



How to Localize Content in Tableau

Author: Russell Christoper
OEM Sales Consultant, Tableau Software
June 2013

While the terms “localization” and “internationalization” are fairly recent buzzwords, the notion of adapting the message to the audience is a long-accepted practice. Savvy cross-cultural communicators learn the subtleties of the audience locale to avoid confusion or worse, offense.

Even in the image-driven world of data visualization, localization plays an important role. Communicating in the audience's preferred language and adjusting formats such as date display or numerical separators (1,000 versus 1 000) can increase viewer comprehension and confidence.

Reports created and displayed in both Tableau Desktop and Tableau Server may be localized in order to support users in different locales. This paper documents the features and behaviors of Tableau's localization functionality and includes samples of the features at work. Some Tableau features rely on JavaScript and presuppose that the user is viewing the report viewed via the browser and Tableau Server. Unless stated otherwise, examples are based on that scenario.

Language vs. Locale

The Windows operating system allows users to set both Language and Locale properties on their computers.

It is good to know the difference between the two settings:

- Language denotes the specific “code page” used to represent the language. Languages may be location-neutral, such as “fr” for French. Languages may also be location-specific, such as “fr-FR” – French as defined in France (as opposed to fr-BE – French in Belgium).
- Locale is a set of language-related user preference information. Locale defines display properties such as:

- Date formatting
- Currency symbols
- Numeric formatting (decimal separators, grouping separators, etc.)

Both Language and Locale can be set in Tableau.

Language support

At the time of writing, Tableau Server and Tableau Desktop have been localized to English, French, German, Brazilian Portuguese, Spanish, Korean, Japanese and simple Chinese. Tableau labels, menus and prompts in both Desktop and Server can be displayed in the languages mentioned above.

It is important to understand that Tableau supports Unicode / double byte character sets. Tableau can therefore display any language, whether or not the language is directly supported in the user interface.

Columns	Rows				Dr
	FrenchDescription	HebrewDescription	JapaneseDescription	EnglishDescription	
FrenchDescription	HebrewDescription	JapaneseDescription	EnglishDescription		Dr
Null	Null	Null	Null	Null	Ab
Acier chromé	כרום (כרום) חלודים	クロム鋼	クロム鋼	Chromoly steel.	Ab
Acier inoxydable / facile à ne.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	ステンレス製で、耐食性を	Stainless steel; designed to		Ab
Adapté à tous les usages, su.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	オオロームにもフローに	Suitable for any type of ridin.		Ab
Aliaje robuste avec moyeu.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	頑丈な合金により、クワット	Sturdy alloy features a quick,		Ab
Amortisseurs en caoutchouc.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	ダンパー・バンプが家車吸収	Rubber bumpers absorb bu.		Ab
Axe très rigide.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	最強の剛性を持ち得るビ	Super rigid spindle.		Ab
Barre d'appui avec tube en	אין-רוס-אין-רוס-אין-רוס-אין-רוס	アルミマシンのアルミニ	Aluminized shaped alumin.		Ab
Barre d'appui en aluminium l.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	長距離ドライブに好む	Alight yet still aluminum l.		Ab
Barre d'appui plate solide.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	フラット・ストロング・エ	Flat bar strong enough for		Ab
Barre d'appui tous usages p.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	オオロームにもフローに	Al-purposure bar for on or off-		Ab
Barres d'appui en alliaje d.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	ダブリンの頑固なアル	Tough aluminum alloy bars f		Ab
Bidon d'eau avec le logo d.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	AWC ロゴ入りの水、容量	AWC logo water bottle - hold.		Ab
Boîtier de protection hermét.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	密封されたカードジが味	Sealed cartridge keeps dri.		Ab
Cadre en alliaje d'aluminium.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	レース・トラックの走行にも	Aluminum-alloy frame provi.		Ab
Cadre profilé pour améliorer.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	より快適な乗り心地のため	Cut-out shell for a more com.		Ab
Cadre renforcé léger pour c.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	軽量合金アルミニ	Lightweight built aluminum.		Ab
Caoutchouc haute densité.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	高密度バンプ	High-density rubber.		Ab
Caoutchouc plus dense que	אין-רוס-אין-רוס-אין-רוס-אין-רוס	超高密度バンプ	Higher density rubber.		Ab
Carbone léger renforcé pour	אין-רוס-אין-רוס-אין-רוס-אין-רוס	炭素の比重で最も高	Carbon fiber carbon reinforce		Ab
Ce vélo est destiné aux ch.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	レース・ウエーに使用する	This bike is ridden by race w.		Ab
Ce vélo offre un excellent r.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	この自転車は室内で高性	This bike delivers a high-lev.		Ab
Chaque cadre est fabriqué a.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	各フレームは、Boethi 工	Each frame is hand-crafte		Ab
Collants chausse en Spandex.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	冬用の短パン/スリムジ	Warm spandex tights for win.		Ab
Combinaison chausse natu.	אין-רוס-אין-רוס-אין-רוס-אין-רוס	天然おれり合皮履物組み	Combination of natural and s		Ab

