КОРПОРАЦИЯ
Галактика

Платформа Xafari.
Состав продукта.
Краткое описание.

Оглавление

Общие сведения

Платформа Xafari представляет собой набор библиотек, которые содержат готовые компоненты и инструментарий для быстрой разработки различных функций для ERP систем, реализующих архитектуру DevExpress XAF.

Ключевыми особенностями платформы 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 пр.

Функционал платформы разделен на 3 группы:

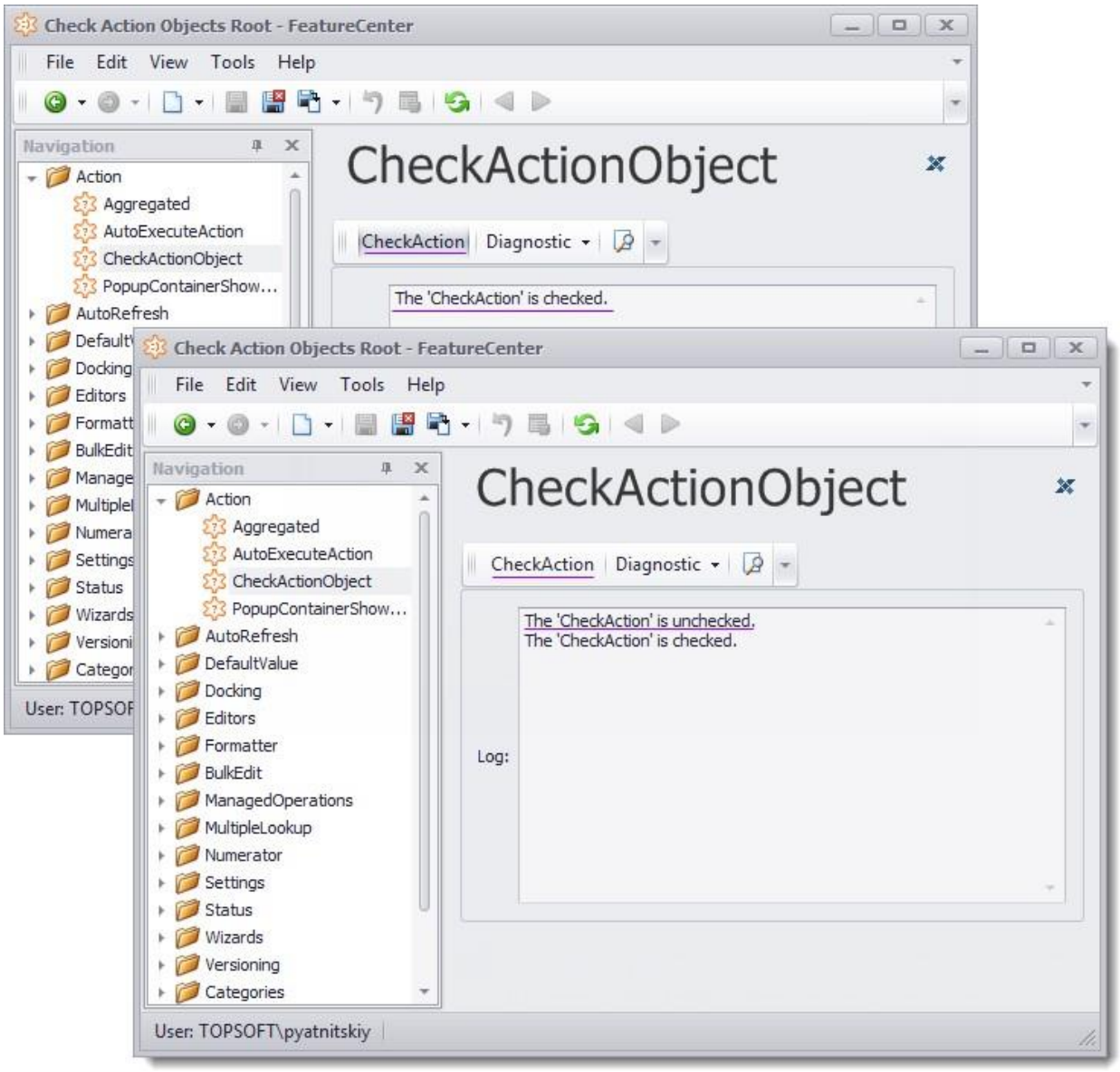
- Компоненты общего назначения
- Бизнес Компоненты
- Компоненты ERP

Компоненты общего назначения

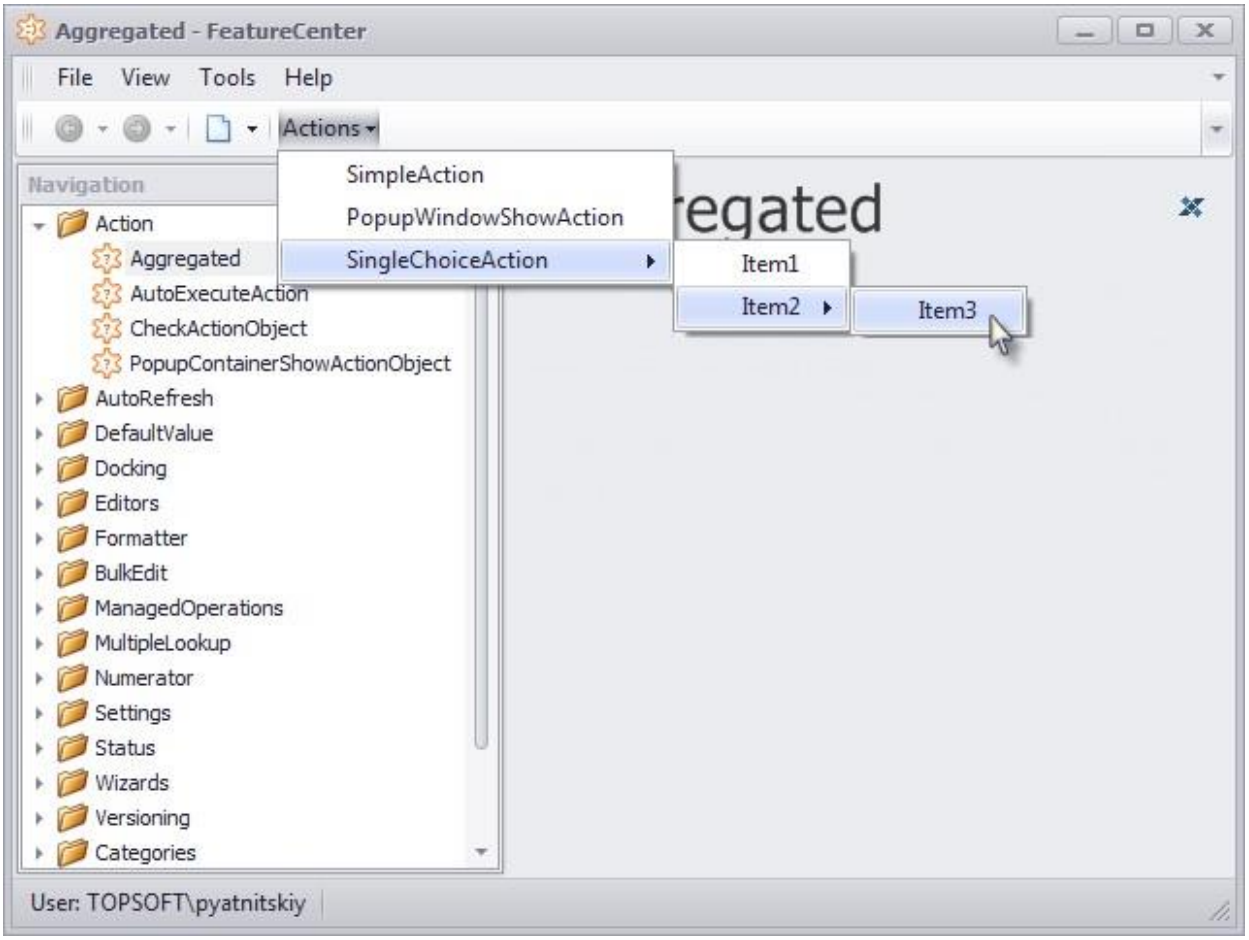
Дополнительные виды действий (Extra Actions)

Action это абстрактное понятие для описания взаимодействия конечного пользователя с интерфейсом приложения. Библиотека Xafari.dll предоставляет следующие типы Action:

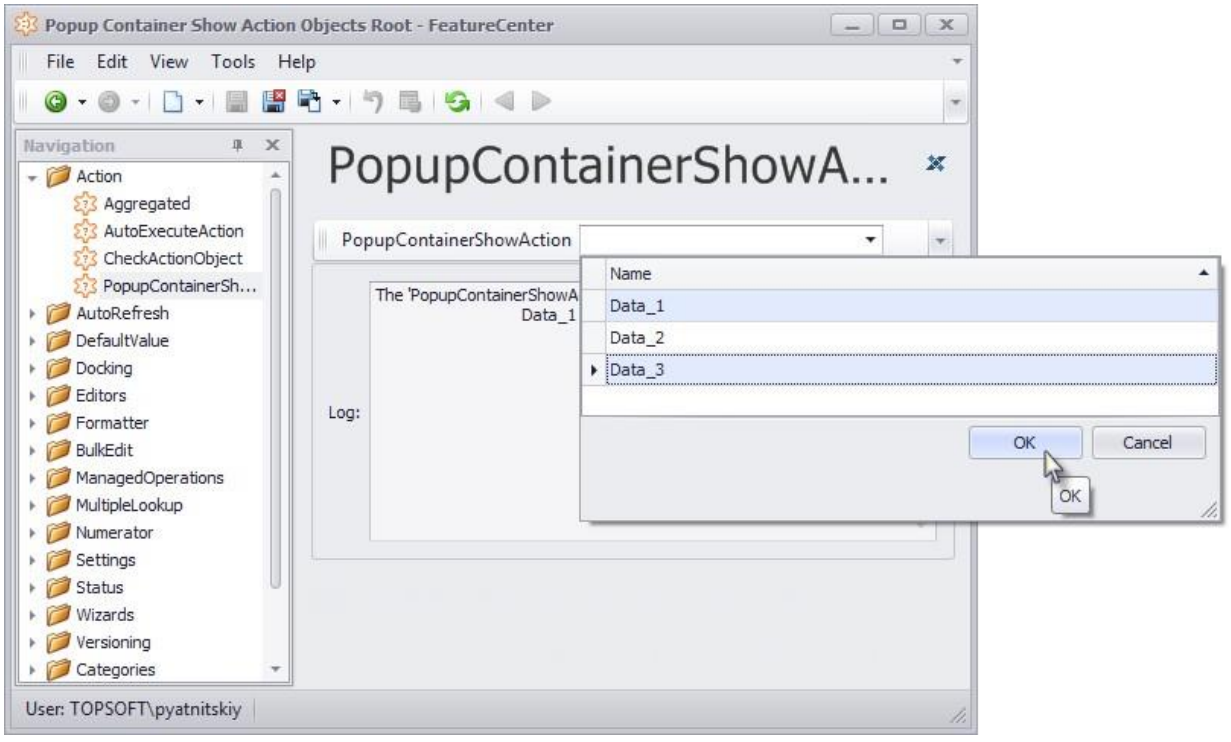
- **CheckAction** – позволяет переключать состояние действия и подписываться на событие смены состояния. Является наследником SimpleAction



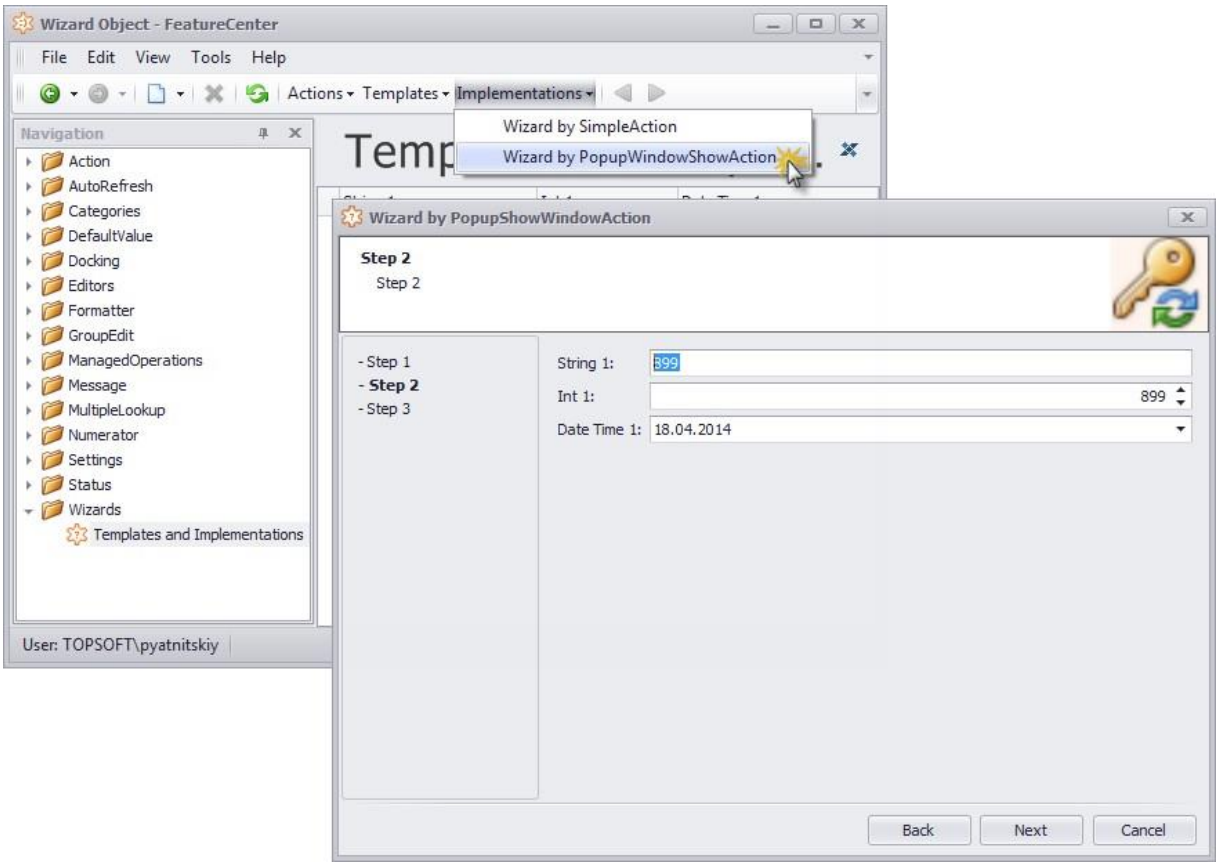
- **AggregatedAction** – является динамическим контейнером для других действий и может быть настроен в модели приложения. Является наследником SingleChoiceAction.



- **PopupContainerShowAction** – наследник PopupWindowShowAction используется для отображения PopupContainer вместо PopupWindow. Пример использования можно найти в Extra Filters.



- **WizardAction** – специальный SimpleAction, который используется для реализации Wizards



Плавающие панели (Dock Panels)

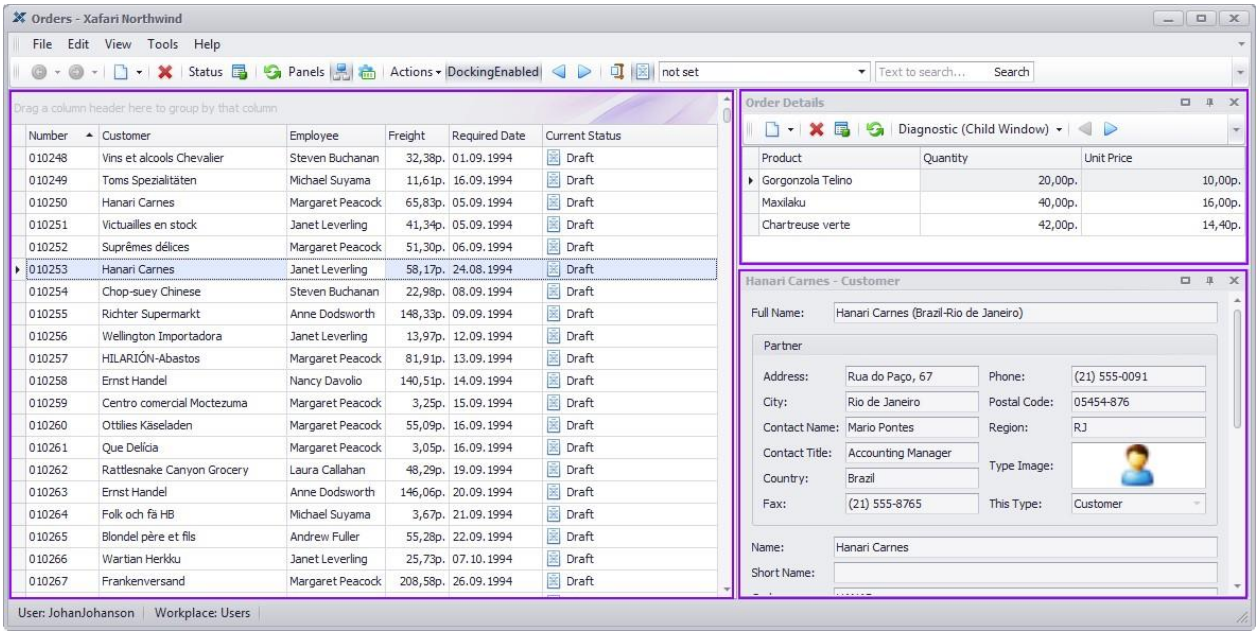
Dock Panels позволяют существенно улучшить эргономику пользовательского интерфейса. Использование панелей позволяет предоставить пользователю больше данных о текущем объекте в виде дополнительных окон. При этом сохраняется возможность того, что пользователь сам может скрыть ненужные ему окна или наоборот показать другие доступные в данный момент.

Панели могут быть использованы как способ усложнения любого View без перенастройки самого View. Такая задача часто возникает, когда новые функции реализованы в дополнительном модуле.

Для работы с панелями требуется добавить компонент DockManager в FrameTemplate. Свойства панелей и правила их отображения описываются в модели приложения (ApplicationModel).

Для панели назначается View, который в дальнейшем используется для отображения данных. Для View в панели и основного View можно задать отношение Master-Detail с помощью какого-либо критерия. В этом случае данные обоих View будут синхронизироваться, в зависимости от параметров отношения Master-Detail. По умолчанию, синхронизация производится через 0.5 секунды после изменения текущего объекта (CurrentObject) или выбранных объектов (SelectedObjects) на MasterView.

Панели использованы при реализации Категорий.

