

# **Aras Innovator 11**

## **Configuring Localization and Internationalization Support**

*Document #: 11.0.02014120801*

*Last Modified: 12/30/2014*



# Copyright Information

Copyright © 2014 Aras Corporation. All Rights Reserved.

Aras Corporation  
300 Brickstone Square  
Suite 700  
Andover, MA 01810

**Phone:** 978-691-8900

**Fax:** 978-794-9826

**E-mail:** [Support@aras.com](mailto:Support@aras.com)

**Website:** <http://www.aras.com>

## Notice of Rights

Copyright © 2014 by Aras Corporation. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, V1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

Distribution of substantively modified versions of this document is prohibited without the explicit permission of the copyright holder.

Distribution of the work or derivative of the work in any standard (paper) book form for commercial purposes is prohibited unless prior permission is obtained from the copyright holder.

Aras Innovator, Aras, and the Aras Corp "A" logo are registered trademarks of Aras Corporation in the United States and other countries.

All other trademarks referenced herein are the property of their respective owners.

## Notice of Liability

The information contained in this document is distributed on an "As Is" basis, without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose or a warranty of non-infringement. Aras shall have no liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this document or by the software or hardware products described herein.

## Table of Contents

<b>Send Us Your Comments .....</b>	<b>5</b>
<b>Document Conventions .....</b>	<b>6</b>
<b>1 Overview .....</b>	<b>7</b>
<b>2 Setup .....</b>	<b>8</b>
2.1 Server Culture Code .....	8
2.2 Time Zones .....	9
2.2.1 Local Time Zone .....	9
2.2.2 Corporate Time Zone .....	9
2.2.3 Update Time Zone Data .....	11
2.2.4 Defining a Language .....	12
2.2.5 Adding the Language Item .....	12
2.2.6 Updating the User Interface (Xml-based resources) .....	13
2.3 Defining the Supported Locales .....	14
<b>3 The User Interface .....</b>	<b>15</b>
3.1 Multi-Lingual Properties .....	15
3.2 Date Properties .....	16
3.3 Locale Neutral Properties .....	17
<b>4 Populating a Language .....</b>	<b>18</b>
4.1 Manually Managing Language Packs in the Database .....	18
4.2 Using the Language Pack Management Utility .....	19
4.2.1 The Command Line .....	20
4.2.2 LanguageTool.exe.config .....	21
4.2.3 Export Configuration .....	21
4.2.4 Import Configuration .....	22
4.3 XML Resources .....	24
4.4 Updating the Resource.Config.xml .....	25
<b>5 Programming Notes .....</b>	<b>26</b>
5.1 AML .....	26
5.1.1 Multi—Lingual Properties .....	26
5.1.2 DateTime Properties .....	27
5.1.3 Locale Neutral Data Formats .....	28
5.2 IOM .....	29
5.3 Date Related SQL Queries .....	34
<b>6 Global Deployment and Performance .....</b>	<b>36</b>
6.1 IIS Compression of Dynamic Content .....	36
<b>7 Appendix (Client Settings) .....</b>	<b>37</b>

7.1	Defining user's Locale in Browser.....	37
7.1.1	<i>Internet Explorer 9</i> .....	37
7.1.2	<i>Internet Explorer 10 and Internet Explorer 11</i> .....	38
7.1.3	<i>Firefox 31 ESR for Windows</i> .....	39
7.1.4	<i>Firefox 31 ESR for Mac</i> .....	41

## Send Us Your Comments

---

Aras Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for future revisions.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where and what level of detail?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, indicate the document title, and the chapter, section, and page number (if available).

You can send comments to us in the following ways:

**Email:**

[Support@aras.com](mailto:Support@aras.com)

Subject: Aras Innovator Documentation

Or,

**Postal service:**

Aras Corporation  
300 Brickstone Square  
Suite 700  
Andover, MA 01810  
Attention: Aras Innovator Documentation

Or,

**FAX:**

978-794-9826  
Attn: Aras Innovator Documentation

If you would like a reply, provide your name, email address, address, and telephone number.

If you have usage issues with the software, visit <http://www.aras.com/support/>

# Document Conventions

The following table highlights the document conventions used in the document:

Table 1: Document Conventions

Convention	Description
<b>Bold</b>	This shows the names of menu items, dialog boxes, dialog box elements, and commands. Example: Click <b>OK</b> .
Code	Code examples appear in <code>courier</code> text. It may represent text you type or data you read.
<code>Yellow highlight</code>	Code with yellow highlight is used to draw attention to the code that is being indicated in the content.
<code>Yellow highlight with red text</code>	Red color text with yellow highlight is used to indicate the code parameter that needs to be changed or replaced.
<i>Italics</i>	Reference to other documents.
<b>Note:</b>	Notes contain additional useful information.
<b>Warning</b>	Warning contains important information. Pay special attention to information highlighted this way.
Successive menu choices	Successive menu choices may appear with a greater than sign (>) between the items that you will select consecutively. Example: Navigate to <b>File</b> > <b>Save</b> > <b>OK</b> .

# 1 Overview

---

Aras Innovator has many features to make it easier for users to view the client in formats and languages specific to each user. This document outlines these features, how to configure them and how to use them in three main areas.

- How to configure Aras Innovator languages, locales, and time zones.
- How to use multi-lingual and DateTime properties.
- How to develop for Aras Innovator in internationalized environments.

## 2 Setup

---

To aid the administrators in deployment, each of the following topics are laid out in the order they should be considered to set-up Aras Innovator multi-lingual, multi-locale use case.

### 2.1 Server Culture Code

The first step in setting up your server is to set the culture code for the web server.

1. Edit the \Innovator\Server\web.config file
2. Change the globalization tag to the correct culture code:

```
<!--
    Use culture attribute to tell Innovator Server which locale to use
    when parsing float numbers.
    Example:    culture="en-US" for english (United States)
               culture="en-GB" for english (United Kingdom)
               culture="de-DE" for german (Germany)
               culture="de-CH" for german (Switzerland)
-->
<globalization requestEncoding="utf-8" responseEncoding="utf-8"
culture="en-US" />
```

becomes

```
<!--
    Use culture attribute to tell Innovator Server which locale to use
    when parsing float numbers.
    Example:    culture="en-US" for english (United States)
               culture="en-GB" for english (United Kingdom)
               culture="de-DE" for german (Germany)
               culture="de-CH" for german (Switzerland)
-->
<globalization requestEncoding="utf-8" responseEncoding="utf-8"
culture="de-DE" />
```



## 2.2 Time Zones

Aras Innovator has two basic configurations for time zone support, Local Time Zone and Corporate Time Zone.

### 2.2.1 Local Time Zone

If all client PC machines accessing Aras Innovator and the server running Aras Innovator are in the same location, or at least in the same time zone, then no time zone configuration is required. All time zones on the user client PC machines Aras Innovator and the Aras Innovator servers must match to use the Local Time Zone. This is the default mode defined after installation of Aras Innovator and is fine for test and development environment, but often may not work for production environments.

Use of the Program Management solution, however, requires that the CorporateTimeZone variable be specified.

### 2.2.2 Corporate Time Zone

If your users access Aras Innovator from multiple time zones, you should configure Aras Innovator to have one corporate time zone that the Innovator Server can reference. This time zone is used for calculating things such as when Project Activities become active.

Setting the Corporate Time Zone should be done before user begin to work with the system in production. If it is done after users have begun to use the system, you have to get all users to log out of the system before enabling the Corporate Time Zone.

**To set the Corporate Time Zone, use the following steps:**

1. Log into Aras Innovator as an Administrator
2. From the TOC, select **Administration --> Variables**
3. From the main toolbar, select **Create a New Item**.

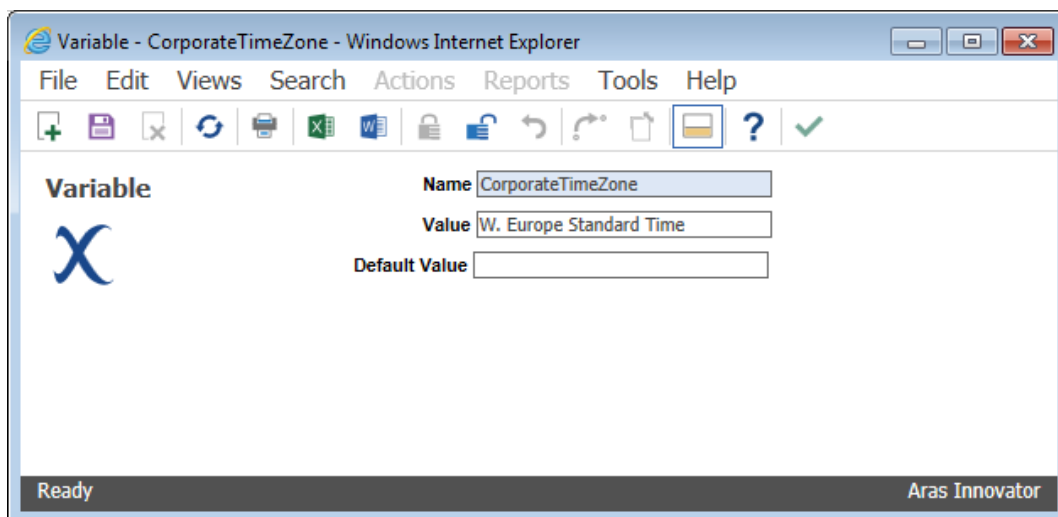


Figure 1.