Figure 1: *Product descriptions in multiple languages*

Setting Language and Locale: Desktop

The Language setting within Tableau Desktop is global across the user interface. You will apply Locale on a workbook-by-workbook basis.

To change the Language setting in Tableau Desktop, use the Help menu and select Choose Language. You must restart Tableau after modifying this setting.

To set Locale, click Workbook Locale on the File menu. You must save the workbook to make this change permanent in the workbook.

If Workbook Locale is set to Automatic (User Locale), then the following settings are checked in the order listed to determine locale:

1. Windows Operating System (OS) Locale
2. Tableau Desktop Language

Example:

- Windows OS Locale: French (France)
- Tableau Desktop Language: English
- Workbook Locale: Automatic (User Locale)

In this example, the workbook will be displayed using French (France) locale settings even though the Tableau user interface is displayed in English. The Windows OS Locale is checked before the Tableau Desktop Language. Please note that for users who have selected a language not supported in Tableau, the application will default to English.

If you choose any value other than Automatic (User Locale) for your Workbook Locale, then your selection will drive all localization behavior for the workbook.

Example:

- Windows Operating System Locale: French (France)
- Tableau Desktop Language: German
- Workbook Locale: English (United States)

The workbook will be displayed using an English (United States) locale even though the Tableau user interface is displayed in German and the Windows OS locale is set to French (France). Workbook Locale takes precedence over all other settings.

Effects of Language/ Locale on Desktop

When changing locale settings, you can expect the following changes:

- Numeric formatting – decimal and group separators:

Month of Order Date	Gross Profit Ratio	Order Quantity	Sales
January 2010	-5.0%	5,467	\$530,288
February 2010	8.2%	4,036	\$318,809

Figure 2: English (United States) Locale

Comma as decimal separator

Space as group separator

Month of Order Date	Gross Profit Ratio	Order Quantity	Sales
janvier 2010	-5,0%	5 467	\$530 288
février 2010	8,2%	4 036	\$318 809

Figure 3: FR (France) Locale

- Map labels



Figure 4: Map label - English (United States) Locale



Figure 5: Map label - French (France) Locale

- Date formatting / translation:

Month of Order Date	Gross Profit Ratio	Order Quantity	Sales
January 2010	-5.0%	5,467	\$530,288
February 2010	8.2%	4,036	\$318,809

Figure 6: Date format - English (United States) Locale

Month of Order Date	Gross Profit Ratio	Order Quantity	Sales
janvier 2010	-5,0%	5 467	\$530 288
février 2010	8,2%	4 036	\$318 809

Figure 7: Date format - FR (France) Locale

- Filter, Quick Filter & page shelf data entry and display:

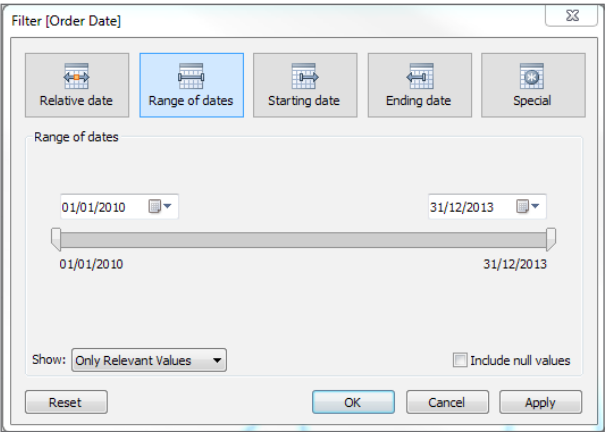


Figure 8: Filter dialog - French (France) Locale

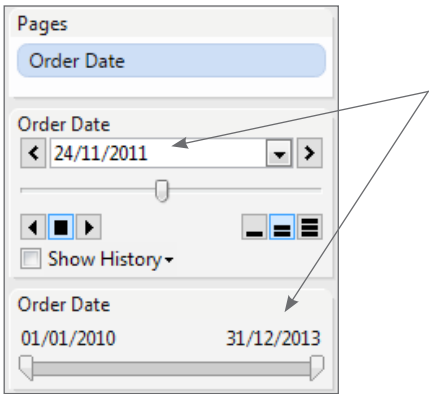


Figure 9: Quick Filter and Page Shelf - French (France) Locale

Some other important features are based on the Tableau Desktop Language. This can be changed by going to Help and choosing Select Language. Your language choice within Tableau Desktop will govern:

- Sheet naming

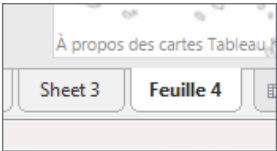


Figure 10: Sheet naming, Tableau language set to French

- Trend line descriptions



Figure 11: Tableau language set to French

- Date part translations

Translated due to Language setting
As noted before, Locale causes Q1 to be displayed as T1

Mois de Order Date	Trimestre de Order Date	Gross Profit Ratio	Order Quantity	Sales
janvier 2010	T1	-5,0%	5 467	\$530 288
février 2010	T1	8,2%	4 036	\$318 809

Figure 12: Date parts impacted by Language and Locale settings - French and French (Fr)

Currency labels (\$, €, £, ¥) are never automatically substituted when locale is changed. This is by design.

When you choose to format a number as currency, the current locale value is leveraged to suggest an appropriate currency symbol – such as a \$ for the United States. After the format mask is applied, currency symbols will not automatically change as the locale is modified.

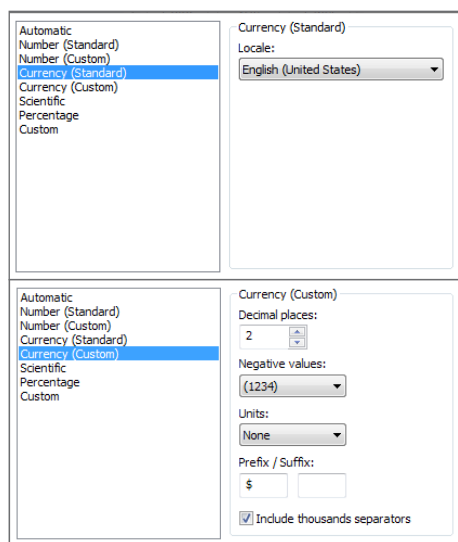


Figure 13: *Currency symbols are not impacted by changes to locale*

Setting Language and Locale: Server

The Workbook Locale setting (if any) established for a specific workbook supersedes all other options. If a report author sets locale of a workbook using Tableau Desktop and publishes it to a Tableau Server, that server will always display the workbook using the locale set by the author.

However, if the author leaves the Workbook Locale set to Automatic (the default), then other settings come into play.

Two settings are available from inside Tableau Server that impact language and locale:

- Default language and locale: set on Maintenance page
- Language and locale: per-user-account settings adjusted by the user through the User Preferences page

An administrator may set default language and locale from the Maintenance page of Tableau Server—these settings are applied to all users unless the user sets his or her own user-specific language preferences in the User Preferences page.

Once a user has selected personal language and locale settings in Server, those preferences will be used. By default, these options are set to “unspecified.”

If users do not choose their own language and locale settings, the locale settings of their browser will drive locale and language of Server and the reports it displays.

If the browser fails to communicate language and locale information, , then language and locale will be determined by the default language and locale settings within Server, followed by the operating system’s language and locale information on the computer on which Tableau Server is installed.

In short, the following options (in the order listed below) ultimately determine how a workbook is localized:

1. Workbook Locale value of the workbook being viewed.
2. Per-user Language and Locale settings in User Preferences
3. The language settings of the user’s browser
4. Default language and locale set on Tableau Server
5. The language and locale setting of the computer on which Tableau Server is installed.

Example #1:

- Windows Operating System Locale: French (France)
- Workbook Locale: German (Austria)
- Browser locale: English (United States)
- User-defined language / locale: Unspecified
- Tableau Server default language / locale: French (France) & French

The workbook will be displayed in German when viewed via Tableau Server and the browser. Workbook Locale takes precedence over all other settings.

Example #2:

- Windows Operating System Locale: French (France)
- Workbook Locale: Automatic(User Locale)
- Browser locale: English (United States)
- User-defined language / locale: Unspecified
- Tableau Server default language / locale: French (France) & French

The workbook will be displayed in English when viewed via Tableau Server and the browser. Since no Workbook Locale or user-defined settings have been specified, browser settings drive localization.