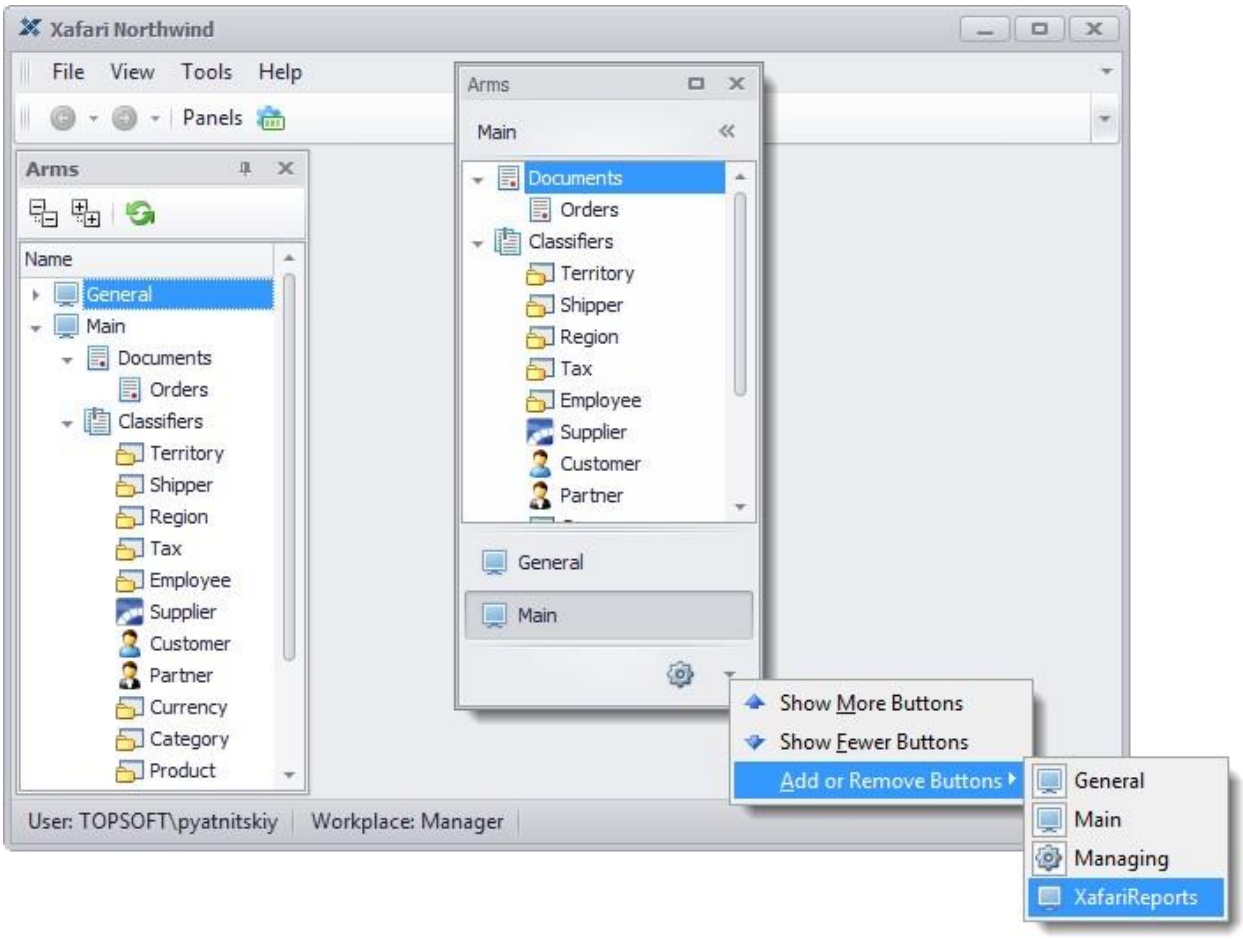


APMы (Alternative Navigation Items)

Панель навигации, легко адаптируемая под нужды конкретного приложения. Стандартное решения такой задачи в архитектуре XAF реализовано в виде Navigation Items, механизм ARMs обладает рядом дополнительных возможностей:

- описание шаблонов для групп элементов навигации.
- Выполнение действий (Action) сразу из панели навигации.
- Специализированные элементы навигации для работы с Xafari Reports
- Добавление собственных типов элементов навигации.
- Управление доступностью к элементам навигации через роли пользователей.

APMы реализованы для обеих платформ: Win, Web, Mvc.

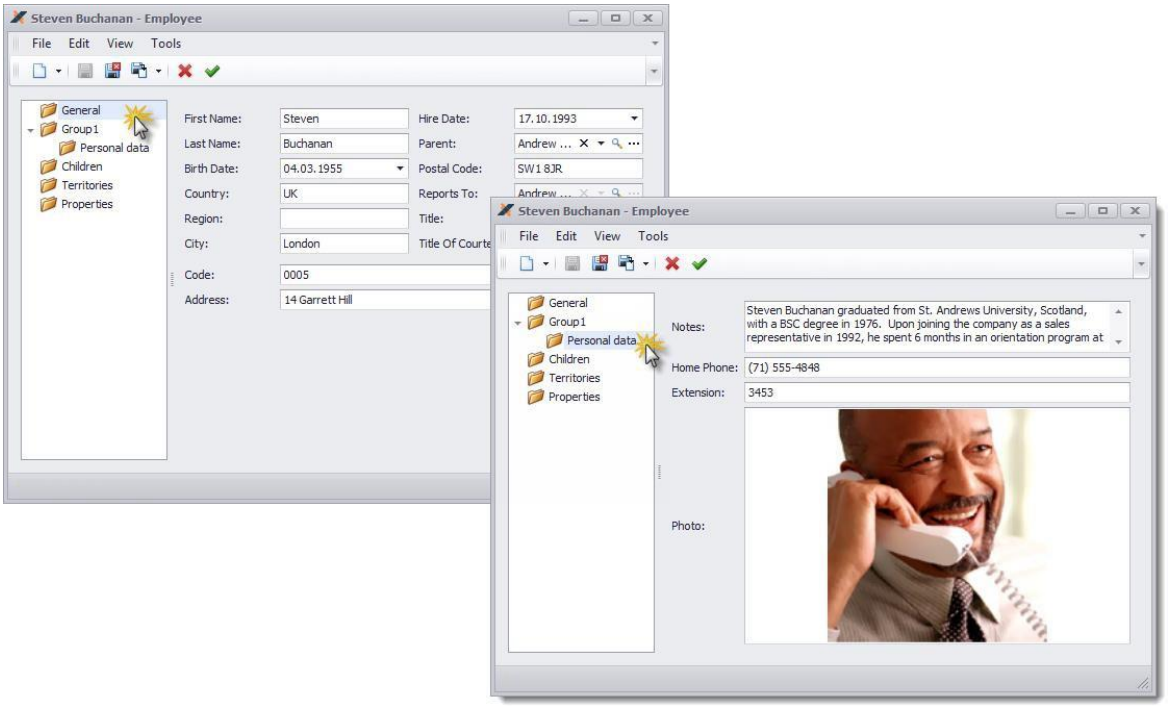


Дополнительные Редакторы Свойств

TabbedDetailPropertyEditor

TabbedDetailPropertyEditor предоставляет способ проектирования макета для Detail View и представляет собой группу иерархических вкладок для переключения между ними и NestedFrame для отображения View, связанных с этими вкладками. Этот редактор может быть использован для сложных DetailView с большим количеством ViewItem, вкладок и коллекций. В отличие от стандартного способа на TabbedLayoutGroup, редактор работает по-другому. В каждый момент времени пользователь работает с текущей вкладкой и простой View вместо одной большой и сложной. Это позволяет решить некоторые проблемы производительности, которые возникают на DetailView с большим (30+) количеством элементов.

Вся настройка для TabbedDetailPropertyEditor производится в модели приложения (ApplicationModel). Настройка заключается в том, чтобы для выбранного DetailViewItem создать иерархию вкладок и настроить простые View для каждой вкладки.

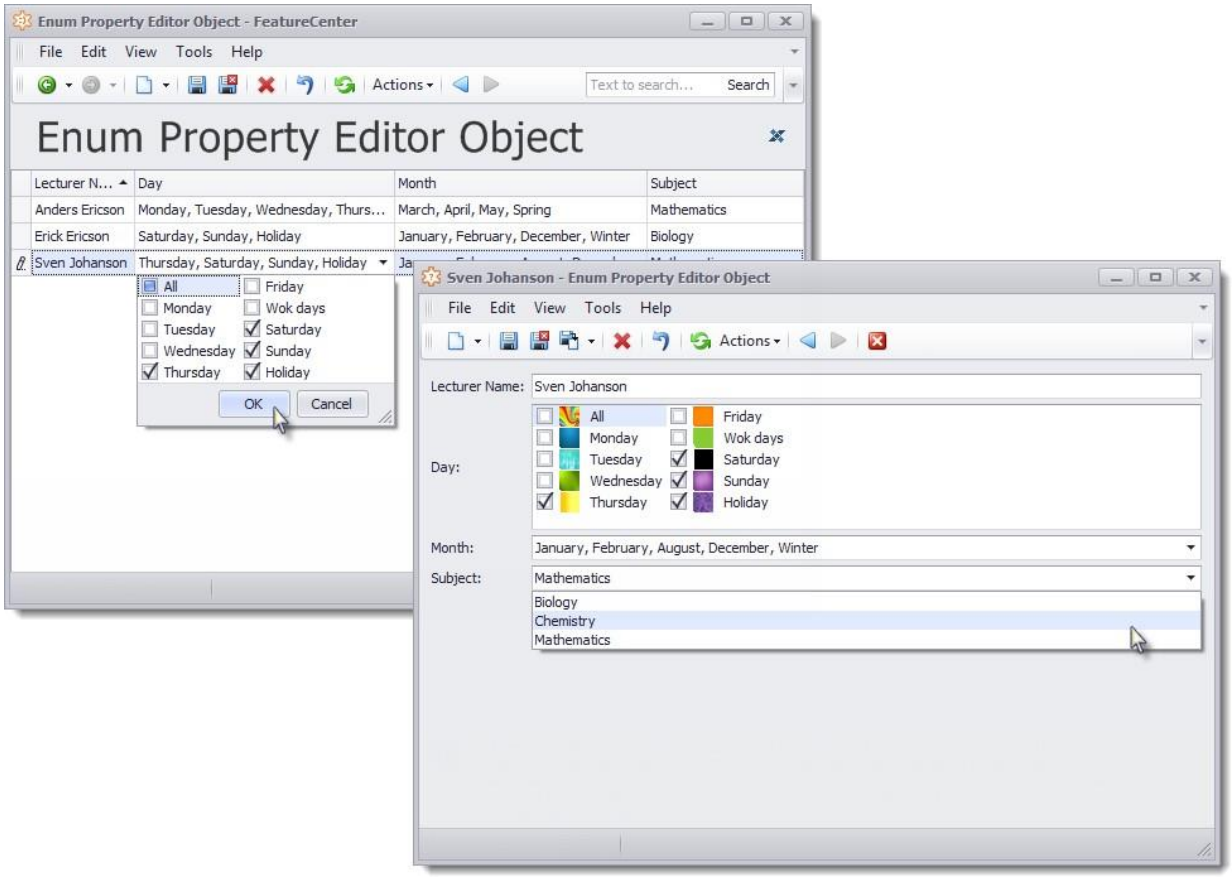


Поддерживаемые платформы: Win, Web, Mvc

EnumPropertyEditor

EnumPropertyEditor является редактором для редактирования свойств типа Enum. В дополнение к стандартному редактору этот редактор позволяет:

- Поддерживает работу с флагами
- Различные способы представления: Standard, CheckedComboBox, CheckedListBox
- Отображение в несколько колонок.

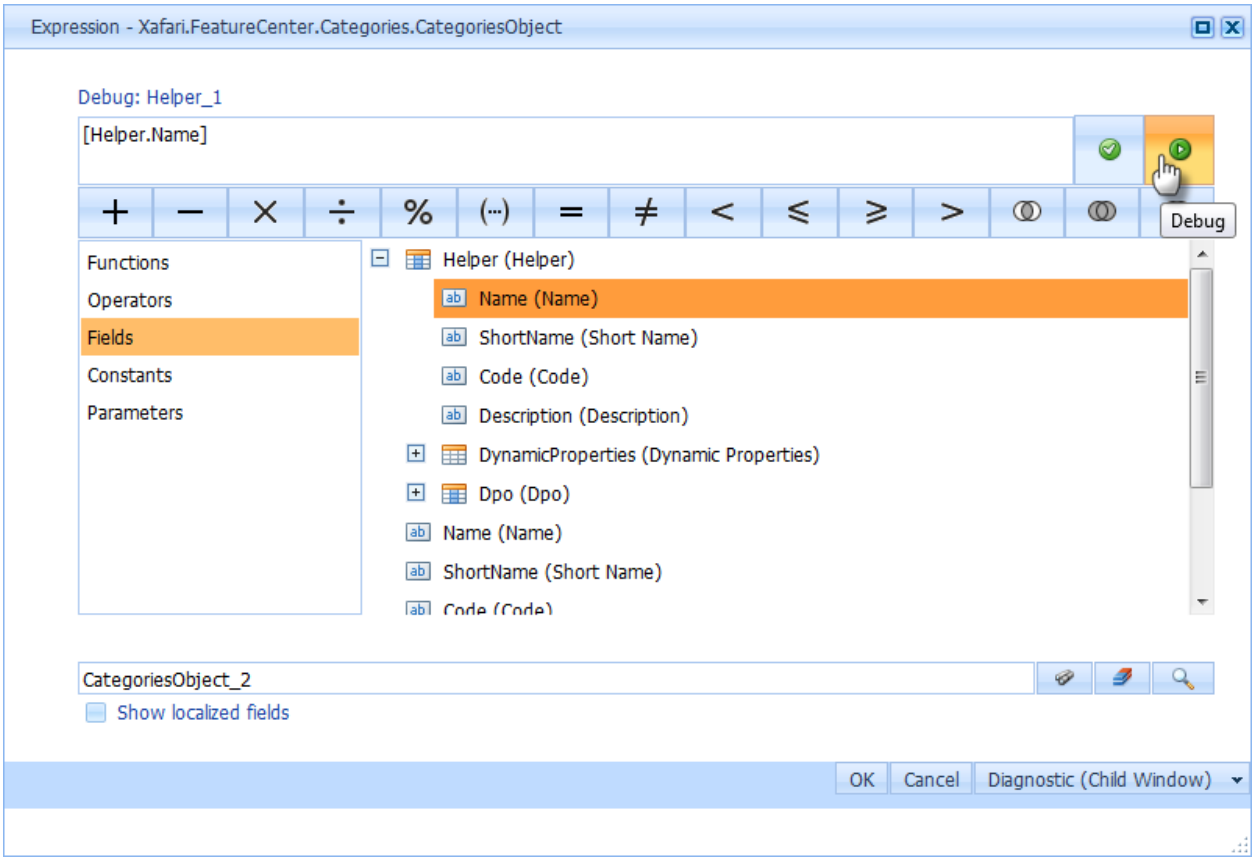


Поддерживаемые платформы: Win, Web, Mvc

ExpressionPropertyEditor

Редактор используется для редактирования свойств типа string, содержащих выражение в формате DevExpress Expression.

Для редактирования выражения используется специфический визуальный элемент.



Поддерживаемые платформы: Win, Web

FolderBrowserPropertyEditor

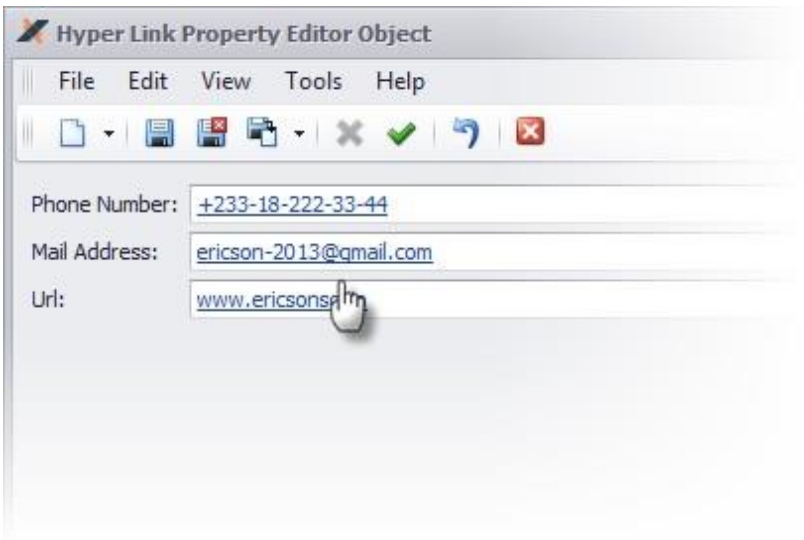
Редактор используется для редактирования свойств типа string, содержащих путь к папке.

Для выбора пути используется визуальный элемент FolderBrowserDialog.

Поддерживаемые платформы: Win

HyperLinkPropertyEditor

Редактор используется для отображения и редактирования свойств типа string, содержащих гиперссылку в формате URI.



Поддерживаемые платформы: Win, Web, Mvc

LabelPropertyEditor

Визуально редактор выглядит как StaticText, но позволяет отображать значения свойств. В отличие от DefaultPropertyEditor в редакторе использован LabelControl.

Поддерживаемые платформы: Win

ObjectSetPropertyEditor

Специализированный редактор, который используется для реализации Множественного выбора

Поддерживаемые платформы: Win

TypeImagePropertyEditor

Редактор является наследником редактора TypePropertyEditor и дополнительно отображает пиктограмму для типа, которая определена в модели приложения.

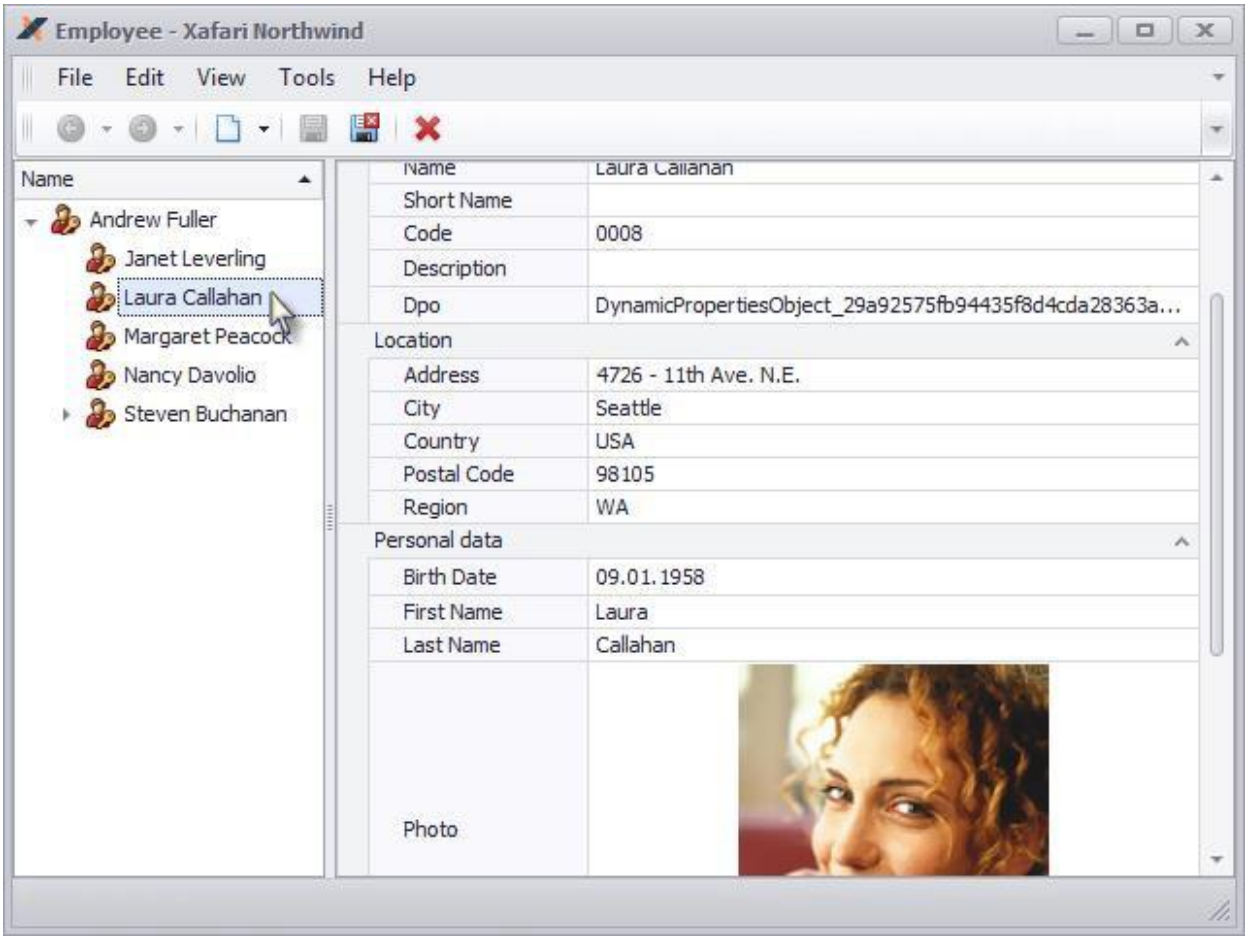
Поддерживаемые платформы: Win

VerticalGridPropertyEditor

Редактор является аналогом DetailPropertyEditor и используется для редактирования ссылочного свойства. Для реализации редактора используется визуальный элемент VGridControl и другие.