4. Set the **Name** = CorporateTimeZone

5. Set the **Value** according to the registry key name of the required time zone.

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Time Zones

Below is a list of the time zone registry key names as of May 2008 for reference:

Afghanistan Standard Time	Egypt Standard Time	Pacific Standard Time
Alaskan Standard Time	Ekaterinburg Standard Time	Pacific Standard Time (Mexico)
Arab Standard Time	Fiji Standard Time	Romance Standard Time
Arabian Standard Time	FLE Standard Time	Russian Standard Time
Arabic Standard Time	Georgian Standard Time	SA Eastern Standard Time
Armenian Standard Time	GMT Standard Time	SA Pacific Standard Time
Atlantic Standard Time	Greenland Standard Time	SA Western Standard Time
AUS Central Standard Time	Greenwich Standard Time	Samoa Standard Time
AUS Eastern Standard Time	GTB Standard Time	SE Asia Standard Time
Azerbaijan Standard Time	Hawaiian Standard Time	Singapore Standard Time
Azores Standard Time	India Standard Time	South Africa Standard Time
Canada Central Standard Time	Iran Standard Time	Sri Lanka Standard Time
Cape Verde Standard Time	Israel Standard Time	Taipei Standard Time
Caucasus Standard Time	Jordan Standard Time	Tasmania Standard Time
Cen. Australia Standard Time	Korea Standard Time	Tokyo Standard Time
Central American Standard Time	Mexico Standard Time	Tonga Standard Time
Central Asia Standard Time	Mexico Standard Time 2	US Eastern Standard Time
Central Brazilian Standard Time	Mid-Atlantic Standard Time	US Mountain Standard Time
Central Europe Standard Time	Middle East Standard Time	Venezuela Standard Time
Central European Standard Time	Montevideo Standard Time	Vladivostok Standard Time
Central Pacific Standard Time	Mountain Standard Time	W. Australia Standard Time
Central Standard Time	Mountain Standard Time (Mexico)	W. Central Africa Standard Time
Central Standard Time (Mexico)	Myanmar Standard Time	W. Europe Standard Time
China Standard Time	N. Central Asia Standard Time	West Asia Standard Time
Dateline Standard Time	Namibia Standard Time	West Pacific Standard Time
E. Africa Standard Time	Nepal Standard Time	Yakutsk Standard Time
E. Australia Standard Time	New Zealand Standard Time	
E. Europe Standard Time	Newfoundland Standard Time	
E. South America Standard Time	North Asia East Standard Time	
Eastern Standard Time	North Asia Standard Time	

6. Save, Unlock, and Close the Variable Item.

7. Log out of Aras Innovator.

- Restart the World Wide Web Publishing service on the server running the Innovator Server tier of Aras Innovator.

The end result is shown in the main Aras Innovator window.

For users whose local time does not match the corporate time, the UI displays both the local time and the corporate time:

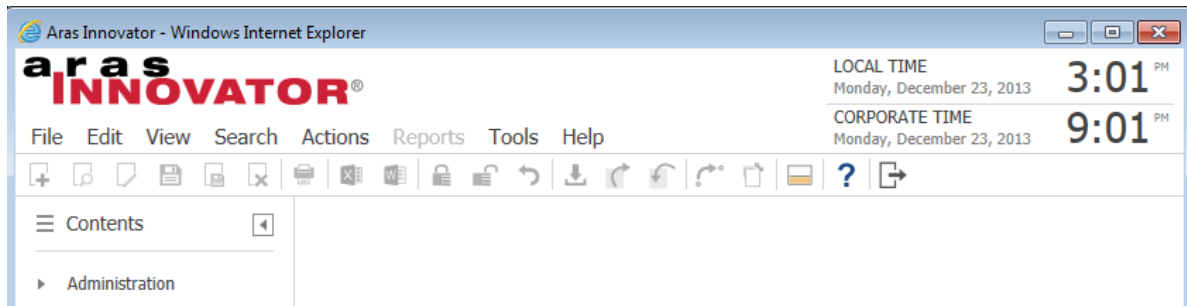


Figure 2.

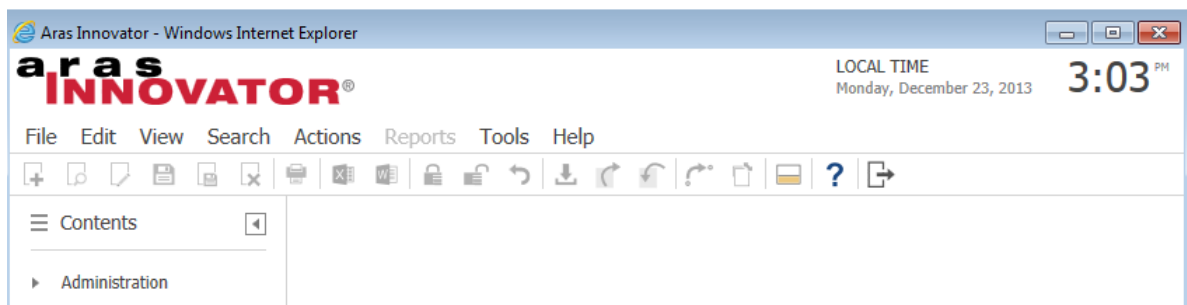


Figure 3.

The difference in time is established when the user first logs into Aras Innovator.

### 2.2.3 Update Time Zone Data

With the periodic changes to Time Zone information as well as Daylight Savings, it is necessary that the time zone data within Aras Innovator be updated. Similar to Windows Updates, this new information needs to be loaded into your system.

**To update the time zone data for your instance, use the following steps:**

- Log into Aras Innovator as an administrator.
- From the main menu, select **Actions --> Update** time zones data.  
This saves the `tzupdate.xml` to your Working Directory
- Once the file finishes downloading, a File Dialog is displayed. Select the `tzupdate.xml` file downloaded.
- Click **OK** when update has completed.

## 2.2.4 Defining a Language

Aras Innovator has the ability to store properties as 'Multi-Lingual Properties'. A multi-lingual property stores multiple values for the same property. The value displayed is based on the language defined in Aras Innovator for your Locale. In most cases, if there is no value for a specific language, the value for the default language (English) is shown.

## 2.2.5 Adding the Language Item

The first step to using multi-lingual properties is to define the Language Item in Aras Innovator.

---

**Warning** Adding of a new language should not be done in a live production system. The new language should be added off hours when no users are logged into the system.

---

1. Log into Aras Innovator as an administrator.
2. From the TOC, select **Administration --> Localizations-> Languages**.
3. From the main toolbar, click **Create a New Item**.

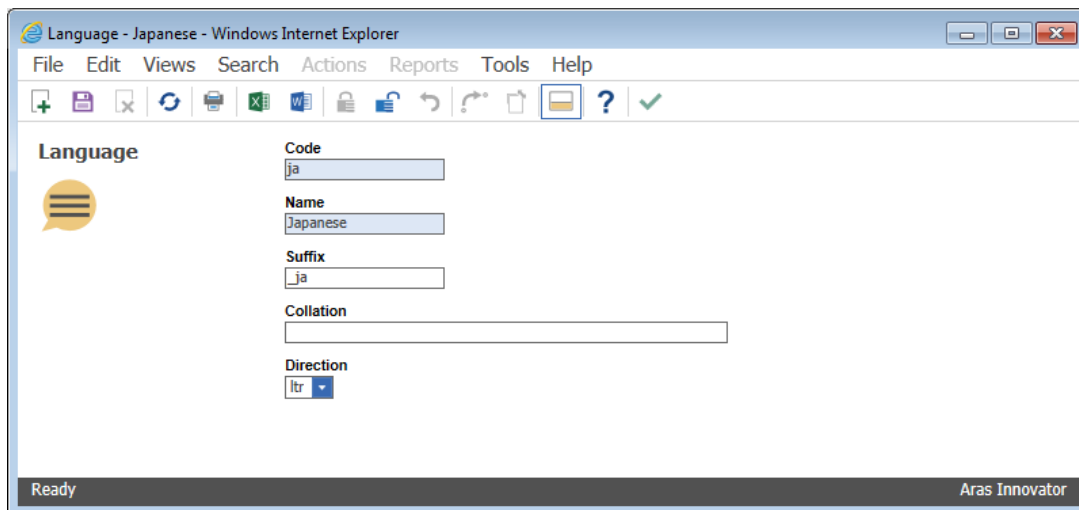


Figure 4.

- **Code** = The letter ISO code that identifies the language. For example, 'ja' for Japan
  - **Name** = The name of the language for reference in the keyed name. For example, Japanese
  - **Suffix** = The identifier for differentiation of one language from another in a multi-lingual property. For example, '\_ja' for Japanese
  - **Collation** = This property is used to set custom sorting rules for each language column at the database level. A null value for collation should simply mean that there is no custom collation for that column.
4. Save, Unlock, and Close.

## 2.2.6 Updating the User Interface (Xml-based resources)

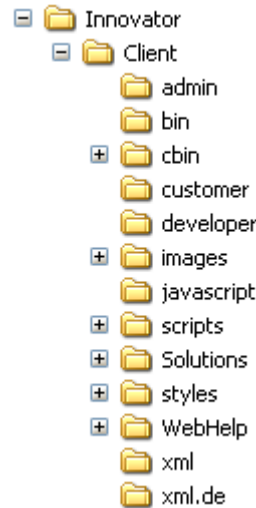
The second step to adding a language is to update the xml resources that define the menus, toolbars, and user interface elements.

The files that make up the majority of the XML-based resources. The language specific XML files need to be put in the /Client folder next to the default (English) counterparts.

Any file that cannot be found in the language specific folder can cause the corresponding file from the default (English) folder to be loaded.

Language specific folders should be in the format xml.{Language Item Code}

Example: xml.ja



Each of the standard Aras Innovator solutions (Program Management, Product Engineering, and Quality Planning) has its own xml resources. Each of these folders should have any language specific files put in a corresponding folder.

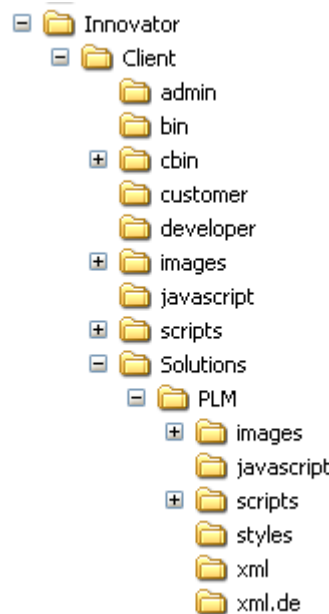
Quality Planning for instance has a folder '/Client/Solutions/QP/xml'. The corresponding folder for Japanese would be '/Client/Solutions/QP/xml.ja'

The three solutions installed with Aras Innovator have the following XML-based resource folders:

\Innovator\Client\Solutions\PLM

\Innovator\Client\Solutions\Project

\Innovator\Client\Solutions\QP



## 2.3 Defining the Supported Locales

Aras Innovator has only the English locale defined by default. Additional locales must be defined in the database.

