串，这是因为压缩十进制值中不能存储连字符 (-)。以下标记对于与十进制值之间的转换有效：

%yyyy（四位数表示的年份）

%yy（两位数表示的年份）

%NNNNyy（截断的两位数年份）

%mm（两位数表示的月份）

%dd（两位数表示的一月中的第几天）

%ddd（三位数表示的一年中的第几天）

字面值数字 0 到 9 都有效。

* **输入**：basedate (date) [, format (string)]
* **输出**：converted\_date (decimal)
* **示例**。如果 mylink.basedate 列包含日期 2012-08-18，那么以下函数会将日期存储为十进制数 18082012：

DateToDecimal (mylink.basedate, "%dd%mm%yyyy")

如果 mylink.basedate 列包含日期 2012-08-18，并且目标列的长度为 10，小数位为 2，那么以下函数会将日期存储为十进制数 201208.18：

DateToDecimal (mylink.basedate)

### StringToDate

以给定的格式返回给定字符串表示的日期。如果字符串包含缺省格式 yyyy-mm-dd 的日期，那么不必指定格式字符串。

* **输入**：string (string) [,format (string)]
* **输出**：result (date)
* **示例**：如果 mylink.mystring 列包含字符串“1958–08–18”，那么以下函数将返回日期 1958–08–18。

StringToDate(mylink.mystring)

如果 mylink.mystring 列包含字符串“18:08:1958”，那么以下函数将返回日期 1958–08–18

StringToDate(mylink.mystring,"%dd:%mm:%yyyy")

### StringToTime

返回给定字符串的时间表示。

* **输入**：string (string)，[format (string)]
* **输出**：result (time)
* **示例**：如果 mylink.mystring 列包含字符串“20:06:58”，那么该函数将返回时间 20:06:58。

StringToTime(mylink.mystring)

如果 mylink.mystring 列包含字符串“20:6:58”，那么该函数将返回时间 20:06:58。

StringToTime(mylink.mystring,"%(h,s):$(n,s):$(s,s)")

### StringToTimestamp

返回给定字符串的时间表示。

* **输入**：string (string) [format (string)]
* **输出**：result (time)
* **示例**：如果 mylink.mystring 列包含字符串“1958–08–08 20:06:58”，那么该函数将返回时间戳记 1958–08–08 20:06:58。

StringToTimestamp(mylink.mystring)

如果 mylink.mystring 列包含字符串“8/ 8/1958 20: 6:58”，那么该函数将返回时间戳记 1958–08–08 20:06:58。

StringToTimestamp(mylink.mystring, "%(d,s)/%(m,s)/%yyyy%(h,s):$(n,s):$(s,s)")

### TimestampToDate

返回给定时间戳记的日期。

* **输入**：timestamp (timestamp)
* **输出**：result (date)
* **示例**：如果 mylink.mytimestamp 列包含时间戳记 1958–08–18 20:06:58，那么以下函数将返回日期 1958–08–18：

TimestampToDate(mylink.mytimestamp)

### TimestampToDecimal

将给定的时间戳记返回为压缩十进制。您可以指定格式字符串来指定如何以十进制数形式存储时间戳记（可选）。缺省格式字符串为“%yyyy%mm%dd%hh%nn%ss”，因此，时间戳记 2009-08-25 14:03:22 将存储为十进制数 20090825140322。格式字符串只能指定包含数字的格式。例如，不能指定诸如“%yyyy/%mm/%dd%hh:%nn:%ss”之类的格式字符串，这是因为压缩十进制值中不能存储斜杠 (/) 和冒号 (:)。以下标记对于与十进制值之间的转换有效：