Благодаря тому, что свойства представлены в виде списка, то добавление новых свойств или исчезновение существующих происходит естественным способом без нарушения общей композиции дизайна формы. Поэтому VerticalGridPropertyEditor использован в Динамических Свойствах, Групповом Редактировании , Анастройках Приложения.

Настройка редактора производится в модели приложения. Для этого необходимо назначить некоторый DetailView. В дальнейшем список ViewItem и их параметры берутся именно из этого DetailView.

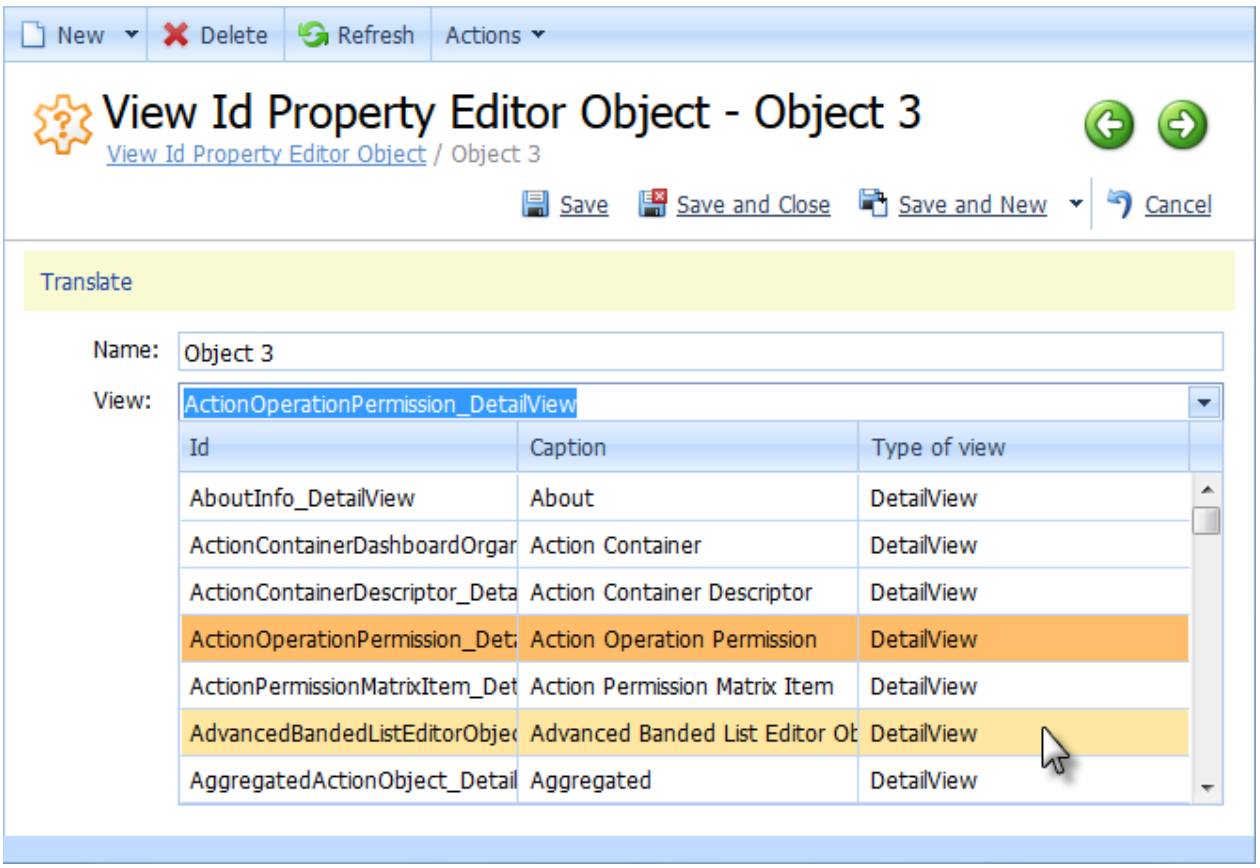


Поддерживаемые платформы: Win, Web, Mvc

ViewIdPropertyEditor

Редактор используется редактирования свойств типа string, в которых содержаться идентификаторы View из модели приложения.

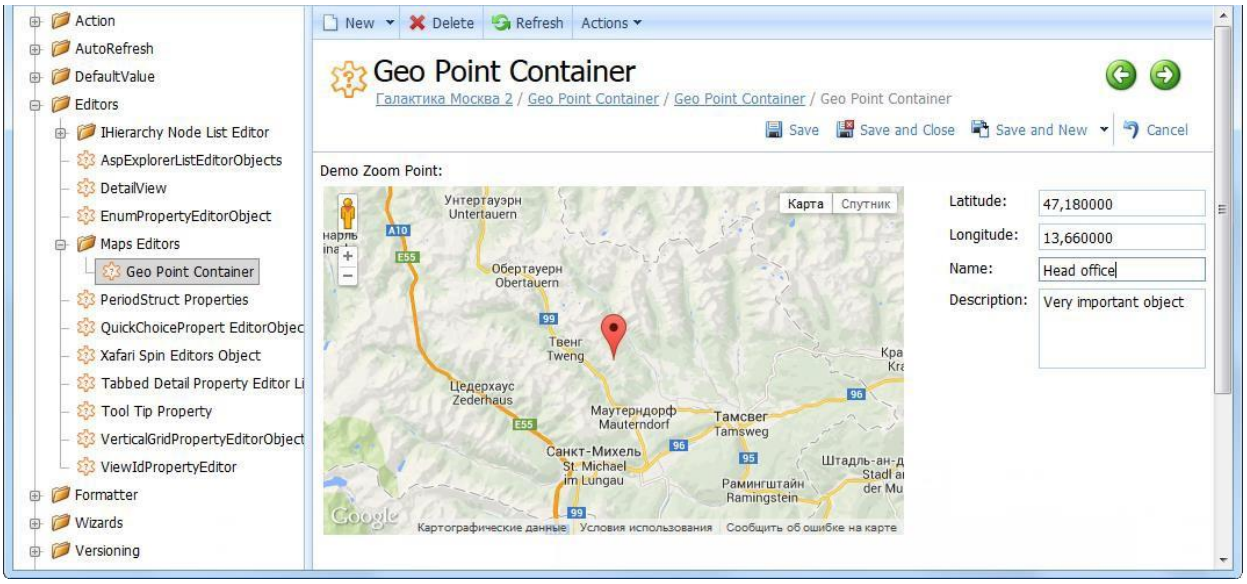
Редактор позволяет выбирать View отдельного вида (ListView, DetailView, DashboardView).



Поддерживаемые платформы: Win, Web

GoogleMapsPropertyEditor

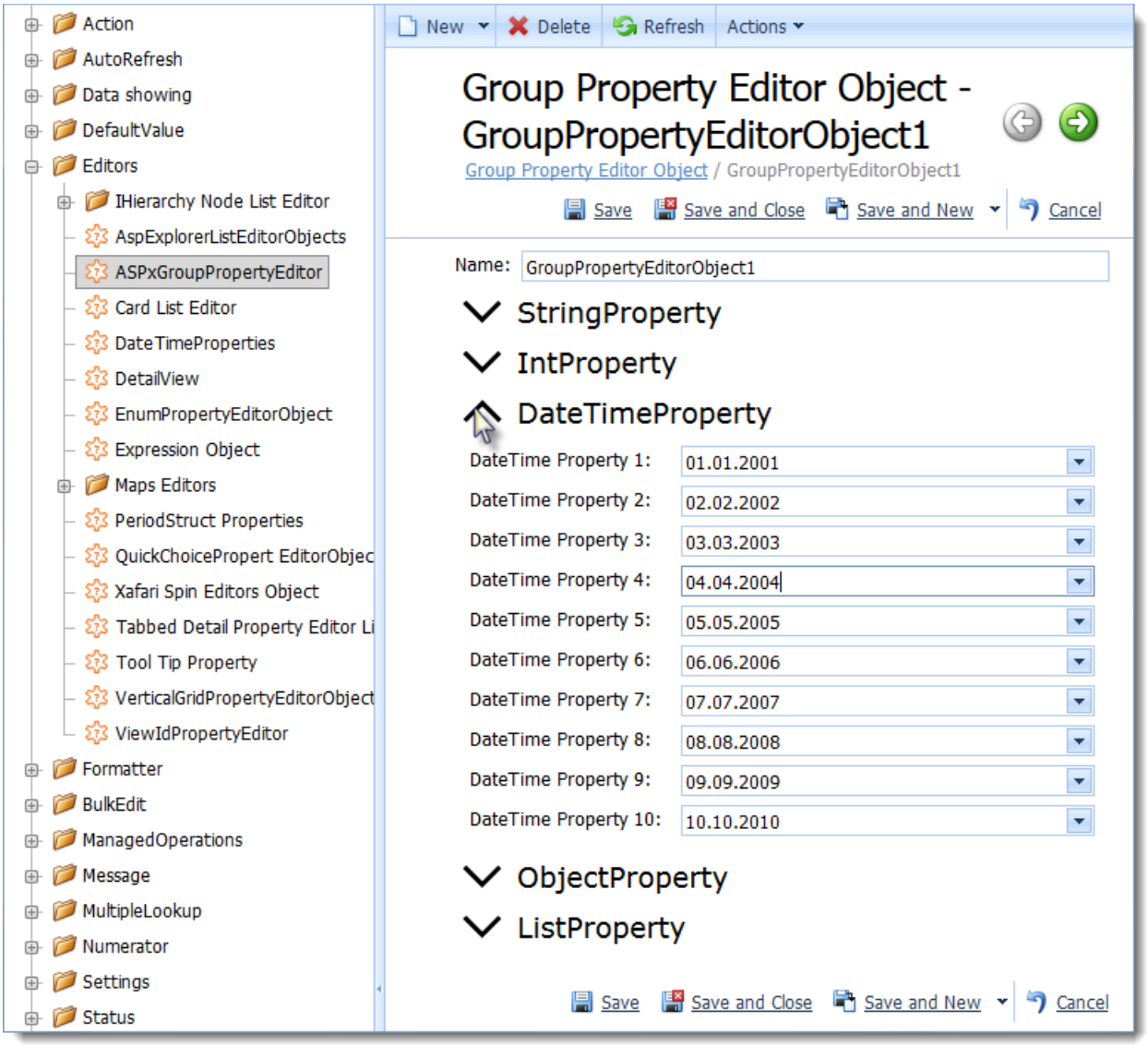
Редактор позволяет редактировать глобальные координаты при помощи позиционировании на карте Google.



Поддерживаемые платформы: Web

GroupPropertyEditor

Редактор позволяет редактировать группу полей на карточной форме в то время как весь DetailView работает в режиме ViewMode. Для настройки поведения редактора используется дополнительный DetailView.



Поддерживаемые платформы: Web

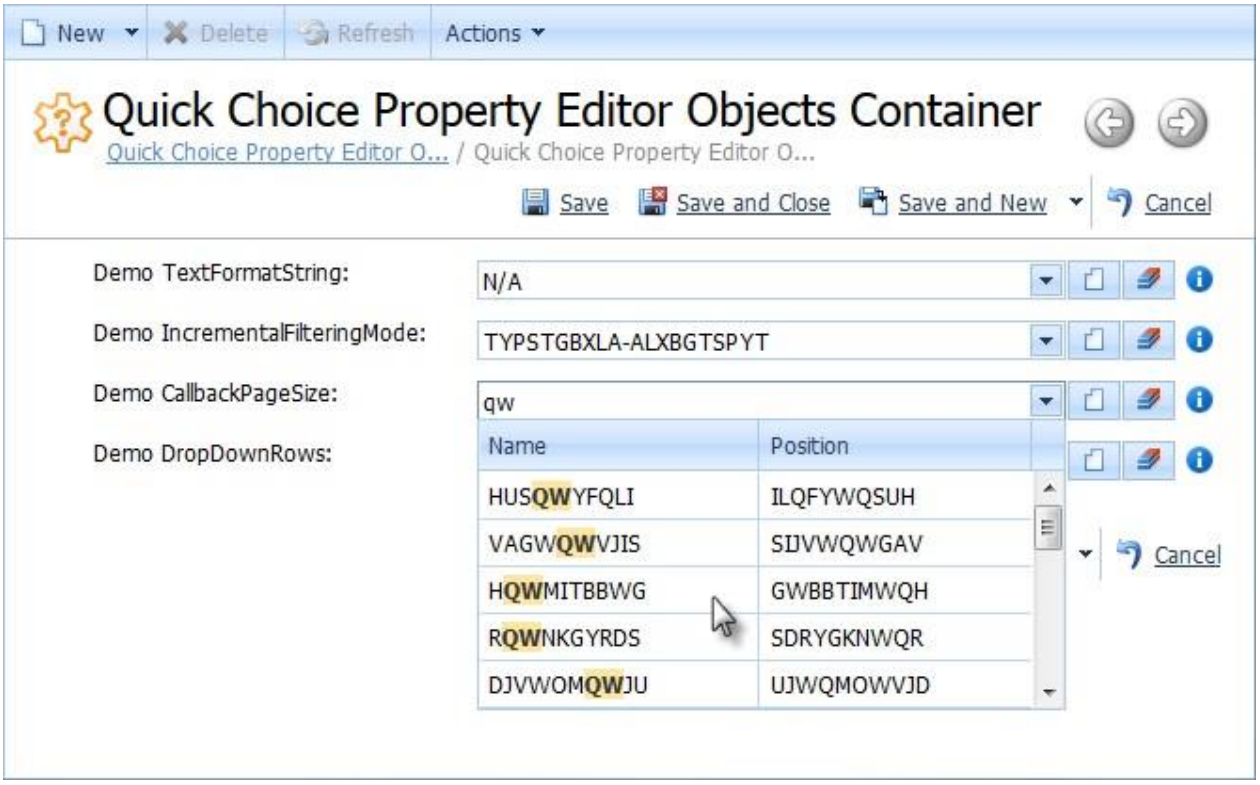
ProgressBarPropertyEditor

Редактор используется для отображения хода исполнения какого-либо процесса и отображает текущее заданное в процентах значение.

Поддерживаемые платформы: Win, Web.

QuickChoicePropertyEditor

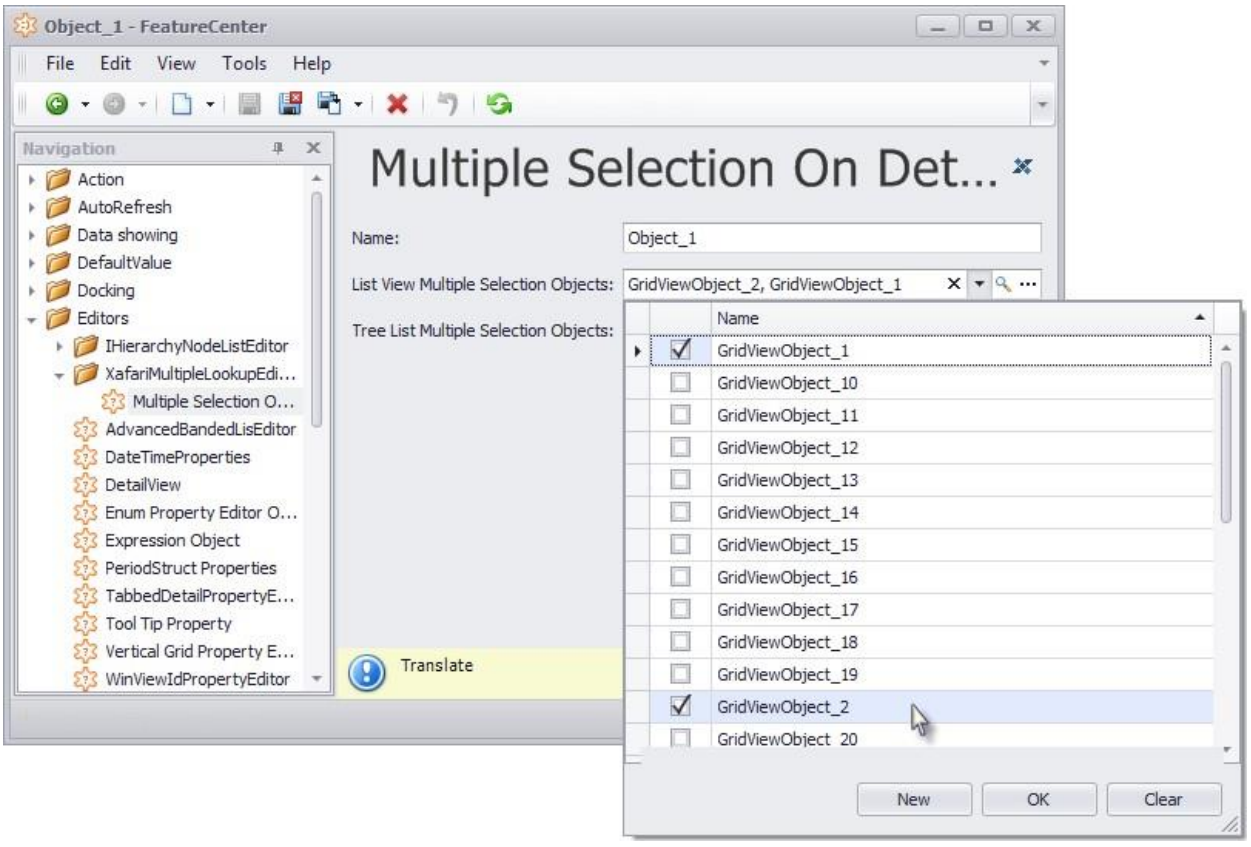
Редактор является альтернативой для стандартного LookupPropertyEditor и используется для быстрого поиска по ключевым полям без отображения дополнительных popup окон. Для реализации редактора использован стандартный визуальный элемент ASPxComboBox.



Поддерживаемые платформы: Web.

XafariMultipleLookupEditor

Редактор используется для редактирования свойств типа IEnumerable и является альтернативой для стандартного редактора ListPropertyEditor в тех случаях, когда список является хранилищем для множества выбранных объектов.



Поддерживаемые платформы: Win.

