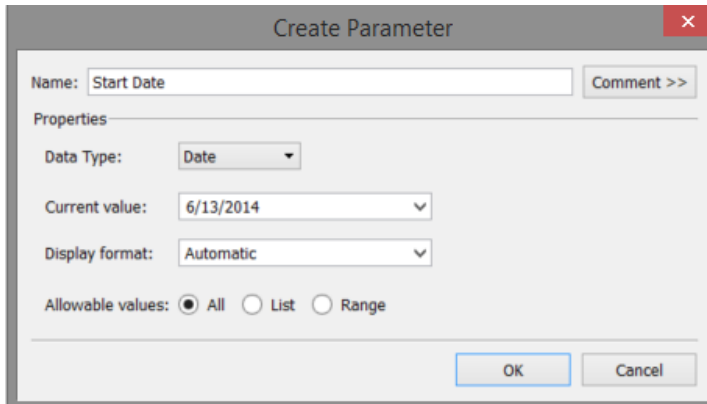


Date Documentation

Filtering by Date Using Parameters

1. In the Data window, click the drop-down arrow at the top right of Dimensions, and then select **Create > Parameter**.



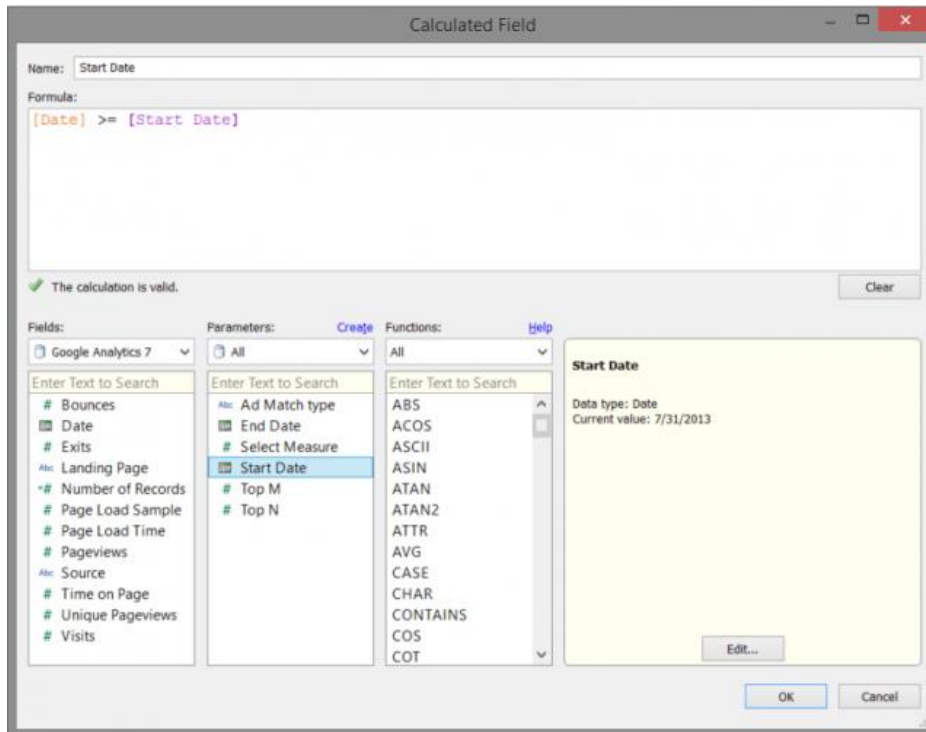
The screenshot shows a dialog box titled "Create Parameter". It has a "Name" field containing "Start Date" and a "Comment >>" button. Below this is a "Properties" section with three dropdown menus: "Data Type" set to "Date", "Current value" set to "6/13/2014", and "Display format" set to "Automatic". At the bottom of the properties section are three radio buttons for "Allowable values": "All" (selected), "List", and "Range". At the very bottom of the dialog are "OK" and "Cancel" buttons.

2. In the Create Parameter dialog box, name the parameter.
3. Under Data Type, select **Date or Date & time**.
4. Under **Current Value** set it to whatever you want.
5. Click **OK**.
6. Right-click the **Start Date** parameter and then select **Show Parameter Control**. A menu will pop up on the right. You can now change the date using a drop-down calendar.
7. Repeat steps 1-6 to create the **End Date** parameter.