To define a supported locale, use the following step:

1. Log into Aras Innovator as an administrator.
2. From the TOC, select **Administration --> Localizations-> Locales**.
3. From the main toolbar, select **Create a New Item**.

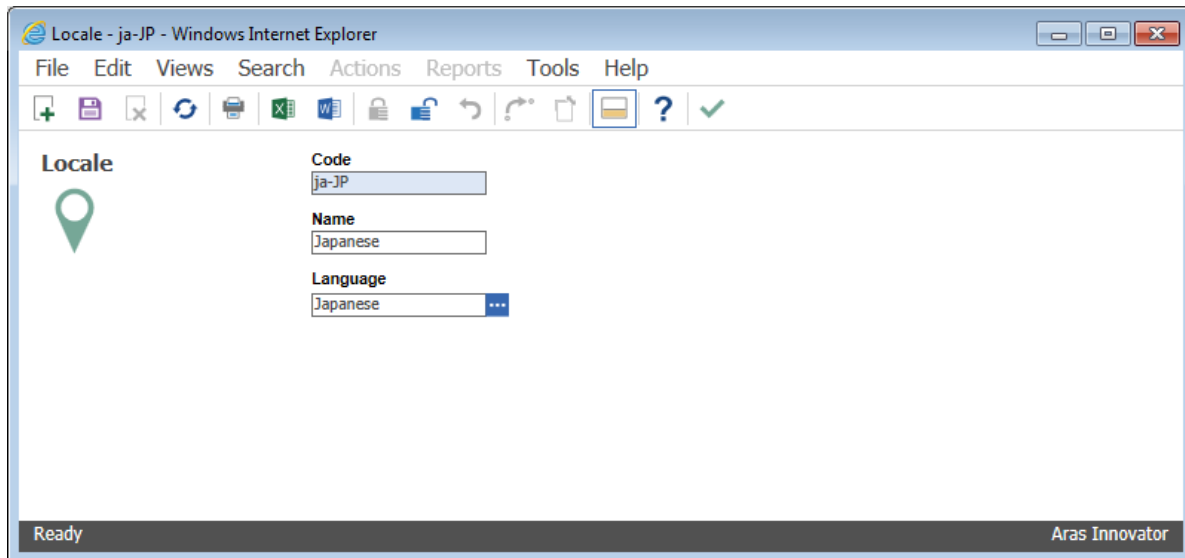


Figure 5.

4. **Code** = The culture code for locale. For example, 'ja-JP' for Japan
5. **Name** = The name of the locale.
6. **Language** = The language item that is referenced by users with this locale on their client PC machines. For example, 'Japanese' from section 2.2.5 [Adding a Language](#).
7. Save, Unlock, and Close.
8. Restart the World Wide Web Publishing service on the server running the Innovator Server tier of Aras Innovator.

## 3 The User Interface

This section outlines how the addition of a supported locale reflects to users.

### 3.1 Multi-Lingual Properties

Starting with version 10, Aras Innovator selects the language based on user's preferences specified in the browser. A multi-lingual property displays either a value in the locale specific language or a value in the default language (English).

When inputting values into a multi-lingual field, the value is saved to the language defined by the session. If the session language does not exist in the database, then the value is saved to the default (English) language.

To edit a language other than the defined by the session locale, the user can either select F2 in the while editing the property, or select the ellipse next to the field.

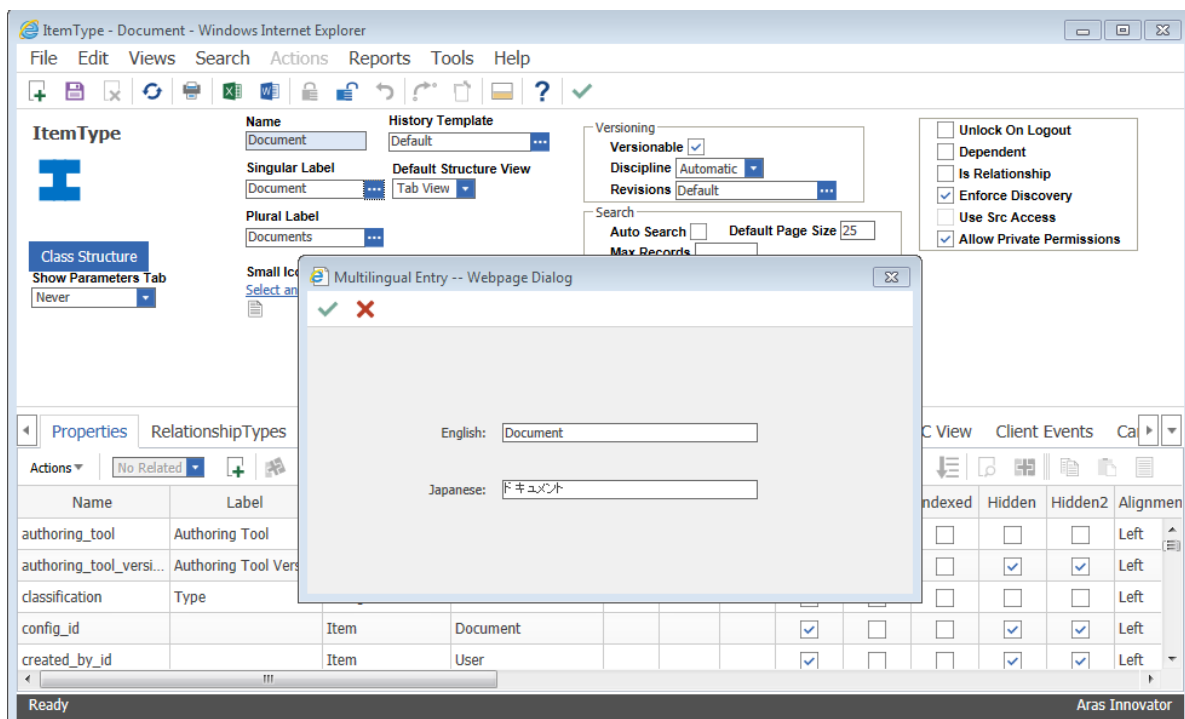


Figure 6.

In Figure 6, we see the multi-lingual dialog for the Singular Label.

**Note:** The Singular Label has an ellipsis button next to the field. Selecting this ellipsis opens the multi-lingual dialog and allows the user to input languages for locales other than the current. The same dialog can be opened by putting the cursor into the field and selecting F2 from your keyboard.

For properties in relationship grids, you need to put the cell into edit mode and then select F2 from your keyboard to open the multi-lingual dialog.

Forms and Grids display the property value based on the locale of the current session:

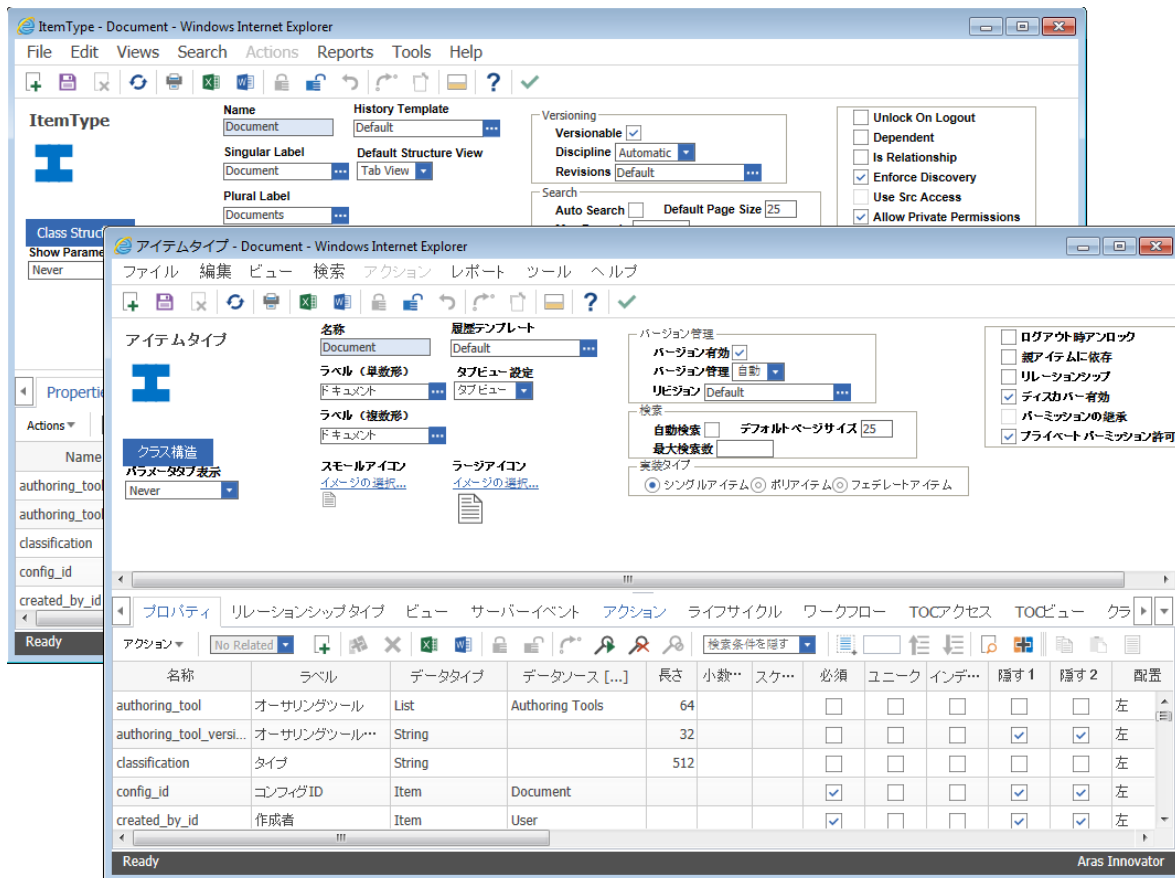


Figure 7.

