



Компания "Консид Решения"

Методология разработки WEB приложений на базе

Open Source HH.WEB

Оглавление

Введение	3
1. Цели и задачи, решаемые на этапах разработки бизнес-приложений для Web	5
2. Этапы разработки бизнес-приложений для Web.....	6
2.1. Постановка технического задания на разработку Web форм	6
2.2. Эскизное (Визуальное) проектирование.....	6
2.3. Разработка хранимых процедур	8
2.4. Тестирование	8
3. Архитектура и общее описание	9
3.1. Используемые термины и сокращения	9
3.2. Архитектура и общее описание системы.....	9
3.3. Спецификация системы.....	10
3.4. Политика безопасности и лицензирование.....	12
4. Функционал приложения OpenSourceHH.WEB.....	13
4.1. Используемые термины и сокращения	13
4.2. Описание структуры приложения Open SourceHH.WEB	13
4.2.1. Элементы управления приложения Open SourceHH.WEB	13
4.2.2. Настройки приложения Open SourceHH.Web	16
4.2.3. Описание бизнес-процессов в конфигурации приложения	17
4.3. Управляющие элементы и свойства.....	19
4.3.1. Управляющие элементы и свойства.....	19
4.3.2. Описание глобальной конфигурации приложения	29
4.3.3. Описание специфических настроек системы.....	30
4.3.4. Описание событий, поддерживаемых контролами генератора web отчетов	31
4.3.5. Требования к разработке хранимых процедур	32
4.3.6. Передача параметров между формами.....	32
4.3.7. Ограничения в определении названия параметров форм.....	33
4.3.8. Требование к написанию хранимых процедур и запросов	33
4.3.9. Генерация отчетов Crystall Reports с использование генератора отчетов	34

Введение

Open Handheld Engine (ONE) - платформа бизнес-приложений

Платформа решений Open HandHeld Engine является ядром линейки всех решений предлагаемых компанией. Реализация платформы обеспечивает высокую производительность и надежность выполняемых операций за счет используемых технологий и глубокой проработки внутренней архитектуры.

Платформа является основой для разработки промышленных бизнес-приложений для автоматизации при помощи конструкторов Архитектуры.

Преимущества:

- высокая производительность
- масштабируемость
- высокая степень адаптируемости
- независимость от поставщика оборудования ТСД (мобильный терминал сбора данных)
- короткие сроки разработки конечных приложений за счет использования визуальных средств разработки

Платформа бизнес-приложений включает в себя:

- WEB-сервер
- Telnet-сервер

Web-сервер обеспечивает работу тонких клиентов на стационарных компьютерах. Web-сервер основан на технологии asp.net, но использует свои управляющие элементы, обеспечивающие большую производительность за счет минимизации времени по учету состояний объектов и использования оптимизированной логики работы. Web-сервер позволяет писать клиентские приложения для доступа и манипулирования данными, хранящимися в БД.

Пользователю нет необходимости обладать знаниями в области разработки web-приложений. Всю внутреннюю обработку сервер берет на себя за счет использования конфигурации.

Конфигурация представляет собой набор xml-файлов, которые компилируются в приложении при запуске системы.

Для разработки конфигурации конечного приложения администратор системы должен обладать только навыками разработки запросов к базам данных и хранимых процедур, и пройти курс обучения технологии разработки WEB-приложений при помощи конструктора Open Source NH.WEB.

Наличие конфигурации сервера и использование конструктора web-приложений обеспечивает следующие преимущества:

- ускоряет разработку бизнес-приложений в разы
- минимизирует набор специфических знаний разработчика
- обеспечивает стабильную работу web-приложений
- сокращение времени на постановку технического задания. Данный подход за счет унификации стандартов разработки позволяет сократить время на постановку технического задания, так как в этом случае будет описываться реализация бизнес-логики, а в постановке задачи технической реализации не будет необходимости.
- сокращение времени тестирования нового функционала. Данный подход за счет унификации стандартов разработки позволяет сократить время тестирования разработанной функциональности конечного приложения, так как, в этом случае, будет тестироваться только реализация бизнес-логики, а тестирование программной реализации отсутствует, в связи с тем, что вся программная логика находится в ядре системы и она уже оттестирована.

Telnet-сервер представляет собой набор стандартных обработчиков, обеспечивающих обмен данными с ТСД.

Telnet-сервер платформы Open HandHeld Engine обеспечивает работу по протоколу telnet с использованием любых клиентов, поддерживающих данный протокол. Как пример, использование WaveLink для ТСД, или использование HyperTerminal на стационарных рабочих местах.

Логика работы Telnet-сервера вынесена в конфигурационный файл формата xml. Это обеспечивает широкие возможности адаптации работы Telnet-сервера без трудоемкой разработки.

Для разработки терминальных приложений под Telnet-сервер используется визуальный конфигуратор Open Source NH.Mobile. Использование конфигуратора ускоряет процесс разработки приложений и не требует специфических навыков персонала. Это сокращает стоимость дополнительных доработок, которые могут потребоваться в будущем. Данный подход также позволяет сократить время на тестирование терминального приложения.

Высокая производительность Telnet-сервера обеспечивается наличием изолированных потоков для каждого тонкого клиента, что исключает возможность “зависания” Telnet-сервера при выполнении ресурсоемких операций пользователями.

В данном документе рассматривается методология разработки Web приложений при помощи визуального средства разработки Open Source NH.WEB.

1. Цели и задачи, решаемые на этапах разработки бизнес-приложений для Web

Open HandHeld Engine (ONE) в своем составе имеет визуальное средство для разработки конфигурации Web-приложений Open Source HandHeld.WEB, предназначенное для облегчения и ускорения процесса разработки бизнес-процессов Web-приложения.

Бизнес-процессы Web-приложения рекомендуется разрабатывать по следующей методике. Это обеспечит быструю и качественную разработку решений.

В разработке решений рекомендуется участие двух человек (возможно совмещение ролей в одном человеке):

- бизнес-аналитика
- разработчика SQL

Задачами бизнес-аналитика являются:

- Согласование проектной документации, сроков разработки Web-форм
- Постановка технического задания на разработку Web-форм. Определение списка основных Web-форм, используемых в разработке и их связь с другими Web-формами (переходы).
- Постановка задач на разработку хранимых процедур и отображений для программиста
- Визуальная разработка форм Web-приложения

Задачами SQL разработчика являются:

- Разработка хранимых процедур в соответствие с требованиями бизнес-аналитика
- Разработка интерфейсов Web форм
- Тестирование бизнес-процессов Web форм приложения
- Внутренняя приемка-сдача разработанных Web-форм процессов

Возможно совмещение обязанностей бизнес-аналитика и программиста в одном лице, но не рекомендуется.

2. Этапы разработки бизнес-приложений для Web

Разработка Web-приложений на платформе ONE включает следующие этапы:

1. Постановка технического задания на разработку Web-форм.
2. Эскизное проектирование. Разработка Web-форм с описанием свойств и событий при помощи конструктора архитектуры Open Source NH.WEB (Визуальное проектирование).
3. Разработка хранимых процедур (SQL процедуры для выборки данных и проводки изменений в БД).
4. Тестирование разработанных Web форм и логики их работы.
5. Сдача в эксплуатацию.

2.1. Постановка технического задания на разработку Web форм

Этот этап методологии разработки Web-форм предполагает плотное общение бизнес-аналитика со стороны компании интегратора (или ИТ-отдела) с заказчиком - бизнес-аналитиком или логистом от производства. На данном этапе определяются и описывается список и логика Web-форм для дальнейшей их реализации в Open Source NH.WEB.

Результатом этого этапа являются техническое задание, которые подписывается обеими сторонами.

Результатом этапа является текстовый документ, который описывает все Web формы приложения, логику их функционирования и переходы между формами.

2.2. Эскизное (Визуальное) проектирование.

На этом этапе все Web-формы и их логика функционирования в соответствии с техническим заданием переносятся в конфигурацию приложения Open Source NH.WEB. Таким образом, определяется последовательность Web-форм и логика их работы.

Средство разработки Open Source NH.WEB имеет встроенное функциональное окно для просмотра отображения разрабатываемых Web-форм.

