Программный продукт Xafari Framework

Состав и назначение

Аннотация

В настоящем руководстве приводится состав и краткое описание основных функций программного продукта Xafari Framework.

Издание 05.2019

Оглавление

Обц	цие	сведения	5
1.	Kov	поненты общего назначения	6
1.1.	Do	ck Panels (плавающие панели)	6
1.2.		ernative Navigation Items (АРМы)	
1.3.		ra Property Editors (редакторы свойств)	
1.3.		TabbedDetailPropertyEditor	
1.3		EnumPropertyEditor	
1.3		ExpressionPropertyEditor	
1.3		VerticalGridPropertyEditor	
1.3	.5.	ViewIdPropertyEditor	
1.3	.6.	GoogleMapsPropertyEditor	
1.3	.7.	ASPxGroupPropertyEditor	11
1.3	.8.	QuickChoicePropertyEditor	11
1.3	.9.	XafariMultipleLookupEditor	
1.3	.10.	PeriodStructPropertyEditor	
	.11.	XafariDatePropertyEditor	
	.12.	XafariLookupEditor	
	.13.	TypelmagePropertyEditor	
	.14.	ObjectSetPropertyEditor	
	.15.	HyperLinkPropertyEditor	
	.16. .17.	Folder Browser Property Editor	
	.17. .18.	ProgressBarPropertyEditor	
	.10. .19.	HTMLPropertyEditor	
		ra List Editors (редакторы списков)	
1.4.		XafariGridListEditor	
1.4		XafariTreeListEditor	
1.4		ExplorerLisEditor	
1.4		HierarchyNodeListEditor	
1.4		CardListEditor	
1.5.	Ter	mplated Web Editors (редактируемые шаблоны для Web)	
1.6.	Ext	ra Filters (фильтр по периоду)	22
1.7.		art Design (стратегия генерации форм)	
1.8.		zions (действия)	
		zards (мастера)	
		· · · · ·	
		полнительные функции модуля Xafari	
	0.1.		
	0.2.	Auto Refresh (автоматическое обновление данных)	
	0.3. 0.4.	Action by Timer (действие по таймеру) Action Executor (контекстное действие)	
	0.4. 0.5.	Criteria Object Creation (критерий для новых объектов)	
	0.5.	Show Success Message (сообщение об успешном выполнении)	
	0.7.	Layout Group Expand Enable (свертывание/развертывание групп на форме)	
		нес-компоненты	
2.1.		earchyNode (иерархические данные)	
2.2.		ject Set (набор выбранных объектов)	
2.3.	Xaf	ari Security (система безопасности)	28
2.4.	eXt	ension Framework (расширение бизнес-объектов)	28
2.5.	Log	gic Controllers (контроллеры бизнес-логики)	29

2.6.	Object Status (статусы объектов)	29
2.7.	Bulk Edit (групповое редактирование)	30
2.8.	Numerators (нумераторы)	31
2.9.	Categories (категории)	31
2.10.	Managed Operations (управляемые операции)	32
2.11.	Dynamic Properties (динамические реквизиты)	32
2.12.	Application Settings (настройки приложения)	32
2.13.	Data Versioning (версионность данных)	33
2.14.	Work Places (рабочие места)	34
2.15.	Data Management (импорт/экспорт данных)	35
2.16.	Xafari Service (сервисы Xafari)	35
2.17.	Business Operations (бизнес-операции)	36
2.18.	ХАЅ (конфигуратор)	37
2.19.	Console Application (консоль приложения)	38
3. ŀ	Компоненты ERP	39
3.1.	AppModule (модуль приложения)	39
3.2.	Audit (аудит)	
3.3.	Docflow (документооборот)	
3.4.	Branches (филиалы)	40
3.5.	Message Queue (очередь сообщений)	40
3.6.	Xafari Server (сервер Xafari)	
3.7.	Xafari Reports (отчеты Xafari)	41

Общие сведения

Программный продукт Xafari Framework (далее — Xafari Framework) представляет собой набор библиотек, которые содержат готовые компоненты и инструментарий для быстрой разработки различных функций для ERP-систем, реализующих архитектуру DevExpress XAF.

Ключевыми особенностями продукта являются:

- поддержка ERP-приложений на различных платформах: WinForms, WebForms, ASP.NET MVC, консольное приложение, сервер расчетов;
- поддержка различных провайдеров баз данных: Postgres Pro, Oracle, Microsoft SQL Server и проч.

Функционал разделен на 3 группы:

- 1) Компоненты общего назначения.
- 2) Бизнес-компоненты.
- 3) Компоненты ERP.

Общие сведения 5

1. Компоненты общего назначения

1.1. Dock Panels (плавающие панели)

Dock Panels — это технология для разработки сложных интерфейсов, параллельно отображающих множество вспомогательных списковых и детальных форм. Наличие дополнительных форм позволяет предоставить пользователю больше данных о текущем объекте.

Дополнительные формы отображаются внутри плавающих контейнеров, которые можно самостоятельно скрывать, прикреплять к различным областям рабочего пространства, откреплять, перетаскивать и т. п.

Данные внутри панели могут быть синхронизированы с объектами на основной форме. Для этого предлагаются несколько типов панелей, поддерживающих отношение Master-Detail. Кроме того, есть возможность вычислять данные внутри панели с помощью синтаксиса Expressions.

Разработка дополнительных панелей осуществляется в Модели приложения. Важно отметить, что разработка дополнительных форм не затрагивает настройки основной формы, дополнительный функционал может быть разработан в другом модуле и поставляться отдельно.

В составе фреймворка Dock Panels использованы в ряде решений, например: <u>категории</u>, <u>документооборот</u>.

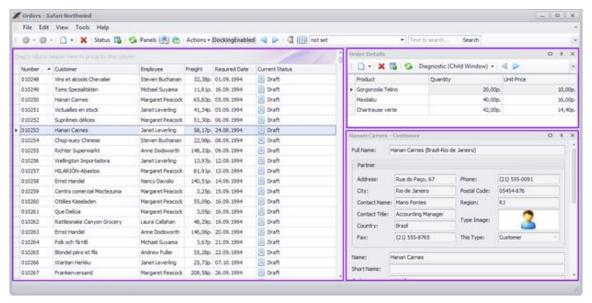


Рис. 1.

Все типы Dock Panels реализованы для платформ WinForm и ASP.NET.

1.2. Alternative Navigation Items (АРМы)

Модуль Xafari.Arms реализует панель навигации, легко адаптируемую под нужды конкретного приложения. Alternative Navigation Items позволяет разрабатывать главное меню для множества различных пользовательских рабочих мест. Элементы этого меню обеспечивают доступ к:

- списковым и детальным формам бизнес-объектов;
- выполнению Action;

- запуску бизнес-операций;
- формированию отчетов.

Указанный перечень элементов АРМ может быть самостоятельно расширен разработчиком. Доступ к тем или иным элементам навигации можно установить через пользовательские роли.

Разработка главного меню осуществляется в Модели приложения. Кроме конкретных пользовательских элементов узел Xafari|ArmDesign может содержать шаблоны — это группы элементов навигации, предназначенные для многократного использования при проектировании меню.

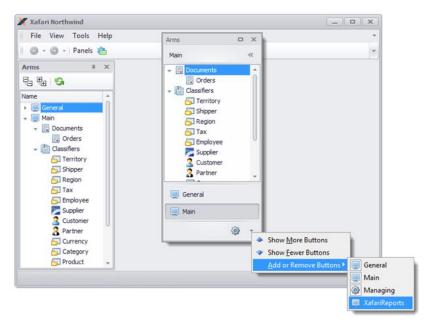


Рис. 2.

APMы реализованы для платформ WinForm, ASP.NET и MVC.

1.3. Extra Property Editors (редакторы свойств)

Редактор свойства обеспечивает привязку поля бизнес-объекта к определенному элементу управления, который обеспечивает отображение и редактирование данного поля на визуальной форме приложения. Редакторы свойств, поставляемые Xafari Framework, можно разделить на следующие группы:

- Редакторы для стандартных типов .NET. Они содержат функционал, обеспечивающий работу с данными по определенному специализированному сценарию. Например, <u>FolderBrowserPropertyEditor</u> отображает данные типа System.String, но применяется только для строк, которые описывают путь к каталогу в файловой системе.
- Редакторы для ссылочных свойств. Они предназначены для отображения объекта по ссылке некоторым специфическим образом. Например, TabbedDetailPropertyEditor.
- Редакторы для новых типов данных, реализованных в составе платформы. Например, GoogleMapsPropertyEditor предназначен для работы с типом GeoPoint.

Конфигурирование редакторов осуществляется в Модели приложения.

1.3.1. TabbedDetailPropertyEditor

Отображает ссылочное свойство. Зачастую применяется для свойства this текущего объекта, что позволяет спроектировать оригинальную детальную форму с применением особенностей редактора. Идея редактора заключается в том, чтобы распределить многочисленные поля бизнесобъекта между несколькими детальными представлениями и отображать объект по частям.