If the multi-lingual property has a value for the language defined by the current session local then the value is displayed; otherwise the value for the default language (English) is displayed. The same rule applies for the xml resource defined UI elements, similar to the toolbar text.

## 3.2 Date Properties

Starting with version 10, Aras Innovator displays date properties based on user's language preferences specified in the browser. Date-Times are stored in the database as UTC (Universal Time Code).

On the client, the date-time is interpreted one of two ways.

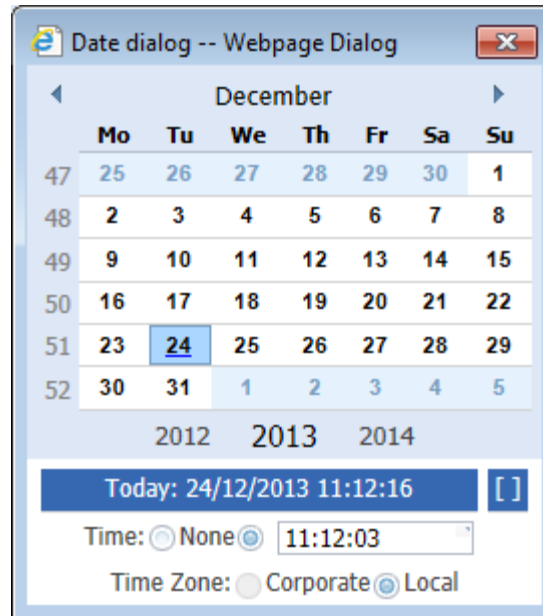
- If the Corporate Time Zone is set, then all DateTime values is displayed and applied in Corporate Time Zone.
- If the Corporate Time Zone is not set, then all DateTime values is displayed and applied in the time zone established by the local client session.



For user input, the new date dialog indicates whether the date-time input is based on local time zone or corporate time zone

Time indicates whether to pass a specific Time Value or to use None, in which case 12AM is used instead, although the UI may not display this option if time is not part of the property's pattern.

The Time Zone set of radio-buttons serves only as an indicator of what time zone is used by the client. If a 'Corporate Time Zone' is used by Aras Innovator then the Corporate radio button is checked and Local radio button is disabled; if a 'Corporate Time Zone' is not used by Aras Innovator then Local radio button is checked and Corporate radio button is disabled.



### 3.3 Locale Neutral Properties

Decimal and float properties are also be stored in the database in a locale Neutral format. The format of the dates and numbers displayed in the Aras Innovator UI is determined from the DOJO toolkit as well as the .Net Localization libraries.

## 4 Populating a Language

Once a language has been defined, there are two main ways to populate the language.

- You can manually enter all language data through the UI.
- You can import a language using the Aras Innovator Language Pack Management Utility.

### 4.1 Manually Managing Language Packs in the Database

All multi-lingual deployments of Aras Innovator eventually need to manually configure languages in Aras Innovator, to account for its specific customization.

Once you have defined a language, refer to section 2.2.4 [Defining a Language](#), you can add values to this language for each of the UI elements.

Table 2:

ItemType	Property	Use	Where to edit in the UI
Action	Label	Displays in the Action menu of the toolbars	Administration\Actions
Activity Template	Label	Text to display the Activity message of a workflow	Administration\Workflow Maps property on the Activity
Activity Template	message	Text to display the Activity label of a workflow	Administration\Workflow Maps property on the Activity
Activity Template Task	description	Text to display the Task description of a Workflow Activity	Administration\Workflow Maps under the Tasks label Relationship of an Activity
Activity Template Variable	Label	Text to display the label of an Activity Variable of a workflow	Administration\Workflow Maps under the Variable label Relationship of an Activity
Business Calendar Exception	description	Text displayed to describe a Business Day exception.	Administration\Calendars on the Exceptions label relationship
Field	Label	Text displayed to label a field in a form	Administration\Forms on the Field Label relationship
Field	legend	Text displayed to label a field border in a form	Administration\Forms on the Field Border relationship
Filter Value	Label	Text displayed in pick lists of a filtered list in the UI	Administration\Lists on the Filter Value relationship
Grid Column	Label	Text displayed in Configurable Grid column headers	Administration\Grids on the Grid Column relationship
ItemType	label_plural	Text used in TOC	Administration\ItemTypes
ItemType	Label	Text used in item properties, and Item window title	Administration\ItemTypes

Life Cycle State	Label	Text displayed to label a state of a lifecycle	Administration->LifeCycles on the Life Cycle State properties
Message	Text	Text displayed for a notification message	Administration->Notification Messages
Message	Title	Title of the notification message	Administration->Notification Messages
PM_ProjectGridLayout	Label	Text displayed in the column header of the Project grid	Administration\Preferences on the Project Grid Layout relationship of the World Preference
Property	Label	Text used in the search grid column headers	Administration\ItemTypes on the Properties relationship
Property	default_value	Text used in the default value of a multi-lingual string property	Administration\ItemTypes on the Properties relationship
RelationshipType	Label	Text used the relationship tabs in Item windows	Administration\RelationshipTypes
Report	Label	Displays in the Report menu of the toolbars	Administration\Reports
SavedSearch	Label	Text use for the name of Saved Search	Administration\Saved Searches
SearchMode	Label	Text used to display the different search modes	ItemType not exposed through UI by default
UserMessage	Text	Displays in custom error messages in the UI	ItemType not exposed through UI by default
Value	Label	Text displayed in pick lists in the UI	Administration\Lists on the Value relationship
Workflow Map	description	Text displayed for description of a workflow	Administration\Workflow Maps
Workflow Map	Label	Text displayed in UI for name of a workflow	Administration\Workflow Maps
Workflow Map Path	Label	Text to display the vote path between two Workflow Activities	Administration\Workflow Maps on a path between two activities
Workflow Map Variable	Label	Text to display the label of a Wokflow variable	Administration\Workflow Maps on the Process Variable relationship

Each row in the table above represents a property in the user interface that has been defined as a multi-lingual string. The translator needs to edit each Item of that type that needs to display a non-English value.

## 4.2 Using the Language Pack Management Utility

To aid in the ability to move language packs from one database to the next, Aras has created a Language Pack Management Utility to export and import language values from a database. Language Pack Management Utility is a command line utility that imports and exports language packs from the Aras Innovator database on a SQL level. The following explains how to use the utility available from the

Language Pack Management Utility folder of the Aras Innovator CD Image or from the Aras Community Projects site at <http://www.aras.com/communityProjects/default.aspx>.

### 4.2.1 The Command Line

There are several elements to the Language Pack Management Utility command line tool. First, is the executable file for the Language Pack Management Utility itself, `LanguageTool.exe`. The details of the various options for running the executable at the command line are as follows:

```
Usage: LanguageTool.exe [-export/-import] [-config_file:<file path>]
[-folder:<folder path>] [-log_file:<file path>] [-log_level:<log level>]
] [-server:<SQL server name>] [-db:<database name>] [-login:<login name>]
[-pwd:<login password>]
```

Where:

- `-export/import`: This is the required parameter to indicate if the operation is to export or import a language.
- `-config_file`: (Optional) The path to the configuration file for current operation, `ImportConfig.xml` or `ExportConfig.xml`.
- `-folder`: (Optional) The path to the data folder. This is the output folder for the export operation, or the input folder for the import operation.
- `-log_file`: (Optional) The path to the log file.
- `-log_level`: (Optional) The logging level (high, medium, low).
- `-server`: (Optional) The name of the Microsoft SQL Server instance where the Aras Innovator database is installed.
- `-db`: (Optional) The name of the Aras Innovator database.
- `-login`: (Optional) The innovator login assigned to the Aras Innovator database.
- `-password`: (Optional) The password for the innovator login assigned to the Aras Innovator database.

**Note:** The optional parameters can be generally included in the `LanguageTool.exe.config` configuration file, but the command line parameters take precedence.

## 4.2.2 LanguageTool.exe.config

The LanguageTool.exe.config configuration file that is referenced by the Language Pack Management Utility is a straightforward xml file that defines the parameters for the batch load activity.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
    <sectionGroup name="userSettings"
      type="System.Configuration.UserSettingsGroup, System,
        Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" >
      <section name="Aras.Tools.I18N.ExpImpMLData.Properties.Settings"
        type="System.Configuration.ClientSettingsSection, System,
          Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
          allowExeDefinition="MachineToLocalUser" requirePermission="false"/>
      </sectionGroup>
    </configSections>
    <userSettings>
      <Aras.Tools.I18N.ExpImpMLData.Properties.Settings>
        <setting name="config_file" serializeAs="String">
          <value>C:\German\ImportConfig.xml</value>
        </setting>
        <setting name="folder" serializeAs="String">
          <value>C:\German\data</value>
        </setting>
        <setting name="log_file" serializeAs="String">
          <value>C:\German\log.txt</value>
        </setting>
        <setting name="log_level" serializeAs="String">
          <value>high</value>
        </setting>
        <setting name="server" serializeAs="String">
          <value>localhost</value>
        </setting>
        <setting name="db" serializeAs="String">
          <value>InnovatorSolutions</value>
        </setting>
        <setting name="login" serializeAs="String">
          <value>innovator</value>
        </setting>
        <setting name="pwd" serializeAs="String">
          <value>innovator</value>
        </setting>
      </Aras.Tools.I18N.ExpImpMLData.Properties.Settings>
    </userSettings>
  </configuration>
```

To help understand how to configure this file, the example above can be broken down line by line:

```
<setting name="config_file" serializeAs="String">
  <value>C:\German\ImportConfig.xml</value>
</setting>
```

This is the path to the path to a configuration file for current operation, ImportConfig.xml or ExportConfig.xml.

```
<setting name="folder" serializeAs="String">  
  <value>C:\German\data</value>  
</setting>
```

The path to the data folder. This is the output folder for the export operation, or the input folder for the import operation.