%yyyy（四位数表示的年份）

%yy（两位数表示的年份）

%NNNNyy（截断的两位数年份）

%mm（两位数表示的月份）

%dd（两位数表示的一月中的第几天）

%ddd（三位数表示的一年中的第几天）

%hh（使用 24 小时制的以两位数表示的小时数）

%nn（两位数表示的分钟数）

%ss（两位数表示的秒数）

%ss.*N*（由两位数表示的秒数加上所允许的小数位数。小数位数从 1 到 6，包括 1 和 6）。

字面值数字 0 到 9 都有效。

如果指定的格式包含微秒（例如，%ss.4），那么将在十进制值中推断小数点的位置。小数点的位置不必与指定的十进制小数位一致（例如，小数位 = 4）。

* **输入**：timestamp (timestamp) [, format (string)]
* **输出**：result (decimal)
* **示例**：如果 mylink.mytimestamp 列包含时间戳记 1958–08–18 20:06:58，那么以下函数将返回十进制值 19580818200658：

TimestampToDecimal(mylink.mytimestamp)

如果 mylink.mytimestamp 列包含时间戳记 1958–08–18 20:06:58，那么以下函数将返回十进制值 200658580818：

TimestampToDecimal(mylink.mytimestamp, "%hh%nn%ss%yy%mm%dd")

### TimestampToString

返回给定时间戳记的字符串。

* **输入**：timestamp (timestamp) [format (string)]
* **输出**：result (string)
* **示例**：如果 mylink.mytimestamp 列包含时间戳记 1958–08–1820:06:58，那么该函数将返回字符串“1958–08–1820:06:58”。

TimestampToString(mylink.mytimestamp)

如果 mylink.mytimestamp 列包含时间戳记 1958–08–1820:06:58，那么该函数将返回字符串“18/08/1958 20:06:58”：

TimestampToString(mylink.mytimestamp, "%dd/%mm/%yyyy %hh:$nn:$ss")

### TimestampToTime

返回给定时间戳记的字符串表示。

* **输入**：timestamp (timestamp)
* **输出**：result (time)
* **示例**：如果 mylink.mytimestamp 列包含时间戳记 1958–08–1820:06:58，那么该函数将返回时间 20:06:58：

TimestampToTime(mylink.mytimestamp)

### TimeToString

返回给定时间的字符串。

* **输入**：timestamp (timestamp) [format (string)]
* **输出**：result (time)
* **示例**：如果 mylink.mytime 列包含时间 20:06:58，那么以下函数将返回字符串“20:06:58”：

TimeToString(mylink.mytime)

如果 mylink.mytime 列包含时间 20:06:58，那么以下函数将返回字符串“58:06:20”：

TimeToString(mylink.mytime, "%ss:$nn:$hh")

### TimeToDecimal

将给定的时间返回为压缩十进制。您可以指定格式字符串来指定如何以十进制数形式存储时间（可选）。缺省格式字符串为“%hh%nn%ss”，因此，时间 14:03:22 将存储为十进制数 140322。格式字符串只能指定包含数字的格式。例如，不能指定诸如“%hh:%nn:%ss”之类的格式字符串，这是因为压缩十进制值中不能存储冒号 (:)。以下标记对于与十进制值之间的转换有效：

%hh（使用 24 小时制的以两位数表示的小时数）

%nn（两位数表示的分钟数）

%ss（两位数表示的秒数）

%ss.*N*（由两位数表示的秒数加上所允许的小数位数。小数位数从 1 到 6，包括 1 和 6）。

字面值数字 0 到 9 都有效。

如果指定的格式包含微秒（例如，%ss.4），那么将在十进制值中推断小数点的位置。小数点的位置不必与指定的十进制小数位一致（例如，小数位 = 4）。

* **输入**：time (time) [, format (string)]
* **输出**：result (decimal)
* **示例**：如果 mylink.mytime 列包含时间 20:06:58，那么以下函数将返回十进制值 200658：

TimeToDecimal(mylink.mytime)

如果 mylink.mytime 列包含时间 20:06:58，那么以下函数将返回十进制值 580620：

TimeToDecimal(mylink.mytime, "%ss%nn%hh")