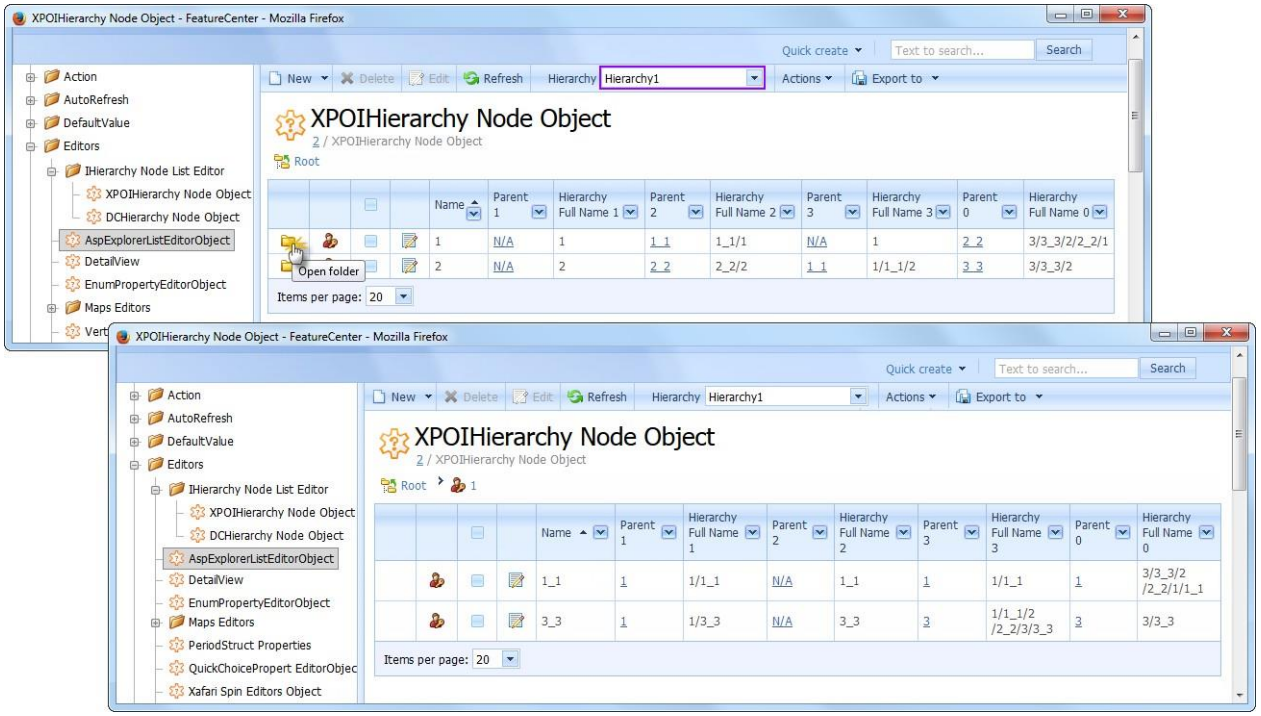
Дополнительные ListEditors

ExplorerLisEditor

Специальный списковый редактор для отображения иерархических данных ITreeNode и IHierarchyNode. Этот редактор является наследником от стандартного GridListEditor, в работе редактора используется дополнительная информация о том, как реализовано отношение Parent-Child на уровне БД. На основании этих данных в CollectionSource добавляется дополнительный критерий. Такой подход позволяет выполнять эффективные запросы для получения данных.

Работа с иерархией организована через дополнительную панель навигации выше списка и дополнительную колонку для идентификации наличия вложенных узлов.

Внешне этот редактор напоминает проводник в ОС Windows (поэтому такое название).



Поддерживаемые платформы: Web.

XafariGridListEditor

Этот редактор предоставляет функционал редактора иерархических структур в табличном виде (TreeListEditor). Для платформы Win добавлена поддержка Множественного выбора. Для платформы Web добавлена поддержка Редактируемых Шаблонов Редакторов Web.

Поддерживаемые платформы: Win, Web.

XafariTreeListEditor

Этот редактор предоставляет функционал редактора иерархических структур в виде дерева (TreeListEditor). Для платформы Win добавлена поддержка Множественного выбора. Для платформы Web добавлена поддержка Редактируемых Шаблонов Редакторов Web.

Поддерживаемые платформы: Win, Web.

CardListEditor

Это специальный редактор списков, который отображает коллекцию объектов не построчно, а в виде матрицы. При этом способ отображения объекта определяется специальным

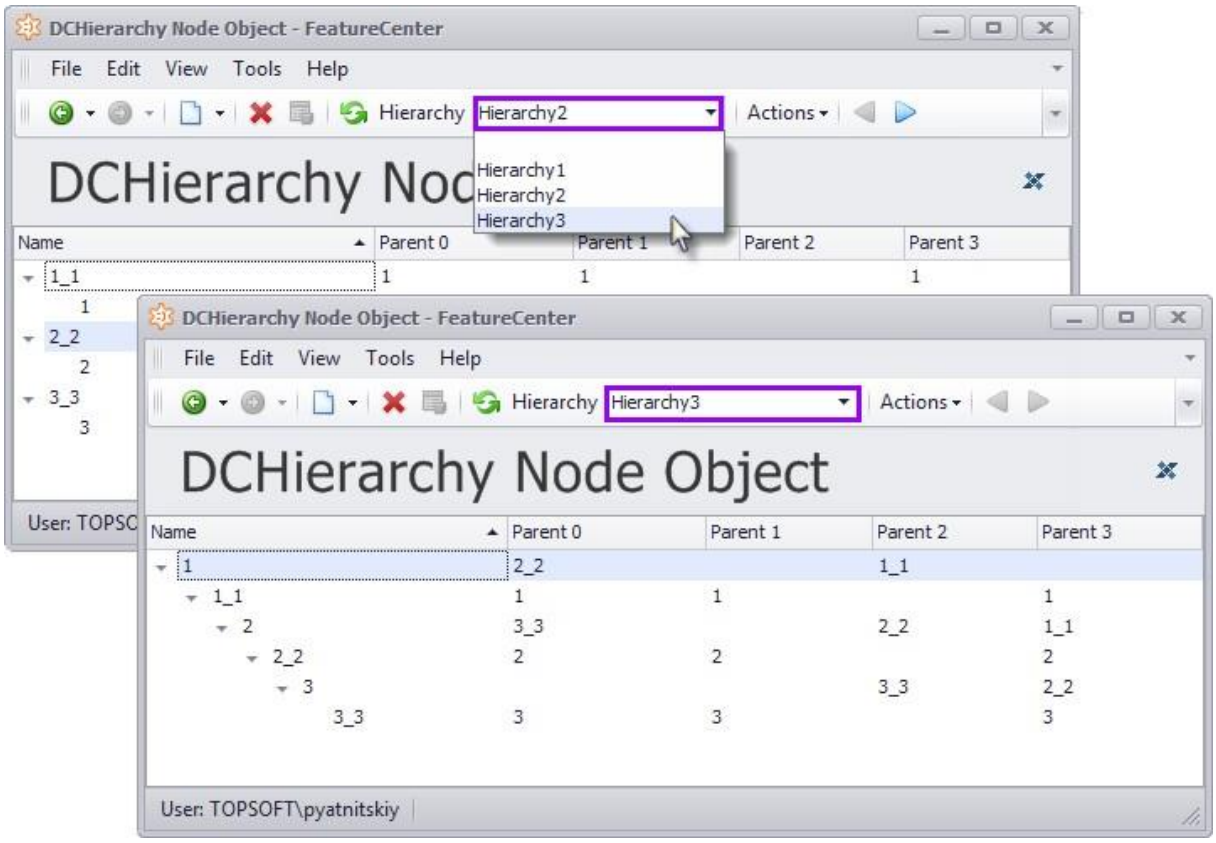
дополнительным DetailView, который может быть настроен стандартным способом в Редакторе Модели.

Поддерживаемые платформы: Web.

HierarchyNodeListEditor

Этот редактор предоставляет функционал редактора иерархических структур, он служит для отображения иерархических данных ITreeNode и IHierarchyNode. В работе редактора используется дополнительная информация о том, как реализовано отношение Parent-Child на уровне БД. На основании этих данных в CollectionSource добавляется дополнительный критерий. Такой подход позволяет выполнять эффективные запросы для получения данных, в то время как стандартный TreeLisEditor всегда загружает абсолютно все данные.

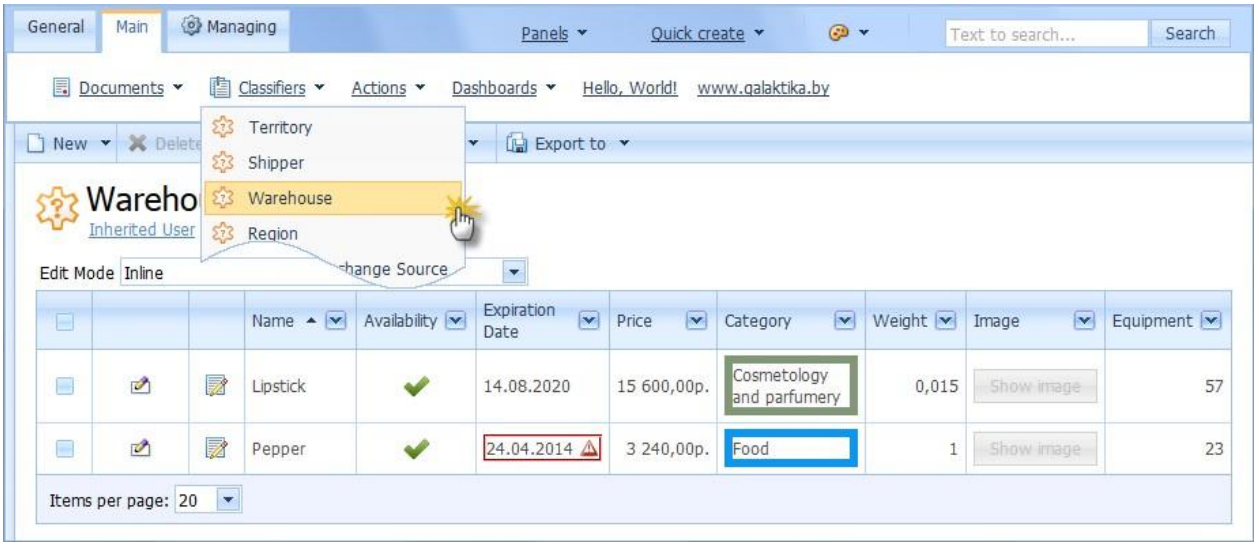
В дополнение к стандартным возможностям этот редактор допускает редактирование структуры иерархических данных при помощи drag'n'drop.



Поддерживаемые платформы: Win, Web.

Редактируемые Шаблоны Редакторов для Web (Templated Web Editors)

Эта технология позволяет изменить оформление практически любого стандартного редактора для Web интерфейса. По умолчанию редактор генерируется стандартным образом, чтобы изменить внешний вид редактора необходимо создать шаблон в виде UserControl и указать ссылку на этот шаблон в Модели Приложения.



Ниже 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

Мастера (Wizards)

Мастера (Wizards) предназначены для последовательного выполнения сложных операций. Наиболее ярким примером использования мастеров является процедура инсталяции. Мастера реализуют аналогичный функционал с учетом некоторых особенностей разработки бизнес приложений с использованием персистентных объектов.

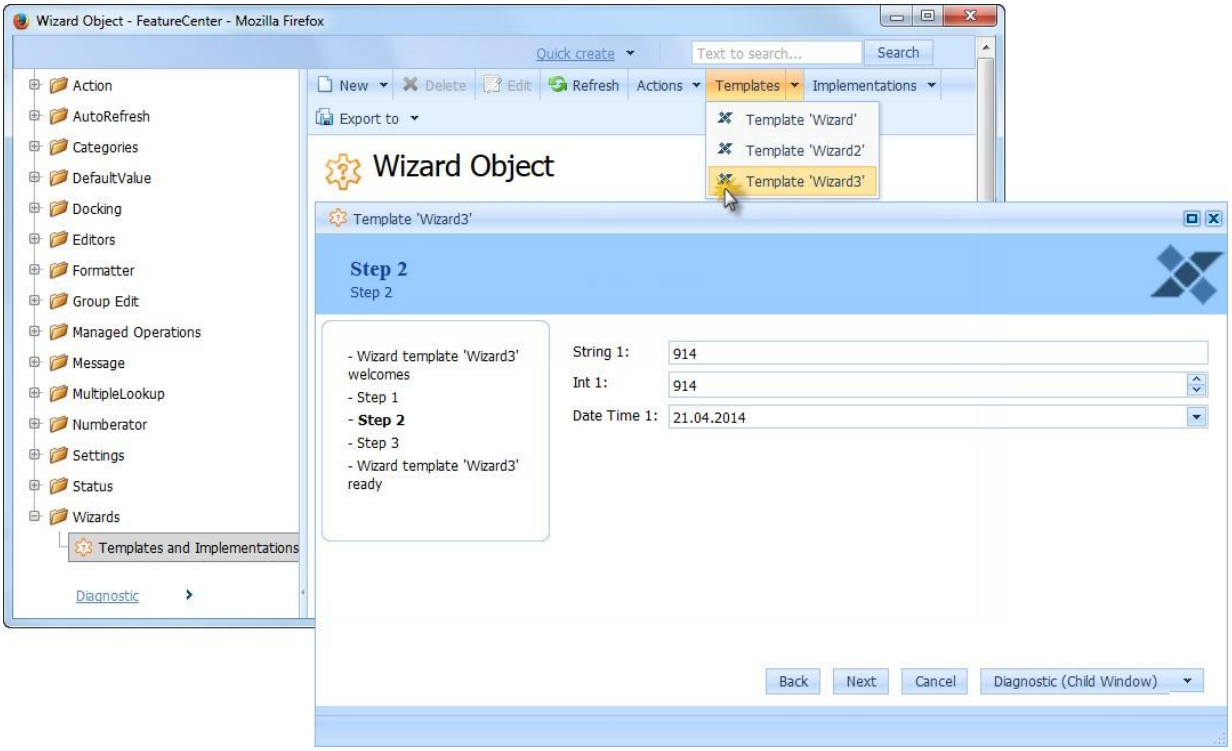
Ключевые особенности:

- Мастер - это аналог PopupWindowShowAction, которые перед выполнением действия предлагает пользователю настроить параметры этого действия в виде редактирования свойств некоторого объекта.
  Мастер выполняет аналогичную роль, за исключением того, что для редактирования объекта с параметрами действия можно сделать несколько последовательных шагов с возможностью пропустить некоторые шаги в зависимости от выполнения условий.
- Любой мастер работает в popup окне
- Шаги мастера оформляются в виде DetailView для одного и того же типа объекта.
- Работа мастера завершается выполнением какого действия или отказом от его выполнения.

Механизм мастеров обладает следующими возможностями:

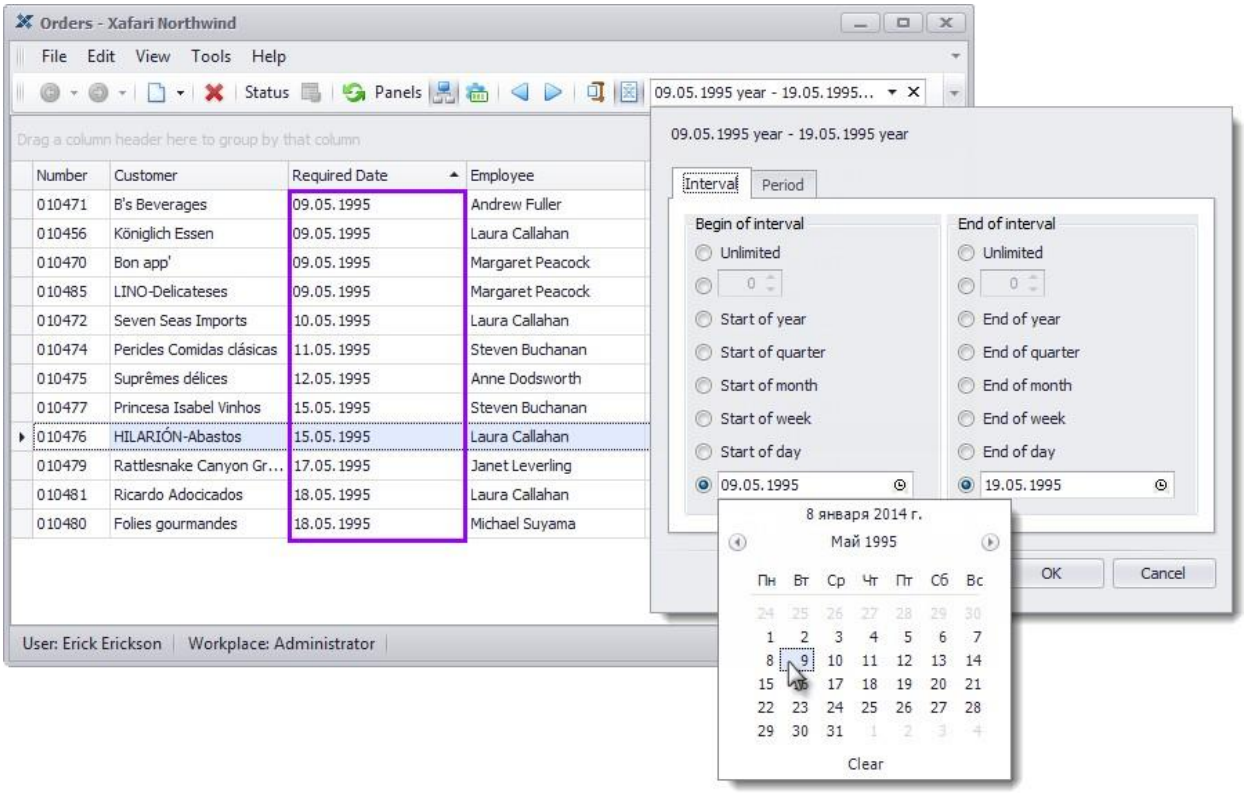
- Поддержка на обеих платформах Win и Web.
- Реализовано 3 вида шаблона представления.
- Используется новый вид действий WizardAction, наследник от ShowPopupWindowAction
- Поддерживается управление свойствами мастера в дизайнере Visual Studio, а также в модели приложения.
- Автоматическая генерация View в модели приложения для каждого этапа мастера.
- Реализована возможность условного перехода между этапами мастера.

Поддержана возможность управления этапами мастера из кода через специальные события.



Дополнительные фильтры (Extra Filters)

В Xafari реализовано общее решение для фильтрации данных по периоду. Для выбора периода используется удобная форма, позволяющая быстро сформировать период по месяцу, кварталу, году, а также произвольный период.