Example #3:

- Windows Operating System Locale: French (France)
- Workbook Locale: Automatic (User Locale)
- Browser Locale: German (Austria)
- User-defined language / locale: German (Austria)
- Tableau Server default language / locale: French (France) & French

The workbook will be displayed in German when viewed via Tableau Server and the browser. While the browser locale has been set to German (Austria), it is the user-defined language/locale settings on the server which drive localization. If the user-defined language/locale settings were not specified, the workbook would still be displayed in German based on browser settings.

Effects of Language/ Locale on Server

The same effects documented in the topic Effects of Language/ Locale on Desktop can be expected in Tableau Server. In addition, Server-specific user interface components will be localized to one of the 8 languages supported by Tableau. These user interface components are:

- Login screen text
- Portal text
- Menu text
- Tableau administration reports (available from the Maintenance page)

Simple Report Localization

The most basic localization process involves presenting a different database field to the user based on locale.



The data source below contains translated fields for Category and SubCategory in Japanese, German and English. A Language parameter allows users to select the language of their choice.

The Language parameter is then leveraged to drive the behavior of three calculated fields:

- Category Label
- SubCategory Label
- Title Label

An example of the expression behind Category Label:

Figure 14: Supporting a multi-lingual report

```
If [Language] = 1 Then [製品カテゴリー]
  ElseIf [Language] = 2 Then [Produktkategorie]
  Else [Department]
End
```

Each one of the calculated fields dynamically displays the Category, SubCategory, and Title (“Sales”) in the correct language.

You may download and explore the same workbook here (requires Tableau):

[Click me to retrieve this artifact.](#)

When this visualization is rendered by Tableau Server, the parameter driving localization can be set automatically on the URL or via JavaScript. This approach will be covered in more depth in the section Automating Localization Actions.

This technique is fast and easy to implement, but has some shortcomings. Note the following potential issues in the screenshot (Figure 15):

1. The Category and SubCategory field labels must be hidden from the user since they will display un-localized text (“Category Label” and “SubCategory Label”)
2. The title of the parameter is not localized and presents “Language”
3. The axis label displays “Sales,” the actual name of the measure; it also is not localized

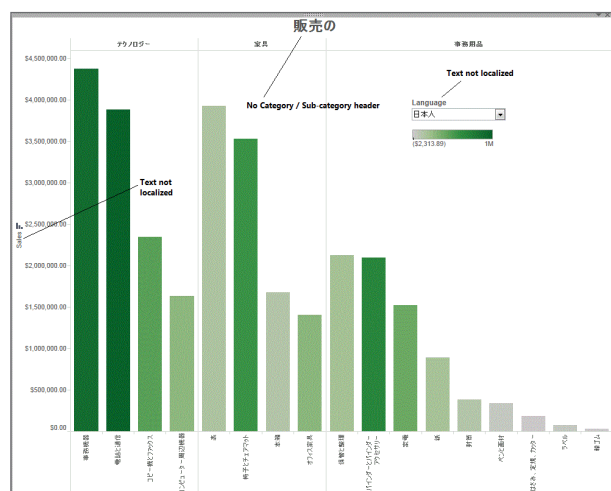


Figure 15: Simple localization

The next topic, Localizing Labels demonstrates techniques which you can use to deal with label text.

Localizing Labels

Tableau supports the ability to localize labels, as well. Figure 16 exhibits a report with correctly translated label text, axis labels and filter / legend labels. Since the Workbook Locale property within the workbook has not been set, dates are being displayed as English (United States), the locale settings of the author's computer.

Localizing labels in a report generally requires a distinct “label lookup” table which contains one field per label to be localized:

	A	B	C	D	E	F
1	Language	Category	Subcategory	Report Title	Color Legend	Language Parameter
2		3 Category	Subcategory	Sales	Profit	Language
3		1 製品カテゴリー	製品サブカテゴリー	販売の	利益	言語
4		2 Produktkategorie	Produkt-Subkategorie	Vertrieb	Profit	Sprache