Соответствующий контрол содержит группы иерархических вкладок слева и вложенный фрейм справа; фрейм отображает форму, выбранную на текущей вкладке. Редактор позволяет существенно увеличить быстродействие интерфейса при отображении громоздких бизнес-объектов.

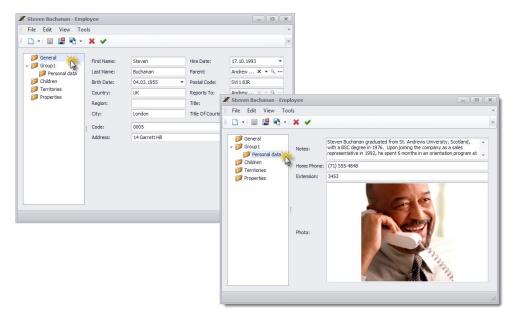


Рис. 3.

Поддерживаемые платформы: Win, Web, Mvc.

1.3.2. EnumPropertyEditor

Отображает и редактирует величины типа Enum, которые отмечены FlagsAttribute. Позволяет отмечать значения с помощью контролов CheckedComboBox или CheckedListBox, поддерживает отображение перечисления в несколько колонок.

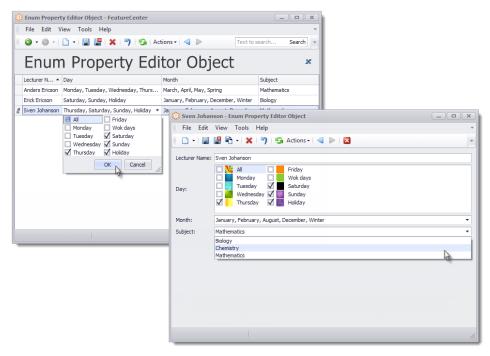


Рис. 4.

Поддерживаемые платформы: Win, Web, Mvc.

1.3.3. ExpressionPropertyEditor

Отображает и редактирует величины типа string. Предполагается, что строка представляет собой формулу на языке Expression. Таким образом, ExpressionPropertyEditor предназначен для оперативной разработки и отладки выражений.

Для редактирования выражения используется специфический визуальный элемент, который позволяет создавать формулы в рантайме, проверять и отлаживать их, свободно перемещаться по дереву метаданных и спискам встроенных функций, операторов, констант и параметров.

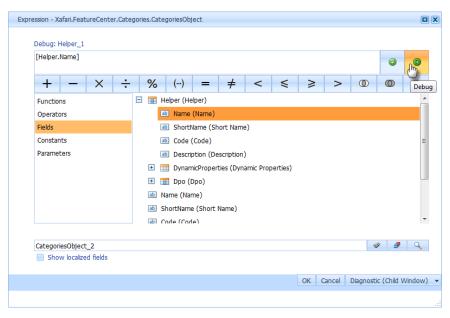


Рис. 5.

Поддерживаемые платформы: Win, Web.

1.3.4. VerticalGridPropertyEditor

Отображает ссылочное свойство. Для визуализации объекта используется VgridControl. Таким образом, все поля объекта представлены в виде списка.

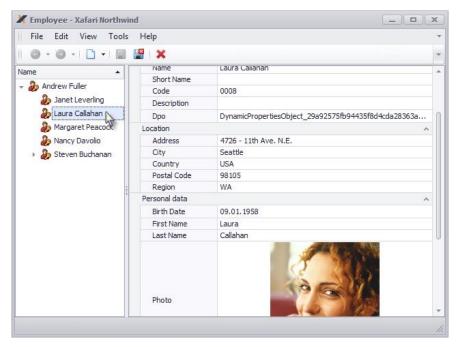


Рис. 6.

При таком подходе изменение набора полей происходит естественным образом, без нарушения общей композиции дизайна формы. В составе платформы VerticalGridPropertyEditor и его наследники используются в динамических свойствах, групповом редактировании, настройках приложения.

Поддерживаемые платформы: Win, Web, Mvc.

1.3.5. ViewIdPropertyEditor

Отображает и редактирует величины типа string, в которых содержатся идентификаторы View из Модели приложения. Редактор поддерживает выпадающий список доступных View, который отображает Id, Caption и Type of View, что облегчает поиск необходимого элемента.

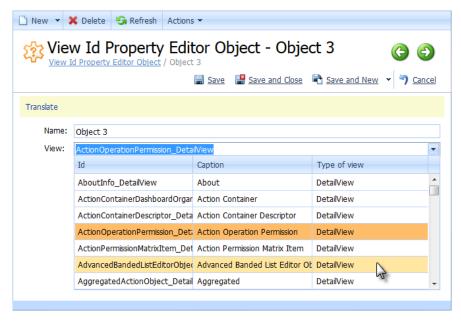


Рис. 7.

Поддерживаемые платформы: Win, Web.

1.3.6. GoogleMapsPropertyEditor

Редактор предназначен для встроенного типа Xafari.Base.GeoPoint. Позволяет отображать и редактировать глобальные координаты объекта при помощи позиционирования на карте Google.



Рис. 8.

Поддерживаемые платформы: Web.

1.3.7. ASPxGroupPropertyEditor

Отображает ссылочное свойство. Разработчик интерфейса может разделить поля бизнесобъекта на несколько групп. Генерируемая редактором детальная форма будет работать в режиме ViewMode, но в то же время раскрытая группа полей будет отображаться в режиме EditMode. Редактор позволяет существенно увеличить быстродействие ASP.NET интерфейса при отображении громоздких бизнес-объектов.

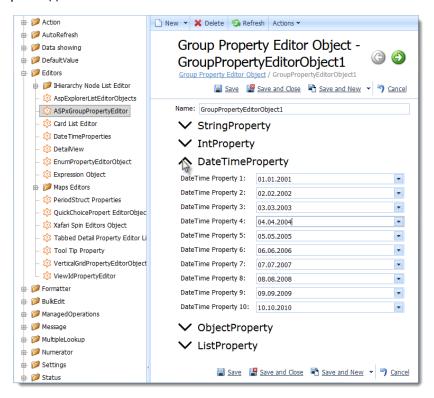


Рис. 9.

Поддерживаемые платформы: Web.

1.3.8. QuickChoicePropertyEditor

Отображает ссылочное свойство и обеспечивает выбор объекта из выпадающего списка с помощью быстрого поиска по ключевым полям.

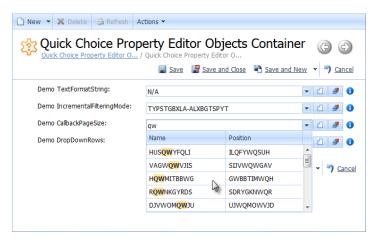


Рис. 10.

Список доступных объектов загружается порционно. Редактор использует визуальный элемент ASPxComboBox.

Поддерживаемые платформы: Web, Mvc.

1.3.9. XafariMultipleLookupEditor

Отображает и редактирует величины типа IList.

Предоставляет выпадающую форму для выбора значений и отображает выбранные элементы в виде компактного перечисления в строке вместо списковой формы. Возможны быстрая очистка выбранного списка, просмотр его в модальном окне.

В режиме выбора поддерживается выделение элементов с помощью чекбоксов, для быстрого управления выделением реализованы сочетания горячих клавиш.

В составе платформы наследник XafariMultipleLookupEditor используется в функционале обработки наборов выбранных объектов.

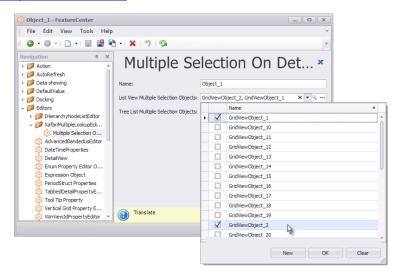


Рис. 11.

Поддерживаемые платформы: Win.

1.3.10. PeriodStructPropertyEditor

Редактор предназначен для встроенного типа Xafari.Base.PeriodStruct, обеспечивает просмотр и редактирование временного интервала. В составе платформы PeriodStructPropertyEditor используется в функционале фильтрации по периоду.

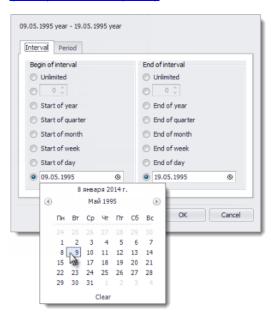


Рис. 12.

Поддерживаемые платформы: Win, Web.

1.3.11. XafariDatePropertyEditor

Отображает и редактирует величины типа DateTime. Поддерживаются два способа редактирования данных:

- выбор даты с помощью выпадающего календаря;
- явный ввод значений с клавиатуры (при этом используется соответствующая маска).

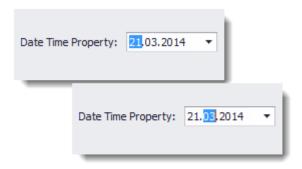


Рис. 13.