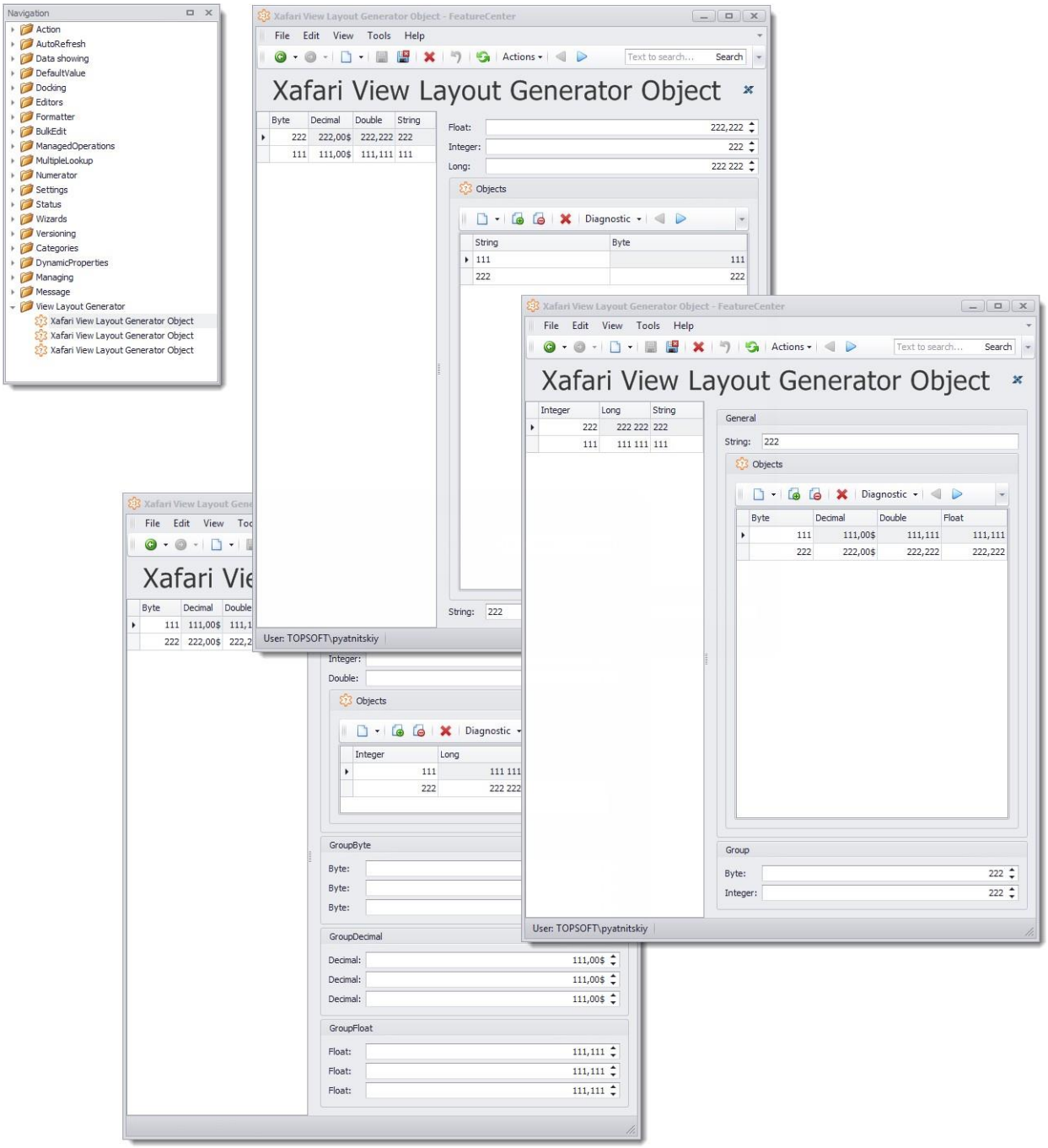
Smart Design

XafariViewLayoutStrategy – это реализация альтернативного способа генерации размещения (Layout) для ListView и DetailView. В основе данного подходя лежит идея, что при генерации представления для бизнес-объекта с большим количеством свойств, не следует все свойства этого объекта размещать на View по умолчанию.

XafariViewLayoutStrategy позволяет следующее:

- Описывать View и его размещение (Layout) в коде.
- Генерировать View в дополнение к стандартным.
- Задавать стратегию по умолчанию на различных уровнях.
- Реализовать и использовать собственную стратегию.

XafariViewLayoutStrategy может использоваться на платформах Win, Web, Mvc.



Контекстная справка (Context Help)

В состав Xafari входит механизм вызова контекстной справки, размещенной на сайте, из View. Все настройки делаются в модели приложения.

Реализованы несколько уровней настроек для указания ссылки на страницу:

- необходимо указать сайт с документацией
- страница по умолчанию для всех View
- шаблон в формате ObjectFormatter для автоматической генерации относительной ссылки на страницу справки
- можно явно задать страницу для View

Контекстная справка может использоваться на платформах Win, Web, Mvc.

Авто обновление данных

Активация для View функциональности автоматического обновления позволяет автоматически обновлять отображаемые данные через определенный интервал времени.

Действие по таймеру

В ряде приложений бизнес-логика может требовать, чтобы Action выполнялся автоматически, по таймеру. WinAutoExecuteController платформы Xafari реализует такую функцию. Этот контроллер позволяет настроить таймер, установить обратный отсчет времени, задать Action для запуска и т.д.

Бизнес Компоненты

Иерархические данные (IHiearchyNode)

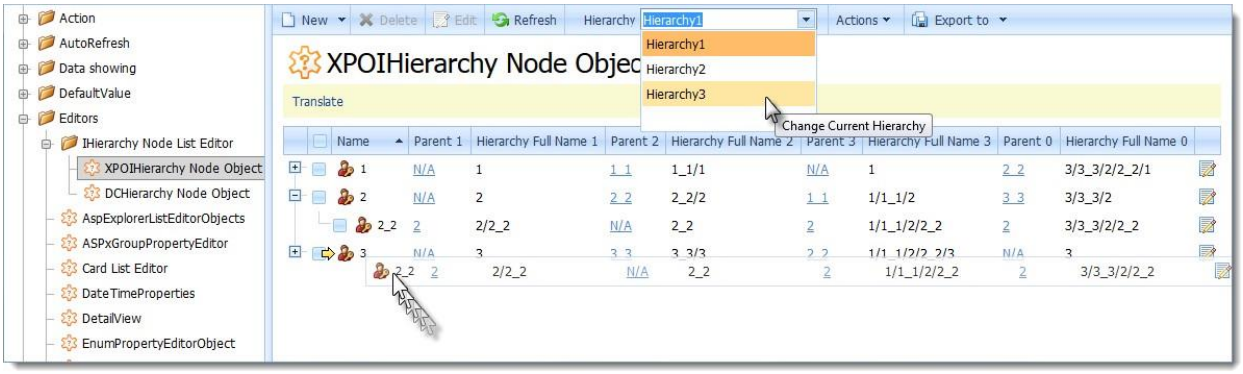
IHierarchyNode это специальный тип для эффективного описания иерархических данных. Данные, реализующие данный интерфейс, поддержаны в системе специальными редакторами и сервисами, которые обеспечевают высокопроизводительную и наглядную обработку информации. IHierarchyNode специально реализован таким образом, чтобы максимально эффективно обрабатывать Parent-Children отношение.

IHierarchyNode позволяет реализовать несколько иерархий, а также переключать представление иерархических данных на ListView.

Также имеется возможность управлять стратегией удаления подчиненных (children) узлов. Можно определить, чтобы при удалении узла происходило следующее:

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

Для работы с IHierarchyNode созданы специальные редакторы (ListEditor) ASPxXafariTreeListEditor, ASPxHierarchyNodeListEditor, ASPxExplorerListEditor, XafariTreeListEditor, WinHierarchyNodeListEditor



Множество выбранных объектов

В Xafari реализовано общее решение по работе с множеством (набором) выбранных объектов. Это решение позволяет избежать создание отдельных персистентных коллекций со специальными персистентными объектами для каждого типа, что приводит к неоправданному усложнению модели данных приложения.

В Xafari использовано универсальное решение с фиксированной структурой БД, не требующее добавления новых персистентных объектов. Также это решение позволяет формировать эффективные запросы к БД без дополнительного кодирования.

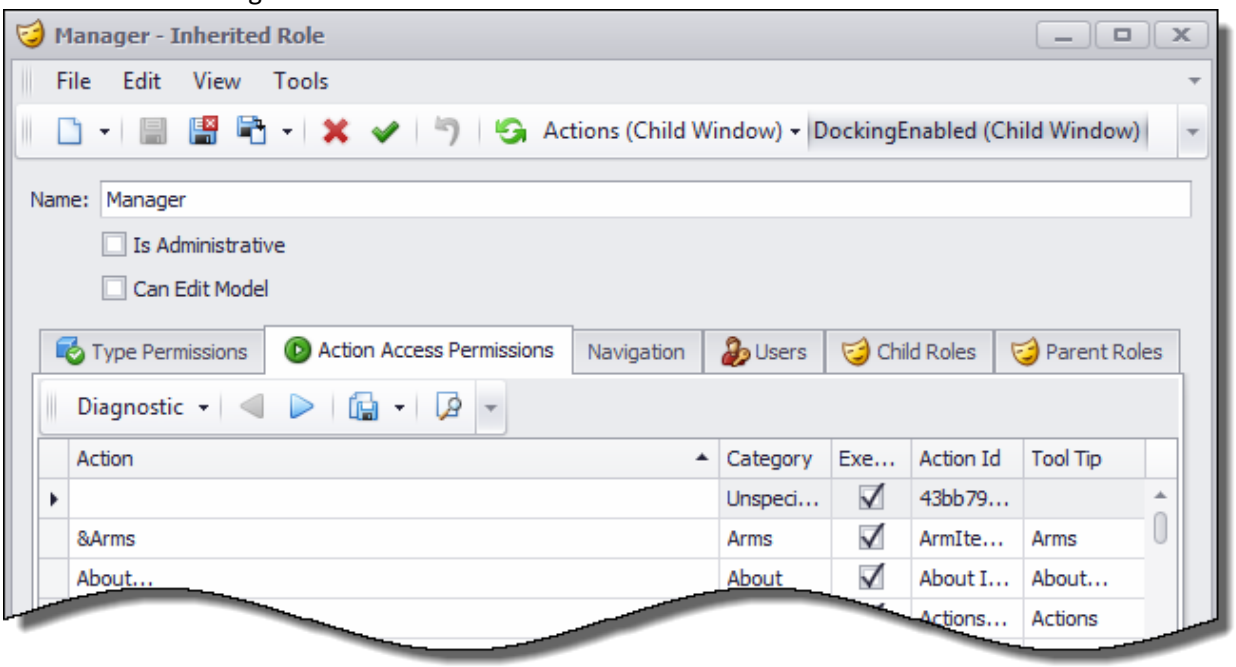
В состав этого решения вошли:

- Персистентные объекты для хранения ссылок на выбранные объекты.
- Специальный объект ObjectSet для реализации множества выбранных объектов. Допускается указание ID конкретных объектов, а также выбор по критерию.
- 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 MSSql и Oracle.

Безопасность Xafari (Xafari Security)

Модули безопасности позволяют разграничивать доступ к данным на основании прав и ролей. Следует обратить внимание на некоторые специальные возможности:

- XafariAutentication позволяет использовать такую форму аутентификации пользователя, которвая позволяет пользователю самостоятельно выбрать желаемый способ аутентификации: Windows или стандартный.
- Модель данных по технологии Domain Components (DC) полностью повторяет стандартную модель данных XPO, а также интегрирована с технологией Extensions Framework.
  Все это позволяет естественным образом подключить систему безопасности к системам, построенным по технологии Domain Components.
- Дополнительные виды прав:
  - ActionPermission позволяет управлять доступностью к Action.
  - NavigationPermission позволяет управлять доступностью к элементам NavigationItems.



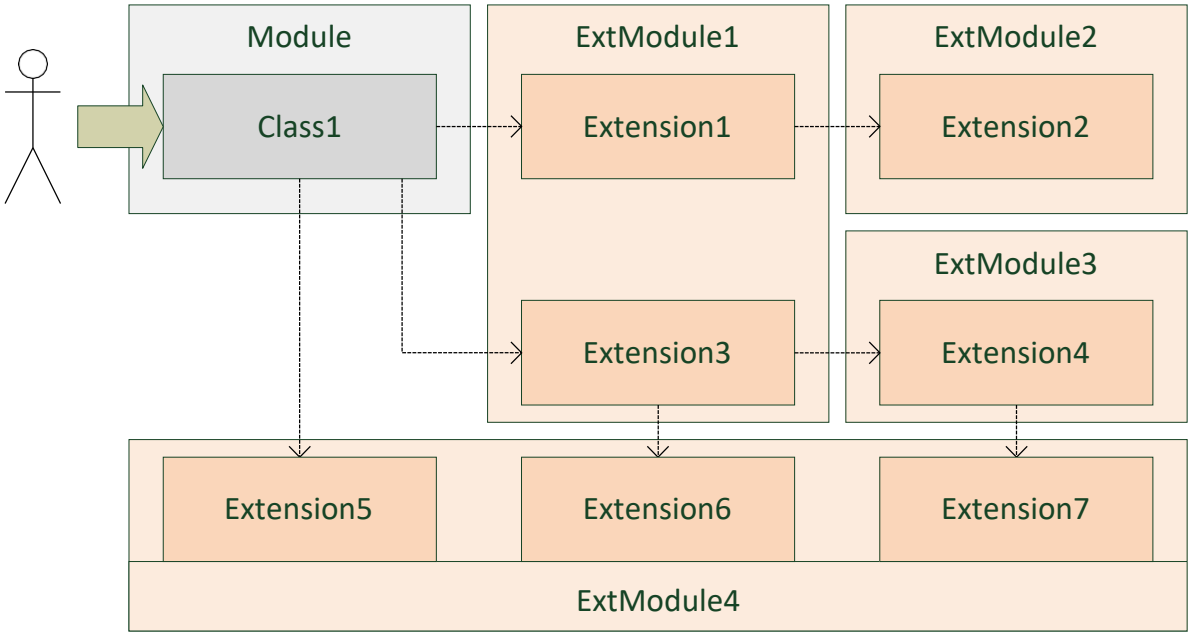
Модуль приложения (AppModule)

AppModule это определенная методика проектирования ERP системы. AppModule аккумулирует весь набор необходимых платформенно-независимых библиотек, настроек и опций. 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 путанницы и неоднозначности в случае разработки приложения для различных платформ (Web, Win, Console и пр.). 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. В случае стандартного XAF приложения, все основные модули и настройки нужно добавлять и инициализировать в каждом платформенно-зависимом модуле.

Расширение бизнес-объектов

Под eXtension Framework (XF) понимается некоторая модель проектирования и разработки сложных приложений. Она включает правила, соглашения и рекомендации по проектированию, а также библиотеку базовых классов и методов расширения компонентов XAF.



Преимущества eXtension Framework:

- Простая и понятная бизнес-модель данных.
- Нет лишних «технологических» бизнес-объектов.
- Модульная стратегия расширения модели данных.

Простая и понятная бизнес-модель данных

Базовыми являются понятия Сущность (Entity) и расширение сущности (Entity Extension). Главной идеей eXtension Framework является способ добавления новых полей в сущности приложений. Если ООП предлагает решать эту задачу через наследование расширений от сущности, то eXtension Framework решает эту задачу через добавление в сущность ассоциаций на расширения.

А использование Domain Components позволяет включать расширение целиком в объект сущности без каких-либо дополнительных конструкций.

Нет лишних «технологических» бизнес-объектов

Использование eXtension Framework позволяет избавиться от такого «неприятного» следствия от использования наследование, как появление новых типов бизнес-объектов в приложении.

В случае ООП первоначальная цель «расширение» решается через создание «новой сущности». В этом смысле новый класс является следствием применения выбранной технологии, поэтому его можно назвать «технологическим».

eXtension Framework четко различает бизнес-сущности и их расширения. Расширение является только расширением и не является сущностью, которую расширило. Поэтому и «технологические» бизнес-объекты отсутствуют в приложении.

Модульная стратегия расширения модели данных

eXtension Framework позволяет создать любую комбинацию расширений из имеющегося набора, которая может потребоваться клиенту. При этом максимально исключены конфликты совместного использования «параллельных» расширений, которые появляются при использовании наследования.

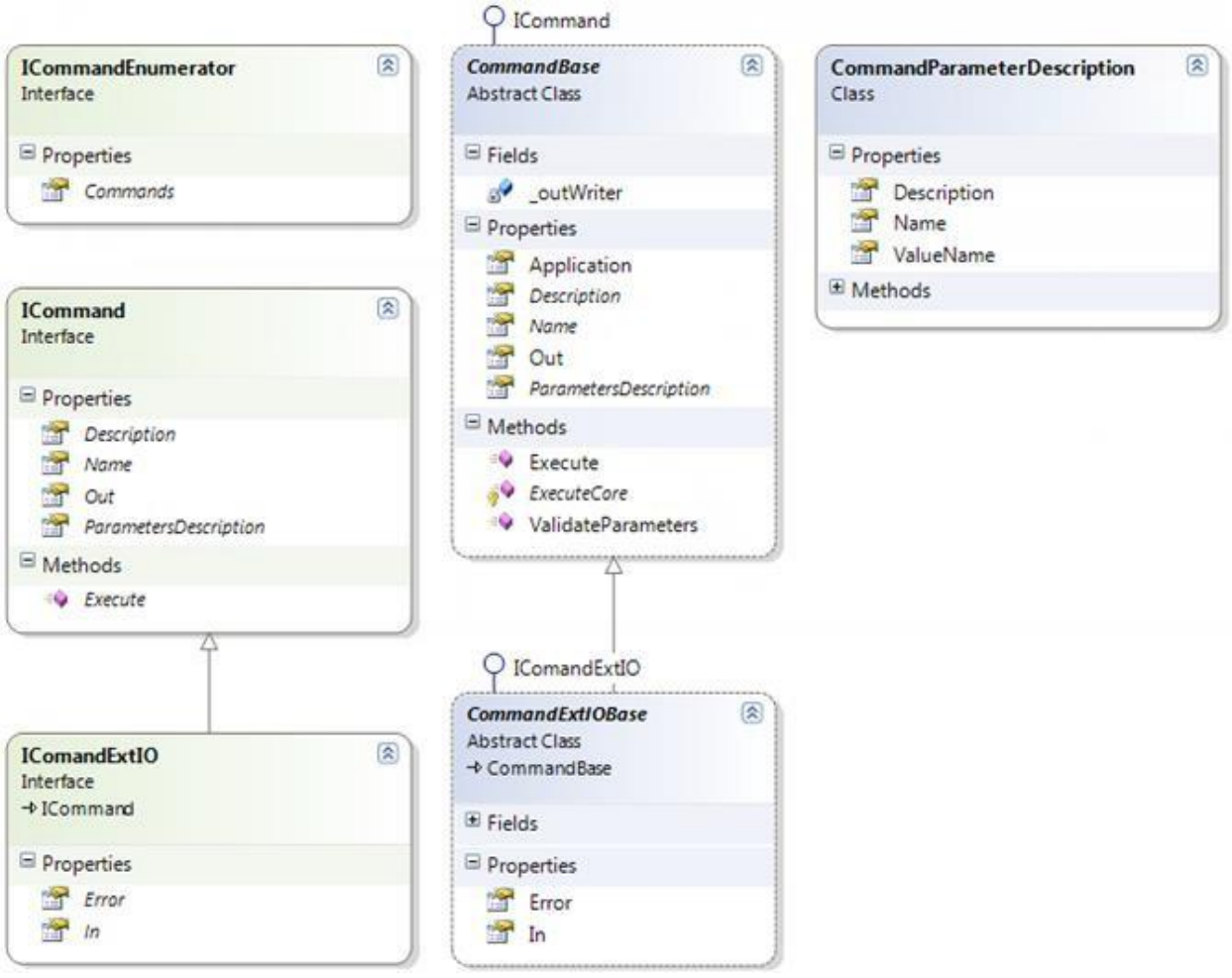
Консоль приложения (Console Application)

При эксплуатации прикладных систем существует ряд задач/функций как, например, импорт/экспорт данных, пакетное формирование отчетов, выполнение других бизнес-функций, которые необходимо выполнять в автоматическом, пакетном режиме, возможно, по расписанию. Взаимодействие с пользователем в таких случаях минимально, или отсутствует вовсе.