### DecimalToDate

将给定的压缩十进制数返回为日期。在将十进制数转换为日期时，会忽略其符号和小数位。您可以指定格式字符串来指定如何以十进制数形式存储日期（可选）。缺省格式字符串为“%yyyy%mm%dd”，因此，日期 2009-08-25 将存储为十进制数 20090825。格式字符串只能指定包含数字的格式。例如，不能指定诸如“%yyyy-%mm-%dd”之类的格式字符串，这是因为压缩十进制值中不能存储连字符 (-)。以下标记对于与十进制值之间的转换有效：

%yyyy（四位数表示的年份）

%yy（两位数表示的年份）

%NNNNyy（截断的两位数年份）

%mm（两位数表示的月份）

%dd（两位数表示的一月中的第几天）

%ddd（三位数表示的一年中的第几天）

字面值数字 0 到 9 都有效。

* **输入**：basedec (decimal) [, format (string)]
* **输出**：date
* **示例**。如果 mylink.mydecdata 列包含值 18082012，那么以下函数将返回日期 2012-08-18：

DecimalToDate (mylink.basedate, "%dd%mm%yyyy")

如果 mylink.mydecdata 列包含值 -201208.18，那么以下函数将返回日期 2012-08-18：

DecimalToDate (mylink.basedate)

### DecimalToTime

将给定的压缩十进制数返回为时间。您可以指定格式字符串来指定如何以十进制数形式存储时间（可选）。缺省格式字符串为“%hh%nn%ss”，因此，时间 14:03:22 将存储为十进制数 140322。格式字符串只能指定包含数字的格式。例如，不能指定诸如“%hh:%nn:%ss”之类的格式字符串，这是因为压缩十进制值中不能存储冒号 (:)。以下标记对于与十进制值之间的转换有效：

%hh（使用 24 小时制的以两位数表示的小时数）

%nn（两位数表示的分钟数）

%ss（两位数表示的秒数）

%ss.*N*（由两位数表示的秒数加上所允许的小数位数。小数位数从 1 到 6，包括 1 和 6）。

字面值数字 0 到 9 都有效。

如果指定的格式包含微秒（例如，%ss.4），那么将在十进制值中推断小数点的位置。小数点的位置不必与指定的十进制小数位一致（例如，小数位 = 4）。

* **输入**：time (time) [, format (string)]
* **输出**：result (decimal)
* **示例**：如果 mylink.mytimedec 列包含十进制值 200658，那么以下函数将返回时间 20:06:58：

DecimalToTime(mylink.mytimedec)

如果 mylink.mytimedec 列包含十进制值 580620，那么以下函数将返回时间 20:06:58：

DecimalToTime(mylink.mytimedec, "%ss%nn%hh")

### DecimalToTimestamp

将给定的压缩十进制数返回为时间戳记。您可以指定格式字符串来指定如何以十进制数形式存储时间戳记（可选）。缺省格式字符串为“%yyyy%mm%dd%hh%nn%ss”，因此，时间戳记 2009-08-25 14:03:22 将存储为十进制数 20090825140322。格式字符串只能指定包含数字的格式。例如，不能指定诸如“%yyyy/%mm/%dd%hh:%nn:%ss”之类的格式字符串，这是因为压缩十进制值中不能存储斜杠 (/) 和冒号 (:)。以下标记对于与十进制值之间的转换有效：

%yyyy（四位数表示的年份）

%yy（两位数表示的年份）

%NNNNyy（截断的两位数年份）

%mm（两位数表示的月份）

%dd（两位数表示的一月中的第几天）

%ddd（三位数表示的一年中的第几天）

%hh（使用 24 小时制的以两位数表示的小时数）

%nn（两位数表示的分钟数）

%ss（两位数表示的秒数）

%ss.*N*（由两位数表示的秒数加上所允许的小数位数。小数位数从 1 到 6，包括 1 和 6）。

字面值数字 0 到 9 都有效。

如果指定的格式包含微秒（例如，%ss.4），那么将在十进制值中推断小数点的位置。小数点的位置不必与指定的十进制小数位一致（例如，小数位 = 4）。

* **输入**：timestamp (timestamp) [, format (string)]
* **输出**：result (decimal)
* **示例**：如果 mylink.mytimestampdec 列包含值 19580818200658，那么以下函数将返回时间戳记 1958–08–18 20:06:58：