Поддерживаемые платформы: Win, Web.

1.3.12. XafariLookupEditor

Отображает ссылочное свойство и позволяет выбирать связанный объект из выпадающего списка. Кроме того, пользователь может работать с отдельной списковой формой. Также реализована возможность открывать детальную форму связанного объекта.

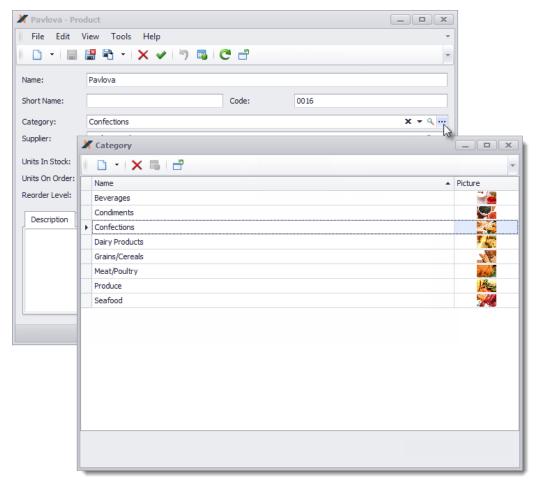


Рис. 14.

1.3.13. TypeImagePropertyEditor

Отображает и редактирует величины типа Туре. Предоставляет пользователю список доступных типов. В списке отображены имя типа и пиктограмма, определенная для этого типа в Модели приложения.

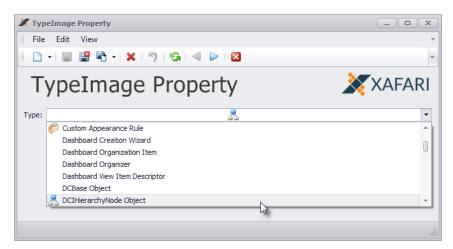


Рис. 15.

Поддерживаемые платформы: Win.

1.3.14. ObjectSetPropertyEditor

Редактор предназначен для встроенного типа Xafari.ObjectSet<>. Данный специализированный редактор является наследником XafariMultipleLookupEditor и используется в функционале обработки наборов выбранных объектов.

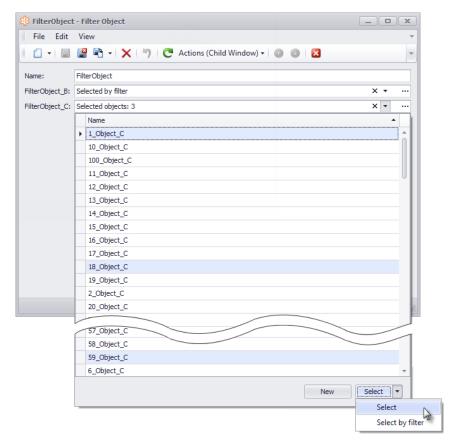


Рис. 16.

Поддерживаемые платформы: Win.

1.3.15. HyperLinkPropertyEditor

Отображает и редактирует величины типа System.String, содержащие гиперссылку в формате URI. Обеспечивает активацию ссылки по щелчку.

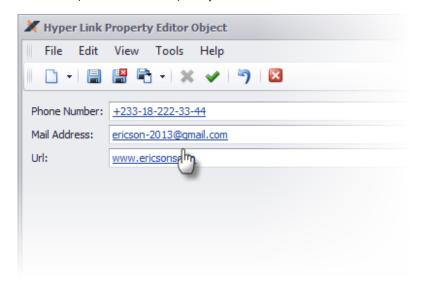


Рис. 17.

Поддерживаемые платформы: Win, Web, Mvc.

1.3.16. FolderBrowserPropertyEditor

Отображает и редактирует величины типа System.String, содержащие путь к каталогу. Для выбора каталога используется визуальный компонент FolderBrowserDialog.



Рис. 18.

Поддерживаемые платформы: Win.

1.3.17. LabelPropertyEditor

Отображает и редактирует величины типа System.String. Для отображения значения применяется визуальный компонент LabelControl, что обеспечивает визуализацию поля в режиме StaticText.

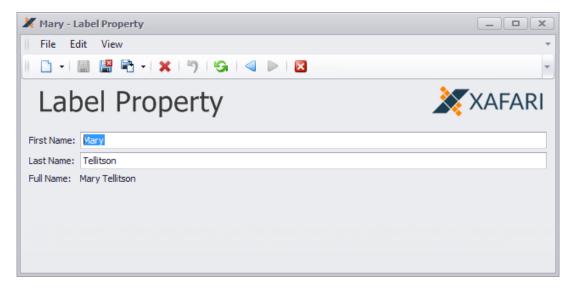
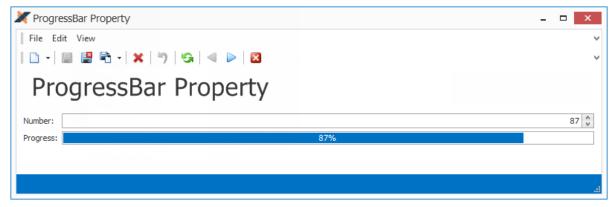


Рис. 19.

Поддерживаемые платформы: Win.

1.3.18. ProgressBarPropertyEditor

Отображает и редактирует величины типа System.Int. Переводит текущее значение в проценты и отображает его с помощью полосы прогресса.



Puc 20

Поддерживаемые платформы: Win, Web, Mvc.

1.3.19. HTMLPropertyEditor

Отображает и редактирует величины типа System.String. Подразумевается, что строка содержит текст в формате HTML.

Редактор предоставляет контрол, который обеспечивает удобное отображение и редактирование такого текста.

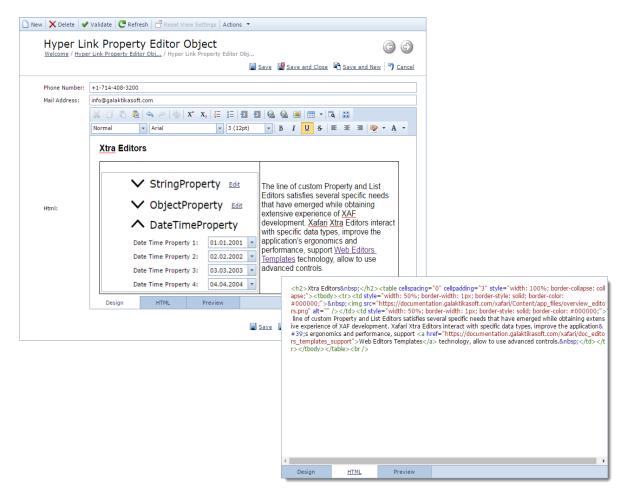


Рис. 21.

Поддерживаемые платформы: Web.

1.4. Extra List Editors (редакторы списков)

1.4.1. XafariGridListEditor

Отображает линейный список бизнес-объектов. Для визуализации данных используется Grid-Control.

Win-версия поддерживает возможность выбора элементов с помощью чекбоксов, Web-версия поддерживает технологию <u>редактируемых шаблонов</u>.

Поддерживаемые платформы: Win, Web.

1.4.2. XafariTreeListEditor

Отображает иерархический список бизнес-объектов в виде дерева.

Win-версия поддерживает возможность выбора элементов с помощью чекбоксов, Web-версия поддерживает технологию редактируемых шаблонов.

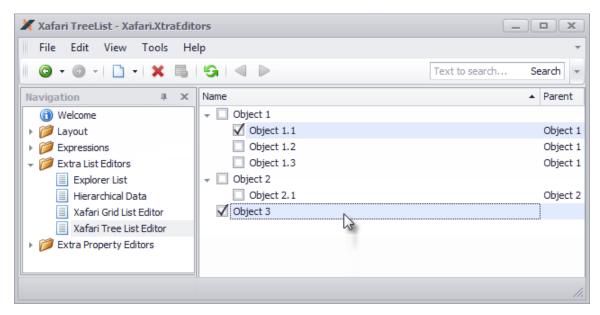


Рис. 22.

Поддерживаемые платформы: Win, Web.

1.4.3. ExplorerLisEditor

Отображает иерархический список данных. Этот компонент является составной частью технологии обработки иерархий в Xafari Framework (см. <u>иерархические данные</u>).

CollectionSource данного редактора имеет дополнительный критерий, который учитывает особенности реализации Parent-Child зависимостей на уровне БД. Такой подход позволил существенно оптимизировать извлечение данных, что обеспечивает высокое быстродействие.

Перемещение пользователя по узлам иерархии организовано по аналогии с проводником в ОС Windows.

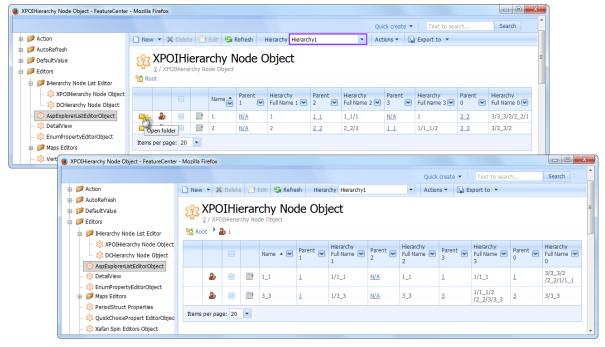


Рис. 23.