Create Date Range Using Parameters

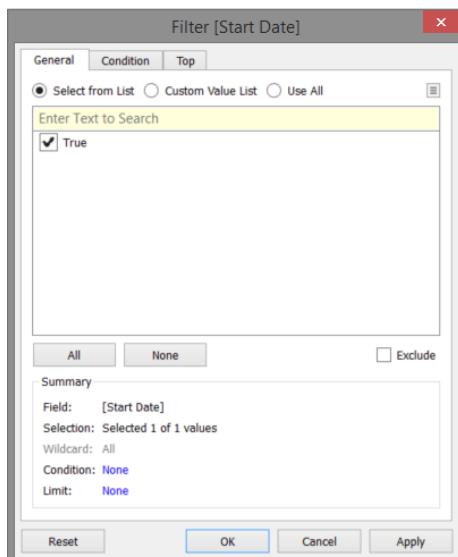
1. Right-click **the parameter we just created** in the Dimensions pane and then select **Create > Calculated Field**.
2. Name the calculated field **Date Range**, and type the following in the **Formula** window, and then click **OK**:

```
[Order Date] >= [Start Date] AND [Order Date] <= [End Date]
```



Add Date Range to Filters

1. Drag **Date Range** to **Filters**, select **True**, and then click **OK**.
2. Right-click **Date Range** on the **Filters** shelf, select **Apply to Worksheets**, and then select **All Using this Data Source**.



Creating Date Selector

First create a parameter for the user to select the date. For this example the user is going to select a month. The parameter has been called DateSelector.

Edit Parameter DateSelector

Name:

Properties

Data type:

Current value:

Display format:

Allowable values: All List Range

Range of values

Minimum:

Maximum:

Step size:

To have this displayed in the format of MMM YY we need to alter the Display Format. Select Custom and type in the format mmm yy.

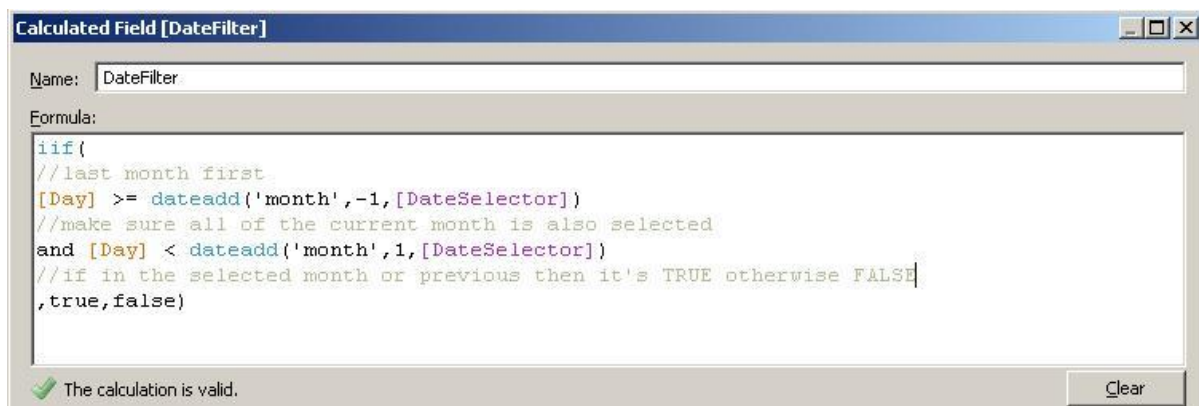
14/03/01
14/3/01
14.3.01
2001-03-14
3/14/2001
3/14/01
03/14/01
03/14/2001
14 March 2001
14 March 2001
Wednesday, March 14, 2001
March 14, 2001
March 2001
March, 2001
2001 Q1
2001 W11
Custom

Custom
Format:

Every month the user selects is actually the first day of that month, which is important to know as future date manipulations need to be taken from that date.

The next step is to create the calculated field which says what dates we are going to use based on the value selected by the user. Remember the parameter only actually selects the 1st of the selected month but the user actually thinks they're selecting the entire month. This means as part of our calculated field we need to specify the entire month is being selected. We also need to select the previous month as that's what is needed as part of this hypothetical report.

This calculated field is our Filter field.



Note: The field named [Day] is the name of the date field in the example dataset I'm using.

Now we have all of our fields you can construct a worksheet. Right click on the parameter DateSelector and 'Show Parameter Control'

Drag the DateFilter into the Filter section and set the value to True. Then pull in the other data you want to display in your report.

Date Functions

DATEADD(date_part, interval, date)

Returns the specified date with the specified number interval added to the specified date_part of that date.

Example

```
DATEADD('month', 3, #2004-04-15#) = 2004-07-15 12:00:00 AM
```

This expression adds three months to the date #2004-04-15#.