Для решения подобного класса задач разработано легковесное консольное приложение - **RunCmd.exe**, к которому может подключаться только требуемый для решения данной задачи набор модулей. Необходимые для решения конкретной задачи команды и их параметры передаются в командной строке.

Разработчики реализуют команды, заключая их в состав модуля. Необходимые для разработки новых команд классы и интерфейсы находятся в сборке Xafari.dll (в пространстве имен Xafari.Commands).

В составе платформы Xafari поставляются встроенные команды для запуска произвольной бизнес-операции, команда очистки БД от удаленных объектов и пр.



Контроллеры бизнес-логики

LogicController – это методика реализации бизнес-логики приложения, которая позволяет существенно сократить число контроллеров в приложении без потери функциональных возможностей.

В основе этой методики лежит класс LogicController. Он является аналогом для ViewController и может быть использован вместо ViewController.

LogicController привязывается к определенному типу бизнес-объекта и активируется только на View для этого бизнес-объекта. LogicController не является наследником от Controller, поэтому для View будут доступны только LogicController привязанные к данному типу.

Таким образом, LogicController позволяет избежать избыточности наследников класса Controller и связанных с этим проблем падения быстродействия.

Статусы объектов

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

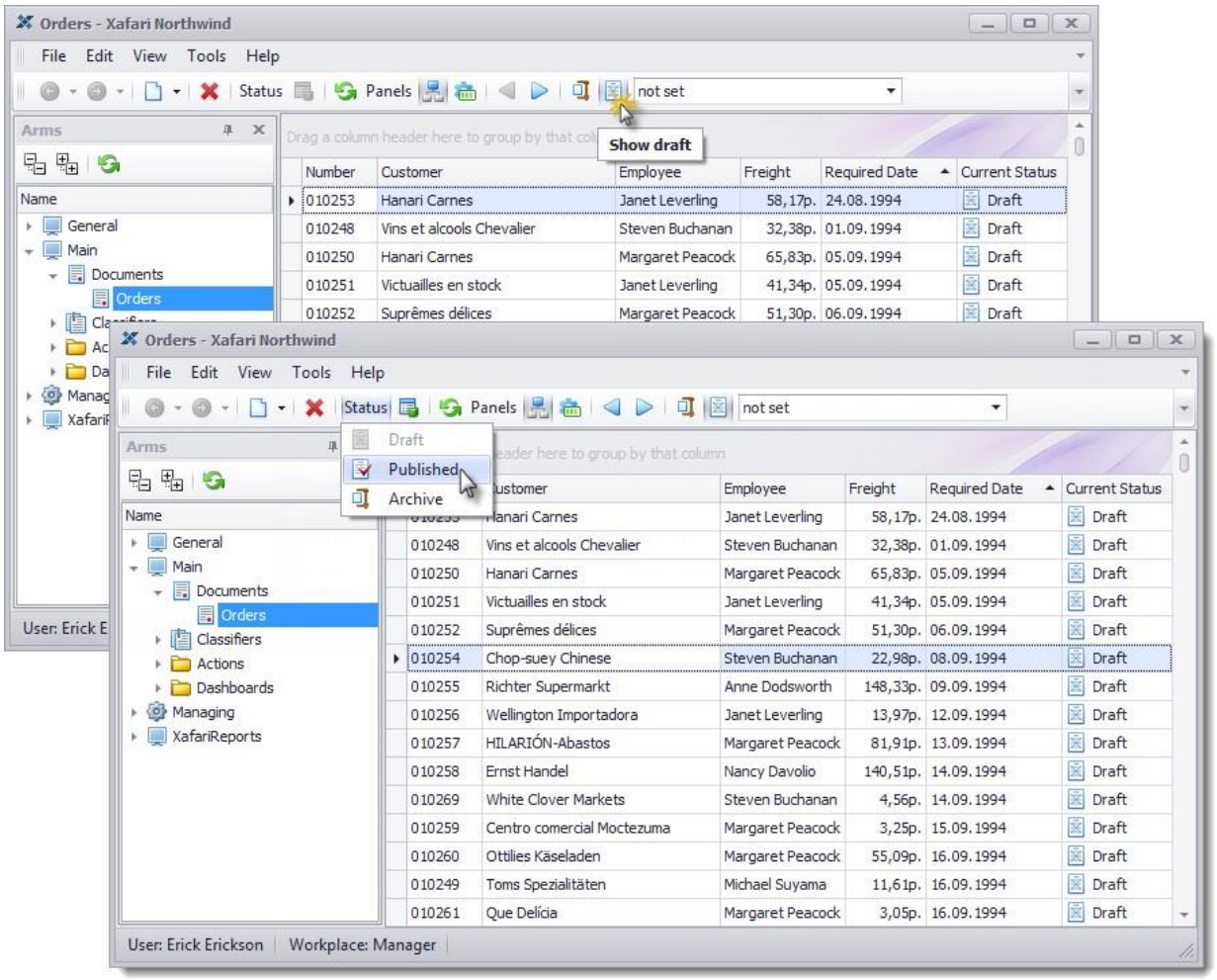
- Черновик (Draft) – объект находится в состоянии редактирования.
- Опубликован (Published) – объект отредактирован и может быть использован.
- Архив (Archive) – устаревший объект.

При отображении объектов со статусами возможности ListView расширяются несколькими **Actions:**

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

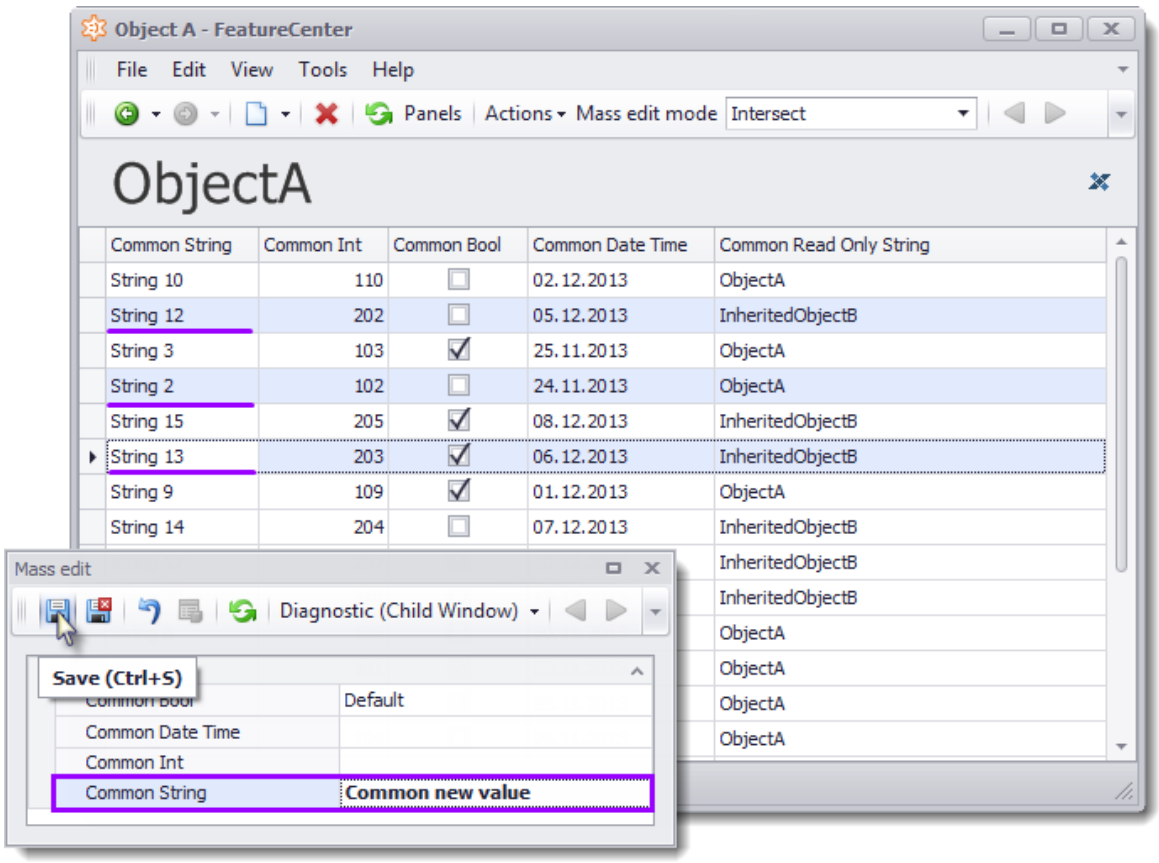
Статус объекта можно изменить также в DetailView.

Статусы объектов использованы для активных блокировок и реализованы для персистентных объектов (XPO) и domain components (DC).



Групповое редактирование (Bulk Edit)

В приложениях, разработанных с использованием платформы Xafari, имеется возможность обеспечить групповую обработку объектов на списковой форме. Такая функциональность востребована в случаях, когда свойство нескольких объектов должно содержать 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 (dock panel).



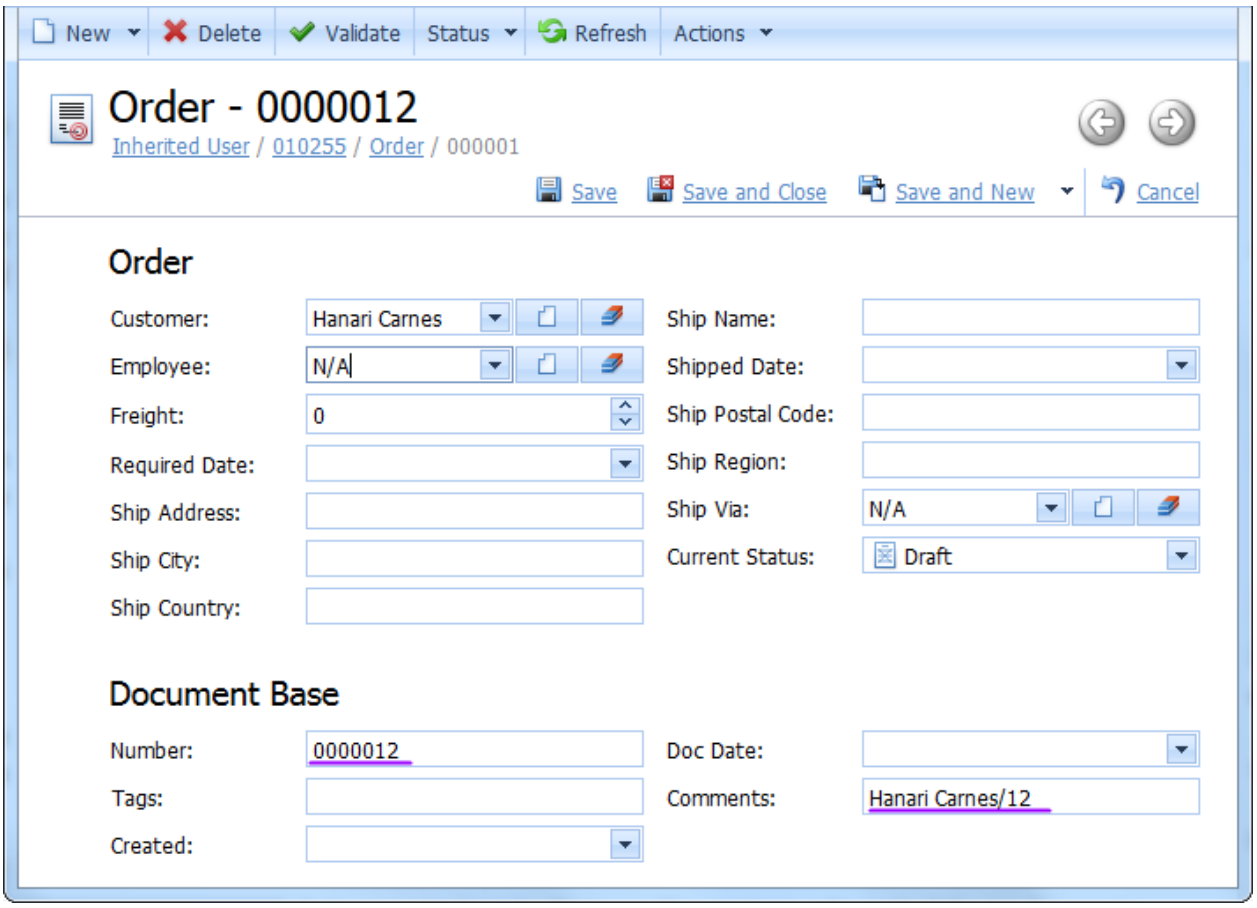
Нумераторы (Numerators)

Сервис нумераторов предназначен для проведения автоматического заполнения полей персистентных объектов с использованием порядковой нумерации. Это можно рассматривать как специфический частный случай вычисляемых свойств.

При автоматическом заполнении полей сервис нумераторов предоставляет следующие возможности:

- установка начального значения индекса (StartValue) и шага (Step)
- установка шаблона номера (Template)
- сквозная нумерация (ThroughNumeration): сквозная нумерация применяется для последовательного нумерования объектов различных типов. Например, все исходящие документы (накладные, платежки и пр.) должны использовать одну цепочку номеров
- использование удаленных номеров (UseDeleteNumbers): использование удаленных номеров 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
- поддержана пакетная обработка документов

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

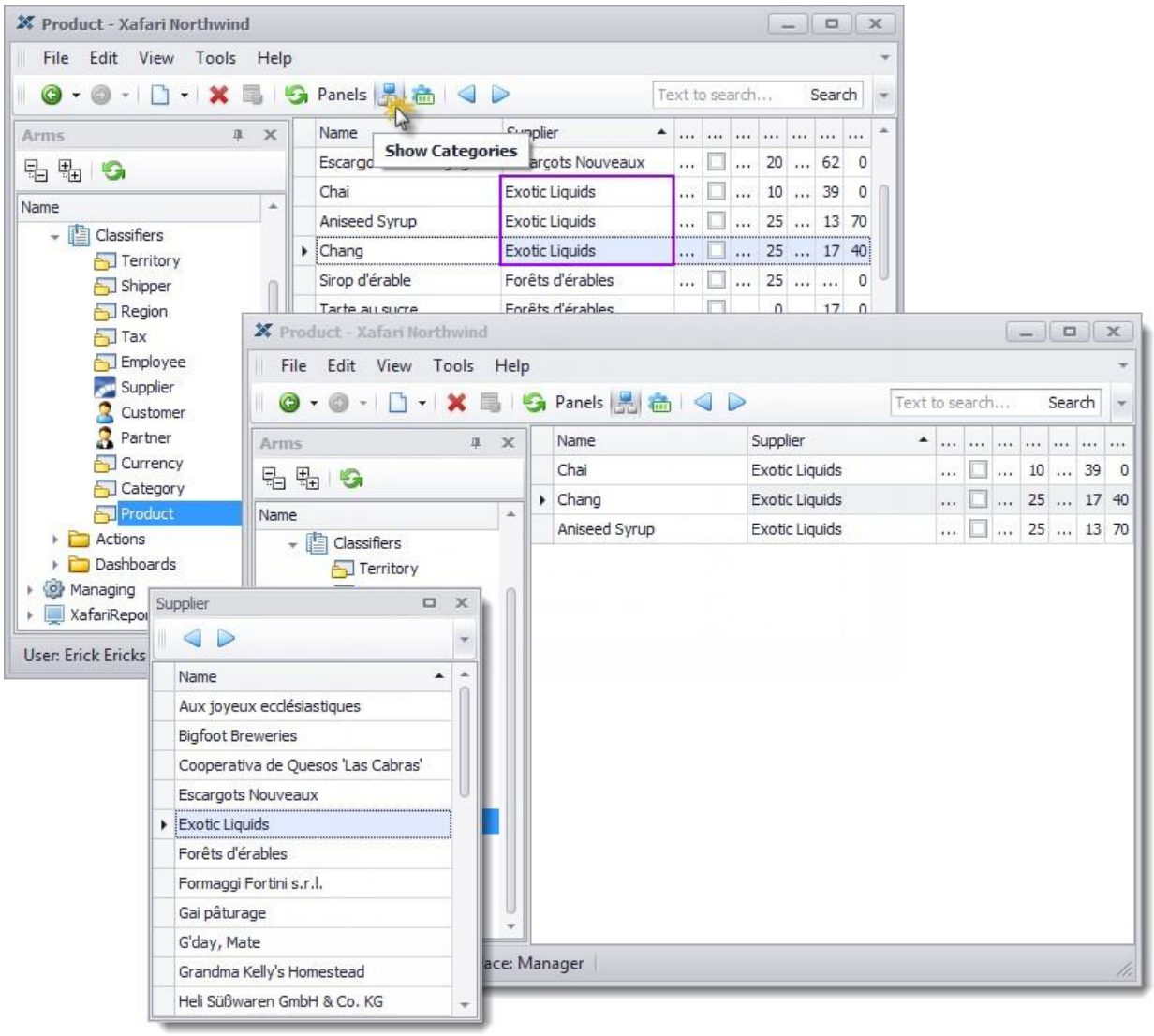


Категории (Categories)

Категории используются для внешней классификации по ссылкам, имеющимся у объектов. Категории (внешние классификации) реализованы как плавающие панели, содержащие список объектов, по которым производится классификация данных текущей списковой формы.

Настройка и подключение категорий для списковой формы производится в модели приложения. Также имеется возможность описать подключение категории прямо в коде посредством атрибута:

[BOCategory(«Customers», typeof(Customer), «Customer.Id = ‘{0:Id}'»)]
public class Order {}
Категории могут быть использованы на платформах Web, Win, Mvc.