Поддерживаемые платформы: Win, Web.

1.4.4. HierarchyNodeListEditor

Отображает иерархический список данных, как и <u>ExplorerLisEditor</u>, является составной частью технологии обработки иерархий в <u>Xafari Framework</u> (см. <u>иерархические данные</u>), обладает аналогичными преимуществами в области быстродействия, поддерживает мультииерархичность.

Данные отображаются в виде древовидной структуры, пользователь имеет возможность править иерархическую структуру методом drag and drop.

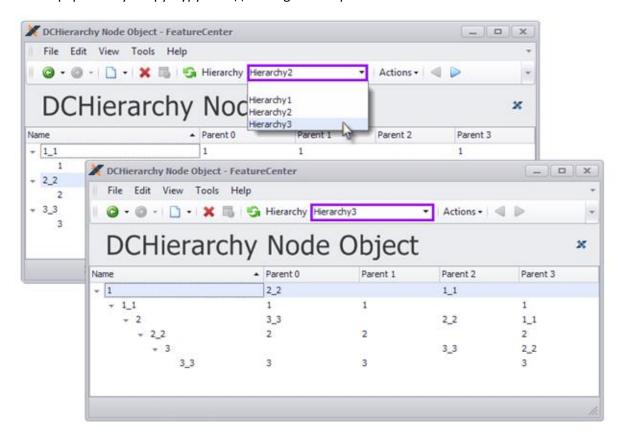


Рис. 24.

Поддерживаемые платформы: Win, Web.

1.4.5. CardListEditor

Этот редактор предназначен для эргономичного отображения коллекции объектов на Webстранице. Вместо построчного вывода элементы коллекции расположены в ячейках матрицы.

Макет отображения объекта внутри ячейки определяется на дополнительной детальной форме в Модели приложения.

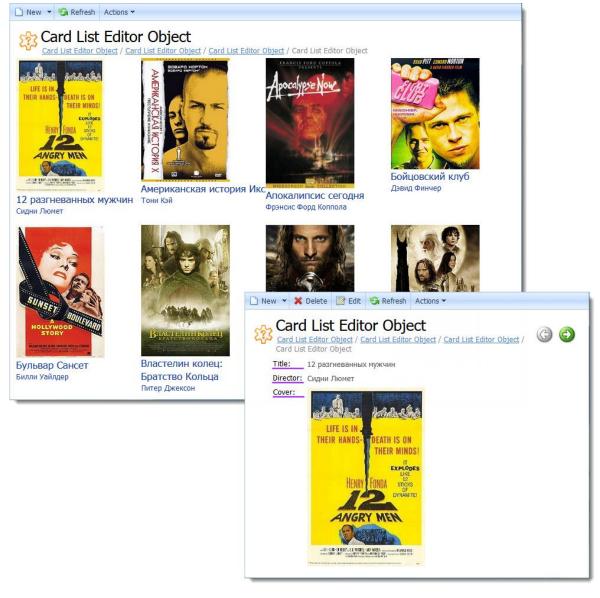


Рис. 25.

Поддерживаемые платформы: Web.

1.5. Templated Web Editors (редактируемые шаблоны для Web)

Эта технология позволяет изменять оформление редактора для ASP.NET интерфейса без вмешательства в код самого редактора.

Все поддерживаемые редакторы поставляются с базовой визуализацией. Чтобы оформить редактор иначе, разработчик использует .ascx компонент из палитры шаблонов Xafari Framework, кодирует в нем новый UI и привязывает его к редактору через Модель приложения.

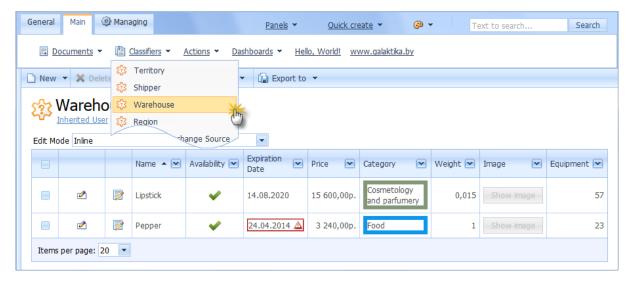


Рис. 26.

Ниже приведен список редакторов, которые поддерживают шаблоны:

- ASPxXafariBooleanPropertyEditor;
- ASPxXafariByteArrayPropertyEditor;
- ASPxXafariBytePropertyEditor;
- ASPxXafariColorPropertyEditor;
- ASPxXafariCriteriaPropertyEditor;
- ASPxXafariDateTimePropertyEditor;
- ASPxXafariDecimalPropertyEditor;
- ASPxXafariDefaultPropertyEditor;
- ASPxXafariDoublePropertyEditor;
- ASPxXafariEnumPropertyEditor;
- ASPxXafariFloatPropertyEditor;
- ASPxXafariHtmlPropertyEditor;
- ASPxXafariImagePropertyEditor;
- ASPxXafariInt64PropertyEditor;
- ASPxXafariIntPropertyEditor;
- ASPxXafariListPropertyEditor;
- ASPxXafariLookupPropertyEditor;
- ASPxXafariObjectPropertyEditor;
- ASPxXafariPopupCriteriaPropertyEditor;
- ASPxXafariProtectedContentPropertyEditor;
- ASPxXafariStringPropertyEditor;
- ASPxXafariTimeSpanPropertyEditor;
- ASPxXafariTreeListEditor;
- ASPxXafariTypePropertyEditor.

1.6. Extra Filters (фильтр по периоду)

Специальный контроллер в совокупности с редактором <u>PeriodStructPropertyEditor</u> выполняет фильтрацию списковой формы по полю типа DateTime. Пользователь может указывать интересующие его периоды и интервалы, а контроллер преобразует их в критерии и выполнит фильтрацию.

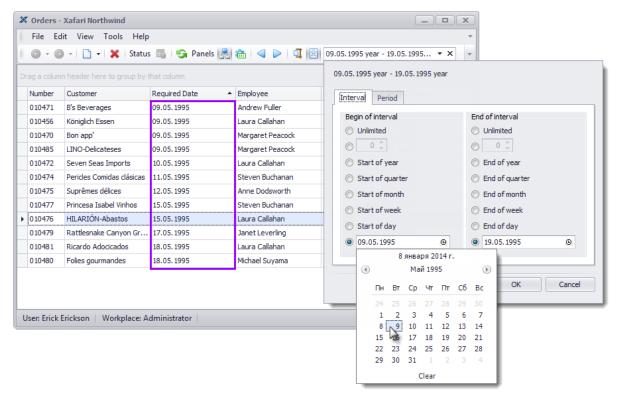


Рис. 27.

Дополнительные фильтры реализованы для платформ WinForm, ASP.NET и MVC.

1.7. Smart Design (стратегия генерации форм)

Mодуль XafariSmartDesign предоставляет возможность управлять стратегией генерации визуальных форм прямо в коде бизнес-класса.

Базовая стратегия автоматически выносит все поля объекта на макет формы. Зачастую это необоснованно и влечет чрезмерные расходы вычислительных ресурсов.

Альтернативная стратегия SmartDesign формирует для бизнес-объекта пустую форму и предлагает разработчику следующие возможности:

- описывать компоновку полей на форме в коде объекта с помощью атрибутов;
- генерировать дополнительные формы наряду с базовыми;
- управлять базовой стратегией на различных уровнях (сборка, пространство имен, класс);
- реализовать и использовать собственную стратегию.

XafariViewLayoutStrategy может использоваться на платформах Win, Web, Mvc.

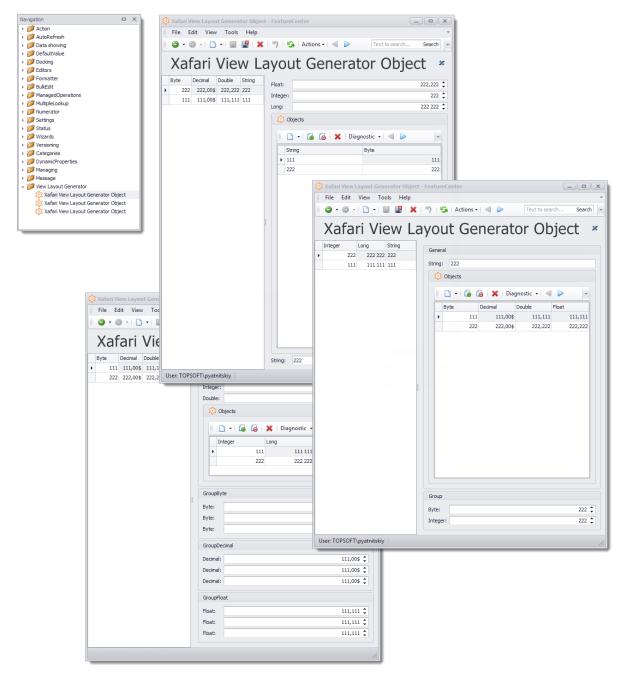


Рис. 28.