DecimalToTimestamp(mylink.mytimestampdec)

如果 mylink.mytimestampdec 列包含十进制值 200658580818，那么以下函数将返回时间戳记 1958–08–18 20:06:58：

DecimalToTimestamp(mylink.mytimestampdec, "%hh%nn%ss%yy%mm%dd")

### IsValidDate

返回给定的值是否为有效的 date 类型。

* **输入**：testdate (date)
* **输出**：result (int8)
* **示例**。如果 mylink.mydate 列包含日期 2011-09-13，那么以下函数将返回值 1。

IsValidDate(mylink.mydate)

如果 mylink.mydate 列包含字符串“380096.06”，那么以下函数将返回值 0，这是因为转换后的字符串不是有效的 date 类型。

IsValidDate(StringTodate (mylink.mydate))

**IsValidDecimal**

返回给定的值是否为有效的 decimal 类型。如果将 allzerosflag 设置为 0，那么全零表示将视为无效。缺省情况下，allzerosflag 设置为 0。

* **输入**：testvalue (decimal) [, allzerosflag (uint8)]
* **输出**：result (int8)
* **示例**。如果 mylink.mynum 列包含值 310007.65，那么以下函数将返回值 1。

IsValidDecimal(mylink.mynum)

如果 mylink.mynum 列包含字符串“wake-robin”，那么以下函数将返回值 0，这是因为转换后的字符串不是有效的 decimal 类型。

IsValidDecimal(StringToDecimal (mylink.mynum))

### IsValidTime

返回给定的时间是否为有效的 time 类型。

* **输入**：testtime (time)
* **输出**：result (int8)
* **示例**。如果 mylink.mytime 列包含时间 23:09:22，那么以下函数将返回值 1：

IsValidTime(mylink.mytime)

如果 mylink.mydate 列包含字符串“IbnKayeed”，那么以下函数将返回值 0，这是因为转换后的字符串不是有效的 time 类型。

IsValidTime(StringToTime (mylink.mytime))

### IsValidTimestamp

返回给定的时间戳记是否为有效的 timestamp 类型。

* **输入**：testtimestamp (timestamp)
* **输出**：result (int8)
* **示例**。如果 mylink.mytimestamp 列包含时间 2011-09-13 23:09:22，那么以下函数将返回值 1：

IsValidTimestamp(mylink.mytimestamp)

如果 mylink.mytimestamp 列包含字符串“one of two”，那么以下函数将返回值 0，这是因为转换后的字符串不是有效的 timestamp 类型。

IsValidTimestamp(StringToTimestamp (mylink.mytimestamp))

### Trim

除去所有前导空格、结尾空格和跳格，并将内部出现次数减少为 1。参数 *stripchar* 指定除空格或跳格之外的字符（可选）。参数 *options* 指定要执行的 Trim 操作的类型，并包含以下某一个或多个值（可选）：

**A** 除去所有出现的 *stripchar*

**B** 除去出现的前导和结尾 *stripchar*

**D** 除去前导、结尾和冗余空格字符

**E** 除去结尾空格字符

**F** 除去前导空格字符

**L** 除去所有出现的前导 *stripchar*

**R** 除去出现的前导、结尾和冗余 *stripchar*

**T**除去所有出现的结尾 *stripchar*

* **输入**：string (string) [stripchar (string)] [options (string)]
* **输出**：result (string)
* **示例**。如果 mylink.mystring 包含字符串“ String with whitespace ”，那么以下函数将返回字符串“String with whitespace”：

Trim(mylink.mystring)

如果 mylink.mystring 包含字符串“..Remove..redundant..dots....”，那么以下函数将返回字符串“Remove.redundant.dots”：

Trim(mylink.mystring,".")