```
<setting name="log_file" serializeAs="String">  
  <value>C:\German\log.txt</value>  
</setting>
```

This is the path to the log file.

```
<setting name="log_level" serializeAs="String">  
  <value>high</value>  
</setting>
```

This is the logging level, 'high' recommended.

```
<setting name="server" serializeAs="String">  
  <value>localhost</value>  
</setting>
```

The name of the Microsoft SQL Server instance where the Aras Innovator database is installed. This value can be found in the Aras Innovator InnovatorServerConfig.xml in DB-Connection tag. (server="localhost")

```
<setting name="db" serializeAs="String">  
  <value>InnovatorSolutions</value>  
</setting>
```

This is the name of the Aras Innovator database in SQL Server. This value can be found in the Aras Innovator InnovatorServerConfig.xml in DB-Connection tag. (database='InnovatorSolutions')

```
<setting name="login" serializeAs="String">  
  <value>innovator</value>  
</setting>
```

This is the login assigned to innovator user of the Aras Innovator database in SQL Server. This value can be found in the Aras Innovator InnovatorServerConfig.xml in DB-Connection tag. (uid='innovator')

```
<setting name="pwd" serializeAs="String">  
  <value>innovator</value>  
</setting>
```

This is the name login assigned to innovator user of the Aras Innovator database in SQL Server. This value can be found in the Aras Innovator InnovatorServerConfig.xml in DB-Connection tag. (pwd='innovator')

Any of the parameters may be left out as long as the command line equivalent parameter is defined.

### 4.2.3 Export Configuration

In order to export a language pack from the Aras Innovator database, you need to define what languages and ItemTypes to export. The definitions of these settings are found in the `ExportConfig.xml`.

```

<export>
  <languages list="en, de" />
  <itemtype name="Action" />
  <itemtype name="Activity Template" />
  <itemtype name="Activity Template Task" />
  <itemtype name="Activity Template Variable" />
  <itemtype name="Business Calendar Exception" />
  <itemtype name="Field" />
  <itemtype name="Filter Value" />
  <itemtype name="Grid Column" />
  <itemtype name="ItemType">
    <properties list="label,label_plural" />
  </itemtype>
  <itemtype name="Life Cycle State" />
  <itemtype name="PM_ProjectGridLayout" />
  <itemtype name="Property" />
  <itemtype name="RelationshipType" />
  <itemtype name="Report" />
  <itemtype name="SavedSearch" />
  <itemtype name="SearchMode" />
  <itemtype name="UserMessage" />
  <itemtype name="Value" />
  <itemtype name="Workflow Map" />
  <itemtype name="Workflow Map Path" />
  <itemtype name="Workflow Map Variable" />
</export>

```

To help understand how to configure this file, the example above can be broken down line by line:

```
<languages list="en, de" />
```

These are the languages by Language Item code property defined in Aras Innovator. The above example shows en (English) and fr (French).

```
<itemtype name="Field" />
```

Each ItemType defined in this file contains all of its multi-lingual string properties exported from the database.

```
<properties list="label,label_plural" />
```

If you want to select some, but not all multi-lingual string properties, add this tag with the properties to export.

#### 4.2.4 Import Configuration

In order to import a language pack into the Aras Innovator database, you first need to define the Language Item in the database, refer to section 2.2.4 [Defining a Language](#). Next, you need to setup the ImportConfig.xml.

```

<import>
  <itemtype name="Action" />
  <itemtype name="Activity Template" />
  <itemtype name="Activity Template Task" />
  <itemtype name="Activity Template Variable" />
  <itemtype name="Business Calendar Exception" />

```

```

<itemtype name="Field" />
<itemtype name="Filter Value" />
<itemtype name="Grid Column" />
<itemtype name="ItemType" />
<itemtype name="Life Cycle State" />
<itemtype name="PM_ProjectGridLayout" />
<itemtype name="Property" />
<itemtype name="RelationshipType" />
<itemtype name="Report" />
<itemtype name="SavedSearch" />
<itemtype name="SearchMode" />
<itemtype name="UserMessage" />
<itemtype name="Value" />
<itemtype name="Workflow Map" />
<itemtype name="Workflow Map Path" />
<itemtype name="Workflow Map Variable" />
</import>

```

Each ItemType defined in this file should have a corresponding XML file containing the language values in the data folder.

## 4.3 XML Resources

The XML resources represent the static content of the Aras Innovator user interface, such as menu options and toolbar buttons. There is no automated tool for updating these resources. They must be done manually in a text editor.

For instance in the \Innovator\Client\xml\item\_menu.xml file

```

<?xml version="1.0" encoding="UTF-8"?>
<menuapplet on_load="onItemMenuLoad" on_select="onClickItem"
on_check="onCheckMenu" show="item_menubar">
  <menubar id="item_menubar">
    <menu id="file_menu" name="File">

```

Would become

```

<?xml version="1.0" encoding="UTF-8"?>
<menuapplet on_load="onItemMenuLoad" on_select="onClickItem"
on_check="onCheckMenu" show="item_menubar">
  <menubar id="item_menubar">
    <menu id="file_menu" name="Datei">

```

In the \Innovator\Client\xml.de\item\_menu.xml file.

Aras Innovator houses all the generic user interface XML resources in the \Innovator\Client\xml folder. However, each solution could have its own xml resources as well. See section 2.2.6 [Updating the User Interface \(Xml-based resources\)](#).



## 4.4 Updating the Resource.Config.xml

The Resource.Config.xml located in \Innovator\Client\XML folder defines the language(s) that can be loaded on the Client. When you define the language pack, you need to update Resource.Config.xml file and add a reference to your language.

Below is an example of adding the German Language to the Resource.Config.xml:

```
<satelliteresources>  
  ...  
  <resource culture="neutral"/>  
  <resource culture="ja"/>  
  <resource culture="de"/>  
  ...
```

Once you have updated the Resource.Config.xml and saved it, you need to Restart IIS on the Server.

## 5 Programming Notes

---

For details on any subjects in this section, refer to the Aras Innovator - Programmer's Guide. This section is only meant to be a reference to areas of interest for programmers. The Aras Innovator UI automatically accounts for language, time, and locale variables for the end user, but the programmer needs to know what to look out for when writing Solutions, Solutions Add-Ons, and Integration projects.

### 5.1 AML

AML must be written to support the multi-lingual properties and the locale neutral data. When querying for AML, you must understand the format of the return. When applying AML, you must be sure to apply multi-lingual values to the correct language, apply DateTimes in the correct time zone, and properties with locale neutral data must have the correct formats.

#### 5.1.1 Multi—Lingual Properties.

Multi-lingual properties should indicate the specific language being passed, if not the default (English). Multi-Lingual properties now contain the xml:lang attribute to indicate what language is returned.

##### 5.1.1.1 Get Queries

Let us first consider a query with action='get' from an English client.

```
<Item type="ItemType" id="B88C14B99EF44982..." action="get" select="label"/>
```

This query returns the label in the language of the client session established at the time of connection value of the label property. If there is no locale matching the client at the time the session is established, the default (English) is returned.

```
<Item type="ItemType" id=" B88C14B99EF44982...">
  <label xml:lang="en">Document</label>
</Item>
```

From a German client, the return would include the German label

```
<Item type="ItemType" id="B88C14B99EF44982...">
  <label xml:lang="de">Dokument</label>
</Item>
```

Requests may specify the language(s) desired in the response by including the 'language' attribute on the Item tag. The value of this attribute may either be a comma-delimited list of language codes or an asterisk (\*) to request all languages.

```
<Item type="ItemType" id="B88C14B99EF44982..." action="get" select="label"
language="en,de"/>
```

To return

```
<Item type="ItemType" id=" B88C14B99EF44982..." >
  <i18n:label xml:lang="de"
xmlns:i18n="http://www.aras.com/I18N">Dokument</i18n:label>
  <i18n:label xml:lang="en"
xmlns:i18n="http://www.aras.com/I18N">Document</i18n:label>
</Item>
```

If the language attribute is not present, the AML should be interpreted to be requesting session-language values for all multilingual properties. The fallback behavior is to return the default-language value for any property with no session-language value.

### 5.1.1.2 Add/Edit/Update Queries

When adding or editing an Item through an AML query, the most important thing to keep in mind is that unless the namespace is defined, then session language is updated. This is either the language defined by the session locale, or the default(English) language for locales not defined in the Aras Innovator Database

```
<Item type="ItemType" id="B88C14B99EF44982..."action="edit">
  <label>Document</label>
</Item>
```

Following is the result for an en-US session:

```
<Item type="ItemType" id=" B88C14B99EF44982..." >
  <i18n:label xml:lang="de" is_null="1" xmlns:i18n=http://www.aras.com/I18N
/>
  <i18n:label xml:lang="en"
xmlns:i18n="http://www.aras.com/I18N">Document</i18n:label>
</Item>
```

While the same query displays the following result for a de-DE session:

```
<Item type="ItemType" id=" B88C14B99EF44982..." >
  <i18n:label xml:lang="de"
xmlns:i18n="http://www.aras.com/I18N">Document</i18n:label>
  <i18n:label xml:lang="en" is_null="1" xmlns:i18n="http://www.aras.com/I18N"
/>
</Item>
```

In order to set a language value, other than the default, of a property you must specify the namespace as a property attribute.