Figure 17: Label Table

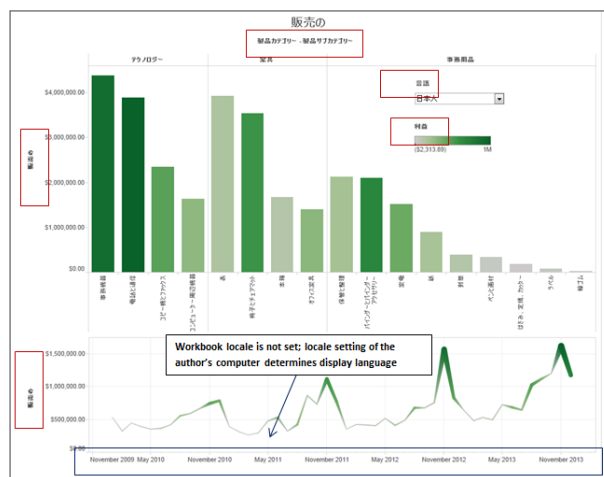


Figure 16: Label localization

Fields from the table are included in a Tableau worksheet and filtered in order to return the correct “language row” based on the value of a [Language] parameter.

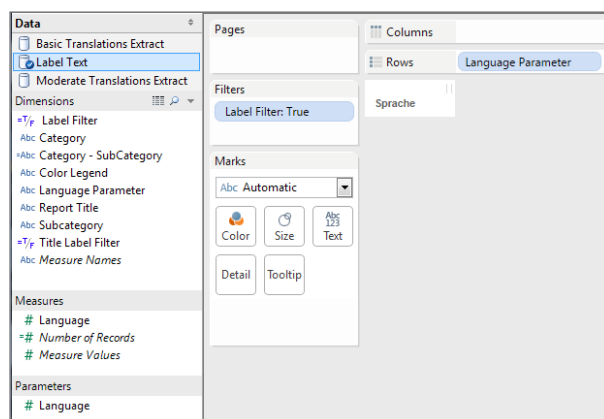


Figure 18: Localizing a label

Figure 18 demonstrates the following:

- The use of a label lookup table
- A single-field worksheet which will appear as though it were a label and can be “floated” into a report
- A filter which ensures the correct language is displayed by the “label” in question

In the example above, the [Language Parameter] dimension represents one of three values:

- Language
- Sprache
- 言語

The correct value is displayed by way of the [Label Filter] calculated field which compares the currently selected language in the [Parameters].[Language] parameter to the value in the label lookup table's [Language] field:

```
[Parameters].[Language] = [Language]
```

The worksheet has also been formatted so that column and row dividers are not displayed.

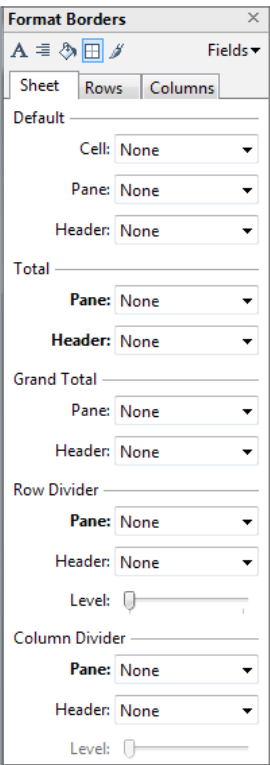


Figure 19: Label formatting

While using an exact dashboard size is considered a best practice in and of itself, you may find it too restrictive.

Furthermore, floating objects are not transparent, so overlapping may become an issue.

The “labels” which you create can be added to the report as a standard “docking” worksheet, or as a (new in Tableau v8) floating object.

Generally, using docked worksheets as labels is preferred. Docked worksheets are easier to position and function seamlessly inside an automatically resizing dashboard.

Floating objects use absolute positioning inside a dashboard. Therefore, you must set the size of the dashboard to an exact value in order to ensure the floating items are always positioned correctly.

Figure 16 - Label localization demonstrates the use of both floating and docked label worksheets. The labels associated with the dashboard’s language selector and color legend title “float.”

Docked worksheets are used to replace the axis labels on the left-hand side of the dashboard. A docked worksheet is also used to display the Category and SubCategory labels directly over the bar chart.

Regardless of which technique you choose to leverage, remember to use the Hide Field Labels for Columns/ Rows functionality of Tableau to remove unlocalized text from your charts. You may also need to Edit Axis to remove Title and Subtitle text.

You may download the sample workbook used in this section here:

[Retrieve this artifact.](#)

Currency and Date formatting

Dates

The techniques demonstrated thus far allow a Tableau report creator to control nearly all label and data field localization.

However, the approaches described will not allow for automatic currency and date formatting.

As mentioned in Effects of Language/ Locale on Desktop, Tableau date formatting and translation is generally controlled by the locale under which the workbook is rendered.

It is not possible to dynamically set locale via a parameter value.

Instead, locale for the workbook is determined by the processes outlined in Setting Language and Locale: Desktop and Setting Language and Locale: Server.