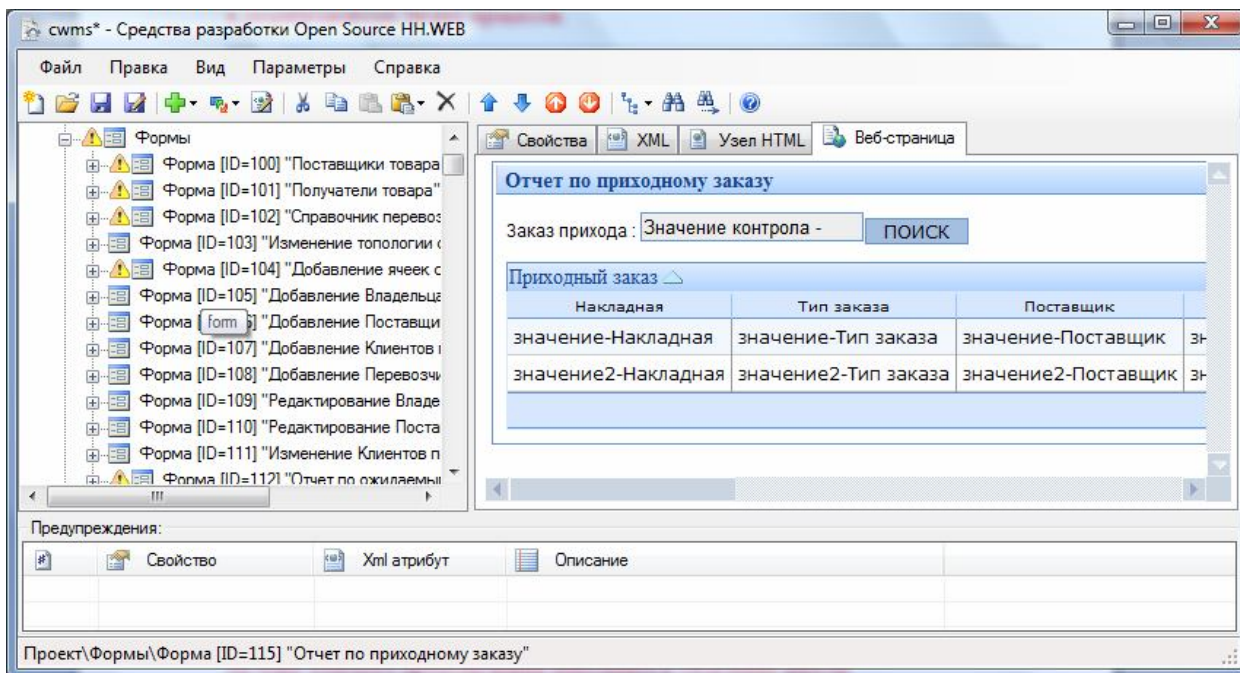


Рис. 2.1. Визуализация разрабатываемой Web формы

На этапе эскизного проектирования выполняются следующие работы:

- создается отображение всех Web форм
- определяются логические переходы между формами
- определяются основные элементы управления Web форм: таблицы, списки выбора, выбор даты, функциональные кнопки и т.д.
- определяются основные события (инициализации, проверки и т.п.) элементов управления формы
- определяются имена хранимых процедур и параметров, которые будут использоваться. Также определяются имена sql отображений (view) для выборки данных из БД
- программистом создаются пустые хранимые процедуры, возвращающие константы. Используется для тестов логики работы бизнес-процесса.
- тестируется логика Web форм в соответствии с ТЗ. Исправляются ошибки и недочеты.

Результатом окончания этапа является – конфигурация приложения с полным описанием Web форм приложения, логики функционирования и хранимых процедур для обработки и связи с базой данных приложения.

2.3. Разработка хранимых процедур

Список хранимых процедур составляется бизнес-аналитиком совместно с разработчиком на этапе постановки технического задания и на этапе эскизного проектирования.

Хранимые процедуры разрабатываются в соответствии с требованиями и ограничениями на основе шаблонов. Каждому шаблону хранимой процедуры соответствует свой набор входных параметров.

2.4. Тестирование

После эскизного проектирования и разработки хранимых процедур можно приступить к тестированию Web-приложения. Для этого необходимо установить Web сервер приложения сервер базы данных, сконфигурировать сервер приложения, перенести конфигурацию приложения. Тестирование бизнес-процессов приложения осуществляется на разработанном приложении, настроенном на базу данных. Результаты выполнения процессов Web-приложения анализируются на предмет правильности функционирования. Для отслеживания ошибок в хранимых процедурах и времени их выполнения в SQL Manager запускается приложение Profiler.

Если в ходе тестирования обнаружены ошибки, то в этом случае анализируются журнал работы приложения и MsSql Profiler, база данных приложения и конфигурация приложения. Найденные ошибки устраняются и далее, продолжается тестирование.

По итогам тестирования подписывается акт приемки сдачи результатов тестирования.

3. Архитектура и общее описание

3.1. Используемые термины и сокращения

Оператор - человек, обученный работе с Комплексом и обладающий навыками работы с оборудованием и Базой.

АРМ - Автоматизированное Рабочее Место оператора

Сервер БД – Компьютер, предназначенный для хранения и обработки данных, поступающих от Модулей и АРМ.

Сервер – часть Программы, находящаяся на персональном компьютере и необходимая для обмена данными между компьютером и базой данных Web приложения.

Событие инициализации – событие, возникающее до появления экранной формы или управляющего элемента на экране терминала.

Событие перехода – событие, возникающее после обработки всех управляющих элементов на экранной форме.

Событие обновления – событие, возникающее после обработки управляющего элемента на экранной форме.

Терминал сбора данных или Терминал - Прибор, представляющий собой микрокомпьютер, оснащенный сканером штрих-кода, процессором, операционной системой, программой, встроенной памятью и системой ввода-вывода информации (экран и клавиатура) в виде, понятном человеку, а также системой обмена данными с компьютером.

Сканер штрих-кода или Сканер - прибор, входящий в состав Терминала или выполненный в виде отдельного прибора. Предназначен для считывания и распознавания (декодирования) штриховых кодов и передачи результатов декодирования в терминал сбора данных или базу данных предприятия.

Принтер штрих-кода или Принтер - прибор, предназначенный для печати Этикеток, и используемый для нанесения на Этикетку необходимых данных.

3.2. Архитектура и общее описание системы

Сервер реализован как часть трехуровневой архитектуры и представляет собой сервер приложений, в котором вся логика работы и описание Web форм вынесены в конфигурацию. Web сервер реализован как Интернет сайт, разработанный в соответствии с технологией .Net .

Сервер приложения представляет собой многопоточное Web-приложение, в котором каждый клиент работает в изолированном потоке. Это позволяет избежать зависания Сервера, если один из Операторов выполняет ресурсоемкую операцию.

Сервер не привязан к Базе Данных. Работа с Базой Данных поддерживается посредством вызова хранимых процедур. В качестве баз данных могут быть использованы MSSQL2000/2005, Oracle и т.п. Администратору, работающему с Сервером необходимо просто описать логику работы с данными в хранимых процедурах и описать их вызов в конфигурации приложения.

Большинство настроек сервера вынесены в файлы конфигурации, и считываются при запуске Сервера. Конфигурация представляет собой xml файлы интуитивно понятного формата. Разработка конфигурации сервера приложения осуществляется в визуальном приложении Open Source NH.Web.

Конфигурация сервера приложений разделена на несколько частей.

В корневой папке располагаются настроечные файлы Web приложения:

- **Ohewebconfig.xml** – содержит системные настройки Web Сервера. В конфигурацию вынесены такие настройки как строка подключения к Базе Данных и т.д.
- **webmenu.xml** – содержит описание иерархического меню Web -приложения
- **OhewebDataConfig.xml** – содержит описание настроек выбора строк для табличных данных Web форм.

В папке **forms** находятся xml файлы с описанием Web форм приложения. Каждый xml файл описывает определенную Web форму. Код Web формы в конфигурации соответствует имени файла.

В папке **reports** находятся шаблоны отчетов Crystal Report Web приложения.

Так же могут находиться папки соответствующие идентификатору пользователя, в которых находятся текущие отчеты для 3D модуля систем.

При возникновении ошибок, связанных с работой Сервера, все исключения и ошибки отслеживаются и записываются в виде текстовых файлов в директорию **Log**, которая находится в инсталляционной директории Web приложения. Файлы логов разбиты по дате появления в них информации и представляют собой текстовые файлы.

3.3. Спецификация системы

Описание пользователей

Все пользователи Web-приложения, которые работают с системой, должны быть зарегистрированы в системе и включены в одну из функциональных групп, таких как администраторы, диспетчеры и т.д.

Описание меню иерархического дерева Web приложения

В конфигурации должно быть описано функциональное меню приложения.

При описании меню задаются свойства пункта меню и переходы на другие пункты меню или вызов Web-формы.

В конфигурации приложения должны быть описаны все Web-формы приложения, состоящие из множества объектов для которых задаются свойства и события.