DATEDIFF(date_part, date1, date2, [start_of_week])

Returns the difference between date1 and date2 expressed in units of date_part.

The start_of_week parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If it is omitted, the start of week is determined by the data source.

Example

```
DATEDIFF('week', #2013-09-22#, #2013-09-24#, 'monday')= 1
```

```
DATEDIFF('week', #2013-09-22#, #2013-09-24#, 'sunday')= 0
```

The first expression returns 1 because when start_of_week is 'monday', then 22 September (a Sunday) and 24 September (a Tuesday) are in different weeks. The second expression returns 0 because when start_of_week is 'sunday' then 22 September (a Sunday) and 24 September (a Tuesday) are in the same week.

DATENAME(date_part, date, [start_of_week])

Returns date_part of date as a string. The start_of_week parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If start_of_week is omitted, the start of week is determined by the data source.

Examples

```
DATENAME('year', #2004-04-15#) = "2004"
```

```
DATENAME('month', #2004-04-15#) = "April"
```

DATEPART(date_part, date, [start_of_week])

Returns date_part of date as an integer.

The start_of_week parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If start_of_week is omitted, the start of week is determined by the data source.

Note: When the date_part is weekday, the start_of_week parameter is ignored. This is because Tableau relies on a fixed weekday ordering to apply offsets.

Examples

DATEPART('year', #2004-04-15#) = 2004

DATEPART('month', #2004-04-15#) = 4

DATETRUNC(date_part, date, [start_of_week])

Truncates the specified date to the accuracy specified by the date_part. This function returns a new date. For example, when you truncate a date that is in the middle of the month at the month level, this function returns the first day of the month. The start_of_week parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If start_of_week is omitted, the start of week is determined by the data source.

Examples

DATETRUNC('quarter', #2004-08-15#) = 2004-07-01 12:00:00 AM

DATETRUNC('month', #2004-04-15#) = 2004-04-01 12:00:00 AM

DAY(date)

Returns the day of the given date as an integer.

Example

DAY(#2004-04-12#) = 12

ISDATE(string)

Returns true if a given string is a valid date.

Example

ISDATE("April 15, 2004") = true

MAKEDATE(year, month, day)

Returns a date value constructed from the specified year, month, and date.

Available for Tableau Data Extracts. Check for availability in other data sources.

Example

```
MAKEDATE(2004, 4, 15) = #April 15, 2004#
```

MAKEDATETIME(date, time)

Returns a datetime that combines a date and a time. The date can be a date, datetime, or a string type. The time must be a datetime. This function is available only for MySQL connections.

Examples

```
MAKEDATETIME("1899-12-30", #07:59:00#) = #12/30/1899 7:59:00 AM#
```

```
MAKEDATETIME([Date], [Time]) = #1/1/2001 6:00:00 AM#
```

MAKETIME(hour, minute, second)

Returns a date value constructed from the specified hour, minute, and second.

Available for Tableau Data Extracts. Check for availability in other data sources.

Example

```
MAKETIME(14, 52, 40) = #14:52:40#
```

MAX(expression) or MAX(expr1, expr2)

Usually applied to numbers but also works on dates. Returns the maximum of a and b (a and b must be of the same type). Returns Null if either argument is Null.

Examples

```
MAX(#2004-01-01# ,#2004-03-01#) = 2004-03-01 12:00:00 AM
```

```
MAX([ShipDate1], [ShipDate2])
```

MIN(expression) or MIN(expr1, expr2)

Usually applied to numbers but also works on dates. Returns the minimum of a and b (a and b must be of the same type). Returns Null if either argument is Null.

Examples

MIN(#2004-01-01# ,#2004-03-01#) = 2004-01-01 12:00:00 AM

MIN([ShipDate1], [ShipDate2])

MONTH(date)

Returns the month of the given date as an integer.

Example

MONTH(#2004-04-15#) = 4

NOW()

Returns the current date and time.

The return varies depending on the nature of the connection:

- For a live, unpublished connection, NOW returns the data source server time.
- For a live, published connection, NOW returns the data source server time.
- For an unpublished extract, NOW returns the local system time.
- For a published extract, NOW returns the local time of the Tableau Server Data Engine. When there are multiple worker machines indifferent time zones, this can produce inconsistent results.

Example

NOW() = 2004-04-15 1:08:21 PM

TODAY()

Returns the current date.

Example

TODAY() = 2004-04-15

YEAR (date)

Returns the year of the given date as an integer.

Example

YEAR(#2004-15#) = 2004