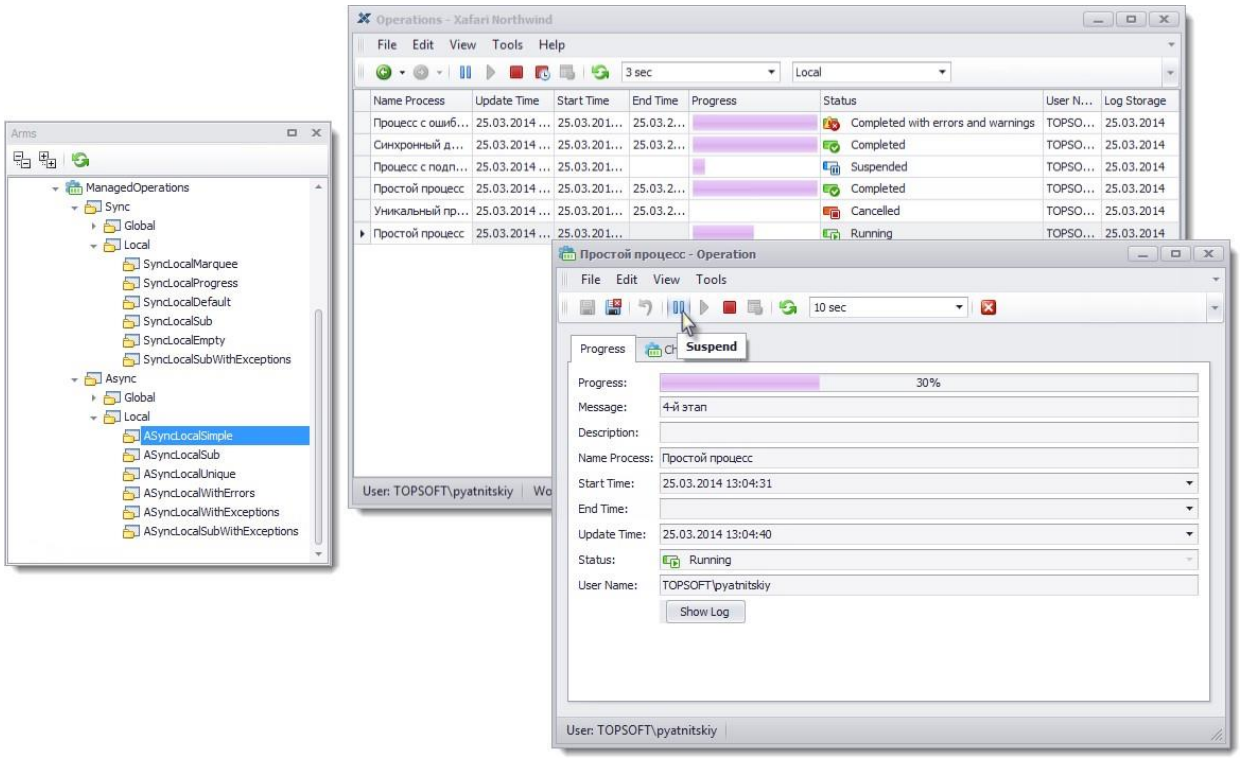
Управляемые операции (Managed Operations)

Модуль *Xafari.ManagedOperations* реализует функциональность управляемых операций. Под управляемыми операциями понимается исполнение некоторого прикладного метода с возможностью его приостановки, продолжения и прерывания, в отдельном или общем потоке. Во время исполнения осуществляется логирование и мониторинг процесса.

Управляемые операции поддерживают несколько вариантов использования:

- Операции могут быть синхронные или асинхронные
- Операции могут быть локальные или глобальные
- Операции могут иметь вложенные операции
- Несколько способов визуализации хода выполнения синхронной операции
    - Проценты
    - Бесконечный цикл
    - Этапы
- Подключаемая плавающая панель администрирования операций

Управляемые операции могут быть использованы на платформах Web, Win, Mvc.



Динамические реквизиты (Dynamic properties)

В Xafari реализованы базовые классы, которые обеспечивают возможности по работе с динамическими реквизитами для бизнес-объектов. Динамические реквизиты – это дополнительные свойства для объектов, которые могут быть использованы для хранения и обработки дополнительных данных, не предусмотренные стандартной конфигурацией прикладного решения.

В отличии от популярного решения Custom Property, динамические реквизиты существуют только для экземпляра объекта. Поэтому два объекта одного типа могут иметь различные динамические реквизиты.

Для работы с динамическими реквизитами рекомендуется использовать специальные редакторы, такие как VerticalGridPropertyEditor и другие.

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

Настройки приложения (Application Settings)

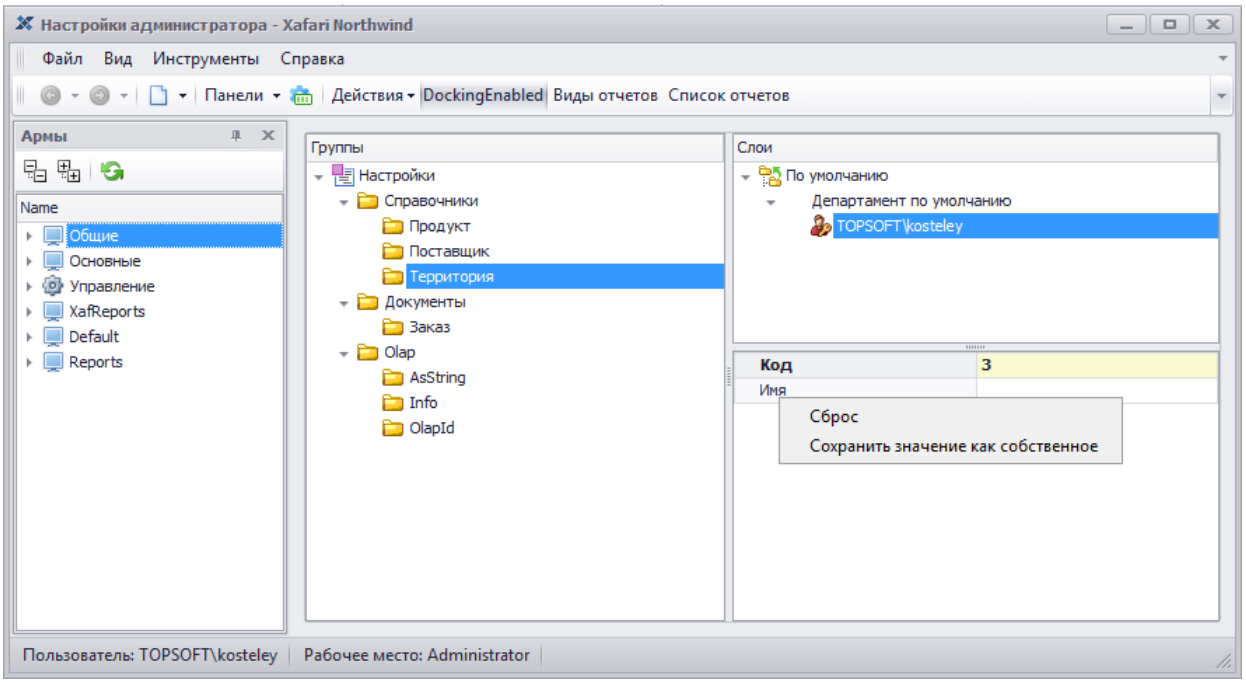
Некоторые приложения настолько сложные и реализуют много вариантов использования, что возникает необходимость ввести настройки для приложения. В состав Xafari входит специальный модуль реализующий настройки приложения.

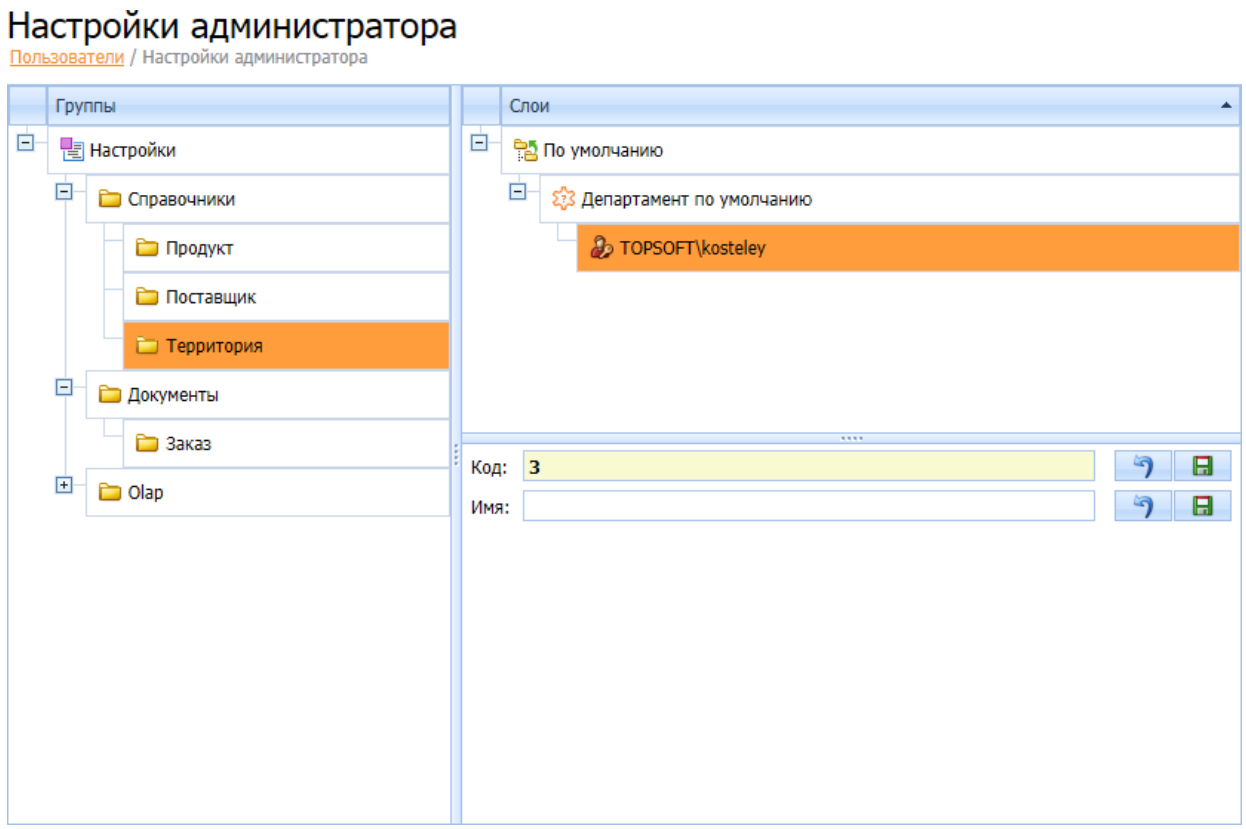
Настройки приложения позволяют следующее:

- Можно спроектировать и описать иерархию настроек и групп любой сложности. Описание структуры настроек производится на C# стандартным способом расширения модели приложения.
- Каждый модуль может содержать собственные настройки, которые будут добавлены в общую систему при подключении данного модуля.
- Можно определить несколько слоев значений настроек, при этом значения последующего слоя перекрывают значения предыдущего слоя. По такому же принципу работают слои в

Модели приложения.
Явно выделен слой системных настроек, дополнительно может быть определен слой пользовательских настроек.

- Управление доступом к настройкам регулируется системой безопасности.
- Возможна конвертация настроек при смене версии приложения.
- Удобный способ работы с настройками в коде.





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

Версионность данных (Versioning)

Данные прикладной системы с течением времени претерпевают существенные изменения. Объект создается, может многократно изменяться, в какой-то момент он может быть удален из системы. Этот объект может участвовать в различных бизнес-процессах, его данные могут учитываться при формировании отчетов или могут быть подвергнуты анализу в специализированных системах. В связи с этим может возникнуть потребность в том, чтобы учитывать состояние анализируемого объекта на определенный момент времени. Для обеспечения именно такого свойства бизнес объекта предназначен модуль Версионность данных (или Versioning).

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).

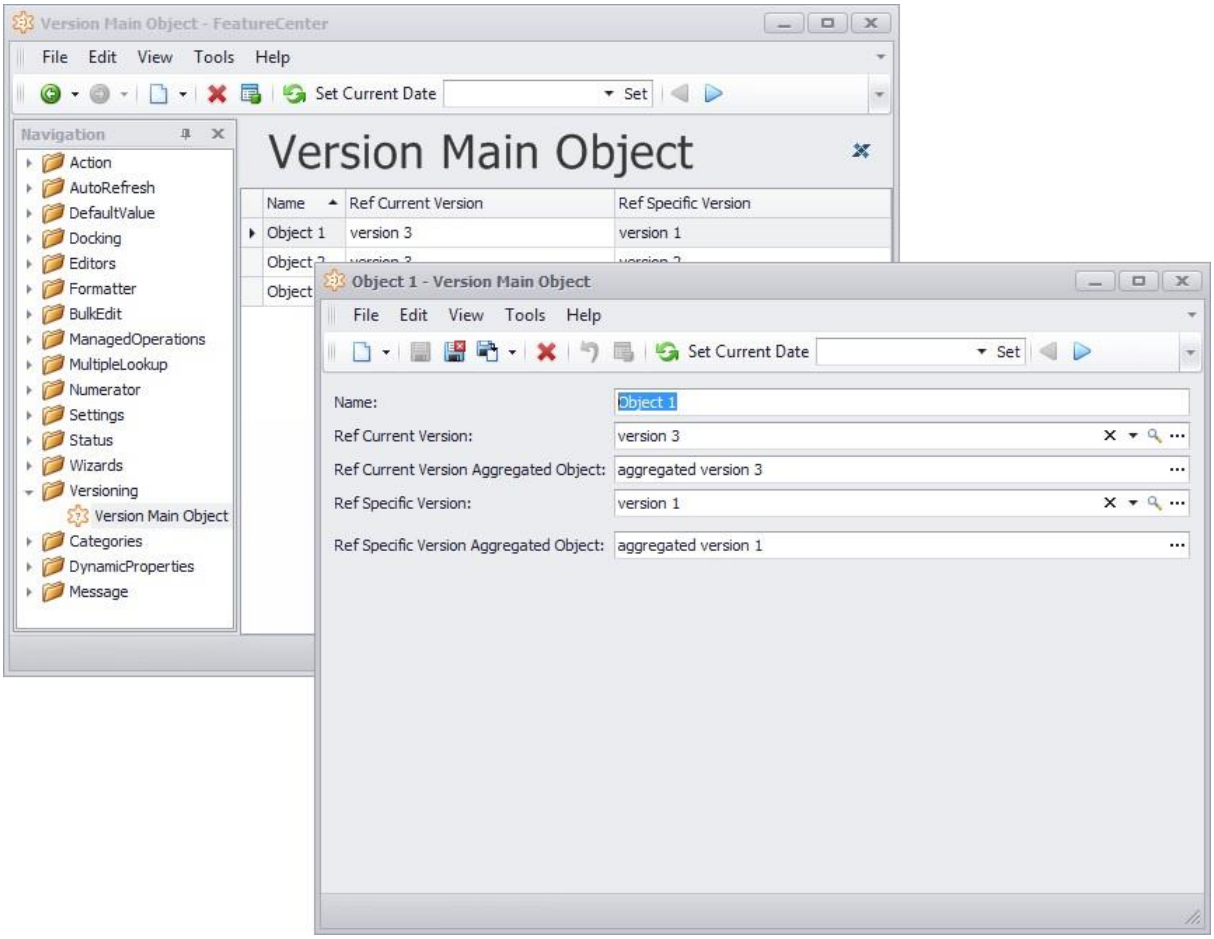
Похожая задача решается системами контроля версий, такими как SVN, TFS и пр.

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 бизнес-объекта.
- 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



Рабочие места (Work Places)

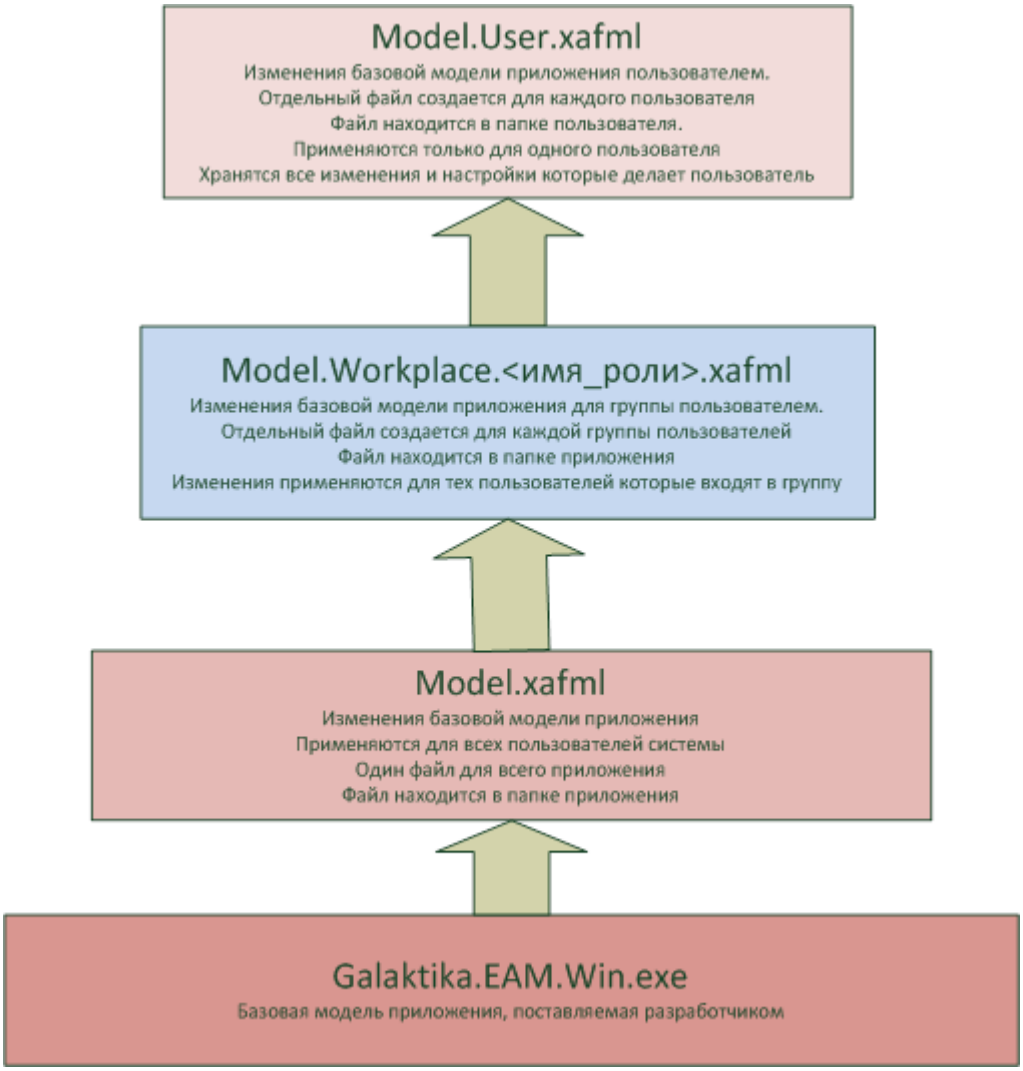
Под рабочим местом пользователя (workplace) подразумевается текущая конфигурация пользовательского интерфейса приложения, видимые пользователю пункты меню, доступные функции.

Способ конфигурирования рабочих мест пользователей основан на идее использования дополнительных слоев модели приложения. В модель приложения вводится слой, который и позволяет осуществлять настройку рабочего места пользователя.

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 использованием наименований назначенных пользователю ролей. Наименования рабочих мест должно соответствовать наименованиям существующих ролей. Т.е. пользователю будет доступно рабочее место, если существует файл рабочего места с именем, соответствующим одной из назначенных пользователю ролей.