Описание объектов (управляющих элементов) Web форм.

Все Web формы состоят из списка управляющих элементов (объектов), с которыми работает пользователь в процессе выполнения бизнес-процессов Web-приложения.

Все управляющие элементы разделяются на типы:

label - текстовая метка. Представляет собой управляющий элемент (контроль) Web формы текстовое поле, которое заполняется текстовым значением константой.

textbox - поле для ввода текста. Представляет собой управляющий элемент (контроль) Web формы текстовое поле, для ввода данных пользователем. Данный контроль может при инициализации по умолчанию заполнен или константой или значением по запросу из БД.

combobox - список выбора. Представляет собой управляющий элемент (контроль) Web формы список выбора одного из нескольких значений. Данный контроль может при инициализации по умолчанию заполнен или значениями константами или значениями по запросу из БД.

ajaxcombobox - список выбора оптимизированный. Представляет собой управляющий элемент (контроль) Web формы список выбора одного из нескольких значений. При работе с данным элементом управления минимизируется список выбора в зависимости от значения введенного текстового значения в области объекта.

datetime - выбор даты. Представляет собой управляющий элемент (контроль) Web формы для выбора даты, которое может заполняться значением по умолчанию.

button - функциональная кнопка. Представляет собой управляющий элемент (контроль) Web формы кнопку, при нажатии на которую выполняется переход на заданную форму и выполняется событие submit, если оно задано.

popupmenu - всплывающее меню. Представляет собой управляющий элемент (контроль) Web формы всплывающее меню. Данный элемент задается для Web-формы и срабатывает по нажатию правой клавишей манипулятора “мышь”. При выборе одного из пунктов всплывающего меню выполняется переход на заданную Web форму.

grid - таблица с данными. Представляет собой управляющий элемент (контроль) Web формы набор данных, представленный в виде таблицы, которая заполняется данными по sql запросу (хранимой процедуры) к базе данных

Все элементы управления формы могут поддерживать события (соответствующие элементам управления)

- **init** - инициализация контроля. Автоматически вызывается при попадании пользователя на форму (если событие определено для контроля).

- change - событие обновления контрола. Автоматически вызывается при введении пользователем данных в приложении (если событие определено для контрола).
- submit - событие перехода. Автоматически вызывается при переходе по ссылке link (колонка таблицы) или нажатии функциональной клавиши button (если событие определено для контрола).

Кроме этого, ввод данных от пользователя проверяется на соответствие типу данных элемента управления.

3.4. Политика безопасности и лицензирование

Файловая система

Все файлы с расширениями xml и файлы *.log должны быть доступны как для чтения, так и для изменений. Временные файлы при работе программы создаются в папке программы, поэтому она должна быть помечена как доступная для изменений.

Сервер должен запускаться от имени пользователя, для которого должен быть обеспечен доступ к папке инсталляции Сервера на чтение и запись.

При Windows аутентификации на MSSQL сервере пользователь должен иметь доступ к Базе Данных.

Лицензирование

Сервер поставляется вместе с комплектом пользовательских лицензий и лицензии программного обеспечения, кроме этого включаются 2 лицензии программного обеспечения для разработчиков.

Лицензия представляет собой xml файл, который содержит информацию о максимальном количестве одновременно работающих пользователей, доступных модулях, информацию о фирме и информацию, об уникальном ключе, сгенерированным Web Сервером.

Одна лицензия соответствует одному серверу приложений. При необходимости использования Web Сервера на нескольких серверах приложений необходимо приобретение нескольких лицензий.

Процесс получения лицензии прост:

Разработчику высылается файл, содержащий ключ и информацию о фирме, который генерируется в процессе установки программного обеспечения. Файл находится в корневом каталоге, на диске где установлена операционная система. Далее файл посылается разработчику. В ответ на присланный ключ разработчик высылает файл с лицензиями для данного сервера. Затем пользователь заменяет сгенерированный Сервером файл на файл, присланный разработчиком.

4. Функционал приложения Open Source HH.WEB.

4.1. Используемые термины и сокращения

Экранная форма (форма) – это содержимое экрана, которое отражается на экране монитора ПЭВМ.

События – реакция на действия, которые произвел пользователь во время работы, или которые возникли автоматически.

Событие инициализации – событие, возникающее до появления экранной формы или управляющего элемента на экране терминала.

Событие перехода – событие, возникающее после обработки всех управляющих элементов на экранной форме.

Событие обновления – событие, возникающее после обработки управляющего элемента на экранной форме.

Событие выхода – событие, возникающее при нажатии пользователем клавиши выхода с формы.

Управляющий элемент экранных форм или контрол – объект Web формы для отображения, ввода и обработки информации.

4.2. Описание структуры приложения Open Source HH.WEB

4.2.1. Элементы управления приложения Open SourceHH.WEB

Описание элементов и областей приложения Open SourceHH.Web

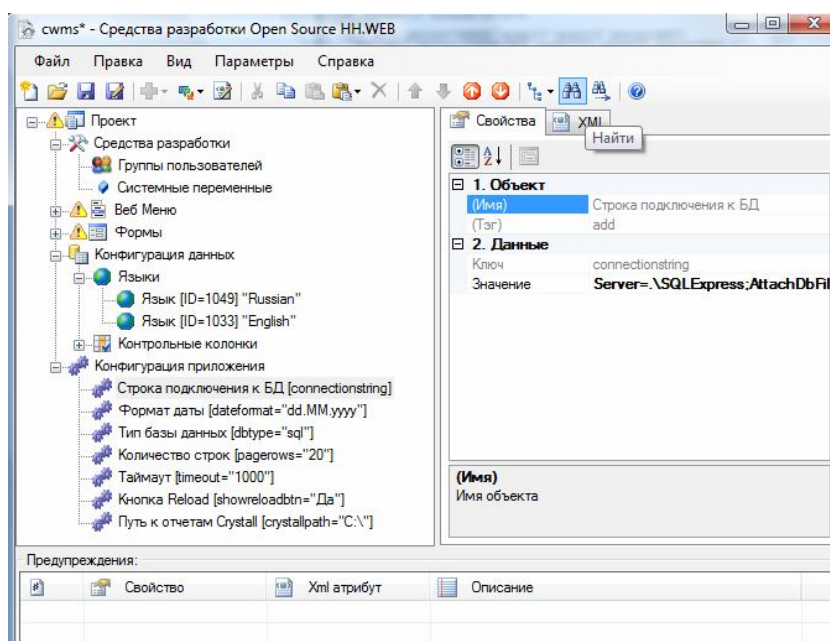
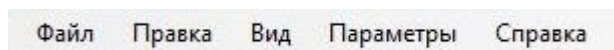


Рис. 4.1. Элементы и области приложения Open SourceHH.Web

Приложение Open SourceHH.Web состоит из следующих областей:

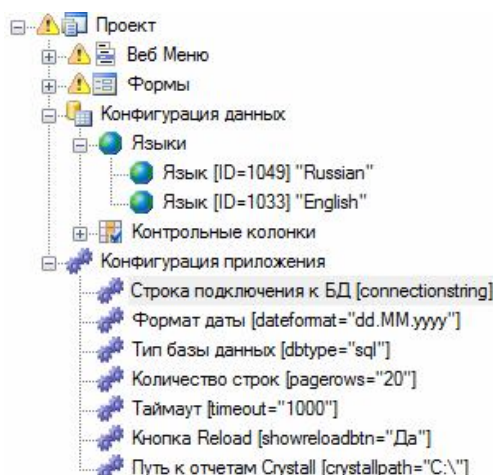
- Функциональное меню. Выбор функциональных действий с объектами проекта и бизнес-процессов.



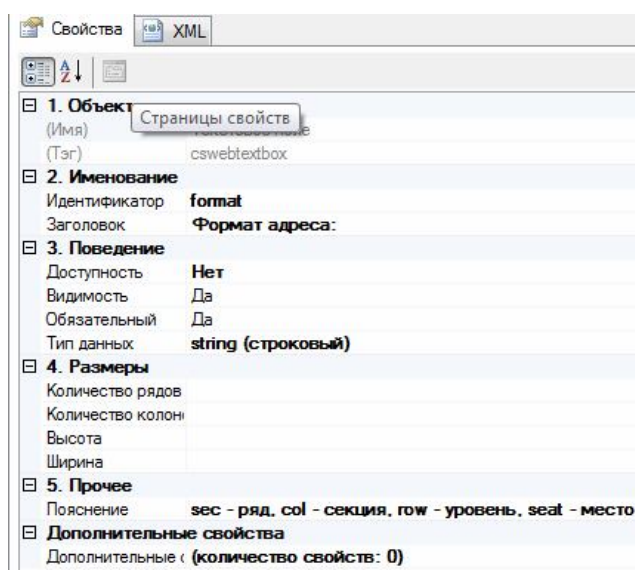
- Панель элементов. Выбор функциональных действий с объектами проекта и бизнес-процессов.