Стратегия генерации форм реализована для платформ WinForm, ASP.NET и MVC.

1.8. Actions (действия)

Действие в терминологии Xafari Framework — это абстрактный компонент, который используется для реализации взаимодействия конечного пользователя с интерфейсом приложения. Действие имеет абстрактную модель и реализацию в виде конкретного контрола на уровне определенной визуальной платформы. Библиотека Xafari.dll предоставляет следующие типы действий:

 CheckAction. Представляет собой некоторый элемент управления с двумя состояниями: включено и выключено. Является наследником SimpleAction. В момент переключения состояний генерирует соответствующее событие.

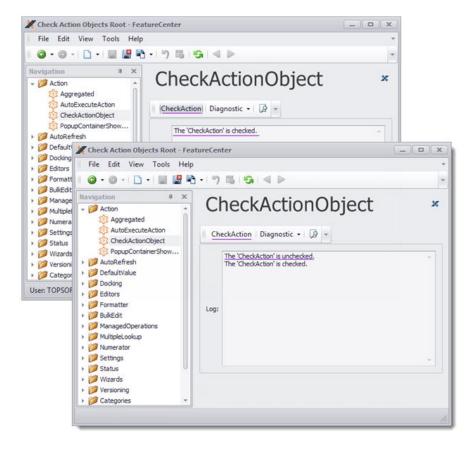


Рис. 29.

• AggregatedAction. Является динамическим контейнером для Action других типов и выступает наследником SingleChoiceAction. Иерархическая структура пунктов и подпунктов может быть полностью спроектирована в Модели приложения.

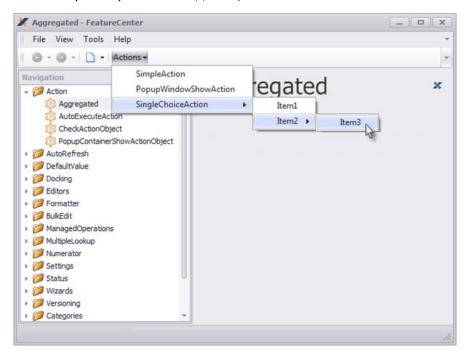


Рис. 30.

• PopupContainerShowAction. Отображает выпадающий контейнер перед выполнением метода Execute(). Контейнер отображает некоторую форму, определенную разработчиком. Является наследником PopupWindowShowAction. Данный Action используется в функционале Extra Filters.

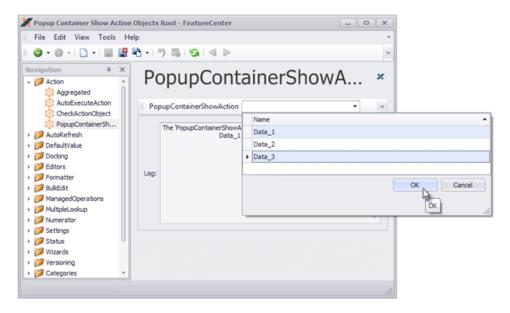


Рис. 31.

Действия поставляются в модулях Xafari, Xafari.Win, Xafari.Web, Xafari.Mvc.

1.9. Wizards (мастера)

Мастера позволяют реализовать поэтапное выполнения сложной операции. Визуально работа мастера напоминает процесс пошаговой инсталляции ПО. Этапы мастера представлены в виде последовательности модальных окон, каждое из которых отображает некоторую детальную форму. Между этапами мастера возможны условные переходы. Работа мастера завершается выполнением Action или отказом от его выполнения.

Конфигурация мастера доступна в дизайнере Visual Studio, в Модели приложения, а также через специальные события.

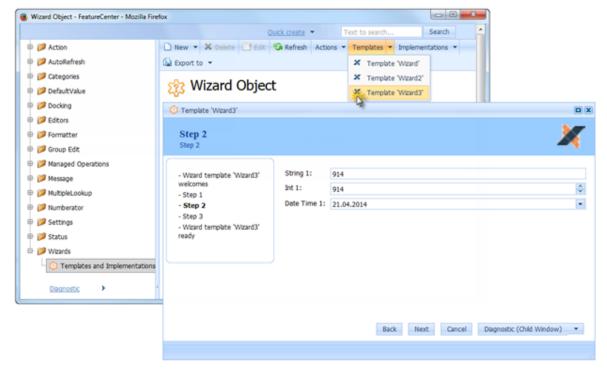


Рис. 32.

Мастера поставляются в модулях Xafari, Xafari.Win, Xafari.Web.

1.10. Дополнительные функции модуля Xafari

1.10.1. Context Help (контекстная справка)

Обеспечивает возможность связать ERP-приложение с сайтом онлайн-документации. Специальный Action на панели инструментов будет обеспечивать переход на сайт с каждой детальной или списковой формы, причем для каждой формы можно генерировать уникальный url с использованием ObjectFormatter.

1.10.2. Auto Refresh (автоматическое обновление данных)

Обеспечивает автоматическое обновление информации на целевой форме через заданный временной интервал.

1.10.3. Action by Timer (действие по таймеру)

WinAutoExecuteController поддерживает настраиваемый таймер с обратным отсчетом времени. После завершения отсчета контроллер вызовет метод Execute() для заданного Action.

1.10.4. Action Executor (контекстное действие)

Реализация интерфейса IActionExecutor позволит использовать один Action для выполнения различных операций в зависимости от контекста вызова.

1.10.5. Criteria Object Creation (критерий для новых объектов)

Данная функция позволяет использовать значения текущего фильтра списковой формы в качестве значений по умолчанию для вновь создаваемых объектов.

1.10.6. Show Success Message (сообщение об успешном выполнении)

Предоставляется возможность уведомлять пользователя о том, что действие успешно выполнено.

1.10.7. Layout Group Expand Enable (свертывание/развертывание групп на форме)

Позволяет управлять видимостью группы свойств бизнес-объекта на детальной форме. Пользователь может самостоятельно по мере надобности сворачивать либо разворачивать группу.

2. Бизнес-компоненты

2.1. IHiearchyNode (иерархические данные)

Интерфейс Xafari.BC.HierarchicalData.lHierarchyNode декларирует методику описания иерархических данных. Объекты, реализующие данный интерфейс, поддержаны в системе специальными редакторами и сервисами, которые обеспечивают высокопроизводительную и наглядную обработку информации.

IHierarchyNode позволяет реализовать несколько иерархий в одном бизнес-классе. Предусмотрены несколько стратегий удаления подчиненных (children) узлов:

- подчиненные узлы удаляются полностью;
- подчиненные узлы переносятся в корень иерархии;
- подчиненные узлы переносятся на уровень удаляемого узла.

Для обработки иерархических данных внутри системы предоставляется специальный сервисный класс Xafari.BC.HierarchicalData.HierarchyManager. Пользовательский интерфейс для иерархических структур строится с использованием специальных редакторов:

- TreeListEditor;
- HierarchyNodeListEditor;
- ExplorerListEditor.

2.2. Object Set (набор выбранных объектов)

Данное абстрактное решение позволяет единым образом хранить и обрабатывать наборы выбранных объектов произвольных типов. Это решение позволяет избежать создания отдельных персистентных коллекций со специальными персистентными объектами для каждого типа, что приводит к неоправданному усложнению модели данных приложения. Универсальная технология ObjectSet предлагает фиксированную структуру БД и возможность составлять эффективные запросы, использующие наборы объектов, без дополнительного кодирования.

В состав решения входят:

- персистентные типы для хранения ссылок на выбранные объекты;
- специальный класс Xafari.BC.Data.ObjectSet<T> для реализации множества выбранных объектов (выборку можно описывать посредством критерия либо указывать конкретные Id объектов);
- специальный редактор для работы со множеством объектов ObjectSet для нескольких объектов или указания критерия отбора;
- CustomFunction для построения эффективных запросов в БД, которые содержат один или несколько фильтров вхождения во множество.

Поддержаны БД MS SQL и Oracle.

2.3. Xafari Security (система безопасности)

Система безопасности Xafari Security позволяет разграничивать доступ к данным на основании прав и ролей. Базовые типы прав (Permissions) позволяют регулировать уровни доступа к типам, свойствам типов, действиям и элементам навигации. Модель данных системы безопасности реализована по двум технологиям: eXpress Persistent Objects и Domain Components. Возможны 3 типа аутентификации:

- явная пользователь должен ввести логин и пароль;
- Windows Autentification;
- смешанная пользователь сам указывает подходящий способ аутентификации.