As a result, reports may at times display “mixed” localization. Note, for example in Figure 16 - Label localization:

- English (United States) date formatting
- Japanese field and label translations

If you wish to guarantee that dates are formatted using a specific locale formatting, you must set the Workbook Locale property of the workbook before it is published to Tableau Server.

Based on your requirements you may find it necessary to create multiple copies of the same workbook, each with a different Workbook Locale. You can use techniques described in the section Automating Localization Actions to automatically render the appropriate version of the workbook on a per-user basis.

Currency: Formatting

To avoid displaying misleading data, Tableau does not implement automatic currency conversion or dynamic, locale-driven currency formatting.

Regardless of the locale of a report, its author must specify a single currency format for each measure displayed in the report.

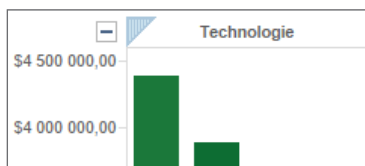


Figure 20: Default currency formatting

For example, the workbook shown in Figure 20 has a Workbook Locale of French (FR) and a Tableau Desktop Language of French. Nevertheless, currency is being displayed in US Dollars – the currency format selected by the original report author. To change currency format, use the Default Number Format dialog (shown in Figure 21).

Figure 21 displays the properties of the Profit measure displayed in Figure 20. Note that the default Currency (standard)—or, in French “Devise (standard)” —locale is actually set to anglais (Etats-Unis), even though the Desktop language is set to French. As you can see, the currency amounts in Figure 20 are displayed in US \$ because of this setting.

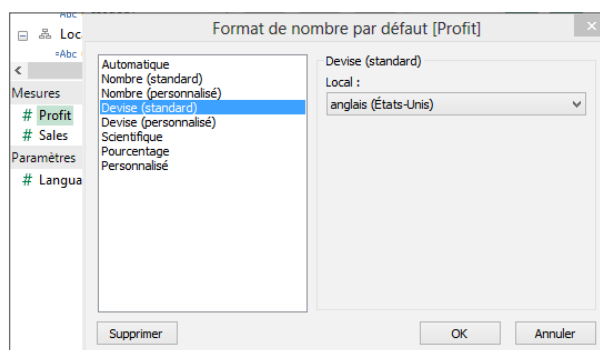


Figure 21: Default Currency format based on OS

When a measure has an unspecified currency format, Tableau recommends a format based on the current language and locale settings of the operating system. If the report author chooses another format—for example français (France)—instead of the suggested format, the measure will reflect the author’s chosen format regardless of the Workbook Locale setting associated with the workbook.

In summary, automatic locale-driven currency formatting is not supported in Tableau. This is by design.

If you wish, you may programmatically forward users to different versions of the same workbook which implement the currency format of your choice. You should not do so unless you have also put a currency conversion process in place.

Currency: Conversion

Overview

Currency conversion is a complex topic and a complete treatment of the subject is beyond the scope of this whitepaper. This section will cover the basics of currency conversion in Tableau, but as currency conversion is an ancillary topic this paper will not dive deeply into the topic.

Currency conversion functionality may be implemented in many ways. A sampling of techniques include:

- Leveraging an Extract, Transform, and Load process

- Using conversion tables and expressions encapsulated in a database and database views
- Implementing conversion expressions inside a report

This topic will focus on the third method.

In-report conversion is not necessarily the best approach as it can lead to “multiple versions of the truth” based on faulty business logic or slightly different implementation of business logic on a per-report basis.

Please review the appendix of this whitepaper for references to internet resources you can leverage to decide which currency conversion technique is best for you.

In-report currency conversion

Tableau report authors can implement simple currency conversion using calculated fields and/or parameters.

The currency-to-currency conversion factor may be stored in a table, or provided dynamically in parameter when the report is executed.

Parameter-driven conversion

In the expression below, a conversion factor is provided via a [Conversion Factor] parameter:

```
//“Converted Sales” Field
//1 USD currently = .78 Euro
[Sales] * [Conversion Factor]
```

The resulting field, [Converted Sales], is used in the report in lieu of [Sales].

The conversion factor can be specified on the URL or via JavaScript when calling the report. In the following example, .78 is passed to the Sales Dashboard report inside the Sales Report workbook.

Table-driven conversion

You can also build a currency conversion table which contains conversion factors and leverage it within Tableau.

While this approach will be hard-pressed to offer completely current conversion values, it can still be useful. Currency conversion tables are often used to deal with scenarios in which the same measure contains values stored in multiple currencies. See the appendix for information on dealing with this challenge.