```
<Item type="ItemType" id=" B88C14B99EF44982..." >
  <i18n:label xml:lang="de"
xmlns:i18n="http://www.aras.com/I18N">Dokument</i18n:label>
  <i18n:label xml:lang="en"
xmlns:i18n="http://www.aras.com/I18N">Document</i18n:label>
</Item>
```

If the namespace is not explicitly defined, then the value is assumed to be the value for the session language.

## 5.1.2 DateTime Properties

DateTime properties are passed to and from the Innovator Server in a neutral data format and the time zone may vary based on a fixed set of rules. DateTime values passed to\from server don't contain milliseconds as fractions of a second are ignored by the Innovator.

### 5.1.2.1 Get Queries

When reading the returns of a query with action="get" there are two main rules to keep in mind.

- If the CorporateTimeZone variable is set in the database, refer to section 2.2 [Time Zones](#), then all DateTime properties are returned in the corporate time zone.
- If the CorporateTimeZone is NOT set, then all DateTime properties are returned in the local time zone of the client session that applied the AML.

This includes AML applied by methods executed as a result of client actions in the UI.

### 5.1.2.2 Add/Edit/Update Queries

When applying AML to Aras Innovator, there are three main rules to keep in mind.

- If the CorporateTimeZone variable is set in the database, refer to section 2.2 [Time Zones](#), then all DateTime properties must be applied in the corporate time zone.
- If the CorporateTimeZone is NOT set, then all DateTime properties must be applied in the local time zone of the client session context that applied the AML.
- The DateTime applied in the AML must conform to the pattern yyyy-MM-dd[Thh:mm:ss]

**Note:** a) the optional portion is enclosed in "[]";  
 b) the format doesn't contain milliseconds as they are ignored by the server).

This includes AML applied by methods executed as a result of client actions in the UI.

There is one special case when applying updates to datetime properties in AML to Aras Innovator. It is possible to specify '\_\_now()' as the value for properties of type datetime, instead of a specific value.

#### Example:

```
<Item type="Document" id="ACD123C46C1BFE8ED4E8BDDE0B009812">
  <effective_date>__now()</effective_date>
</Item>
```

When a request similar to this is received by the server, it writes the value of the current moment of time into the database for the date property effective\_date.

## 5.1.3 Locale Neutral Data Formats

Date, decimal, and float properties are displayed in the Aras Innovator UI based on the patterns in the client sessions Regional and Language settings, however the AML does not follow this rule. AML for these property types are sent in a neutral data format.

### 5.1.3.1 Get Queries

An AML query with action="get" returns date, decimal, and float properties in a data neutral format. Conversion of these values to a locale specific format should be done through client javascript.

```
<Item type="Test Item" id="D95C15B649AC46C1BFE8ED4E8BDDE0B0" action="get"/>
```

#### Returns

```
<Item type="Test Item" id="D95C15B649AC46C1BFE8ED4E8BDDE0B0">
  <date_prop>2006-12-01T08:12:45</date_prop>
  <decimal_prop>1234.56</decimal_prop>
  <float_prop>-1234.56</float_prop>
```

```
</Item>
```

There are some external solutions where the AML returned must be in the patterned format, rather than the neutral data format. For these cases, you can use `action='do_l10n'` `initial_action='get'` to convert dates to the client session patterns. This is not recommended over the use of javascript on the client page, as it could have performance costs.

Example:

If you run the following query from a client with locale of ru-RU

```
<Item type="Test Item" id="D95C15B649AC46C1BFE8ED4E8BDDE0B0" action="do_l10n"
initial_action="get"/>
```

Aras Innovator returns

```
<Item type="Test Item" id="D95C15B649AC46C1BFE8ED4E8BDDE0B0">
  <date_prop neutral_value="2006-12-01T08:12:45">01.12.2006</date_prop>
  <decimal_prop neutral_value="1234.56">1234,56</decimal_prop>
  <float_prop neutral_value="-1234.56">-1234,56</float_prop>
</Item>
```

### 5.1.3.2 Add/Update/Edit Queries

Date, decimal, and float properties must be in a locale neutral data format for any applied AML query to Aras Innovator.

```
<Item type="Test Item" id="D95C15B649AC46C1BFE8ED4E8BDDE0B0" action="add">
  <date_prop>2006-12-01T08:12:45</date_prop>
  <decimal_prop>1234.56</decimal_prop>
  <float_prop>-1234.56</float_prop>
</Item>
```

## 5.2 IOM

Here are few statements that always have to be kept in mind when dealing with locale specific types using IOM:

- The format of property values that are local specific ('date', 'float', 'decimal') are always presented in an AML in the neutral format.
- Property values of type 'date' are always presented in AML in the time zone of the current session
- Dates are always stored in the database in UTC (no time zone)

In order to support conversion to/from neutral format as well as get information about the current session locale\language\time zone\etc. IOM. Innovator class has a new method `getI18NSessionContext()` that returns an instance of a class that implements `I18NSessionContext` interface. The interface allows to get information about the session locale\language\etc. (`getLocale()`, `getLanguageCode()`, etc.) as well as contains methods for converting to/from neutral format (`ConvertToNeutral(..)`, `ConvertFromNeutral(...)`) and methods specific for converting to/from UTC date-time (`ConvertUtcDateTimeToNeutral(..)`, `ConvertNeutralToUtcDateTime(..)`).

Here is the exact format of the conversion methods:

```
string ConvertToNeutral (string value, string type, string format )
```

where,

- 'value' is a string that has to be converted (e.g. '12/25/2003')
- 'type' is name of an Innovator type which representation is locale dependent; this parameter can be one of the following 'date' | 'float' | 'decimal'
- 'format' – format of the 'value'. If the parameter is either null or empty string then an attempt is made to parse passed 'value' assuming that the 'value' has a format that is valid for the session locale; if parsing failed then an exception is thrown.

```
string ConvertFromNeutral( string value, string type, string format )
```

where,

- 'value' is a string in neutral format that has to be converted (in most cases the string would be a property value taken from AML)
- 'type' is name of an Innovator type which representation is locale dependent; this parameter can be one of the following 'date' | 'float' | 'decimal'
- 'format' – the parameter is taken into account only if 'type'='date' in which case it's one of the following: 'short\_date' | 'long\_date' | 'short\_date\_time' | 'long\_date\_time' | 'long\_time' | 'short\_time'.

**Note:** In each case, the string representation of the 'value' returned by the method depends on the locale of the session (in other words: conversion to, let's say, 'short\_date' of a particular 'value' results into different short date format representations of the 'value' depending on the session locale).

```
string ConvertUtcDateTimeToNeutral( string utcDateTime, string inFormat );
```

the method returns string that represents 'utcDateTime' in the time zone of the session and in neutral format. Parameters are:

- 'utcDateTime' is a string that is interpreted either as UTC date-time if it doesn't have an offset or has offset 0 or as the exact moment of time if it has an offset.
- 'inFormat' defines the format of the 'utcDateTime' string.

**Note:** If 'inFormat' is null then an attempt is made to parse passed 'utcDateTime' assuming that it has a valid invariant culture format. If 'inFormat' is not null then it has to represent a valid invariant culture date\time format.

```
string ConvertNeutralToUtcDateTime( string neutralDateTime, string outFormat ) ;
```

the method returns a string that represents the corresponding UTC date-time in the specified format.

- 'neutralDateTime' is a string that represents a date-time value in the session time zone in neutral format
- 'outFormat' is the format in which the resulting UTC date-time is returned.

Let's consider several specific use-cases in order to understand i) what the above statements mean in each situation; ii) how I18NSessionContext methods can help.

### Example 1

User writes a client method which makes request to the server and then converts obtained value of the received property 'modified\_on' of type 'date' into the 'long\_date' format of the locale of the current session in order to display it in UI

```
...
// Make request to the server and get the value of the "modified on" property
var request = this.newItem("Issue", "get");
request.setProperty("id", "4A6478F0C0CB46F1B24D9526223FA14F");
var response = request.apply();
var issue_date_str = response.getProperty("modified_on");

// At this point the value of "issue_date_str" is something like:
// 2007-12-25T15:23:30
// It has to be converted to the format that appropriate for the
// locale of the session:
var cntx = this.getInnovator().getI18NSessionContext();
var new_value = cntx.ConvertFromNeutral(issue_date_str, 'date', 'short_date');

// At this point the "new_value" would look as the following (let's assume
// that the session locale is "en-US"; also remember that we used
// 'short_date'
// format so time portion was cut):
// 12/25/2008
...
```

### Example 2

User writes a client method that reads a form field that contains a date; the user wants to build an AML putting the value read from the field into the AML property 'effective\_date'.

```
...
// Read the value from the field
var duedate = document.forms[0].new_due_date.value;

// The "duedate" looks as the following (let's assume that session locale is
// "fr-FR" (French(France)): 25/12/2007
// Convert to neutral format because according to the statement a.
// all dates in AML must be in neutral format:
var cntx = this.getInnovator().getI18NSessionContext();
var dd_neutral = cntx.ConvertToNeutral(duedate, 'date', '');

// The value of "dd_neutral" would look like: 2007-12-25
// Now create item and set its property 'effective_date'
var newPart = this.newItem("Part", "add");
newPart.setProperty("item_number", "xxx");
newPart.setProperty("effective_date", dd_neutral);
...
```

### Example 3

User writes a stand-alone application using IOM to access Aras Innovator and need to form a correct AML for properties of type 'date'

```

...
// Instance of IOM.Innovator is available as 'inn'
I18NSessionContext cntx = inn.getI18NSessionContext();

Item my_part = inn.newItem( 'Person', 'add' );

// Let's assume that user of the application entered the date-time value
// in a text field; this value is a) in the time zone of the machine which is
// also
// the time zone of the current Aras Innovator session; b) in the
'long_date_time'
// format of the session locale (let's assume it's "en-US"). Before the
conversion
// the date looks like "Friday, October 10, 2008 01:00:00 PM"; after the
conversion -
// "2008/10/10T13:00:00"
string dob_val = , cntx.ConvertToNeutral(inputText.Text, 'date', '');

// Finally set the property
my_part.setProperty( 'date_of_birth', dob_val );
...

```

### Example 4

User writes a stand-alone application using IOM to access Aras Innovator and need to show dates obtained in AML in the short-date-time format of the machine locale (which is also the locale of the current session).

```

...
// Instance of IOM.Innovator is available as 'inn'
I18NSessionContext cntx = inn.getI18NSessionContext();

// Make the request to get data
Item request = this.newItem( 'Part', 'get' );
request.setProperty( 'item_number', 'xxx' );
Item response = request.apply();
string part_date_str = response.getProperty( 'modified_on' );

// 'modified_on' date is in the time zone of the current session (according
// to the statement b) and in neutral format (according to the statement a).
// So before the conversion the date would like: "2005/09/17T11:22:35",
// after the conversion (let's assume that the locale is "fr-FR"):
// "17/09/2005 11:22:35"
inputText.Text = cntx.ConvertFromNeutral( part_date_str, 'date', 'short_date'
);

```



## Example 5

User writes a server 'onBefore' method in which the user needs to compare the created\_on property in the AML with some other date that needs to be obtained from the database.

