Aras Innovator
Configuring Localization and Internationalization Support

Aras Innovator 9.1

Document #: 9.1.02232009

Last Modified: *3/26/2009*

ARAS CORPORATION

Copyright © 2009 Aras Corporation. All rights reserved

Aras Corporation
300 Brickstone Square
Suite 904
Andover, MA 01810

Phone: 978-691-8900
Fax: 978-794-9826

E-mail: Support@aras.com
Website: http://www.aras.com

Notice of Rights

Copyright © 2009 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

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.

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

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

You can send comments to us in the following ways:

- o **Email:**
  Support@aras.com
  Subject: Aras Innovator Documentation

Or,

- o **Postal service:**
  Aras Corporation
  300 Brickstone Square
  Suite 904
  Andover, MA 01810
  Attention: Aras Innovator Documentation

Or,

- o **FAX:**
  978-794-9826
  Attn: Aras Innovator Documentation

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

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

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 will outline 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 administers in deployment, each of the following topics will be 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 will not work for production environments.

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 will be used for calculating things like 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 user have begun to use the system, you will have to get all user to log out of the system before enabling the Corporate Time Zone.

To set the Corporate Time Zone please 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"
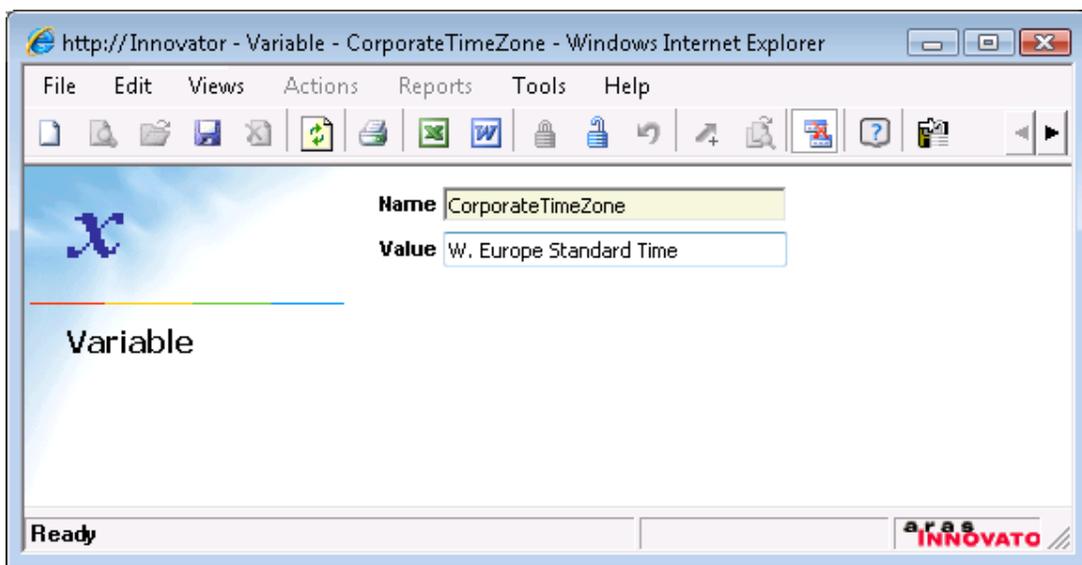
4) Set the Name = CorporateTimeZone

5) Set the Value according to the registry key name of the time zone desired.

   a. 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

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

The end result will be most obviously shown in the main Aras Innovator window.

For users whose local time does not match the corporate time the UI will display both the local time and the corporate time:
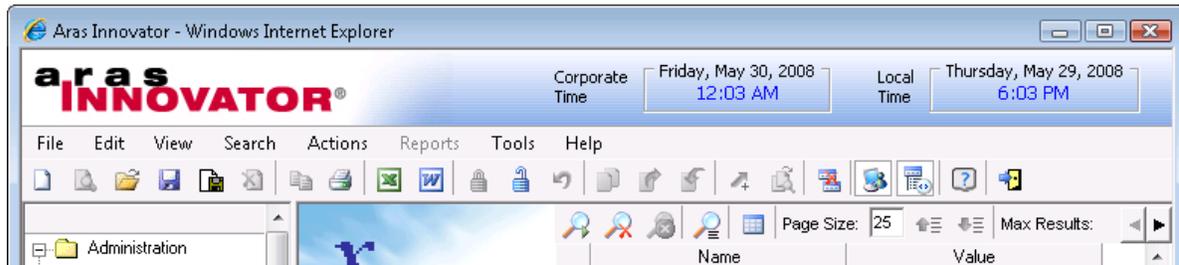
For users who are have no difference between local time and corporate time, the UI will display only one time:

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

2.3 Defining a Language

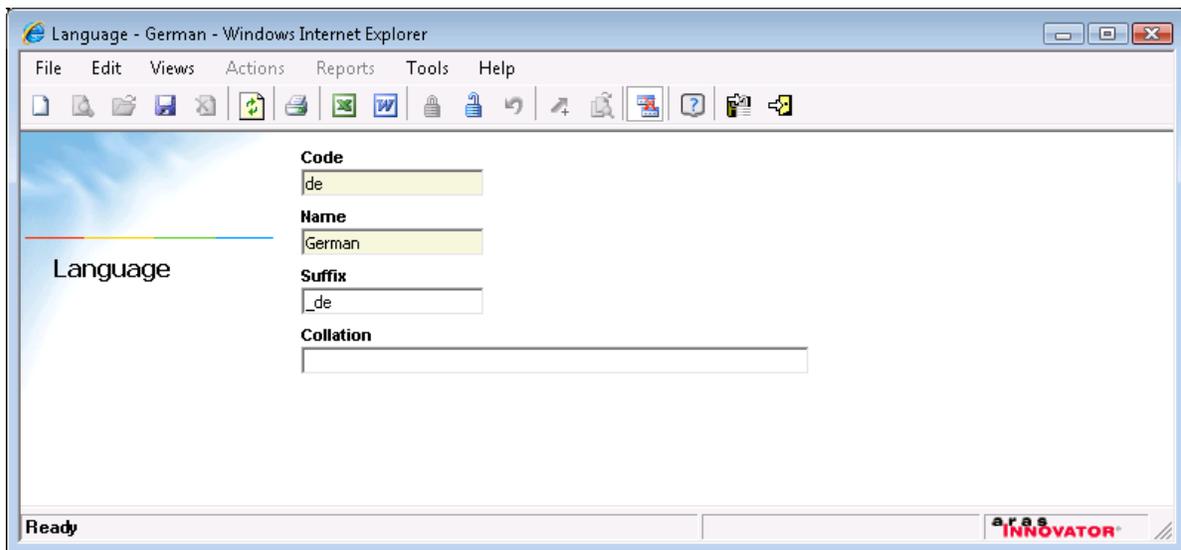Aras Innovator has the ability to store properties as "Multi-Lingual Properties". A multi-lingual property will store multiple values for the same property. The value displayed will be based on the language defined in Aras Innovator for your Locale. In most cases, if there is no value for a specific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

2.3.1 Adding the Language Item

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

1) Log into Aras Innovator as an Administrator

2) From the TOC, select Administration\Languages

3) From the main toolbar, select "Create a New Item"

4) Code = The letter ISO code that identifies the language.

   a. "ru" for Russian

5) Name = The name of the language for reference in the keyed name.

   a. Russian

6) Suffix = The identifier for differentiation of one language from another in a multi-lingual property.

   a. "_ru" for Russian

7) 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.