- Дерево проектов. Выбор объектов проекта или бизнес-процессов для просмотра и изменения свойств объектов.

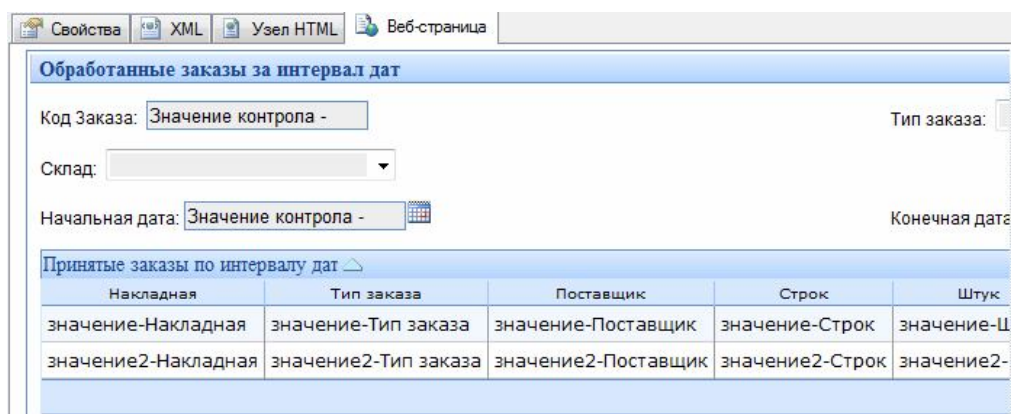


- Окно свойств объектов. Изменение свойств объектов проекта или бизнес-процессов

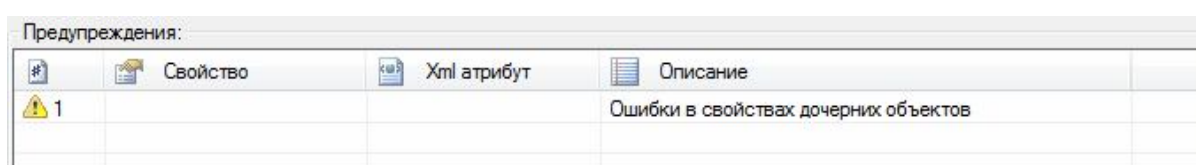


- Окно XML-кода. Просмотр XML-кода объектов проекта или бизнес-процессов

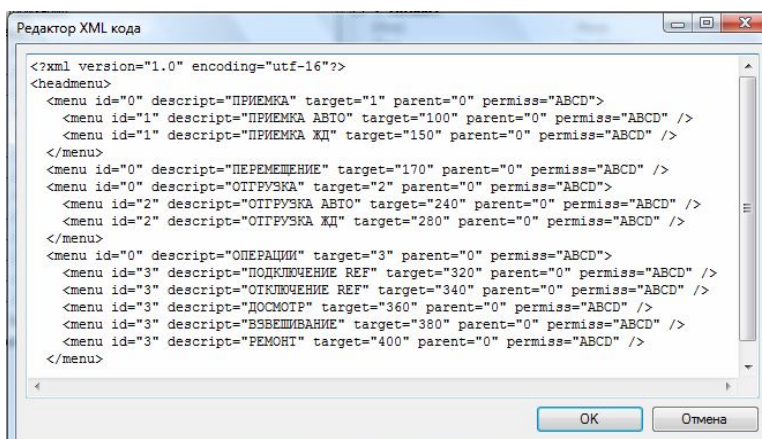
- Окно Web-страница. Просмотр визуального отображения разрабатываемой Web-формы



- Окно предупреждений и ошибок конфигурации



- Окно редактирования XML-кода. Редактирование XML-кода объектов проекта или бизнес-процессов



4.2.2. Настройки приложения Open SourceHH.Web

- **Описание групп пользователей системы по категориям**

В конфигурации приложения Open Source HH.Mobile описываются группы пользователей по категориям: администраторы, операторы и т.д.

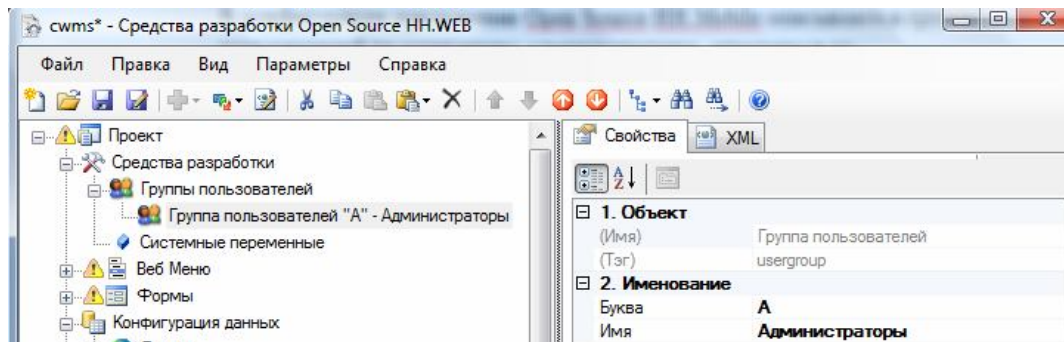


Рис. 4.4. Группы пользователей.

- **Настройка системных переменных - системного файла Ohewebconfig.xml**

В конфигурации приложения Open Source HH.Web описываются системные настройки Web-сервера приложений, такие как:

- Строка подключения к БД
- Формат даты
- Тип базы данных
- Количество строк в таблице
- Тайм-аут. Время ожидания сброса сессии
- Кнопка Reload. Отображение на панели Web-приложения кнопки перезагрузки конфигурации приложения

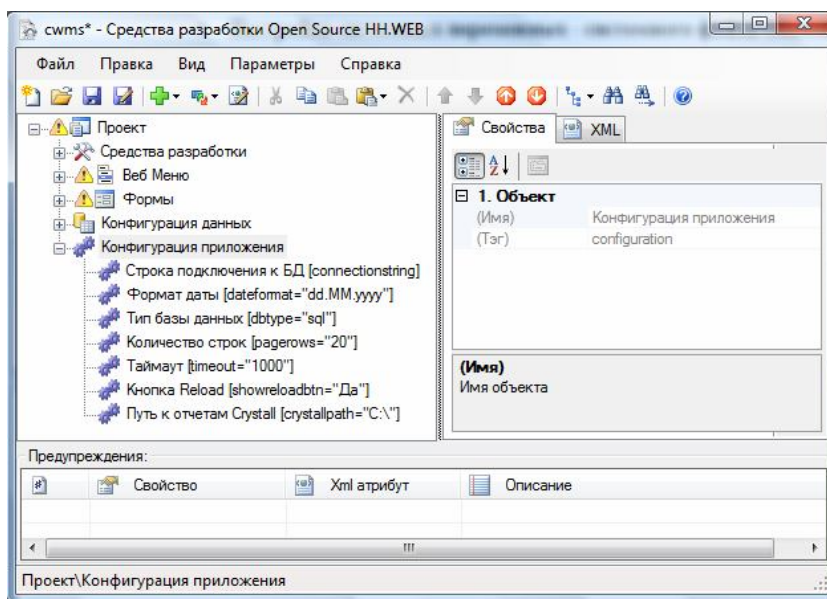


Рис. 4.5. Настройка файла Ohewebconfig.xml

4.2.3. Описание бизнес-процессов в конфигурации приложения

- **Описание пунктов меню иерархического дерева Web-приложения**

Конфигурация всех пунктов меню находится в ветке дерева проектов - "Меню".