如果 mylink.mystring 包含字符串“Remove..all..dots....”，那么以下函数将返回字符串“Removealldots”：

Trim(mylink.mystring,".","A")

如果 mylink.mystring 包含字符串“Remove..trailing..dots....”，那么以下函数将返回字符串“Remove..trailing..dots”：

Trim(mylink.mystring,".","T")

### TrimB

除去字符串中所有结尾空格和跳格。

* **输入**：string (string)
* **输出**：result (string)
* **示例**。如果 mylink.mystring 包含字符串“too many trailing spaces    ”，那么以下函数将返回字符串“too many trailing spaces”：

TrimB(mylink.mystring)

### TrimF

除去字符串中所有前导空格和跳格。

* **输入**：string (string)
* **输出**：result (string)
* **示例**。如果 mylink.mystring 包含字符串“    too many leading spaces”，那么以下函数将返回字符串“too many leading spaces”：

TrimF(mylink.mystring)

### TrimLeadingTrailing

除去字符串中所有前导空格、结尾空格和跳格。

* **输入**：string (string)
* **输出**：result (string)
* **示例**。如果 mylink.mystring 包含字符串“ too many spaces ”，那么以下函数将返回字符串“too many spaces”：

TrimLeadingTrailing(mylink.mystring)

## 日期和时间函数

您可以使用日期和时间函数对 Transformer 阶段中的日期和时间执行各种操作。

在自变量中指定日期、时间或时间戳记的函数使用具有特定格式的字符串：

* 对于日期，格式为 %*yyyy*-%*mm*-%*dd*
* 对于时间，格式为 %*hh*:%*nn*:%*ss*。如果扩展为包含微秒，那么格式为 %*hh*:%*nn*:%*ss.x*，其中 *x* 是为秒指定的小数位数。
* 对于时间戳记，格式为 %*yyyy*-%*mm*-%*dd* %*hh*:%*nn*:%*ss*。如果扩展为包含微秒，那么格式为 %*yyyy*-%*mm*-%*dd* %*hh*:%*nn*:%*ss.x*，其中 *x* 是为秒指定的小数位数。

在自变量中包含星期几的函数使用字符串来指定星期几。星期几可指定为三个字母的缩写，也可指定为完整名称。例如，字符串“thu”和“thursday”都是有效的。

以下函数位于表达式编辑器的日期和时间类别中。方括号表示参数是可选的。以下示例按照 Transformer 阶段的“派生”字段中所示来显示这些函数。

### CurrentDate

以日期格式返回作业运行的日期。

* **输入**：-
* **输出**：date
* **示例**。您可以使用该函数将包含日期的新列添加到 Transformer 阶段输出的数据：

CurrentDate()

### CurrentTime

以时间格式返回作业运行的时间。

* **输入**：-
* **输出**：time
* **示例**。您可以使用该函数将包含时间的新列添加到 Transformer 阶段输出的数据：

CurrentTime()

### CurrentTimeMS

以时间格式返回作业运行的时间，包含微秒。

* **输入**：-
* **输出**：time
* **示例**。您可以使用该函数将包含时间的新列添加到 Transformer 阶段输出的数据。必须将列元数据中的扩展字段设置为微秒以包含完整的时间：

CurrentTimeMS()

### CurrentTimestamp

以时间戳记格式返回作业运行的时间戳记（指定日期和时间）。

* **输入**：-
* **输出**：timestamp
* **示例**。您可以使用该函数将包含时间戳记的新列添加到 Transformer 阶段输出的数据：

CurrentTimestamp()