```
...
// The value is the date in the session time zone (according to the statement
b)
// which is presented in neutral format (per statement a).
string created_on_str = this.getProperty( 'created_on' );

// Convert this to a date-time object
DateTime created_on_dt = DateTime.Parse(created_on_str);

// Build request to the database and get another date with which we need to
compare
// Note that interested us property value of type 'date' is also in the
neutral
// format (according to statement a) and in the time zone of the session
(statement b)
Item request = this.newItem( 'Part', 'get' );
request.setProperty( 'item_number', 'xxx' );
Item response = request.apply();
string part_date_str = response.getProperty( 'modified_on' );
DateTime part_dt = DateTime.Parse( part_date_str);

// Because both created_on_dt and part_dt are DateTime objects
// in the same time zone they could be compared
if( DateTime.Compare( created_on_dt, part_dt ) != 0 )
...

```

## Example 6

Let's say user writes a server method that has to set property `closed_on` of type 'date'. So, the format of the property value must be in neutral format and the value must be in the time zone of the session in which the method is called. Here is a problems though: an attempt to create an instance of `DateTime` on server would return a date\time object in the time zone of the server which might be different from the time zone of the current session; from other side methods for getting `DateTime` in particular time zone and/or converting from a time zone to another time zone are available only in .NET 3.5. In this case the best option is to use method `ConvertUtcDateTimeToNeutral(...)` available in `I18NSessionContext` class:

```
...
// First, get session context
I18NSessionContext cntx = this.getInnovator().getI18NSessionContext();

// Get 'now' moment as date-time in UTC in SortableDateTimePattern
// which is: "yyyy'-'MM'-'dd'T'HH':'mm':'ss"
String utc_now = DateTime.UtcNow.ToString( "s" );

// Get the same moment of time in the time zone of the session
// and in neutral format.
String pval = cntx.ConvertUtcDateTimeToNeutral( utc_now, null );

// Finally set the property value
this.setProperty( 'closed_on', pval );
...

```

**Note:** Similar to the above case in many cases it's required to set 'now' moment of time as the value of a property. In this cases an alternative simpler approach would be to put `__now()` into the property `closed_on`. Note that can be done only in request AML as the `__now()` is interpreted by the Innovator server and replaced on the equivalent date-time representation of 'this moment of time'. So that can be done in let's say a method that is called `onBeforeUpdate` as the resulting AML is processed by the update action of server and the `__now()` is interpreted correctly. At the same time if the method is invoked on, let's say, `onAfterUpdate` putting `__now()` would be a wrong thing to do because this syntax is allowed only in AML requests (not responses (!) and `onAfterUpdate` event works on a response AML that is about to be sent back to the client) of add/update type (i.e. `action="add" | "edit" | "merge" | "update" | etc.`) – refer to section 3.2 [DateTime Properties](#) for more details.

### Example 7

Let's say a user triggers a Server Event from Client machine in the United States locale (en-US) but the server is located on a German locale with a German operating system (de-DE). This German server by default handles decimals in the format 123.456,78 whereas the US client handles decimals 123,456.78. Because of this mismatch in decimal handling on the client and server the server method must be able to handle an invariant culture for decimals. Here is some sample code written in C# to parse a string (returning a decimal property) and convert that string to Decimal:

```
string dval_str_enUS = "123,456.78"; // US Decimal String
string dval_str_deDE = "123.456,78"; // German Decimal String

//Convert US Decimal string to Decimal value
decimal dval_enUS = Decimal.Parse( dval_str_enUS,
CultureInfo.InvariantCulture );

//Convert German Decimal string to Decimal Value
decimal dval_deDE = Decimal.Parse( dval_str_deDE,
CultureInfo.InvariantCulture );

...//Perform some computations on decimal values...

this.setProperty("my_prop", cntx.ConvertToNeutral(dval_enUS.ToString(), "decimal", ""));

...
```

## 5.3 Date Related SQL Queries

AML is the language of Aras Innovator, but some external add-ons do connect to Aras Innovator on a direct SQL level. Because Aras Innovator stores DateTimes in SQL Server as UTC, we have provided an add-on function for Microsoft SQL Server that allows the easy conversion of DateTime to and from UTC.

To convert DateTime values from UTC to a specific time zone, use the function `ConvertToLocal({value},{Time Zone})`. The time zone should be specified according to the registry key name of the time zone desired. (Refer to section 2.2.2 [Corporate Time Zone](#).) To use the DEFAULT parameter for Time Zone it is required that a value be set for `CorporateTimeZone`.

### Example:

```
select item_number, created_on
```

```
from innovator.Document
```

Would be written as

```
select item_number, innovator.ConvertToLocal(created_on, DEFAULT) as  
CreadtedOn
```

```
from innovator.Document
```

Or

```
select item_number, innovator.ConvertToLocal(created_on, 'Eastern Standard  
Time') as CreadtedOn
```

```
from innovator.Document
```

To convert DateTime values from a specific time zone to UTC, use the function `ConvertFromLocal({value},{Time Zone})`. The time zone should be specified according to the registry key name of the time zone desired. (Refer to section 2.2.2 [Corporate Time Zone](#)) To use the DEFAULT parameter for Time Zone it is required that a value be set for CorporateTimeZone.

### Example:

```
update innovator.Document
```

```
set effective_date = '1/1/2007 00:00:00.00'
```

Would be written as

```
update innovator.Document
```

```
set effective_date = innovator.ConvertFromLocal('1/1/2007  
00:00:00.00', DEFAULT)
```

Or

```
update innovator.Documnet
```

```
set effective_date = innovator.ConvertFromLocal('1/1/2007  
00:00:00.00', 'Eastern Standard Time')
```

This allows users to write queries in a familiar time zone context.

```
select item_number, innovator.ConvertToLocal(release_date, DEFAULT) as  
ReleaseDate
```

