Локализация

Локализация применяется к следующим элементам:

* к определенным полям конфигураций (например, названия ссылок в навигационной панели, заголовки колонок коллекции, текст виджета Label и др.);
* к текстовым данным, не относящимся к конфигурациям (например, сообщения об ошибках);
* а также к названиям доменных объектов и их полей, областей поиска и их полей.

**Формат текстовых ресурсов**

Текстовые ресурсы для локализации хранятся в .properties файлах.  
*Ключ1=значение1  
Ключ 2=значение2  
  
Формат ключа*

При локализации *конфигураций*, ключи соответствуют атрибутам элементов в xml-файлах конфигураций. (См. Приложение А – Список локализуемых атрибутов).Выбор формата ключаостается за программистом-конфигуратором.   
Система локализации гарантирует поддержку хотя бы одного языка, т.е. если значение по заданному ключу не найдено, сам ключ будет использоваться в качестве значения. Такой подход позволяет использовать в качестве ключа целевой текст на одном из языков (на основном языке). Например, для локализации текста виджета, описанного в xml-конфигурации как

<list-box id="8a">  
 <field-path value="user"/><pattern value="Имя пользователя: {name})"/>  
</list-box>

в файле *.properties* для английского языка может быть указано:  
 **Имя\ пользователя\:\ {name}**= "{name}" (User name)

(В properties файлах пробелы и другие символы-разделители в ключах должны экранироваться обратным слешем).  
Также можно использовать ключи произвольного вида, что удобно если одному и тому же слову на основном языке могут соответствовать разные переводы или если целевое значение слишком длинное:  
 **CorrectErrorsMessage**=Поля на форме не заполнены или заполнены некорректно.\n Исправьте ошибки заполнения перед сохранением.  
  
Для текстовых элементов, *которые* *не вынесены в конфигурационные файлы*, используются ключи, предопределенные в коде (См. Приложение Б). Для сообщений об ошибках ключом служит имя класса исключения (exception.getClass().getSimpleName()).   
При этом в коде заданы значения по умолчанию, которые используются, если ключ отсутствует в .properties файле.

Для локализации таких текстовых элементов как *имена доменных объекты, областей поиска, их полей* используется более сложная система формирования ключей – с учетом контекста использования. Это позволит сопоставить одной и той же строке разные значения, например: название доменного объекта Country может быть переведено как «Страна», а название области поиска Country - как «Поиск стран».

Сложный ключ имеет следующую структуру:   
 [*классификатор*](контекст1, …контекст2)ключ

* *классификатор*  определяет тип контента – DOMAIN\_OBJECT, FIELD (поле доменного объекта), SEARCH\_AREA, SEARCH\_DOMAIN\_OBJECT, SEARCH\_FIELD
* список контекстов является необязательным элементом, служыт для дальнейшей детализации.
* ключ для доменного объекта – это его имя, ключ поля – имя\_доменного\_объекта.имя\_поля. Аналогично для областей поиска и их полей.

Пример:   
  
[DOMAIN\_OBJECT]country=Страна  
[FIELD]country.name=Название  
[FIELD]country.population=Население  
[FIELD]country.square=Площадь  
[FIELD]country.capital=Столица  
[SEARCH\_AREA]Countries-Search-Area=Поиск стран  
[SEARCH\_DOMAIN\_OBJECT]Employee=Сотрудник  
[SEARCH\_FIELD]Employee.FirstName=Имя  
[SEARCH\_FIELD]Employee.LastName=Фамилия  
[SEARCH\_FIELD]Employee.EMail=e-mail

При этом значение в правой части выражения само является ключом в локализационном файле.

**Загрузка текстовых ресурсов**

Для подключения .properties файлов к модулю проекта их нужно прописать в *cm-module.xml* в следующем формате:

<localization-files>  
 <localization-file locale="ru">path\_to\_file1</localization-file>  
 <localization-file locale="en-GB">path\_to\_file2</localization-file>   
</localization-files>

Локаль указывается в атрибуте locale в формате соответствующему страндарту BCP 47.