   a. "Cyrillic_General_CI_AI" for Russian

8) Save, Unlock, and Close

2.3.2 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 will 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 will 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.ru	Innovator 　Client 　　admin 　　bin 　　cbin 　　customer 　　developer 　　images 　　javascript 　　scripts 　　Solutions 　　styles 　　WebHelp 　　xml 　　xml.de
Each of the standard Aras Innovator solutions (Program Management, Product Engineering, and Quality Planning) has it's 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 Russian would be "/Client/Solutions/QP/xml.ru" The three solutions installed with Aras Innovator have the following XML-based resource folders: \Innovator\Client\Solutions\PLM \Innovator\Client\Solutions\Prpject \Innovator\Client\Solutions\QP	Innovator 　Client 　　admin 　　bin 　　cbin 　　customer 　　developer 　　images 　　javascript 　　scripts 　　Solutions 　　　PLM 　　　　images 　　　　javascript 　　　　scripts 　　　　styles 　　　　xml 　　　　xml.de
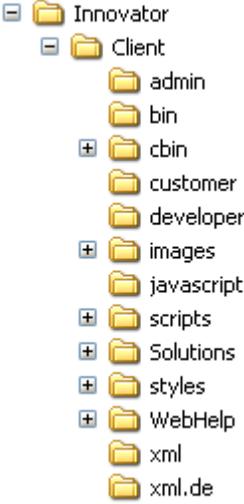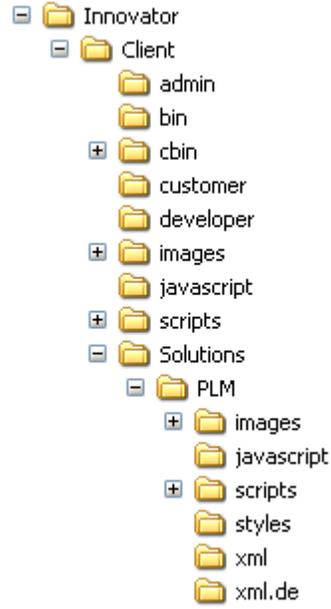
2.4   Defining the Supported Locales

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\Locales

3) From the main toolbar, select "Create a New Item"
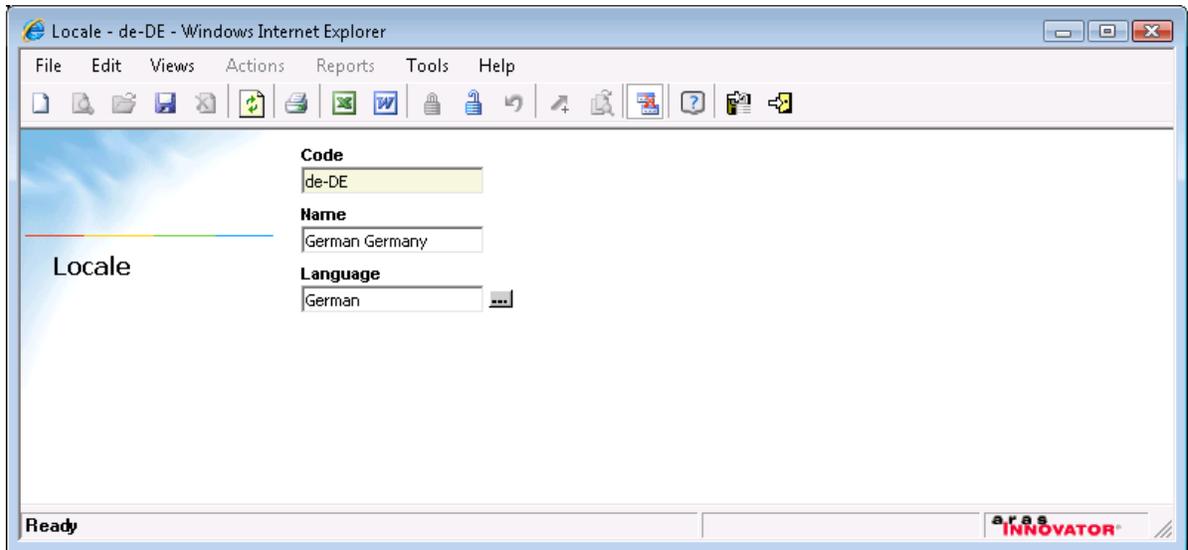
4) Code = The culture code for locale.

   a. "ru-RU" for Russia

5) Name = The name of the locale.

6) Language = The language item that will be referenced by users with this locale on their client PC machines.

   a. "Russian" from 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 will outline how the addition of a supported locale will be reflected to users.