### CurrentTimestampMS

以时间戳记格式返回作业运行的时间戳记（指定日期和时间），时间部分包含微秒。

* **输入**：-
* **输出**：timestamp
* **示例**。您可以使用该函数将包含时间戳记的新列添加到 Transformer 阶段输出的数据。必须将列元数据中的扩展字段设置为微秒以包含完整的时间戳记：

CurrentTimestampMS()

### DateFromDaysSince

通过向基线日期添加一个整数来返回一个日期。该整数可以是负数，这样会返回比该基线日期更早的日期。

* **输入**：number (int32), [baseline\_date (date)]
* **输出**：date
* **示例**。如果 mylink.myintcol 包含整数 18250，mylink.mydatecol 包含日期 1958–08–18，那么以下三个函数是等效的，都返回日期 2008–08–05：
* DateFromDaysSince(18250,"1958-08-18")
* DateFromDaysSince(mylink.myintcol,"1958-08-18")

DateFromDaysSince(mylink.myintcol,mylink.mydatecol)

如果 mylink.mynegintcol 包含整数 -1，mylink.mydatecol 包含日期 1958–08–18，那么以下三个函数是等效的，都返回日期 1958–08–17：

DateFromDaysSince(-1,"1958-08-18")

DateFromDaysSince(mylink.mynegintcol,"1958-08-18")

DateFromDaysSince(mylink.mynegintcol,mylink.mydatecol)

### DateFromComponents

根据由三个单独的值指定的年月日来返回一个日期。

* **输入**：(int32), months (int32), dayofmonth (int32)
* **输出**：date
* **示例**。如果 mylink.yearcol 包含值 2010，mylink.monthcol 包含值 12，mylink.dayofmonthcol 包含值 2，那么以下两个函数是等效的，都返回日期 2010–12–02。
* DateFromComponents(2010, 12, 2)

DateFromComponents(mylink.yearcol, mylink.monthcol, mylink.dayofmonthcol)

### DateFromJulianDay

根据指定的儒略日返回一个日期。

* **输入**：julianday (uint32)
* **输出**：date
* **示例**。如果 mylink.myjulcol 包含值 2454614，那么以下两个函数是等效的，都返回日期 2008–05–27。
* DateFromJulianDay(2454614)

DateFromJulianDay(mylink.myjulcol)

### DateOffsetByComponents

返回指定的日期应用了指定的年偏移量、月偏移量和日偏移量（每个偏移量都以单独的值指定）之后所得到的日期。每个偏移值都可以是正数、零或负数。

* **输入**：basedate (date), year\_offset (int32), month\_offset (int32), dayofmonth\_offset (int32)
* **输出**：date
* **示例**。如果 mylink.basedate 包含 2011-08-18，mylink.yearos 包含值 2，mylink.monthos 包含值 0，mylink.dayofmonthosol 包含值 0，那么以下两个函数是等效的，都返回日期 2013–08–18。
* DateOffsetByComponents("2011-08-18", 2011, 8, 18)
* DateOffsetByComponents(mylink.basedate, mylink.yearos, mylink.monthos,

mylink.dayofmonthos)

* 如果 mylink.basedate 包含 2011-08-18，mylink.yearos 包含值 -2，mylink.monthos 包含值 0，mylink.dayofmonthosol 包含值 0，那么以下两个函数是等效的，都返回日期 2009–08–18。
* DateOffsetByComponents("2011-08-18", 2011, 8, 18)
* DateOffsetByComponents(mylink.basedate, mylink.yearos, mylink.monthos,

mylink.dayofmonthos)

### DaysSinceFromDate

返回源日期到指定日期之间的天数。

* **输入**：source\_date, given\_date
* **输出**：days\_since (int32)
* **示例**。如果 mylink.mysourcedate 包含日期 1958–08–18，mylink.mygivendate 包含日期 2008–08–18，那么以下两个函数是等效的，都返回整数值 18263。
* DaysSinceFromDate(mylink.mysourcedate, mylink.mygivendate)