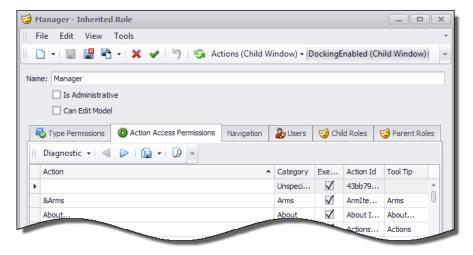


Рис. 33.

2.4. eXtension Framework (расширение бизнесобъектов)

eXtension Framework описывает методологию проектирования и разработки модели данных в сложных приложениях, а также предоставляет набор базовых классов и сервисов для реализации предложенных методик. eXtension Framework подходит только для систем, построенных с использованием Domain Components.

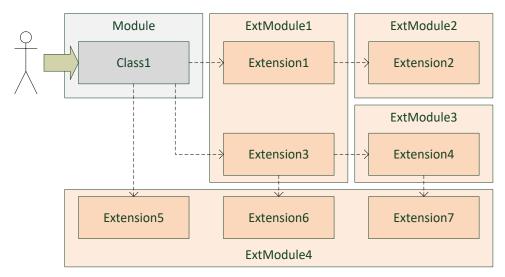


Рис. 34.

В основе концепции лежит отказ от наследования для дополнения бизнес-класса новыми функциями. Вводятся базовые понятия сущность (Entity) и расширение сущности (Entity Extension). Если к сущности необходимо добавить новый набор полей, то это следует сделать с помощью нового расширения: создать расширение, описать в нем требуемый функционал и добавить к сущности ассоциацию на расширение. После регистрации расширения объект сущности и его новый функционал будет является цельной монолитной конструкцией.

Такой подход имеет ряд важных преимуществ при поэтапном наращивании функционала и расширении ERP-системы.

Сохраняется понятность и простота модели данных, она не засоряется лишними технологическими классами. Конечный пользователь будет взаимодействовать всегда с одной и той же сущностью, независимо от состава ее расширений.

Сущности и расширения могут разрабатываться в различных модулях и компоноваться по мере необходимости. При этом максимально исключены конфликты совместного использования параллельных расширений, которые появляются при использовании наследования.

2.5. Logic Controllers (контроллеры бизнеслогики)

Инструмент для реализации бизнес-логики приложения, который позволяет существенно сократить количество используемых контроллеров. Вместо добавления нового ViewController зачастую достаточно воспользоваться классом Xafari.BC.LogicControllers.LogicControllerService.

Контроллер бизнес-логики не может является контейнером для действий, поэтому может быть использован в сценариях без действий.

Контроллер бизнес-логики будет привязан только к определенному типу бизнес-объекта и, соответственно, будет активироваться только на форме этого бизнес-объекта. Поскольку LogicController не является наследником от Controller, то форма будет видеть только контроллеры для текущего типа.

Таким образом, контроллер бизнес-логики позволяет избежать избыточности наследников класса Controller и связанных с этим проблем падения быстродействия.

2.6. Object Status (статусы объектов)

Статус объекта отражает его текущее состояние по степени готовности к его использованию. Доступны следующие статусы:

- Черновик (Draft) объект находится в состоянии редактирования;
- Опубликован (Published) объект отредактирован и может быть использован;
- Архив (Archive) устаревший объект.

При отображении объектов со статусами возможности списковой формы расширяются несколькими дополнительными действиями:

- Show draft включить/выключить отображение черновиков;
- Show archive включить/выключить отображение устаревших объектов;
- Status выбор другого статуса.

Статус объекта можно изменить также на детальной форме.

Статусы объектов использованы для активных блокировок и реализованы для персистентных объектов (XPO) и Domain Components.

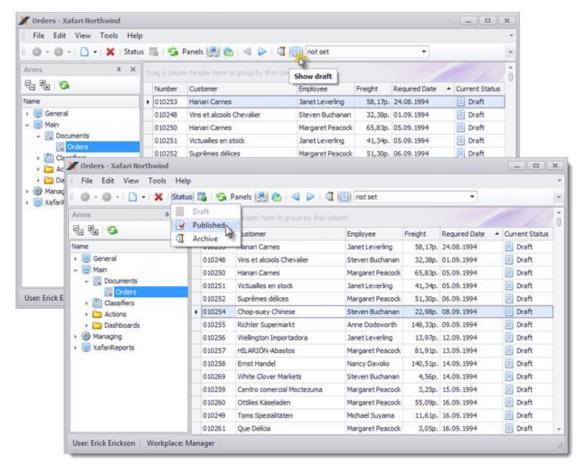


Рис. 35.

2.7. Bulk Edit (групповое редактирование)

Функциональность востребована в случаях, когда свойствам нескольких объектов необходимо одновременно присвоить одинаковое значение. Пользователь может выделить эти объекты и одновременно редактировать это свойство для всей группы. Обработка общих значений проводится в специальном редакторе, размещенном на плавающей панели.

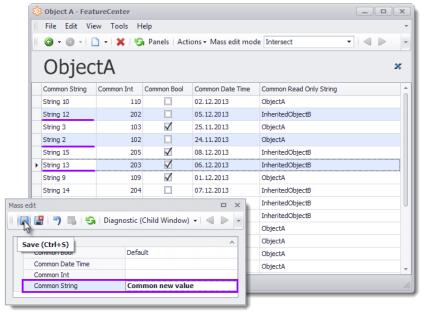


Рис. 36.

2.8. Numerators (нумераторы)

Сервис нумераторов генерирует значения свойств по настраиваемым правилам с использованием порядковой нумерации.

При автоматическом заполнении полей сервис нумераторов предоставляет следующие возможности:

- установка начального значения индекса (StartValue) и шага (Step);
- установка шаблона номера (Template);
- сквозная нумерация (ThroughNumeration) применяется для последовательного нумерования объектов различных типов. Например, все исходящие документы (накладные, платежки и проч.) могут использовать одну цепочку номеров;
- использование удаленных номеров (UseDeleteNumbers) предполагает, что все номера нумератора будут использованы. Если какой-то объект удаляется, то его номер не пропадает, а используется при очередном обращении к сервису нумераторов;
- нумераторы работают как на визуальных формах, так и при создании объектов в коде; поддержана пакетная обработка документов.

Нумераторы реализованы для персистентных объектов (XPO) и Domain Components.

2.9. Categories (категории)

Категории используются для внешней классификации по ссылкам, имеющимся у объектов. Категории могут дополнять любую списковую форму в виде <u>плавающей панели</u>, которая содержит список классификаторов.

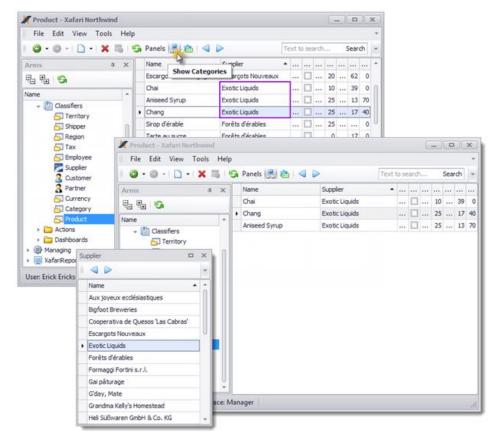


Рис. 37

Категории могут быть использованы на платформах Web, Win, Mvc.

2.10. Managed Operations (управляемые операции)

Управляемая операция является программной оберткой для произвольного прикладного метода, предоставляя ряд полезных функций в процессе выполнения:

- выполнение в отдельном или общем потоке;
- визуализация процесса, возможность приостановки, прерывания и продолжения;
- логирование и мониторинг процесса;
- вариативность операций: локальные, глобальные, вложенные.

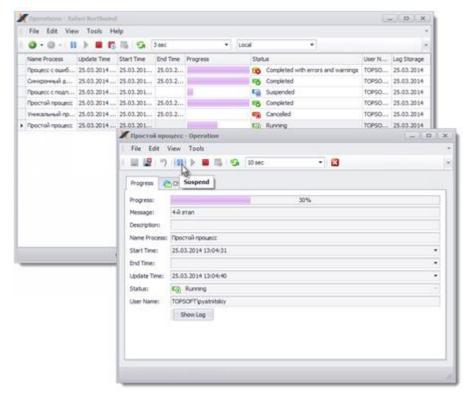


Рис. 38.

2.11. Dynamic Properties (динамические реквизиты)

Динамические реквизиты — это дополнительные поля бизнес-объектов, которые не предусмотрены при проектировании модели данных. Динамические реквизиты существуют только для экземпляра объекта. Поэтому два объекта одного типа могут иметь различные динамические реквизиты.

Динамические реквизиты реализованы для персистентных объектов (XPO) и Domain Components.

2.12. Application Settings (настройки приложения)

Модуль Xafari.BC.Settings предоставляет инфраструктуру для разработки и поддержки системы настроек приложения. Настройка — это некоторая величина, значение которой учитывается приложением на определенном этапе работы. Настройки могут быть организованы в виде

иерархической структуры групп. Значения настроек могут быть одинаковыми для всех экземпляров приложения, а могут быть рассчитаны для каждого экземпляра индивидуально с использованием определенных правил. Концептуально настройки приложения аналогичны настройкам ОС Windows для различных пользователей.