3.1 Multi-Lingual Properties

Each Client session of Aras Innovator has a locale that is defined by the Windows "Region and Language Options" settings of the client PC machine. A multi-lingual property will display 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 will be saved to the language defined by the session. If the session language does not exist in the database, then the value will be 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.
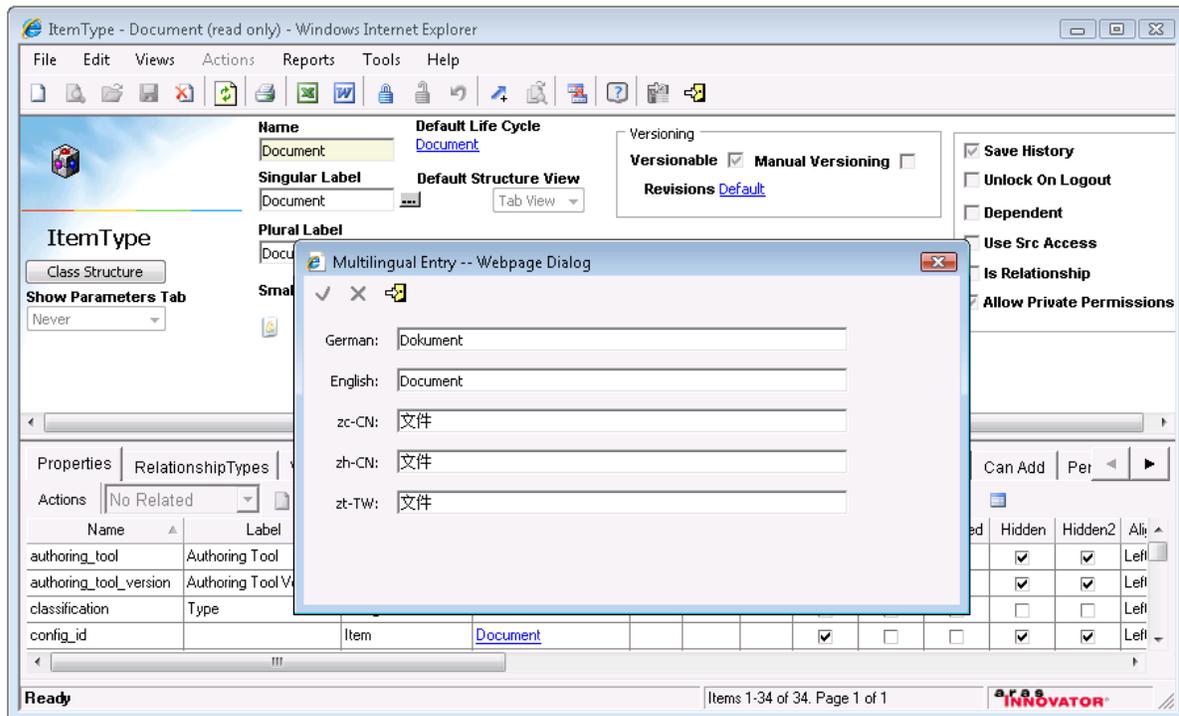
In the picture above, we see the multi-lingual dialog for the Singular Label. Note that the Singular Label has an ellipses button next to the field. Selecting this ellipses will open the multi-lingual dialog and allow the user to input languages for locales other than the current. The same dialog can be opened be putting the cursor into the field and selecting F2 from your keyboard.