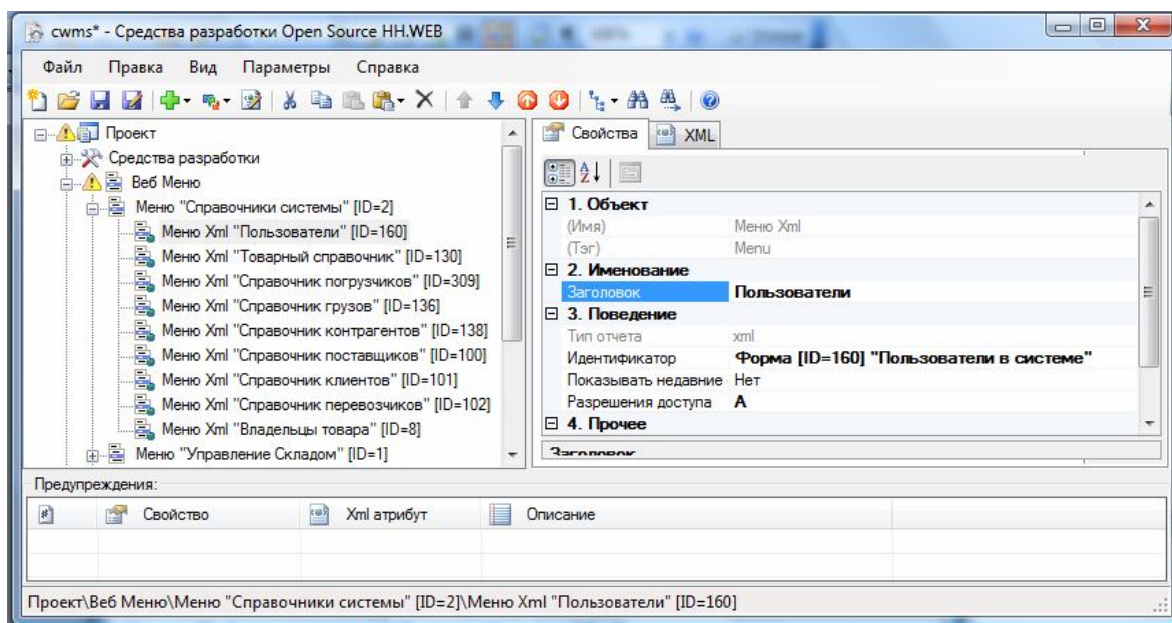


Рис. 4.7. Редактирование пунктов меню из окна свойств.

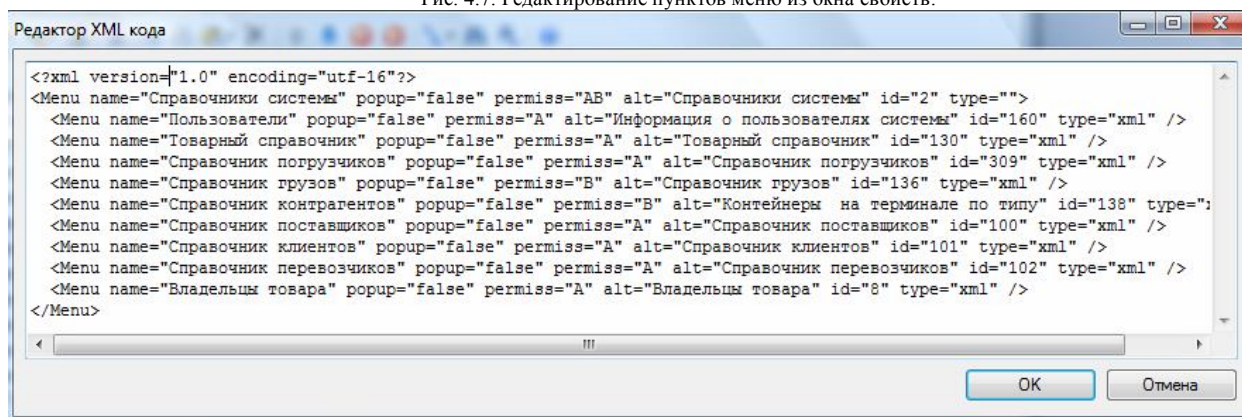


Рис. 4.8. Редактирование пунктов меню из окна редактирования XML кода.

- **Описание Web-форм и логики Web-приложения**

Конфигурация всех разрабатываемых Web-форм находится в ветке дерева проектов - "Формы".

Каждая Форма состоит из совокупности управляющих элементов (контролов). Каждый управляющий элемент описывается набором свойств и событий. В приложении Open Source HH.Web доступно ограниченное число стандартных и аjax-объектов. По согласованию с компанией "Консид Решения" в Open Source HH.Web могут быть добавлены новые объекты.

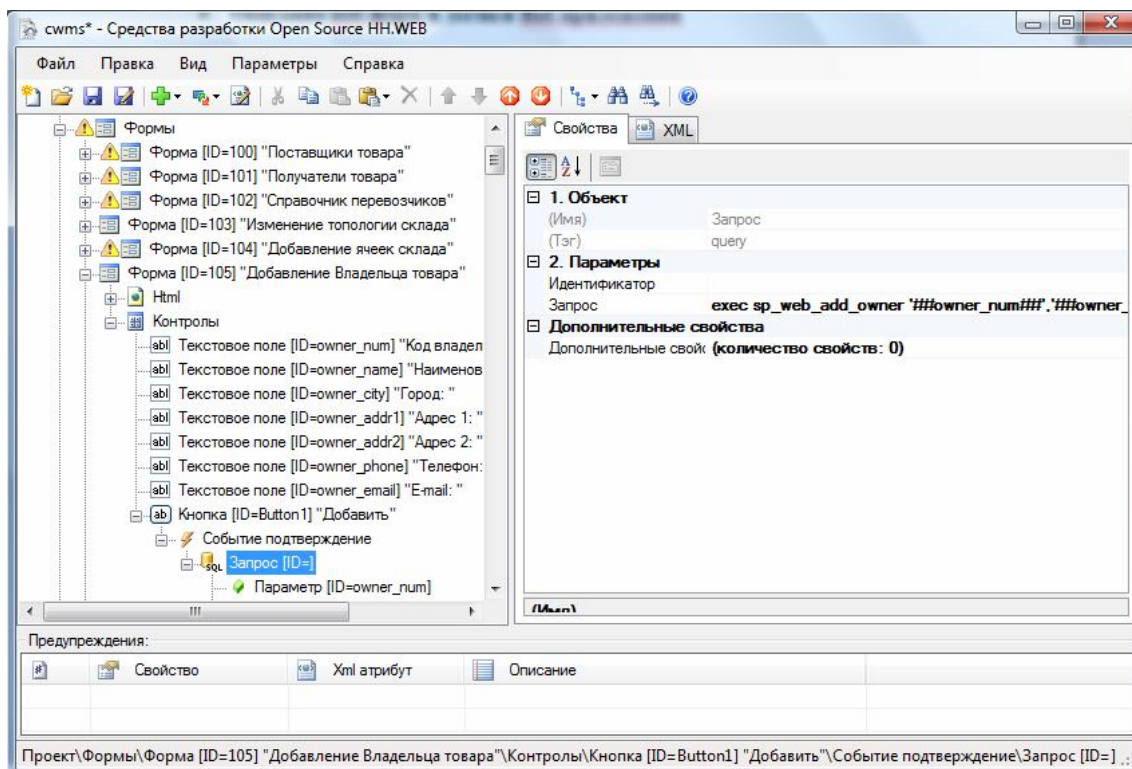


Рис. 4.9. Редактирование свойств Web-формы из окна свойства.

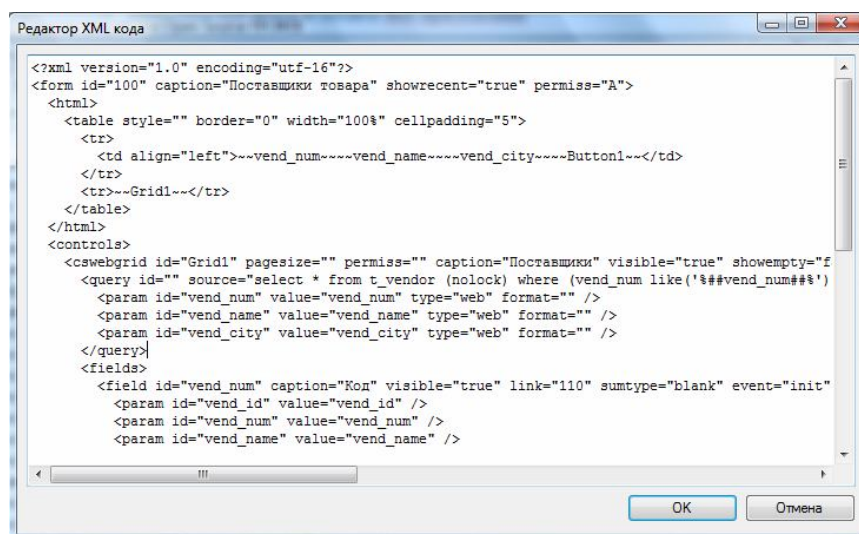


Рис. 4.10. Редактирование Web-формы из окна редактирования XML кода.