DaysSinceFromDate("1958-08-18","2008-08-18")

### DaysInMonth

返回指定的基本日期中月份的天数。

* **输入**：basedate (date)
* **输出**：daysinmonth (int32)
* **示例**。如果 mylink.mysourcedate 包含日期 1958–08–18，那么以下两个函数是等效的，都返回整数值 31。
* DaysInMonth(mylink.mysourcedate)

DaysInMonth("1958-08-18")

### DaysInYear

返回指定的基本日期中年份的天数。

* **输入**：basedate (date)
* **输出**：daysinyearh (int32)
* **示例**。如果 mylink.mysourcedate 包含日期 2012–08–18，那么以下两个函数是等效的，都返回整数值 366。
* DaysInYear(mylink.mysourcedate)

DaysInYear("2012-08-18")

如果 mylink.mysourcedate 包含日期 2011–08–18，那么以下两个函数是等效的，都返回整数值 365。

DaysInYear(mylink.mysourcedate)

DaysInYear("2011-08-18")

### DateOffsetByDays

返回指定的日期应用了指定天数之后所得到的日期。偏移值可以是正数、零或负数。

* **输入**：basedate (date), dayoffset (int32)
* **输出**：date
* **示例**。如果 mylink.basedate 包含 2011-08-18，mylink.dayoffset 包含值 2，那么以下两个函数是等效的，都返回日期 2011–08–20。
* DateOffsetByDays("2011-08-18", 2)

DateOffsetByDays(mylink.basedate, mylink.dayoffset)

* 如果 mylink.basedate 包含 2011-08-18，mylink.dayoffset 包含值 -31，那么以下两个函数是等效的，都返回日期 2011–07–18。
* DateOffsetByDays("2011-08-18", -31)

DateOffsetByDays(mylink.basedate, mylink.dayoffset)

### HoursFromTime

返回时间的小时部分。

* **输入**：time
* **输出**：hours (int8)
* **示例**。如果 mylink.mytime 包含时间 22:30:00，那么以下两个函数是等效的，都返回整数值 22。
* HoursFromTime(mylink.mytime)

HoursFromTime("22:30:00")

### MinutesFromTime

返回时间的分钟部分。

* **输入**：time
* **输出**：minutes (int8)
* **示例**。如果 mylink.mytime 包含时间 22:30:52，那么以下两个函数是等效的，都返回值 30：
* MinutesFromTime("22:30:52")

MinutesFromTime(mylink.mytime)

### MonthDayFromDate

返回指定日期的日部分。

* **输入**：date
* **输出**：day (int8)
* **示例**。如果 mylink.mydate 包含日期 2008-08-18，那么以下两个函数是等效的，都返回值 18：
* MonthDayFromDate("2008-08-18")

MonthDayFromDate(mylink.mydate)

### MonthFromDate

返回指定日期的月份。

* **输入**：date
* **输出**：month\_number (int8)
* **示例**。如果 mylink.mydate 包含日期 2008-08-18，那么以下两个函数是等效的，都返回值 8：
* MonthFromDate("2008-08-18")

MonthDayDate(mylink.mydate)

### TimeDate

以格式字符串形式返回系统时间和日期。

* **输入**：-
* **输出**：system\_time\_and\_date (string)
* **示例**。如果作业在 2008 年 6 月 20 日下午 4:21 运行，那么以下函数将返回字符串“16:21:48 20 Jun 2008”。

TimeDate()

### TimestampFromDateTime

返回指定的日期和时间的时间戳记。

* **输入**：date, time
* **输出**：timestamp
* **示例**。如果 mylink.mydate 包含日期 2008–08–18，mylink.mytime 包含时间 22:30:52，那么以下两个函数是等效的，都返回时间戳记 2008–08–18 22:30:52：
* TimestampFromDateTime("2008–08–18","22:30:52")

TimestampFromDateTime(mylink.mydate,mylink.mytime)

## Null 处理函数