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

Forms and Grids will display the property value will based on the locale of the current session:
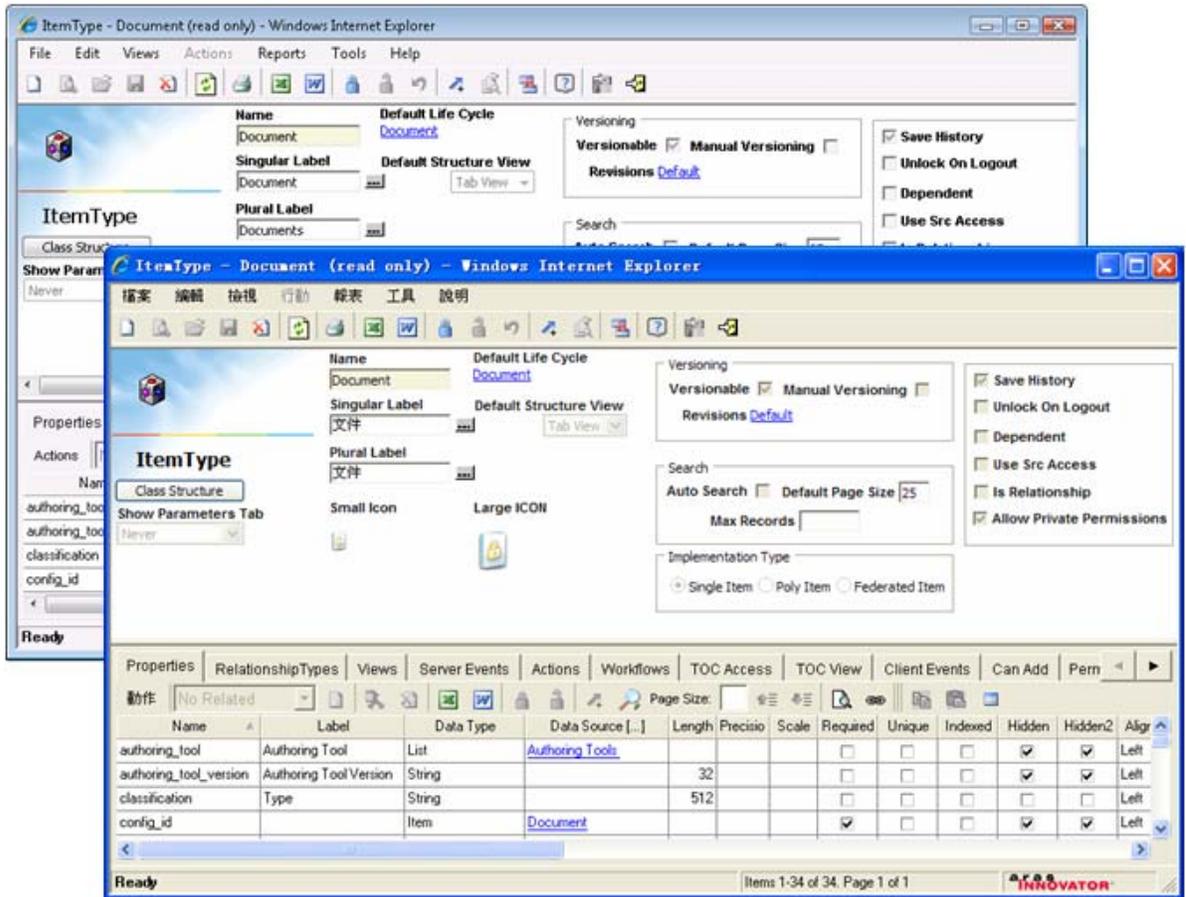
If the multi-lingual property has a value for the language defined by the current session local then the value is displayed; otherwise the value for the default language(English) is displayed. The same rule applies for the xml resource defined UI elements like the toolbar text.

3.2 Date Properties

Date properties will display based on Region and Language settings of the client. Date-Times are stored in the database as UTC (Universal Time Code).

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

1) If the Corporate Time Zone is set, then all DateTime values will be displayed and applied in Corporate Time Zone.

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

The Date properties will display based on the Region and Language Options of the client session based on one of four patterns. Short Date Long Date Short Date + Time Long Date + Time	**Regional and Language Options** Formats | Location | Keyboards and Languages | Administrative To change the way your computer displays numbers, currencies, dates, and time, select an entry from the format list. Current format: English (United States) Examples of how data is displayed using this format: Number: 123,456,789.00 Currency: $123,456,789.00 Time: 12:23:19 AM Short date: 6/7/2008 Long date: Saturday, June 07, 2008 Customize this format... For additional formats, keyboards, and tools, go to the Microsoft website. OK Cancel Apply