```
from innovator.Document
```

```
where created_on >= innovator.ConvertFromLocal('1/1/2007  
00:00:00.00', DEFAULT)
```

## 6 Global Deployment and Performance

---

When dealing with a global deployment of Aras Innovator, one of the most common concerns is performance at remote locations. To help address this issue Aras provides two standard solutions for global deployments.

### 6.1 IIS Compression of Dynamic Content

The easiest way to improve performance when using Aras Innovator is to enable IIS compression on the web server. Aras provides a utility that enables the compression of dynamic content requested from the server (.asp and .aspx pages), but does not compress dll's, exe's, or static content. The difference is most notable to users connecting through slower connections to Aras Innovator, but all users benefit from this adjustment to the server.

The compression.bat utility is available for download from the Aras Project site or from the Utilities folder of the Aras Innovator CD Image for community members with an active subscription package. Community members without an active subscription should contact Aras at [info@aras.com](mailto:info@aras.com) to discuss options.

## 7 Appendix (Client Settings)

### 7.1 Defining user's Locale in Browser

#### 7.1.1 Internet Explorer 9

1. Open Internet Explorer browser and go to **Tools --> Internet Options**.
2. In the **General** tab, click **Languages**.

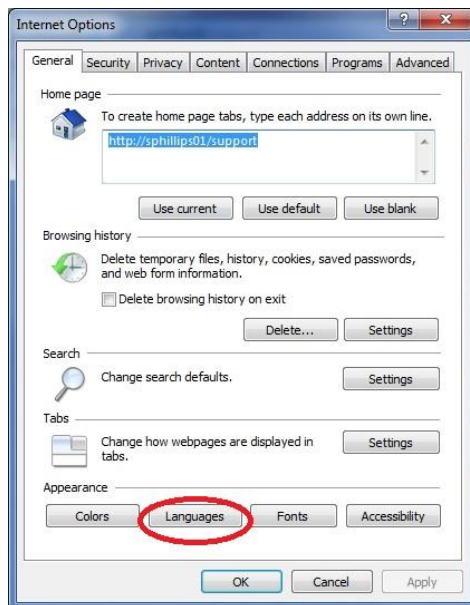


Figure 8.

3. In the **Language Preference** dialog box, click **Add**.



Figure 9.

4. Close the browser.
5. Login to Aras Innovator as administrator/innovator and confirm that Japanese Language is applied.

## 7.1.2 Internet Explorer 10 and Internet Explorer 11

1. Open Internet Explorer browser and go to **Tools --> Internet Options**.
2. In the **General** tab, click **Languages**.

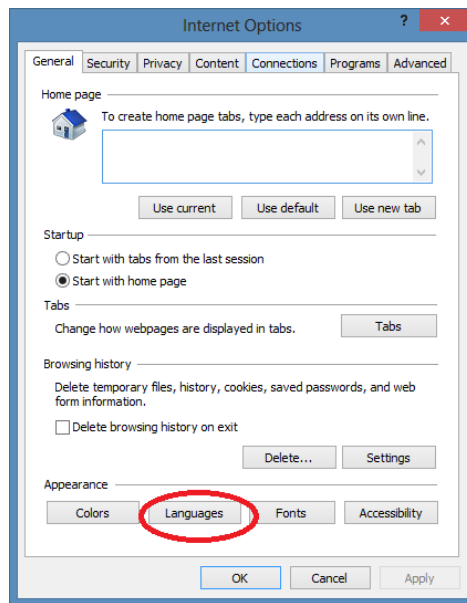


Figure 10.

3. In the **Language Preference** dialog box, click **Set Language Preferences**.

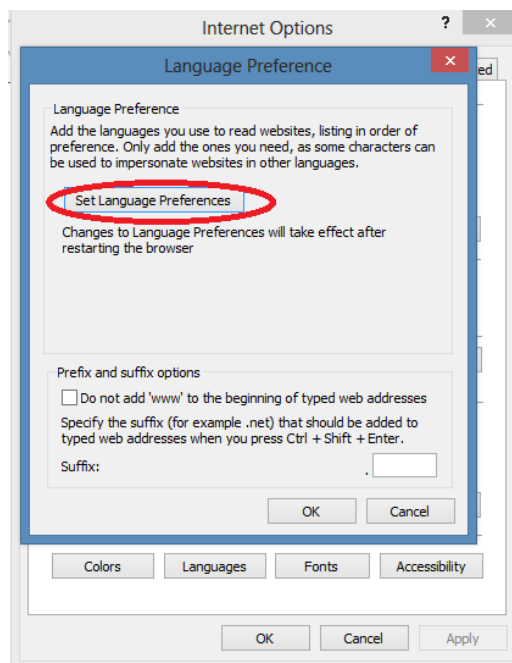


Figure 11.

4. In the **Language** dialog box, click **Add a Language**.
5. Select a language such as Japanese [ja-JP].
6. Click **Add** (bottom right corner).

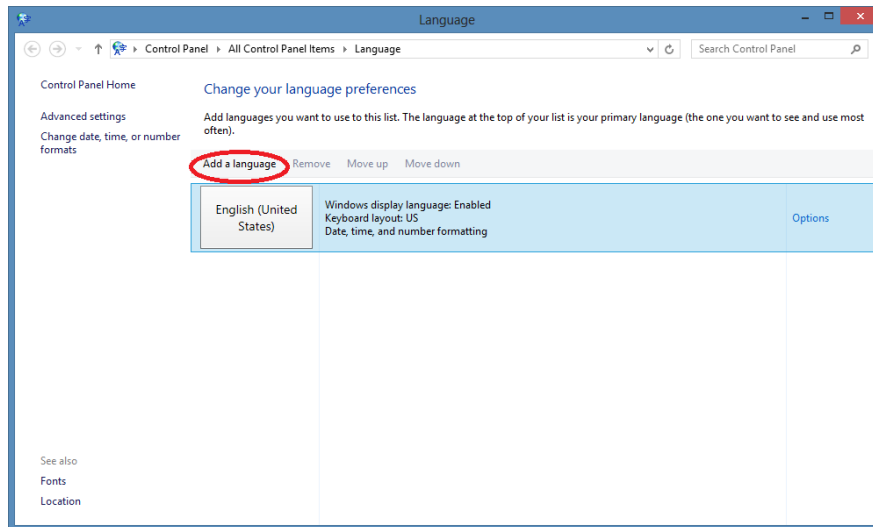


Figure 12.

7. Select **Japanese Language Box** and click **Move up** (move it to the top).

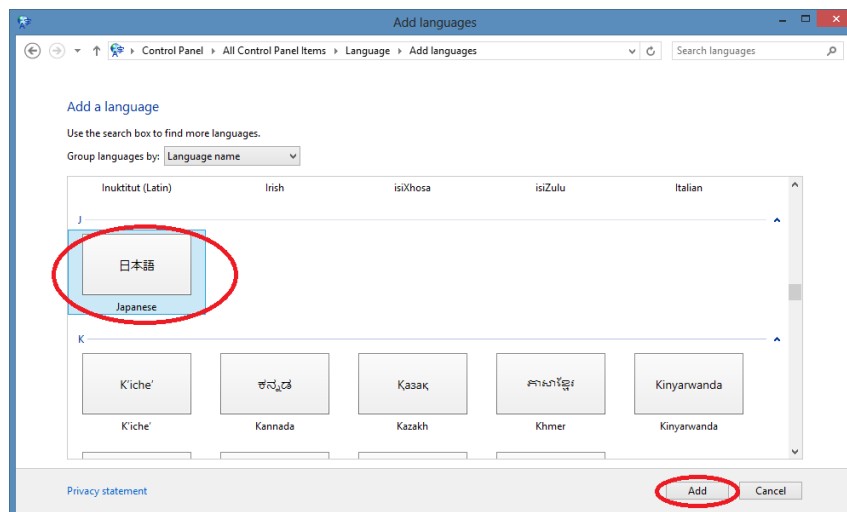


Figure 13.

8. Close the browser.
9. Login to Aras Innovator as administrator/innovator and confirm that Japanese Language is applied.

### 7.1.3 Firefox 31 ESR for Windows

1. Open Firefox browser and go to **Tools --> Options**.
2. Select the **Content** tab.
3. In the **Languages** Box, click **Choose**.

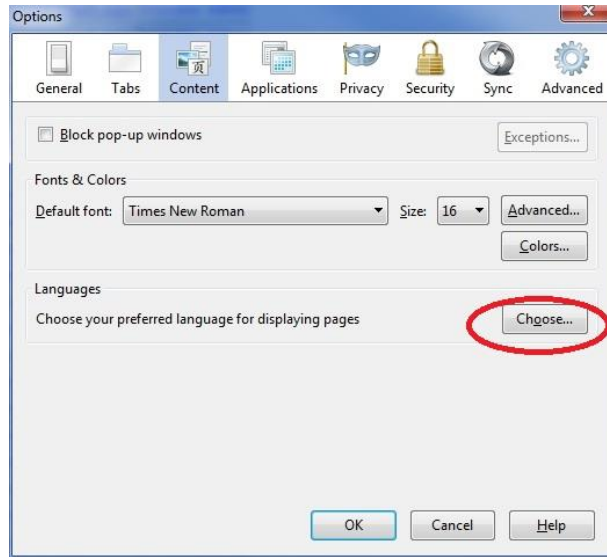


Figure 14.

4. In the **Languages** dialog box, click **Select a language to add...**
5. Select Japanese [ja] and click **Add**.
6. Move it to the top.

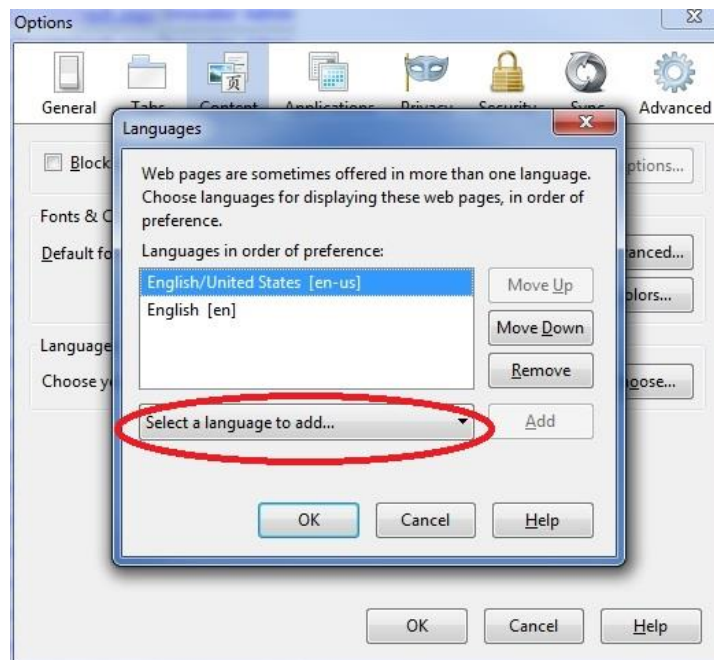


Figure 15.

7. Close the browser.
8. Login to Aras Innovator as administrator/innovator and confirm that Japanese Language is applied.



### 7.1.4 Firefox 31 ESR for Mac

1. Open Firefox browser and go to **Firefox --> Preferences.**

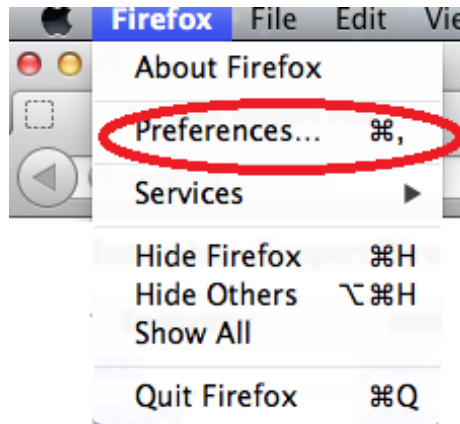


Figure 16.

2. Select the **Content** tab.
3. For **Languages**, click **Choose.**

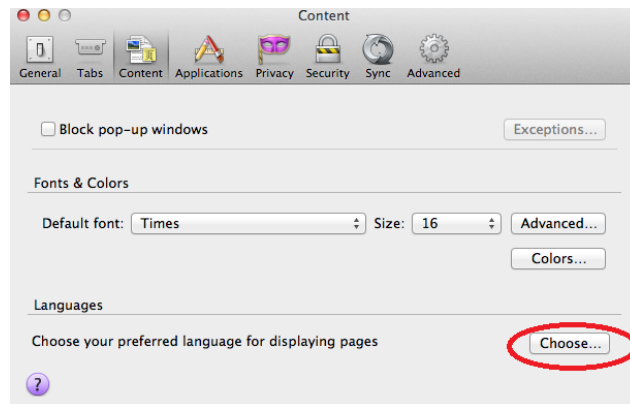


Figure 17.

4. Click **Select a language to add....**



Figure 18.

5. Select a language. For example, Japanese [ja-JP].

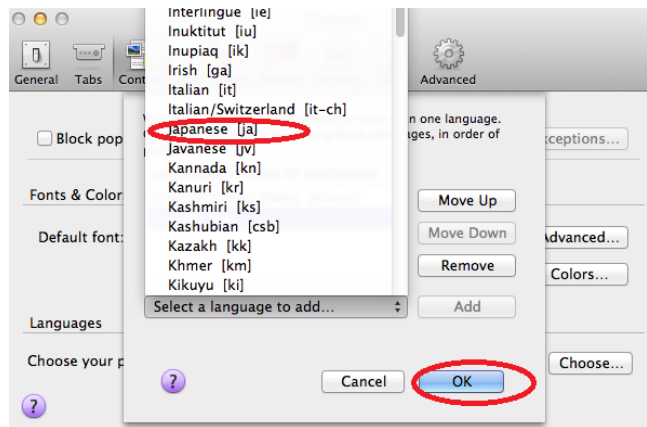


Figure 19.

6. Click **Add** and move it to the top.
7. Close the browser.
8. Login to Aras Innovator as admin/innovator and confirm that Japanese Language is applied.