Загрузку properties -файлов на старте приложения выполняет класс *LocalizationLoader*. Он инициирует класс *MessageResourceProvider*, который выполняет поиск значения по заданному ключу (возможно с контекстом) и локали.   
Основные методы класса *MessageResourceProvider*:

**public static** Map<String, String> getMessages(String locale);

**public static** String getMessages(String key, String locale);

**public static** String getMessage(String key, String locale, String defaultValue)

**public static** String getMessage(MessageKey messageKey, String locale);  
**public static** Set<String> getAvailableLocales();

Класс *MessageKey* – это бин, содержащий информацию о сложном ключе:  
**public class** MessageKey {  
 **private** String **key**;  
 **private** String **classifier**;  
 **private** Map<String, ? **extends** Object> **context**;

...  
}

Предусмотрена подстановка значений в поля для заполнения (placeholders) – класс *PlaceholderResolver*.

**Определение текущей локали пользователя**

Текущая локаль определяется из профиля пользователя:  
  
PersonProfile profile = **profileService**.getPersonProfile();  
String locale = profile.getString(ProfileService.***LOCALE***);

Если локаль не указана в профиле, используется значение из новой настройки   
<global-settings>

<default-locale name="ru"/>

Для локализации окна логина используется только локаль по умолчанию.

**Локализация полей конфигураций**

Бóльшая часть текста, который нужно локализовать, - это определеннные поля в классах конфигураций. Такие поля помечаются специальной аннотацией *@Localizable*. На этапе создания кэшей конфигураций производится рекурсивное сканирование полей в классах конфигураций, и значения, помеченные аннотацией, заменяется на значения, определенные для каждой из локалей.

Сами классы конфигураций верхнего уровня, подлежащие локализации, должны реализовать интерфейс *LocalizableConfig*. Это позволит оптимзировать процесс обработки аннотаций, т.к. локализация не будет выполняться там, где она неприменима в принципе - в серверных конфигурациях (доменные объекты, матрицы доступа и т.п),   
В классе *ConfigurationStorage* вводится новые структуры данных – *localizedConfigsMap*, где ключом служит пара <класс конфигурации, локаль>, а значением - Map<имя конфигурации, конфигурация>. *ConfigurationStorageBuilder* заполняет ее для всех доступных локалей (т.е. для которых есть текстовые ресурсы) и всех конфигураций верхнего уровня, реализующих интерфейс *LocalizableConfig*. Для таких конфигураций выполняется рекурсивный перебор всех полей. Если строковое поле помечено аннотацией *@*Localizable, его значение передается в метод класса *MessageResourceProvider* как ключ для поиска. Если значение найдено, оно заместит текущее значение в копии конфигурации для соответствующей локали.

В интерфейсах *ConfigurationExplorer* и *ConfigurationService* добавляются методы для получения конфигураций с учетом локали:

**public** <T> T getLocalizedConfig(Class<T> type, String name, String locale);  
**public** <T> Collection<T> getLocalizedConfigs(Class<T> type, String locale);

Эти методы используются в классах, которые работают с локализованными конфигурациями (*FormResolver, NavigationTreeResolver*, *ActionConfigBuilder,* PluginHandlerHelper, ExtendedSearchPluginHandler и др.) Структура кэшей в классах *FormResolver* и *NavigationTreeResolver* изменена – вместо самих конфигураций там теперь хранятся только их класс и имя. Для получения конфигурации эти данные вместе с текущей локалью передаваются в соответствующий метод *ConfigurationExplorer* .

**Локализация других элементов текста**

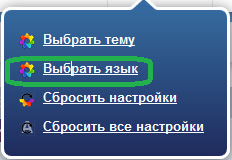
Кроме полей конфигурации, в различных частях системы существуют и другие текстовые данные, подлежащие локализации (например, сообщения об ошибках и др. системные сообщения, а также некоторые элементы визуального интерфейса). Везде, где это окажется целесообразным, такие данные будут вынесены в конфигурацию. Локализация системных сообщений и других данных, которые не относятся к конфигурациям, выполняется непосредственно в точках применения. Локализация имен доменных объектов, областей поиска, их полей и т.п. технически выполняется так же, с отличием только в формировании ключей для поиска.

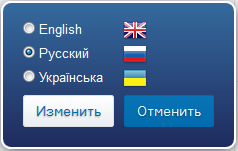
На стороне сервера - с помощью вызова методов класса *MessageResourceProvider*.

На стороне клиента – при инициализации *BusinessUniverse*, текстовые ресурсы для текущей локали сохраняются в классе *Application*, откуда они могут быть доступны для всех клиентских классов.

**Изменение локали из UI**

Добавлена возможность изменять локаль из Настроек приложения. Значение, выбранное пользователем сохраняется в пользовательском профиле.