For user input the new date dialog will indicate weather the date-time input will be 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 will be 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 will also be stored in the database in a locale Neutral format. When displayed in the Aras Innovator UI, the locale settings for Number in the Region and Language Options will be used to determine the format to display these values.
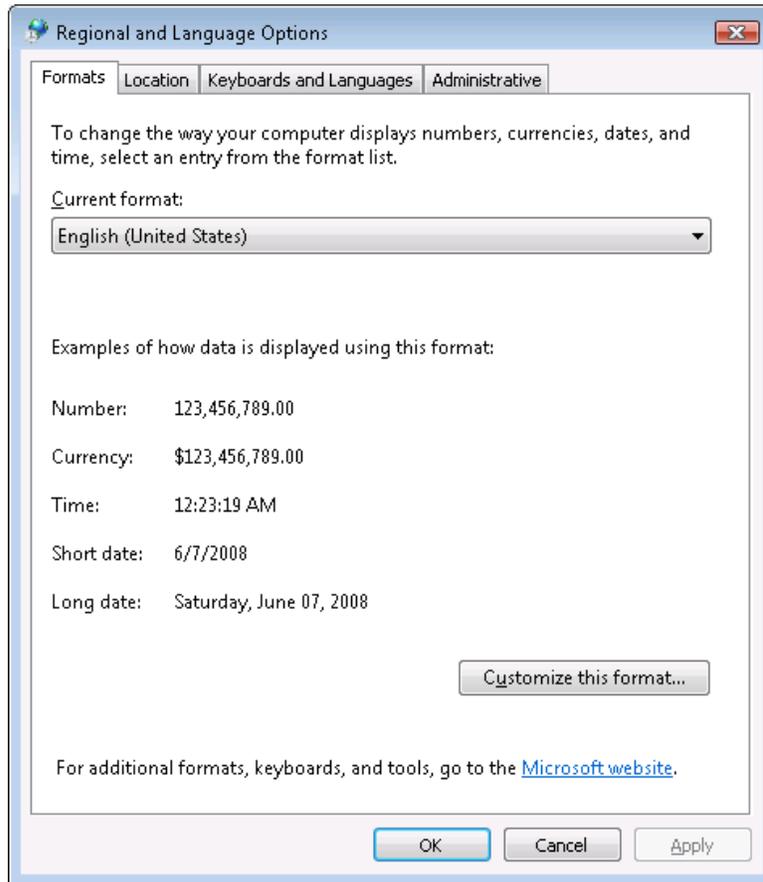
4 Populating a Language.

Once a language has been defined, there are two main ways to populate the language. First, you can manually enter all language data through the UI. Second, you can import a language using the Aras Innovator Language Pack Management Utility.

4.1 Manually Managing Language Packs in the Database

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

Once you have defined a language, see section Defining a Language, you will be able to add values to this language for each of the UI elements.

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 an 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
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 seach 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 worflow	Adminstration\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 will need to edit each Item of that type that will need 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 will explain 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 will be 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 that the optional parameters can be generally included in the LanguageTool.exe.config configuration file, but the command line parameters will 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
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="config_file" value="C:\German\ImportConfig.xml" />
    <add key="folder" value="C:\German\data" />
    <add key="log_file" value="C:\German\log.txt" />
    <add key="log_level" value="high" />
    <add key="server" value="localhost" />
    <add key="db" value="InnovatorSolutions" />
    <add key="login" value="innovator" />
    <add key="pwd" value="innovator" />
  </appSettings>
</configuration>
```

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

```xml
<add key="config_file" value="C:\German\ImportConfig.xml" />
```

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

```xml
<add key="folder" value="C:\German\data" />
```

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

```xml
<add key="log_file" value="C:\German\log.txt" />
```

This is the path to the log file.

```xml
<add key="log_level" value="high" />
```

This is the logging level, "high" recommended.

```xml
<add key="server" value="localhost" />
```

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")

```xml
<add key="db" value="InnovatorSolutions" />
```

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")

```
<add key="login" value="innovator" />
```

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")

```
<add key="pwd" value="innovator" />
```

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 will need to define what languages and ItemTypes to export. The definition of these settings are found in the ExportConfig.xml.