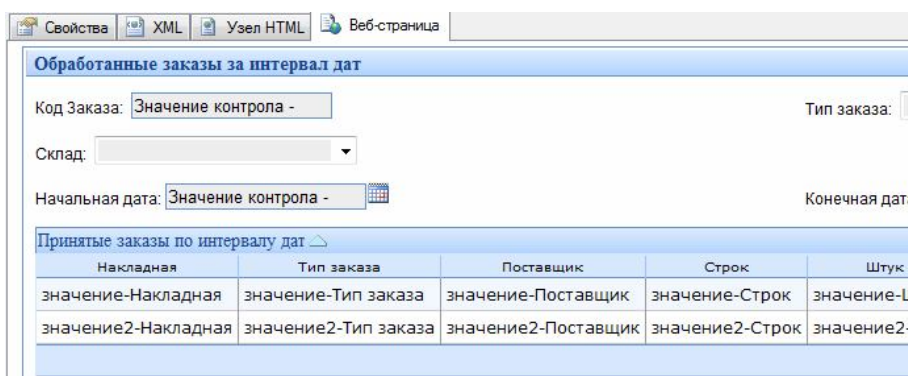


Рис. 4.11. Просмотр разрабатываемой Web-формы.

4.3. Управляющие элементы и свойства

4.3.1. Управляющие элементы и свойства

1. Форма – представляет собой отчет в приложении. Это xml файл, название которого является уникальным идентификатором отчета.

Все отчеты хранятся в определенной папке, задаваемой в конфигурации приложения.

Совокупность форм составляет конфигурацию приложения.

Форма состоит из набора управляющих элементов (контролов).

Все свойства формы представляют собой атрибуты xml файла.

Тег формы = «form»

Пример `<form caption="Расхождения по приходному заказу" id="117" permiss="A" showrecent="1">`

Таблица 4.1. Атрибуты контрола – форма (form)

Название атрибута	Возможные значения	Обязательный / необязательный	Описание
id	число	Да	Уникальный идентификатор отчета. Совпадает с именем файла отчета.
caption	Текст	Нет	Заголовок формы
permiss	Латинские буквы	Да	Определяет привилегию пользователя для пользования таблицей.
showrecent	Да/нет, Логическая константа	Нет	Определяет показывать недавние ссылки на другие страницы пользователя. По умолчанию true
Html	Комплексный тип	Нет	Описывает кусок html кода, который добавляется на страницу. Используется для улучшения визуального отображения контролов приложения и для расширения функционала приложения за счет стандартных html элементов. Для расстановки контролов web-генератора, они должны заключаться в символы ~~-идентификатор контрола~~
controls	Комплексный тип	Да	Содержит xml представление встроенных контролов генератора отчетов
popup	Да/нет, Логическая константа	Нет	Определяет показывать недавние ссылки на другие страницы пользователя. По умолчанию false

Замечания и дополнения:

Все атрибуты формы пишутся малыми латинскими буквами.

id – должен генерироваться автоматически при создании новой формы.

Логическая константа – bool значение, **здесь и ниже** может принимать значения (true/false или 1/0). Без разницы.

permis – латинские буквы, представляют собой список букв, которые соответствуют привилегии пользователя.

Комплексный тип – дополнительный узел в xml файле.

Необходимо предусмотреть возможность добавления дополнительных атрибутов в форму.

Управляющие элементы формы

Контроль типа TextBox (xml представление узла - **cswebtextbox**). Представляет собой textbox для ввода данных. Данные могут ограничиваться определенным типом. Контроль поддерживает все события. Описание событий смотри ниже.

Таблица 4.2. Свойства контроля cswebtextbox

Название атрибута	Возможные значения	Обязательный / необязательный	Описание
id	Текст	Да	Уникальный Идентификатор контроля в разрезе формы.
caption	Текст	Нет	Описание контроля
visible	Логическая константа (Да, нет)	Да	Виден ли контроль для пользователей
type	Список доступных типов для контроля см.ниже	Да	Тип данных контроля
required	Логическая константа (Да, нет)	Да	Требуется или нет ввод данных пользователя.
enable	Логическая константа (Да, нет)	Да	Определяет доступен ли контроль для редактирования или нет.
rows	Число	Нет	Количество рядов
columns	Число	Нет	Количество колонок
height	Число	Нет	Высота контроля
width	Число	Нет	Ширина контроля
alt	Текст	Нет	Пояснения в контроле, дополнительный текст для улучшения понимания

Замечания и дополнения:

Здесь и ниже указаны типы данных. Все значения – **малые буквы**.

"INT64", "CHAR", "VARCHAR", "XML", "DATETIME", "DECIMAL", "MONEY", "DOUBLE", "FLOAT", "INT", "INT32", "STRING", "INT8", "INT16", "BYTE", "BOOLEAN", "BOOL", "UNIQUEIDENTIFIER"

Контроль типа DateTime (cswebdatetimepicker) – выбор даты.

Описание событий смотри ниже.

Таблица 4.3. Свойства контроля cswebdatetimepicker

Название атрибута	Возможные значения	Обязательный / необязательный	Описание
id	Текст	Да	Уникальный Идентификатор контроля в разрезе формы.
caption	Текст	Нет	Описание контроля
visible	Логическая константа (Да, нет)	Да	Виден ли контроль для пользователей
requier	Логическая константа (Да, нет)	Да	Требуется или нет ввод данных пользователя.
enable	Логическая константа (Да, нет)	Да	Определяет доступен ли контроль для редактирования или нет.
height	Число	Нет	Высота контроля
width	Число	Нет	Ширина контроля
Alt	Текст	Нет	Пояснения в контроле, дополнительный текст для улучшения понимания

Контроль типа ComboBox (cswebcombobox) – вывод данных из списка.

Описание событий смотри ниже.

Таблица 4.4. Свойства контроля cswebcombobox

Название атрибута	Возможные значения	Обязательный / необязательный	Описание
id	Текст	Да	Уникальный Идентификатор контроля в разрезе формы.
caption	Текст	Нет	Описание контроля
visible	Логическая константа (Да, нет)	Да	Виден ли контроль для пользователей
requier	Логическая константа (Да, нет)	Да	Требуется или нет ввод данных пользователя.
enable	Логическая	Да	Определяет доступен ли контроль для редактирования

	константа (Да, нет)		или нет.
displayfield	Текст	Да	Имя поля, которое отображается
valuefield	Текст	Да	Имя поля, которое используется как значение
height	Число	Нет	Высота контрола
width	Число	Нет	Ширина контрола
Alt	Текст	Нет	Пояснения в контролу, дополнительный текст для улучшения понимания

Замечания и Дополнения:

Значения последних двух свойств берутся из запроса в событиях.

Для списка констант используется комплексный тип **options**

<options>

<option displayfield="A" valuefield="1"/>

<option displayfield="B" valuefield="2"/>

</options>

В контроле может использоваться вместе и запрос и константы.

Контроль типа Label (csweblabel) просто текст для вывода, поддерживает вывод данных из базы, см. события ниже

Таблица 4.5. Свойства контрола csweblabel

Название атрибута	Возможные значения	Обязательный / необязательный	Описание
id	Текст	Да	Уникальный Идентификатор контрола в разрезе формы.
caption	Текст	Нет	Описание контрола
visible	Логическая константа (Да, нет)	Да	Виден ли контрол для пользователей
height	Число	Нет	Высота контрола
width	Число	Нет	Ширина контрола
alt	Текст	Нет	Пояснения в контролу, дополнительный текст для улучшения понимания

Контроль типа PopUpMenu - Всплывающее меню (**menuitems**) – меню по правой кнопке мыши.

Таблица 4.6. Список параметров контроля **menuitems** – всплывающее меню

Название атрибута	Возможные значения	Обязательный / необязательный	Описание
caption	Текст	Да	Название действия
permis	Латинские буквы	Да	Привилегия пользователя для просмотра отчета
id	Текст	Да	Идентификатор отчета, куда надо перейти
type	Значения: xml – идентификатор встроенного отчета aspx – имя файла пользовательского отчета	Да	Идентификатор отчета, куда надо перейти, после выбора пункта меню
popup	Логическая константа (Да, нет)	Да	Определяет открывать ли следующий отчет в новом окне или нет

Контроль типа Button - Кнопки (**cswebbutton**)– кнопки используются для создания события submit, перехода из отчета в отчет.