Список доступных локалей задается с помощью новой настройки languages:

<business-universe>

...

<settings-popup>

...

<languages>  
 <language name = "en" display-name="English" img="images/flags/United Kingdom(Great Britain).png"/>  
 <language name = "ru" display-name="Русский" img="images/flags/Russian Federation.png"/>  
 <language name = "ua" display-name="Українська" img="images/flags/Ukraine.png"/>  
</languages>

**Приложение А – Список локализуемых атрибутов xml-конфигураций**

|  |  |  |  |
| --- | --- | --- | --- |
| **Xml элемент** | **Атрибут** | **Java-класс** | **поле** |
| <action> | text | ActionConfig | text |
| tooltip | tooltip |
| <child-collection-viewer> | bread-crumb | ChildCollectionViewerConfig | breadCrumb |
| <column> | name | CollectionColumnConfig | name |
| <tab> | name | TabConfig | name |
| <tab-group> | name | TabGroupConfig | name |
|  | value | AllValuesEmptyMessageConfig | value |
| <boolean-format> | true-text | BooleanFormatConfig | trueText |
| false-text | falseText |
| null-text | nullText |
| <display> | tooltip | ColumnDisplayConfig | tooltip |
| <text> | value | ColumnDisplayTextConfig | value |
| <form-table-column>, <summary-table-column> | header | ColumnParentConfig | header |
| <map> | display-text | EnumMapConfig | displayText |
| <label> | text | LabelConfig | text |
| <node-collection-def> | title | NodeCollectionDefConfig | title |
| <pattern> | value | PatternConfig | value |
| <root-node-link> | title | RootNodeLinkConfig | title |
| <action> | tooltip | SummaryTableActionColumnConfig | tooltip |
| <add-button>, <clear-all-button> | text | ButtonConfig | text |
| <created-object> | text | CreatedObjectConfig | text |
| <child-links> | group-name | ChildLinksConfig | groupName |
| <link> | display-text | LinkConfig | displayText |
| <language> | display-name | LanguageConfig | displayName |
| <theme> | display-name | ThemeConfig | displayName |