```
<export>
  <languages list="en, de" />
  <itemtype name="Field" />
  <itemtype name="Grid Column" />
  <itemtype name="ItemType" />
  <itemtype name="PM_ProjectGridLayout" />
  <itemtype name="Property" />
  <itemtype name="RelationshipType" />
</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 will have all of its multi-lingual string properties exported from the database.

4.2.4 Import Configuration

In order to import a language pack into the Aras Innovator database, you will first need to define the Language Item in the database, see Defining a Language. You will next need to setup the ImportConfig.xml.

```
<import>
  <itemtype name="Field" />
  <itemtype name="Grid Column" />
  <itemtype name="ItemType" />
  <itemtype name="PM_ProjectGridLayout" />
  <itemtype name="Property" />
  <itemtype name="RelationshipType" />
</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 Updating the User Interface (Xml-based resources)

5 Programming Notes

For details on any subjects in this section, please 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 will 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 will return 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) will be 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 will be 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>
```

Will have the following 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 will have 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 will 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.

1) If the CorporateTimeZone variable is set in the database, see [Time Zones](), then all DateTime properties will be returned in the corporate time zone.

2) If the CorporateTimeZone is NOT set, then all DateTime properties will be 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.

1) If the CorporateTimeZone variable is set in the database, see [Time Zones](), then all DateTime properties must be applied in the corporate time zone.

2) 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.

3) The DateTime applied in the AML must conform to the pattern yyyy-MM-dd[Thh:mm:ss] (NOTES: 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 dattime 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 like 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 will be 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" will return 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"/>
```

Will return

```
<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 will return

```
<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:

1) The format of property values that are local specific ('date', 'float', 'decimal') are always presented in an AML in the neutral format.

2) Property values of type 'date' are always presented in AML in the time zone of the current session

3) 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:*

1) *'value' is a string that has to be converted (e.g. '12/25/2003')*

2) *'type' is name of an Innovator type which representation is locale dependent; this parameter can be one of the following 'date' | 'float' | 'decimal'*

3) *'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:*

1) *'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)*

2) *'type' is name of an Innovator type which representation is locale dependent; this parameter can be one of the following 'date' | 'float' | 'decimal'*

3) *'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 that 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' will result 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:*

1) *'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.*

2) *'inFormat' defines the format of the 'utcDateTime' string. Note that 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.*

1) *'neutralDateTime' is a string that represents a date-time value in the session time zone in neutral format*

2) *'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-along 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 convertion –
```

// "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 he\she 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 just 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 will be processed by the *update* action of server and the __*now()* will be 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.*) – see <u>DateTime Properties</u> 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 will allow 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. (See Corporate Time Zone.) If the Time Zone parameter is set to DEFAULT and the Corporate Time Zone is set, the value in the CorporateTimeZone variable will be used. If the Time Zone parameter is set to DEFAULT and the Corporate Time Zone is not set, the value in the time zone of the database server will be used.

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. (See Corporate Time Zone.) If the Time Zone parameter is set to DEFAULT and the Corporate Time Zone is set, the value in the CorporateTimeZone variable will be used. If the Time Zone parameter is set to DEFAULT and the Corporate Time Zone is not set, the value in the time zone of the database server will be used.

Example:

```
update innovator.Documnet
set effective_date = '1/1/2007 00:00:00.00'
```

Would be written as

```
update innovator.Documnet
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 will allow 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)
```

To install these functions, the Microsoft SQL Server DateTime Add-On conversion 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 to discuss options.

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 will enable 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 will be most notable to users connecting through slower connections to Aras Innovator, but all users will 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 to discuss options.

6.2   Remote /Client Server Installations

When connecting to Aras Innovator over slower connection, the performance can be improved by using a remote /Client server. Connecting through a remote /Client server will allow users to connect to a local server for all the user interface elements, like images (.gif) or controls (.dll), and the main Aras Innovator server for all data elements, like data used to populate the grids.

For more details on setting up a remote /Client server, please see the *Aras Innovator – Installation Guide*.