In Figure 22 - Adding a currency conversion factor, the label table (shown earlier as Figure 17) has been extended to include a conversion factor. The value saved in this field can be used instead of dynamic Parameter:

//“Converted Sales” Field
[Sales] * [Conversion Factor Field]

C	D	E	F	G
Subcategory	Report Title	Color Legend	Language Parameter	Conversion Factor Field
Subcategory	Sales	Profit	Language	1
製品サブカテゴリ販売の	利益	言語		93.85
Produkt-Subkategor Vertrieb	Profit	Sprache		0.78

Figure 22: Adding a currency conversion factor

The same filter which guarantees the appropriate labels are displayed in the report returns the correct conversion factor.

Time Zone Conversion

Tableau does not support implicit time zone conversion of datetime values based on the user's location. If you wish to apply a time zone-related offset to a datetime value, you must:

- Know the time zone of the stored data
- get the local time zone of the user
- Calculate an offset between the two values
- Apply the offset in a calculation

Example:

```
//“Local Time” Field
DateAdd(“minute”, [Offset in Minutes], [Stored Date])
```

Automating Localization Actions

Setting options on the URL

It is common for web applications and portals which support different languages to host multiple (translated and localized) versions of each HTML page on the site. For example, the “landing” page of your site might have many language-specific versions:

- index_en.html
- index_fr.html
- index_jp.html
- index_de.html

Displaying correctly localized Tableau dashboards on each page can be accomplished easily by changing the [Language] parameter that drives the report.

For example, the German index page will call a Tableau dashboard from within an iframe using this markup:

```
<iframe height=800 width=1400 frameborder=0
src="http://server/views/myWorkbook/Dashboard?Language=2"></iframe>
```

The Japanese page will use the same markup with a different Language parameter value:

```
<iframe height=800 width=1400 frameborder=0
src="http://server/views/myWorkbook/Dashboard?Language=1"></iframe>
```

A simple example of this technique can be found [here](#).

Files for the sample above can be [downloaded here](#).

In addition, a currency conversion factor can also be passed on the URL (assuming that having a real-time value isn't necessary):

```
<iframe height=800 width=1400 frameborder=0
src="http://server/views/myWorkbook/Dashboard?Language=1&ConversionFactor=.78"></iframe>
```

Manipulating options via JavaScript

If you wish to provide a more dynamic experience, you can leverage the Tableau JavaScript API to set label localization options.

In the example below, the user is given the ability to choose which language they'd like to see a report localized to:

```
<script type="text/javascript">var radioArray=[null];</script>
<input name="lang" value="3" type="radio" onclick="radioChange('lang',3,radioArray);"/>English
<input name="lang" value="2" type="radio" onclick="radioChange('lang',2,radioArray);"/>German
<input name="lang" value="1" type="radio" onclick="radioChange('lang',1,radioArray);"/>Japanese
```

Based on the selection made by the user, a different [Language] value is passed to a JavaScript function which either renders or re-renders the dashboard.

```
var mainViz;
function radioChange(radioSet,radioButton,radioArray)
{
    var oldButton=radioArray[0];
    //Initial Render
    if(radioArray[0] == null)
    {
        radioArray[0] = radioButton;
        // Define variables for viz
        var mainVizDiv = $("#mainViz");
        var mainWorkbookUrl =
            "http://localhost/views/LabelLocalization/Dashboard";
        var mainVizOptions = {
            hideTabs: true,
            hideToolbars: true,
            width: mainVizDiv.parent().innerWidth() + "px",
            height: mainVizDiv.parent().innerHeight() + "px",
            language: radioButton, // define language
            onFirstInteractive: function () {
                mainWorkbook = mainViz.getWorkbook();
            }
        };
        // Create viz
        mainViz = new tableauSoftware.Viz(mainVizDiv[0], mainWorkbookUrl, mainVizOptions);
    }
    else
    {
        //Change parameter value on existing viz
        mainWorkbook = mainViz.getWorkbook();
        // Hack to deal with DBCS strings as parameter values.
        // In the real world, it is unlikely you'll display
        // language names directly to the user.
        var languageHack;
        switch (radioButton) {
            case 1:
                languageHack="日本人";
                break;
            case 2:
                languageHack="Deutsch";
                break;
            case 3:
                languageHack="English";
                break;
        }
        // Change Language parameter
        mainWorkbook.changeParameterValueAsync('Language', languageHack);
    }
}
```

You can [try the sample here](#). The complete html page is [available here](#) for download.