Решение включает следующие возможности:

- определять и группировать настройки посредством специальных интерфейсов, расширять модель приложения узлом настроек; каждый модуль может содержать собственные настройки, которые будут интегрированы в общую систему при подключении данного модуля;
- формирование правил распределения значений настроек между экземплярами приложения посредством структуры слоев; простейшим вариантом является слой системных настроек (общих) и слой пользовательских настроек (для каждого пользователя имеются собственные значения);
- конвертация настроек при смене версии приложения;
- генерация соответствующей формы для редактирования настроек в UI;
- обработка настроек в приложении посредством сервиса Xafari.BC.Settings.SettingsAccessor.

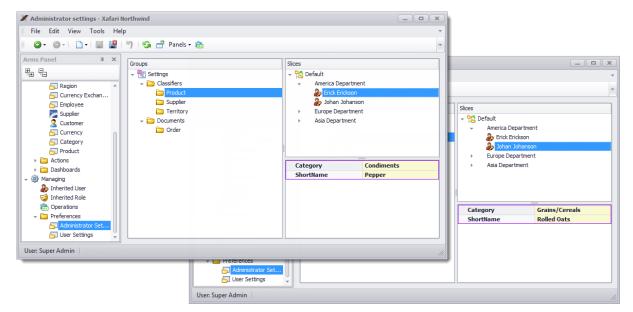


Рис. 39.

Настройки приложения могут быть использованы на платформах Web, Win, Mvc.

2.13. Data Versioning (версионность данных)

Под версионностью бизнес-объекта подразумевается его способность привязать свое состояние к временным отсечкам, а также обеспечение возможности получить доступ к состоянию объекта (его версии) для указанной временной точки (момента времени). Похожая задача решается системами контроля версий, такими как SVN, TFS и проч.

Поддержка версионности для бизнес-объектов основана на следующих концепциях:

- Поддержка версий может быть использована только для бизнес-объектов, реализованных с помощью технологии Domain Components.
- Для хранения версий бизнес-объектов используется сам Domain Component этого бизнесобъекта.
- Для версионных бизнес-объектов существует поле VersionId, содержащее одинаковое значение для всех версий одного бизнес-объекта.

 Для ссылок на версионные бизнес-объекты необходимо использовать VersionId и вычисляемые поля.

Для работы с версиями объекта предусмотрены несколько операций:

- сохранение новой версии объекта (Save as new version);
- получение последней версии бизнес-объекта (Get Latest Version);
- получение конкретной версии бизнес-объекта на заданные дату и время;
- просмотр истории изменений объекта (Show history);
- установка текущей даты для работы с версиями.

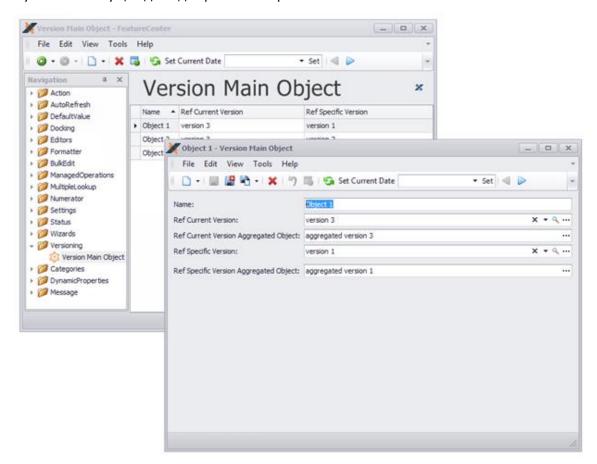


Рис. 40.

2.14. Work Places (рабочие места)

Рабочие места — это дополнительный промежуточный уровень Модели приложения, в котором хранится конфигурация модели для групп пользователей.

Каждому пользователю системы может быть доступно несколько конфигураций рабочих мест. В свою очередь пользовательские изменения модели сохраняются индивидуально для конкретного пользователя и конкретного рабочего места.

Связь рабочего места с пользователем выполняется с применением наименований назначенных пользователю ролей. Наименования рабочих мест должны соответствовать наименованиям существующих ролей. То есть пользователю будет доступно рабочее место, если существует файл рабочего места с именем, соответствующим одной из назначенных пользователю ролей.

Для создания и редактирования рабочих мест используется специальное приложение <u>Xafari</u> <u>Applications Support</u>.

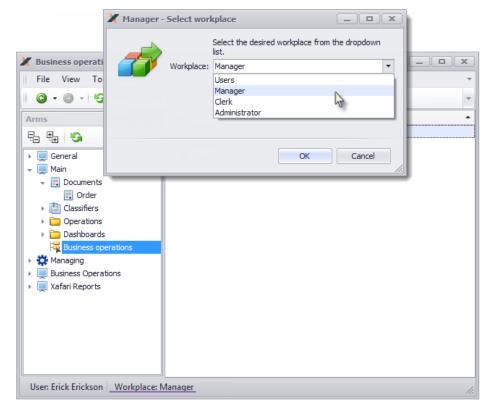


Рис. 41.

Рабочие места могут быть использованы на платформах: Web, Win, Mvc.

2.15. Data Management (импорт/экспорт данных)

Данное решение позволяет синхронизировать персистентные данные со сторонним хранилищем.

Импорт или экспорт данных может потребоваться при решении следующих задач:

- начальная загрузка данных;
- интеграция со сторонними приложениями в обе стороны;
- переход на следующую версию приложения.

Задачи такого рода всегда возникают при внедрении и эксплуатации приложений у заказчика. **Xafari Framework** предлагает целостный подход для их решения. Разработчику следует имплементировать интерфейс Xafari.Data.IDataSource и указать параметры миграции. После этого процедуры импорта/экспорта можно будет запускать автоматически, в т. ч. и в консольном режиме.

В состав платформы включены некоторые популярные типы источников данных: Access, XML, CSV, Excel, Галактика и другие.

2.16. Xafari Service (сервисы Xafari)

Сервис Xafari — это шаблон проектирования, предназначенный для разработки бизнес-логики приложения. Сервисы позволяют быстро разрабатывать бизнес-функции, обладающие следующими характеристиками:

- Функция является синглтоном ХАГ-приложения.
- Функцию можно активировать и деактивировать целиком или только в конкретном контексте.

- Функция может иметь свои собственные настройки, описанные единым образом. Параметры открыты для настройки администратором.
- Функция имеет определенный жизненный цикл, схожий с контроллером.
- Функция может быть связана с текущим пользователем.

2.17. Business Operations (бизнес-операции)

Бизнес-операция является одним из инструментов для разработки бизнес-логики приложения. БО предназначены для кодирования сложных расчетов с последующим изменением состояния бизнес-объектов. Примерами БО являются процедура расчета зарплаты, формирование накладной на основании ДО, получение прайс-листа, процедура МRP и проч. В целом для бизнес-операции характерно преобразование одних данных (входов) в другие (выходы). Разработчику достаточно добавить в код приложения наследника класса Xafari.BC.BusinessOperations.BusinessOperationBase, настроить его с помощью атрибутов и реализовать бизнес-логику. Все необходимые элементы UI будут сгенерированы автоматически. Бизнес-операции позволяют подменять используемый алгоритм без перекомпиляции кода, тем самым поддерживается возможность гибкого конфигурирования конечного приложения.

Бизнес-операции могут выполняться как независимо (без контекста, например, пакетное формирование расходных ордеров, проверка корректности БД), так и в контексте определенного объекта (например, пересчет сумм документа выполняется для экземпляра накладной).

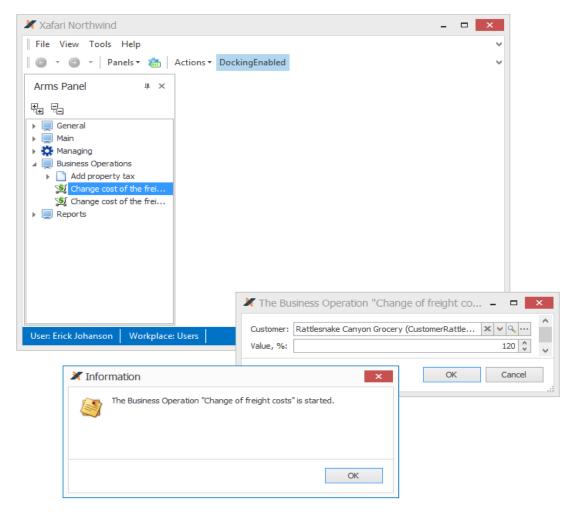


Рис. 42.

Бизнес-операции могут быть использованы на платформах Web, Win, Mvc.