Таблица 4.7. Свойства управляющего элемента **cswebbutton**

Название атрибута	Возможные значения	Обязательный / необязательный	Описание
id	Текст	Да	Уникальный Идентификатор контроля в разрезе формы.
caption	Текст	Нет	Описание контроля
visible	Логическая константа (Да, нет)	Да	Виден ли контроль для пользователей
type	Может содержать следующие значения: gpt,link,close	Да	Тип кнопки
height	Число	Нет	Высота контроля
width	Число	Нет	Ширина контроля
alt	Текст	Нет	Пояснения в контроле, дополнительный текст для улучшения понимания
target	Число	Нет	Определяет уникальный идентификатор отчета на который надо перейти после выполнения всех действий.
event	Текст	Нет	Указывает событие, которое необходимо вызывать при переходе на следующую форму

			(init, submit). По умолчанию init.
popup	Логическая константа (Да, нет)	Да	Определяет, открывать ли следующий отчет в новом окне или нет

Контроль типа Grid - Контроль таблица (cswebgrid)

Таблица 4.8. Атрибуты и параметры контроля – таблица (cswebgrid)

Название атрибута	Возможные значения	Обязательный / необязательный	Описание
id	Текст	Да	Идентификатор контроля уникальный на уровне формы
pagesize	Число	Нет	Определяет количество строк выводимых в таблице. Если данное значение не задано – берется из конфигурации приложения.
showempty	Да/нет, Логическая константа	Нет	Определяет признак видимости таблице, если она не содержит данных. По умолчанию пустая таблица не показывается
caption	Текст	Нет	Заголовок таблицы (описание)
permis	Латинские буквы	Нет	Определяет привилегию пользователя для пользования таблицей
visible	Да/нет, Логическая константа	Нет	Определяет признак видимости таблицы
height	Число	Нет	Высота контроля
width	Число	Нет	Ширина контроля
alt	Текст	Нет	Пояснения в контроле, дополнительный текст, для улучшения понимания
showheader	Да/нет, Логическая константа	Нет	Показывать заголовок таблицы. По умолчанию ДА.
showfooter	Да/нет, Логическая	Нет	Показывать подвал таблицы. По умолчанию ДА.

	константа		
query	Комплексный тип	Да	Определяет запрос, который используется для заливки таблицы
fields	Комплексный тип	Да	Определяет список полей видимых для таблицы
reports	Комплексный тип	Нет	Определяет список отчетов Crystall Reports, которые доступны для контроля

Таблица 4.9. Список параметров для узла **query** контроля cswebgrid

Название атрибута	Возможные значения	Обязательный / необязательный	Описание
id	Текст	Нет	Идентификатор запроса, который читается из конфигурации. Если пустой должен быть заполнен параметр source. (Пока не используется)
source	Текст	Нет	Запрос sql который используется для заливки данных в контрол. Если пустой, то должен быть заполнен параметр id
param	Комплексный тип	Нет	Определяет список параметров для запроса
Узел param			
id	Текст	Да	Идентификатор параметра используется как параметр в запросе. Должен быть заключен в ##id##.
value	Текст	Да	Идентификатор контроля с формы, с которого берутся значения или константа, если параметр имеет соответствующий тип.
type	Список из значений web, const,sys	Да	Тип параметра. Отслеживаются три типа web – параметр, который берется со страницы, const – константа sys – системный параметр, определяемый при авторизации пользователя. Идентификатор начинается с «SYS_»
format	Текст	Нет	Значение, которое надо обернуть Например «116?po_num={0}&ropur=1»

Таблица 4.10. Список параметров для узла **reports** контроля cswebgrid

Название атрибута	Возможные значения	Обязательный / необязательный	Описание
id	Текст	Да	Идентификатор отчета Crystall reports
reportfile	Текст	Да	Название отчета
caption	Текст	Да	Заголовок отчета
visible	Да/нет, Логическая константа	Да	Виден ли отчет в конфигурации или нет
permmiss	Латинские буквы	Да	Привилегия пользователя для просмотра отчета
type	Список из значений web, Const	Да	Тип параметра. Отслеживаются два типа
query	Комплексный тип	Да	Определяет запрос, который используется для заливки отчета (описание смотри таблицу выше)

Таблица 4.11. Список параметров для узла **fields** контроля cswebgrid

Название атрибута	Возможные значения	Обязательный / необязательный	Описание
id	Текст	Да	Имя колонки из таблицы
caption	Текст	Да	Название колонки
visible	Да/нет, Логическая константа	Да	Определяет признак видимости таблице
link	Текст	Нет	Идентификатор отчета который вызывается при нажатии на колонку
popup	Да/нет, Логическая константа	Нет	Определяет показывать недавние ссылки на другие страницы пользователя в новом окне. По умолчанию false. Используется только если определен с параметром Link.
sumtype	Текст. Принимает значения (blank,sum,count)	Нет	Типы суммирования для колонок таблицы blank – суммирование нет sum – суммирование всех значений колонки count – подсчет всех строк колонки

param	Комплексный тип	Нет	Определяет список параметров, передаваемых в отчет
event	init	Нет	Имя события, которое используется при переходе на другой отчет. Если значение не указано = submit, если = init происходит переход на обновление данных по строке.
Узел param			
id	Текст	Да	Название параметра в запускаемом отчете
value	Текст	Да	Имя колонки, из которой берется значение параметра

Элементы управления Ajax

Контроль типа AjaxComboBox (cswebajaxcombobox) – динамический вывод данных списка в асинхронном режиме.

Поддерживает только события обновления (change).

Для корректной работы контроля необходимо чтобы в параметрах контроля была ссылка на сам идентификатор контроля, чтобы динамически менялось содержимое.

Работает только с параметрами только текст. С параметрами других типов работает только на условии «=>»

Таблица 4.12. Свойства контроля cswebajaxcombobox

Название атрибута	Возможные значения	Обязательный / необязательный	Описание
id	Текст	Да	Уникальный Идентификатор контроля в разрезе формы.
caption	Текст	Нет	Описание контроля
visible	Логическая константа (Да, нет)	Да	Виден ли контроль для пользователей
requier	Логическая константа (Да, нет)	Да	Требуется или нет ввод данных пользователя.
enable	Логическая константа (Да, нет)	Да	Определяет, доступен ли контроль для редактирования или нет.
displayfield	Текст	Да	Имя поля, которое отображается
valuefield	Текст	Да	Имя поля, которое используется как значение. Это значение передается другим контролам. И вводится в текстовом окне.
height	Число	Нет	Высота контроля

width	Число	Нет	Ширина контрола
alt	Текст	Нет	Пояснения в контролу, дополнительный текст, для улучшения понимания

```
<cswebajaxcombobox id="login" caption="Выберите пользователя: " enable="true" type="string"
visible="true" requier="1" displayfield="name" valuefield="id" height="" width="" alt="">
```

```
<change>
```

```
<query id="test" source="select id,name from t_user (nolog) where id like '##login##%'">
```

```
<param id="login" value="login" type="web" format="" />
```

```
</query>
```

```
</change>
```

```
</cswebajaxcombobox>
```

Контрол типа Menu - Контрол меню приложения.

Используется для генерации меню приложения. Содержится в xml файле webmenu.xml.

Меню может поддерживать неограниченную вложенность.

Пример XML TreeView:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<webmenu>
```

```
<Menu id='1' name='Управление Складом' alt='Система управления складом (Base)' permiss='A'>
```

```
<Menu id='3' name='Товарный запас' alt='Товары на складе' permiss='AB'>
```

```
<Menu id='130' name='Товарный справочник' alt='Справочник товаров на складе' type='xml' popup='0'
permiss='A' />
```

```
<Menu id='136' name='Запас по товару' alt='Товарный запас по товару' type='xml' popup='0' permiss='A' />
```

```
<Menu id='138' name='Запас по ГМ' alt='Товарный запас по ГМ' type='xml' popup='0' permiss='A' />
```

```
<Menu id='140' name='Запас по Ячейке' alt='Товарный запас по Ячейке' type='xml' popup='0' permiss='A'
/>
```

```
<Menu id='142' name='Запас по Серийному номеру' alt='Товарный запас по серийному номеру' type='xml'
popup='0' permiss='A' />
```

```
<Menu id='Default5.aspx' name='Отчет по товарам ASPX' alt='Подключение отчета aspx' type='aspx'
popup='0' permiss='A' />
```