Using JavaScript to detect and react to Locale

JavaScript makes it possible to detect the locale settings of the user's browser and automatically present him or her with the correctly localized dashboard.

The following JavaScript detects and returns locale information from the browser:

```
function detectLocale() {
    var tempLocale = (navigator.language || navigator.userLanguage).substring(0,2);
    switch(tempLocale) {
        case "en":
            //1033
            return 3; break;
        case "de":
            //1031
            return 2; break;
        case "jp":
            //1041
            return 1; break;
        default:
            return 3; break;
    }
}
```

Usage notes on the navigator object's language and userLanguage properties may be found here:

- [navigator.language](#)
- [navigator.userLanguage](#)

Note: It is important to understand that Internet Explorer returns locale information differently than FireFox, Chrome, and Safari. Internet Explorer does not support navigator.language.

Instead, you must rely on navigator.userLanguage which returns the locale setting for the user's operating system. navigator.language returns language preference information set by the user in the browser itself, not in the operating system.

Once this information is available, it can be used to render a visualization without user intervention by providing language labeling information to the visualization when it is rendered.

You may [test an example here](#) or [download it](#). Make sure to change the language settings in Chrome or FireFox to either Japan or Germany in order to see the functionality in action:

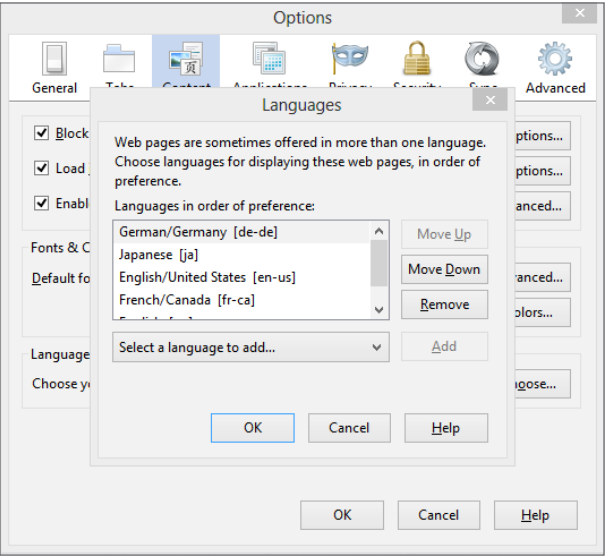


Figure 23: Changing language options in FireFox

The approach described above is appropriate when it is not necessary to automate the formatting or localization of dates and date parts. As described in the Dates topic, dates are formatted independently of any label localization technique you may implement.

On occasion, you may find it necessary to dynamically forward the user to a different version of the report to fully meet your localization needs.

This tactic is typically required when you must hardcode a Workbook Locale in order to achieve your goals or if you need to implement currency conversion and show different currency symbols. You will create multiple copies of the same workbook, each with a hard-coded Workbook Locale and/or different currency symbol.

JavaScript can help you here, too. A simple switch() can be used to build the appropriate URL which points at the correct workbook to launch based on the user's locale:

```
userLocale=detectLocale();  
  
// Change report to render based on user's locale  
switch(userLocale) {  
  case 3: // English  
    var mWBL = "http://localhost/views/LabelLocalizationEN/Dashboard";  
    break;  
  case 2: // German  
    var mWBL = "http://localhost/views/LabelLocalizationDE/Dashboard";  
    break;  
  case 1: // Japanese  
    var mWBL = "http://localhost/views/LabelLocalizationJA/Dashboard";  
    break;  
  default:  
    var mWBL = "http://localhost/views/LabelLocalizationEN/Dashboard";  
    break;  
}
```

The string used in the mWBL variable is eventually used to dynamically render the visualization.

You know what happens now. You can try out a [sample](#) or [download it](#).

Appendix

- [Multi-Currency Best Practice & Implementation, 2010](#)
- [Oracle Corporation, \(n.d.\) Designing and Building Currency Conversion Applications](#)

About Tableau

Tableau Software helps people see and understand data. Tableau helps anyone quickly analyze, visualize and share information. More than 12,000 customer accounts get rapid results with Tableau in the office and on-the-go. And tens of thousands of people use Tableau Public to share data in their blogs and websites. See how Tableau can help you by downloading the free trial at www.tableausoftware.com/trial.

Tableau and Tableau Software are trademarks of Tableau Software, Inc. All other company and product names may be trademarks of the respective companies with which they are associated.