**Приложение Б - ключи, предопределенные в коде**

|  |  |
| --- | --- |
| **Ключ** | **Значение по умолчанию** |
| ForToday | "За сегодня" |
| ForLastWeek | "За неделю" |
| ForLastYear | "За год" |
| ForYesterday | "За вчера" |
| ChoosePeriod | "Выбрать период" |
| TaskComplete | "Задача успешно завершена" |
| RowDeleted | "Строка удалена" |
| ReportUploaded | "Отчет успешно загружен" |
| CorrectValidationErrors | "Поля на форме не заполнены или заполнены некорректно.\nИсправьте ошибки заполнения перед сохранением." |
| Saved | "Успешно сохранено" |
| MessageSent | "Событие отправлно" |
| SystemError | "Ошибка системы" |
| BusinessUniverseContextError | "BusinessUniverseContext не получил конфигурацию для поиска" |
| DoneSuccessfully | "Успешно выполнено" |
| ProcessStarted | "Процесс запущен" |
| CannotSaveUserSettings | "Невозможно сохранить настройки пользователя" |
| LogoutError | "Ошибка выхода из приложения" |
| ExtendedSearchError | "Ошибка расширенного поиска " |
| WrongDateFormat | "Неверный формат времени! Попробуйте " |
| DataNotSavedConfirm | "Изменения данных не сохранены.\nПродолжить выполнение команды ?" |
| ChooseTheme | "Выбрать тему" |
| ResetSettings | "Сбросить настройки" |
| ResetAllSettings | "Сбросить все настройки" |
| Exit | "Выход" |
| CoreVersion | "Версия платформы:" |
| Version | "Версия:" |
| Info | "Справка" |
| Login | "Логин:" |
| FirstName | "Имя:" |
| LastName | "Фамилия:" |
| EMail | "EMail:" |
| Administrator | "Администратор" |
| ChooseLanguage | "Выбрать язык" |
| OkButton | "OK" |
| CancelButton | "Отменить" |
| CancellationButton | "Отмена" |
| CloseButton | "Закрыть" |
| OpenInFullWindowButton | "Открыть в полном окне" |
| DoneButton | "Готово" |
| SaveButton | "Сохранить" |
| ChangeButton | "Изменить" |
| FindButton | "Найти" |
| ContinueButton | "Продолжить" |
| Search | "Поиск" |
| NoResults | "Результаты отсутствуют" |
| ClearForm | "Очистить форму" |
| SizeActionTooltip | "Распахнуть/Свернуть" |
| FavoriteActionTooltip | "Показать/Скрыть избранное" |
| FilterTooltip | "Отобразить/Скрыть фильтры" |
| ColumnsDisplayTooltip | "Отображение видимых колонок" |
| ColumnsWidthTooltip | "Отобразить все колонки на экране" |
| NoNewNotifications | "Новых уведомлений нет" |
| AddConfigFiles | "Добавьте файлы конфигурации:" |
| AddReportFiles | "Добавьте файлы шаблона отчета:" |
| MonthSwitchButton | "Задачи дня" |
| WeekSwitchButton | "Суббота, Воскресенье" |
| ActionExecutionError | "Ошибка выполнения действия: " |
| DateFormatError | "Ошибка в формате даты: " |
| ServerValidationException | "Server-side validation failed" |
| SystemException | "Системная ошибка при выполнении ${commandName}, обратитесь к администратору." |
| GuiException.CommandNotFound | "Команда ${commandName} не найдена" |
| GuiException.CommandCallError | "Ошибка вызова команды: " |
| GuiException.CommandExecutionError | "Команда не может быть выполнена: " |
| GuiException.MultipleFieldPaths | "Multiply fieldPaths should be all reference type or all backreference type" |
| GuiException.SingleFieldPath | "Only single field-path is supported" |
| GuiException.ObjectNotSaved | "Объект ещё не сохранён" |
| GuiException.NoProcessType | "Не задано тип процесса" |
| GuiException.NoProcessName | "Не задано имя процесса" |
| GuiException.ProcessNotSupported | "Process '${processType}' not supported." |
| GuiException.RefPathNotSupported | "Reference ${path} not supported" |
| GuiException.UnknownUrl | "Неизвестный url" |
| GuiException.ManyDefaultForms | "There's more than 1 default form for type: " |
| GuiException.ObjectNotExist | "Object with id: ${objectId} doesn't exist" |
| GuiException.ReportFormError | "Конфигурация формы отчета не найдена или некорректна! Форма: '${formName}', отчет: '${reportName}'" |
| GuiException.ReportNameNotFound | "Имя отчета не сконфигурировано ни в плагине, ни форме!" |
| GuiException.WidgetIdNotFound | "Widget, id: ${widgetId} is not configured with Field Path" |
| GuiException.SearchError | "Ошибка при поиске:\\n" |
| GuiException.CollectionViewError | "Collection view config has no display tags configured" |
| GuiException.HierarchCollection | "Ошибка в конфигурации иерархической коллекции" |
| GuiException.SortingFieldNotFound | "Couldn't find sorting '${field}'" |
| GuiException.VersionError | "Ошибка получения версии: " |
| UserName | "Имя пользователя" |
| Password | "Пароль" |
| SignOn | "Войти" |
| RememberMe | "Запомнить меня" |
| AuthorizationError | "Ошибка авторизации. " |
| WrongPswError | "Ошибка авторизации. Проверте правильность введенных данных." |
| NoConnectionError | "Ошибка авторизации. Невозможно подключиться к серверу." |
| validate.not-empty | "Поле ${field-name} не должно быть пустым!" |
| validate.max | "Поле ${field-name} не может быть больше чем ${value}!" |
| validate.integer | "${value}' должно быть целым! " |
| validate.decimal | "'${value}' должно быть десятичным! " |
| validate.positive-int | "'${value}' должно быть целым положительным!" |
| validate.positive-dec | "'${value}' должно быть десятичным положительным! " |
| validate.unique-field | "Поле ${field-name} со значением '${value}' уже существует! " |
| validate.pattern | "Поле ${field-name} должно соответствовать шаблону ${pattern}!" |
| validate.length.not-equal | "Длина поля ${field-name} должна быть равна ${length}" |
| validate.length.too-small | "Длина поля ${field-name} не может быть меньше чем${min-length}" |
| validate.length.too-big | "Длина поля ${field-name} не может быть больше чем ${max-length}" |
| validate.range.too-small | "Значение поля ${field-name} не может быть меньше чем ${range-start}" |
| validate.precision | "Значение поля ${field-name} должно иметь точность ${precision}" |
| validate.scale | "Значение поля ${field-name} должно иметь ${scale} знаков после запятой" |