2.18. XAS (конфигуратор)

Конфигуратор или Xafari Applications Support (XAS) — это автономное приложение для Windows, которое обеспечивает администрирование ERP-системы, базирующейся на Xafari Framework. XAS — это специальный инструмент для адаптации фреймворка к потребностям заказчика. Использование XAS позволяет ограничить количество административных функций в самом бизнес-приложении, поместив эту функциональность в отдельное приложение. XAS оказывает положительный эффект как на требования к системе (меньше кода, меньше расход памяти и загрузки процессора) так и на пользовательский интерфейс (формы администрирования не перегружают приложение). С помощью отдельного административного приложения также можно значительно повысить уровень безопасности корпоративного программного обеспечения.

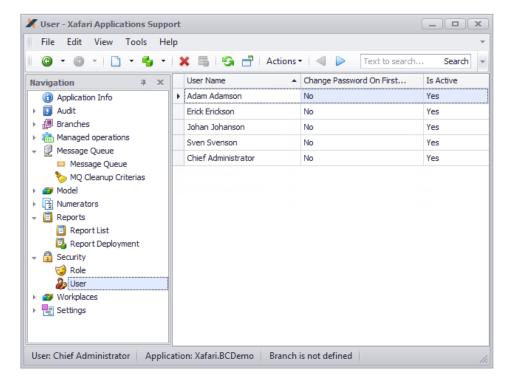


Рис. 43.

XAS редактирует Модель приложения и обеспечивает администрирование следующих функций:

- Audit;
- Branches;
- Managed Operations;
- Message Queue;
- Numerators;
- Reports;
- Security;
- Workplaces;
- Settings.

2.19. Console Application (консоль приложения)

Консольное приложение запускает экземпляр класса XafApplication, инициализация данного экземпляра производится на основании настроек, указанных в <u>AppModule</u> или в config-файле, в частности, производится загрузка лишь указанного перечня модулей, а настройки безопасности запущенного экземпляра соответствуют правам пользователя, выполнившего запуск.

Таким образом, пользователь получает урезанный вариант ХАГ-приложения, пригодный, одна-ко, для выполнения некоторых задач. Диалог пользователя с приложением осуществляется посредством командной строки. Данный сценарий актуален для импорта/экспорта данных, пакетного формирование отчетов и прочих функций, допускающих выполнение в автоматическом, пакетном режиме, возможно, по расписанию. Взаимодействие с пользователем в таких случаях минимально или отсутствует вовсе.

Консольное приложение может выполнять <u>бизнес-операцию</u> или любую другую задачу, закодированную с учетом интерфейса Xafari.BC.Commands.ICommand.

3. Компоненты ERP

3.1. AppModule (модуль приложения)

AppModule — это определенная методика проектирования ERP-системы. AppModule аккумулирует весь набор необходимых платформенно-независимых библиотек и настроек. Далее при проектировании платформенно-зависимого модуля следует сослаться только на AppModule, это избавляет от необходимости дублировать весь набор основных библиотек и настроек.

Данный подход позволяет избежать путаницы и неоднозначности в случае разработки приложения для различных платформ (Web, Win, Console и проч.). Обычно основу приложения составляет определенный набор платформенно-независимых модулей. Этот набор модулей будет одинаковым для всех платформ и будет дополняться в каждом модуле определенными платформенно-зависимыми библиотеками и настройками.

3.2. Audit (аудит)

Модуль аудита позволяет протоколировать изменения данных и вызов действий и бизнесопераций. Модуль имеет гибкие настройки и высокое быстродействие и устойчивость при массивных изменениях объектов (1000 и более объектов в транзакции).

Поддерживается протоколирование следующих событий:

- удаление, создание и модификация объектов включая их поля и коллекции;
- выполнение контекстных и независимых бизнес-операций (БО) включая их откат;
- выполнение пользовательских действий (Actions): контекстных и независимых;
- формирования отчетов;
- вход и выход пользователей.

Протоколирование реализовано на уровне СУБД с использованием триггеров. Это обеспечивает высокую производительность на больших объемах данных и транзакций.

3.3. Docflow (документооборот)

Подсистема документооборота является средством автоматизации широкого спектра бизнеспроцессов. С одной стороны, модуль содержит инструменты для формального описания жизненного цикла объекта-документа, с другой — предоставляет механизмы, обеспечивающие и контролирующие прохождение этого цикла объектами в процессе эксплуатации системы.

Полагается, что бизнес-процесс фиксируется в состояниях специального объекта, именуемого документом, который обрабатывается исполнителями в соответствии с регламентом.

Регламент описывает последовательность этапов, которые проходит документ, перечень и последовательность исполняемых работ. Возможно описывать как строго фиксированные регламенты наподобие Workflow, так и гибкие регламенты в стилистике Adaptive Case Management. В течение жизненного цикла вся связанная с документом информация накапливается и агрегируется.

Регламент детализирует и уточняет ведение документа в соответствии с требованиями конкретной прикладной задачи. ERP-система, применяющая данный модуль, становится адаптив-

Компоненты ERP

ной и приобретает свойства робастности (подстройки под изменяющиеся, вновь появившиеся требования).

Проектирование регламентов абсолютно не требует кодирования на С#, все необходимые этапы, работы, исполнители, переходы и прочие детали описываются в Модели приложения. После запуска системы сервис Docflow анализирует модель, генерирует необходимые элементы и обеспечивает весь жизненный цикл документа.

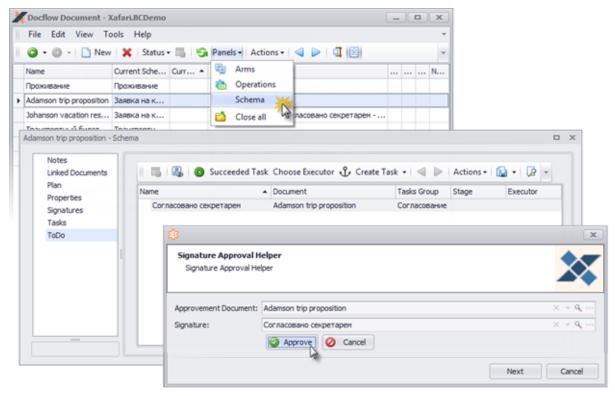


Рис. 44.

3.4. Branches (филиалы)

Модуль филиалов предназначен для частичной изоляции данных различных организационных структур. Использование данного модуля позволяет прозрачно автоматизировать компании холдингового типа. При этом обеспечивается возможность получения консолидированной отчетности по корпорации в целом с группировкой и фильтрацией данных в разрезе филиалов.

Функционал модуля позволяет эффективно управлять правами доступа к информации: в единой базе собственные данные для филиала полностью открыты, доступ к данным остальных филиалов регламентируется в соответствии с установленными правами. Несанкционированный доступ к информации филиалов исключается.

3.5. Message Queue (очередь сообщений)

Очереди сообщений необходимы для связывания независимых приложений и сервисов и их совместной работы. Приложения могут генерировать различные виды сообщений для различных сервисов, которые обрабатывают сообщения и возвращают некий результат приложению. Такая схема позволяет перенести выполнение сложной бизнес-логики с экземпляра приложения на ответственность сервисов, уменьшая потребляемые приложением ресурсы и время.

Все сообщения хранятся в той же базе данных, с которой работает приложение, и должны различаться по типу. Сообщения обрабатываются асинхронно на стороне сервера.

40 Компоненты ERP

3.6. Xafari Server (сервер Хаfari)

Сервер Xafari служит для мониторинга очереди сообщений и распределения сообщений соответствующим обработчикам. Работает сервер асинхронно, как отдельный экземпляр приложения. В процессе работы он выбирает ожидающие сообщения, блокирует их и передает подходящим обработчикам. Сервер использует собственный провайдер безопасности, который обеспечивает окружение для обработки сообщения и асинхронное взаимодействие с базой данных посредством пула подключений.

Возможна работа сервера совместно с приложением, в новом потоке, в этом случае используется локальная очередь для хранения сообщений (или выделенный сервер). При использовании локальной очереди все сообщения удаляются по завершении работы приложения. В режиме выделенного сервера сообщения хранятся в базе данных приложения.

На основе сервера расчетов реализован сервер отчетов, который генерирует отчеты на выделенном ресурсе. Благодаря провайдеру безопасности в отчет попадают лишь те данные, которые были доступны пользователю, инициировавшему сообщение.

3.7. Xafari Reports (отчеты Xafari)

Данная подсистема предоставляет возможность разработки отчетов в коде на этапе проектирования и во время выполнения. Разработчик имеет возможность реализовывать собственные оригинальные алгоритмы для получения исходных данных. Поддержаны форматы печатного отчета, графика, сводной таблицы, выгрузки в файл, Excel. Решение о том, в какой именно форме получать отчет, принимает сам пользователь. Спроектированный отчет можно генерировать с учетом параметров, заданных пользователем, наборы параметров можно сохранять и использовать повторно.

Подсистема также включает сервер отчетов, который генерирует финальный отчет удаленно и сохраняет его в базе данных. Полученный отчет доступен для просмотра через специальную списковую форму.

Компоненты ERP 41