```
</Menu>
```

```
</Menu>
```

```
</webmenu>
```

Описание конфигурации menu – узел описывающий пункт меню

Аттрибуты меню

- name – Название как будет выглядеть в дереве
- alt - описание, появляется при наведении мышки
- id - уникальнны идентификатор меню (отчета). Если используются стандартные отчеты из xml – то число, если подключаются отчеты из aspx файла то имя файла.
- type – тип отчета (используется aspx и xml). Если значение type отсутствует считается что это просто подменю приложения.
- permiss – привилегия меню пользователя. Одна или несколько букв латинского алфавита.
- rorup – признак открытия отчета в новом окне, логическая константа.

4.3.2. Описание глобальной конфигурации приложения

Файл конфигурации приложения содержит глобальные настройки генератора, которые могут меняться и добавляться.

Необходимо предусмотреть возможность добавления в файл бесконечного количества настроек, их редактирования, удаления.

Пример файла

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <add key="connectionstring" value="Data Source=192ю168ю0ю14;Initial Catalog=cwms; Asynchronous
Processing=true;MultipleActiveResultSets=True;User Id=sa;Password=123456" />
  <add key="dateformat" value="dd.MM.yyyy" />
  <add key="dbtype" value="sql" />
  <add key="pagerows" value="20" />
  <add key="timeout" value="1000" />
  <add key="showreloadbtn" value="1" />
  <add key="crystallpath" value="C:\" />
</configuration>
```

Пока поддерживаются следующие настройки:

- connectionstring – строка соединения с базой данных
- dateformat - пользовательский формат даты
- dbtype – тип базы данных (принимает значения – oracle, oledb, odbc, sql(по умолчанию))

- pagerows – количество строк на страницы для контроля таблицы.
- timeout – timeout пользователя.
- showreloadbtn – показывать кнопку перезагрузки конфигурации приложения.
- **crystallpath – имя папки с отчетами crystall reports (устаревшее значение)**
- maxrows – максимальное количество рядов, которые выводятся в таблице

4.3.3. Описание специфических настроек системы

Эти настройки находятся в файле конфигурации OhewebDataConfig.xml. И представляют собой predefined настройки для данных в отчетах.

Пример файла конфигурации

```
<?xml version="1.0" encoding="utf-8"?>
<OhewebData>
  <languages>
    <language id="ru" name="Russian" />
    <language id="en" name="English" />
  </languages>
  <checkcolumns>
    <checkcolumn table="v_wms_user" column="user_id"/>
    <checkcolumn table="v_wms_user" column="user_id"/>
    <checkcolumn table="v_wms_pohead" column="po_id"/>
    <checkcolumn table="t_trancode" column="tran_num"/>
    <checkcolumn table="v_wms_sohead" column="so_id" />
  </checkcolumns>
</OhewebData>
```

На данный момент используются следующие структуры, см. пример.

- language список языков, поддерживаемый приложением.
- checkcolumn – список таблиц и колонок контроля таблица(грид) для которых используется контроль checkbox для обработки нескольких строк грида.

Поддерживаемые атрибуты

Table – имя таблицы

Column – название колонки.

Необходимо иметь возможность добавления, редактирования, удаления соответствующих управляющих элементов.

4.3.4. Описание событий, поддерживаемых контролами генератора web отчетов

Контролы web генератора могут поддерживать несколько событий.

- Событие инициализации – init
- Изменения – change
- Событие выполнения – submit

Список элементов, поддерживающих события следующий:

- Textbox – все события
- Combo – все события
- DatePicker – все события
- Button – событие submit
- Ajax – событие change

Все остальные контролы приложения не поддерживают событий.

Все события контролов описаны в конфигурации и заключаются в комплексный тип.

См. пример.

```
<init>
  <query source ="select '##start##'">
    <param id="start" value="start_date" type="web"/>
  </query>
</init>
<change>
  <query source ="select getdate()-30" />
</change>
<submit>
  <query source ="select getdate()-30" />
</submit>
```

Где

query(source) – узел определяющий запрос (храняемая процедура, или кусок кода SQL), который необходимо выполнить.

param – список параметров, которые надо передать в запрос. Описание параметров аналогична параметрам в контроле grid.

Все параметры используемые в запросе должны заключаться в символы: ##

4.3.5. Требования к разработке хранимых процедур

Все хранимые процедуры, используемые в генераторе отчетов, могут возвращать результат: запрос, константу, параметр.

Признаком успешного выполнения процедуры является возвращение таблицы, константы или параметра, которые не содержит колонки с названием **error!!!**

Если хранимая процедура возвращает колонку `error`, то система понимает, что в результате выполнения обращения к базе данных произошла логическая ошибка и генерирует исключение с описанием этой ошибки.

Предопределенные хранимые процедуры

К списку предопределенных хранимых процедур можно отнести процедуру авторизации пользователей в системе – **sp_wed_user_logon**.

Эта процедура жестко зашита в приложении и принимает три параметра

- идентификатор пользователя в системе
- пароль пользователя
- дополнительный параметр.

Результатом работы этой процедуры является таблица со списком параметров (имена колонок) начинающихся на "SYS_" который в дальнейшем используется в сессии пользователя как глобальные параметры сессии пользователя.

4.3.6. Передача параметров между формами.

Перемещение между формами может происходить несколькими способами:

1. При помощи полей ссылок на гриде (поля типа `link`). Поддерживаемые события `init`, `submit`.
2. При помощи кнопок на форме (кнопки типа `prt` – переход сразу на другую форму, кнопки `submit` – переход на форму после выполнения `submit` формы)
3. Ссылка меню пользователя.

При переходе из формы в форму способом 1 список параметров с родительской формы может браться только из таблицы (из запроса). Параметры берутся даже в том случае, если таблица не содержит видимых колонок, но колонка возвращается в запросе.

При передаче параметров способом 2 генератор отчетов берет все имена параметров на текущей странице и подставляет в хранимую процедуру к событию `submit`. Кроме этого метод использует все скрытые параметры, которые находятся на текущей странице и возможно, пришли с предыдущей. Рекомендуется использовать этот тип передачи, если необходимо передавать параметры между отчетами. Используется как основной тип обмена данными между страницами.

Для передачи параметров между страницами, имена параметров, передаваемых способами 1,2 должны соответствовать именам параметров в запросах и именам контролов поиска для формы назначения (смотри ограничения в названиях параметров).

При использовании способа 3 считается, что форма пришла первый раз и никаких параметров не передается.

4.3.7. Ограничения в определении названия параметров форм

Платформа использует несколько предопределенных контролов, которые нельзя использовать в качестве названий пользовательских элементов. Ниже приведен список предопределенных контролов.

Кроме этого не рекомендуется использовать названия контролов совпадающие с предопределенными названиями параметров Internet Explorer (IE)

№ п.п.	Имя	Назначение
1	added	Checkbox добавления нового фильтра в таблица
2	filtertype	Условие для фильтра таблицы (больше, меньше и т.п.)
3	filter	Значение фильтра таблицы
4	filterbtn	Кнопка фильтра таблицы
5	close	Кнопка закрытия окна
6	hidscroll	Позиция скроллинга по вертикали
7	hidscrollleft	Позиция скроллинга по горизонтали
8	length	Длина (Используется как системный параметр IE)
9	width	Ширина (Используется как системный параметр IE)
10	height	Высота (Используется как системный параметр IE)

4.3.8. Требование к написанию хранимых процедур и запросов

Ограничения

Приложение не поддерживает хранение возвращаемых значений из хранимой процедуры в глобальном Кеше. Поэтому при необходимости отображения возвращаемых параметров необходимо использовать контролы (таблица, текст, комбобокс).

Хранимые процедуры должны выполнять следующие требования:

1. Не содержать параметров output.

Хранимые процедуры могут использоваться для проверки параметров поиска на форме. Все проверки осуществляются средствами sql с использованием хранимых процедур.

Допускаются следующие утверждения:

Возвращается **true**, если есть хотя бы одна строка в запросе и отсутствует колонка с названием **error**, иначе возвращается **false**

Если хранимая процедура не возвращает ни одной строки. Или есть колонка с названием error. При наличии этой ошибки, она выводится в окно, и видна при наведении мыши на контрол. Контрол подсвечивается красным цветом.

Все события проверки корректности введенных параметров срабатывают только после нажатия на кнопку поиска (submit)

4.3.9. Генерация отчетов Crystall Reports с использованием генератора отчетов

Все отчеты генерируются из таблицы с данными. Нет возможности генерации отчета без использования таблицы.

Для генерации отчета используется событие crystall таблицы. При генерации этого события выводится список доступных отчетов для грида и пользователь может выбрать один из отчетов для вывода на экран.

Если для грида используется один отчет, то он автоматически генерируется на экран.

Для генерации отчета используется специальная страница **OneWebReports.aspx**. При вызове в нее передаются два параметра – имя отчета и запрос для генерации данных (может быть хранимая процедура).

Из этой страницы генерируется DataSet, который впоследствии заливается в отчет с данными.