Для создания и редактирования рабочих мест выполняется с помощью специального редактора модели - Xafari.Workplace.ModelEditor.exe



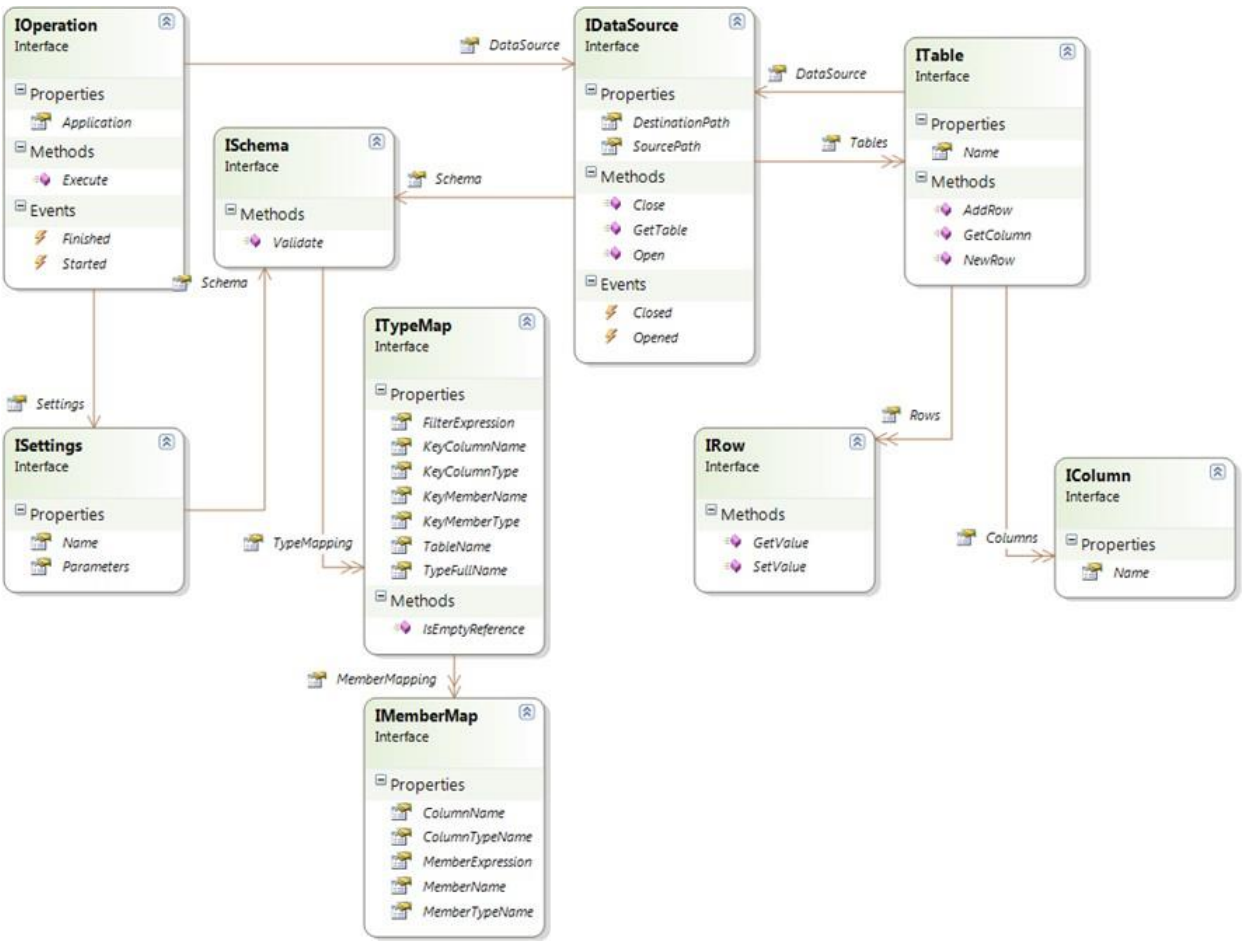
Импорт/Экспорт данных (Data Management (Import/Export))

В состав Xafari входит специальное решение для импорта/экспорта персистентных данных из стороннего хранилища. 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 предлагает стандартный подход для их решения.

В состав Xafari включены несколько типов источников данных (Access, XML, CSV, Excel, Галактика ERP и другие). При этому имеется возможность создавать новые источники данных.



Сервисы Xafari

Сервисы Xafari это программный компонент, реализующий определенную бизнес-функцию в приложении. Примером такой функции является Условное Форматирование , используемое для динамической настройки формы в зависимости от заданных правил.

Сервисы Xafari реализуют различные бизнес-свойства платформы Xafari.

Сервисы Xafari-это шаблон проектирования, предназначенный для разработки бизнес-логики приложения. Роль сервисов в реализации бизнес-логики приложения можно сравнить с ролью, которую играют контроллеры XAF в пользовательском интерфейсе.

Любой сервис имеет некоторые типичные особенности:

- Каждый сервис является синглтоном XAF приложения.
- Его можно активировать и деактивировать целиком или только в конкретном контексте.
- Все сервисы могут иметь свои собственные настройки, описанные единым образом. Параметры открыты для настройки администратором.
- Все службы имеют одинаковый жизненный цикл.
- Сервис может быть связан с текущим пользователем.

Конфигуратор (XAS)

Конфигуратор или Xafari Applications Support (XAS) - это автономное приложение для Windows, основанное на Xafari, которое отвечает за функции, связанные с администрироуванием системы. 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 на 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.

Компоненты ERP

Бизнес-операции (Business Operations)

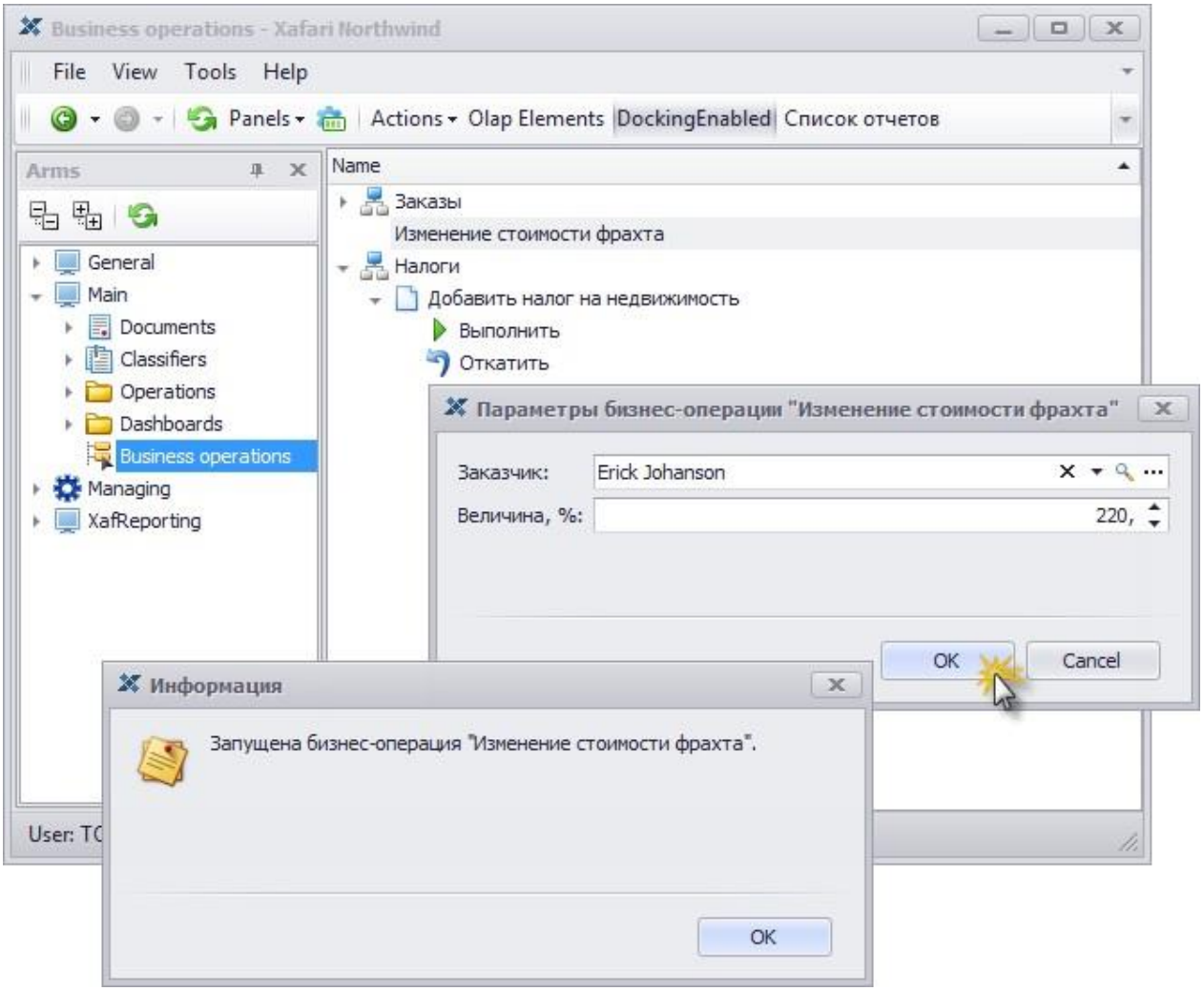
Бизнес-операции предназначены для преобразования одних данных (входов) в другие (выходы). Одни БО могут выполнять сложные расчеты с последующим изменением состояния бизнес-объектов, т.е. могут изменять данные системы. Другие же работают только «на чтение», т.е. получают параметры в качестве входных данных и на основании данных системы возвращают результат. Примерами БО являются процедура расчета зарплаты, формирование накладной на основании ДО, получение прайс-листа, процедура MRP и пр.

Бизнес-операции являются составной частью интерфейса взаимодействия подсистем.

Бизнес-операции реализуют специальный механизм замены их реализации без перекомпиляции кода, тем самым они могут быть использованы как точки конфигурирования конечного приложения.

Бизнес-операции используются для поддержания совместимости между различными версиями подсистем.

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



Аудит Xafari (Xafari Audit)

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

Поддерживается протоколирование:

- 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.

Документооборот (Docflow)

Модуль документооборота содержит механизмы и компоненты обеспечивающие формальное описание жизненного цикла документа или объекта. Основным понятием является *Регламент.*

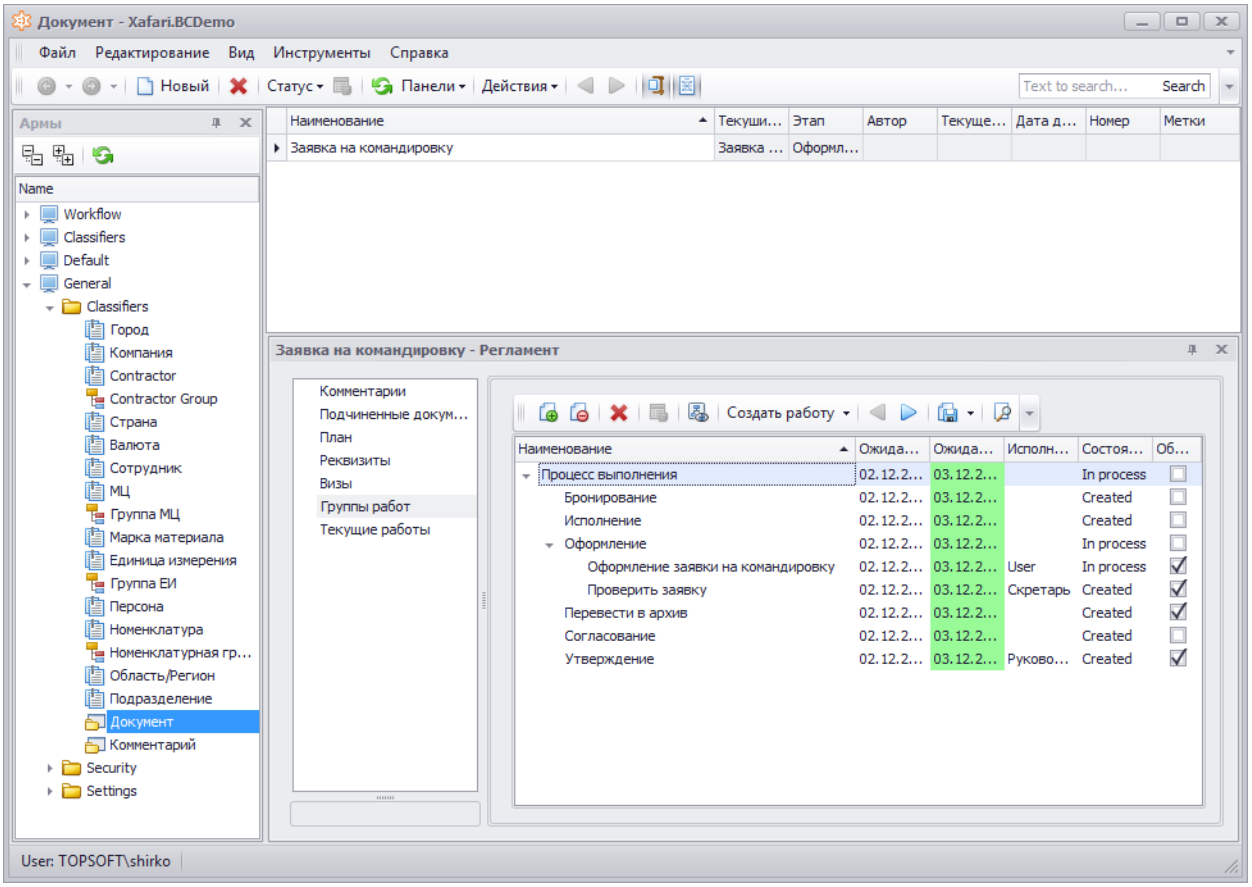
Регламент описывает последовательность этапов, которые проходит документ, перечень и последовательность исполняемых работ. Возможно описывать как «фиксированные» *регламенты* – в духе workflow, так и «гибкие/живые» *регламенты* – как это принято в ACM

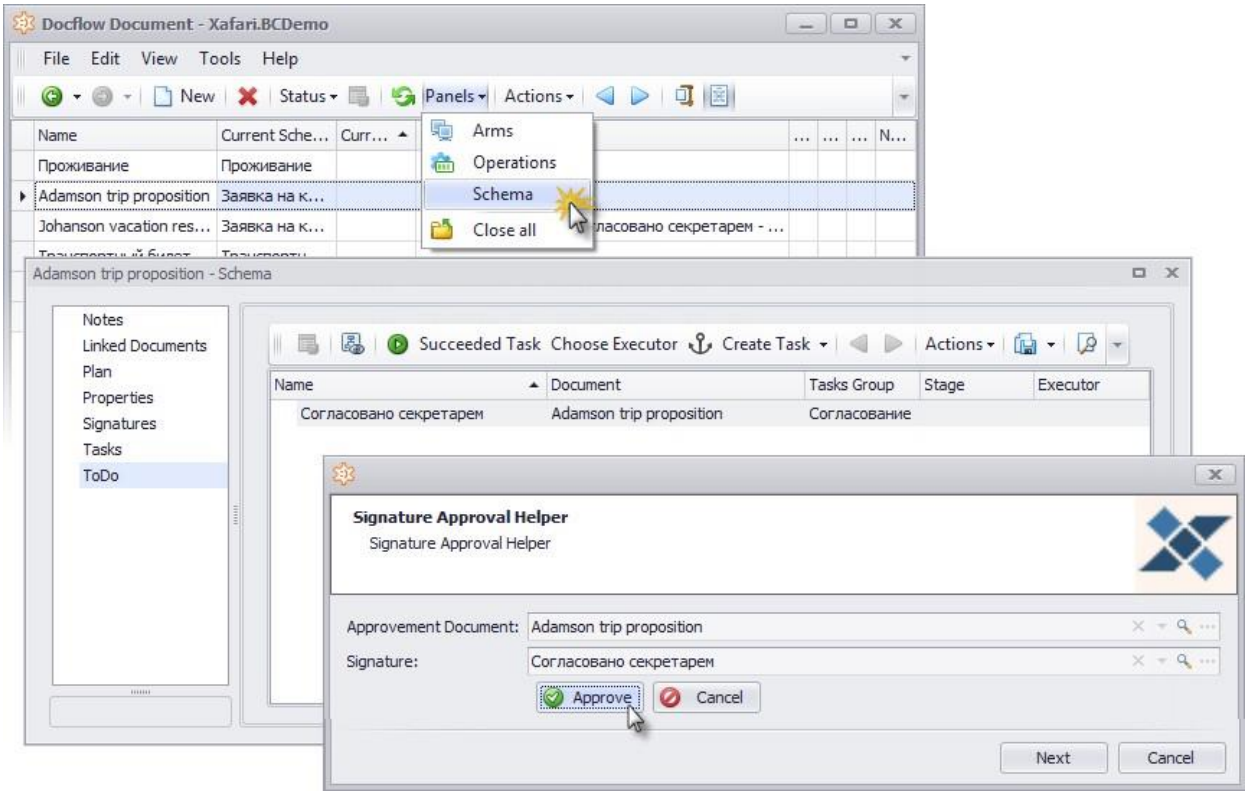
(Adaptive Case Management). В течении жизненного цикла, вся связанная с документом информации накапливается и агрегируется.

Регламент фактически является подтипом документа, который детализирует/уточняет поведение документа под требования конкретной прикладной задачи. Система включающая данный модуль становится адаптивной приобретая свойства робастности (подстройки под изменяющееся, вновь появившиеся требования).

Вся настройка поведения (*регламента*) производится в модели приложения или в конфигураторе. Схема регламента доступна для изменения в момент эксплуатации системы, для этого не требуется программирование с использованием C#.

Для одного типа объекта (доменного компонента) может быть определено несколько регламентов. Так разные экземпляры одного типа могут обрабатываться различными регламентами.





Филиалы (Branches)

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.

Список задач (Task List)

Данный модуль предоставляет возможности для формирования списка задач для исполнителя, и их исполнения. Главным требованием при создании модуля было удобная и прозрачная работа со списком задач для конечного пользователя, для этого были созданы интуитивно понятные и простые формы, адаптируемые для различных целей.
Модуль создан с учетом потребностей прикладных задач и является универсальным и гибким.
Для использование модуля имеется специальный API.
Модуль Список задач активно используется в решении документооборот.

Очереди сообщений (Message Queue)

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 бизнес логику с 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.

Сервер Xafari (Xafari Server)

Этот компонент служит для мониторинга очереди сообщений и распределения сообщений соответствующим обработчикам. Работает сервер асинхронно, как отдельное XAF приложение. Во время работы, выбирает необходимые для обработки сообщения, блокирует и отправляет соответствующим по типу обработчикам. Работает сервер со своим провайдером безопасности, который обеспечивает окружение для обработки сообщения и асинхронное взаимодействие с базой данных, используя пул подключений.

Может работать вместе с приложением, работая в новом потоке и используя локальную очередь для хранения сообщений или на выделенном сервере. При использовании локальной очереди, после завершении работы приложения все сообщения удаляются. В режиме выделенного сервера сообщения хранятся в базе данных приложения.

На основе сервера расчетов реализован сервер отчетов, для построения отчетов на выделенном сервере. Строит отчеты на сервере и сохраняет сформированный отчет в базу данных, которые можно просмотреть в журнале отчетов приложения. Отчеты формируются от пользователя, генерирующего сообщение, это значит, что в сформированном отчете будут только те данные, которые видны пользователю.

Отчеты Xafari (Xafari Reports)

Новая система отчетности для XAF приложений, расширяющая стандартный функционал системы отчетности. Позволяет строить отчеты на базе нескольких бизнес-объектов со сложной структурой данных и алгоритмов наполнения этой структуры. Возможность сохранять наборы параметров отчетов для последующего использования. Предоставляет разработчику возможность для создания произвольного алгоритма для формирования отчетности.

Отчеты Xafari позволяют определять несколько представлений отчета (печатный отчет, график, сводная таблица, выгрузка в файл и пр.). Решение о том, в какой именно форме получать отчет, принимает сам пользователь.

ASP.NET MVC платформа

Архитектура XAF предусматривает одновременную поддержку бизнес-приложения на платформах WinForms и ASP.NET. Xafari реализует дополнительно поддержку для ASP.NET MVC приложений. Методика разработки приложений при этом не изменяется. Особенностью является принципиально новый механизм генерации html-кода, основанный на принципах ASP.NET MVC.

Первые сравнительные тесты производительности MVC-приложения показали на 30% лучший результат в сравнении с аналогичным Web-приложением.

Познакомится с примерами приложений можно на странице Demos