# Картографическая подсистема ГИС.

# Описание, предназначение, принципы работы – для разработчиков.

Наименование проекта – **ГИС-подсистема** или **Картографическая подсистема**

Проект в TFS – **$\ГИС (Cool\Common)**

Caseberry – **ИИС\ГИС**

Wiki – [**http://wiki.ics.perm.ru/GIS.Default.aspx**](http://wiki.ics.perm.ru/GIS.Default.aspx)

ГИС (географическая информационная система) - это система сбора, хранения, анализа и графической визуализации пространственных данных и связанной с ними информацией о необходимых объектах ([Википедия](http://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0)).

# Зачем проект нужен?

Картографическая подсистема представляет собой компонент аналитической системы, используемый для отображения, управления, создания и анализа географической информации.

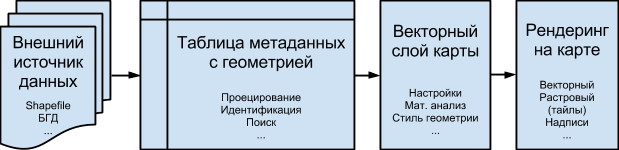
Основное предназначение системы – это создание и настройка интерактивных карт на основе имеющихся геоинформационных данных о районах края и города, водных ресурсах, кварталах, улицах, зданиях, территориях предприятий, автодорогах, железнодорожных и трамвайных путях и т.д.

Но никому не интересно, что карта может отображать домики, улицы и речки. Для этого подойдут 2GIS, Яндекс.Карты и пр.

Особенность нашего проекта в том, что на карте можно отображать объекты шторма (DataObject). Более того, не просто отображать, а делать особые визуализации, такие как количественный анализ, анализ по категориям, визуализация в виде диаграмм и интерполяция. Т.е. размер и изображение точки на карте могут динамически определяться в зависимости от значения какого-либо свойства прикладного объекта.

# Как работает?

В основе лежит opensource-библиотека [SharpMap](http://sharpmap.codeplex.com/), которая на основе стандартных источников данных ГИС ([shape file](http://en.wikipedia.org/wiki/Shapefile), база геоданных и др.) умеет отрисовывать карту в System.Drawing.Image.



Общая последовательность программных модулей, участвующих при рендеринге стандартного картографического слоя.

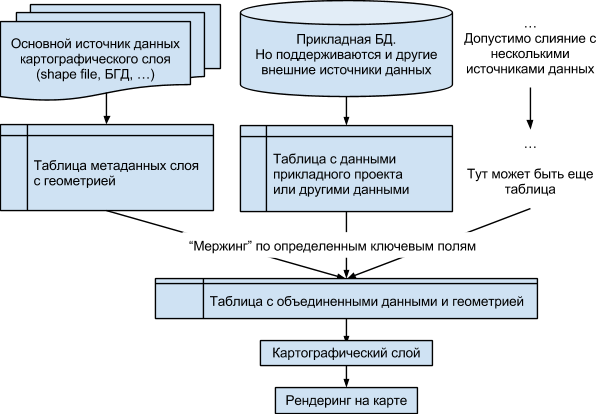
Эта диаграмма охватывает большую часть кода “ядра” ГИС. Большинство из этих фич были реализованы в рамках проекта. Остальная часть, которая была уже реализована в SharpMap (чтение внешних данных, проецирование, задание стиля отображения геометрий, рендеринг) была сильно доработана и изменена.

Прошу обратить внимание, что вышеприведенная диаграмма сильно упрощена, чтобы отразить суть рендеринга стандартного картографического слоя. Для отображения слоя с прикладными объектами данных необходимы изменения.

Первое основное “вмешательство” нашего кода происходит на стадии формирования таблицы геоданных (таблицы метаданных с геометрией). По определенным настройкам, заранее заданными в проекте ГИС, прикладная таблица данных и таблица данных слоя (из внешнего источника) соединяются.

Например, таблица “Заявки” из проекта Новогор соединяется с таблицей векторного слоя “Дома жилые” по совпадающим значениям в столбцах Район, Улица, НомерДома и т.п. В итоге получается одна большая таблица с наличием прикладных данных и столбца геометрии. За работу этого на первый взгляд немудреного механизма ответственно огромное количество кода в проекте. Этот механизм очень гибок и имеет большое количество настроек.

Так сложилось, что данный механизм (соединение слоя с прикладными данными и отображение полученного слоя на карте) называется построением или отображением отчета на карте. А настройки, на основе которых соединяются данные - соединениями отчета.



Объединение данных при построении отчета на карте.

Второе по важности “вмешательство” нашего кода в механизм SharpMap происходит при рендеринге карты.

Отображение векторного слоя на карте гибко настраивается: от статических иконок для точечных данных и заливок для поверхностей до математически вычисленных цветов заливок и размеров иконок. На основе настроек слоя может быть выполнен статистический анализ прикладных данных (количественный анализ слоя) и анализ по атрибутам (анализ по категориям), а также данные слоя могут быть визуализированы в виде диаграмм (2D и 3D).

В паре с векторным слоем, может быть отображен слой надписей. Надписи также имеют большое количество настроек: стиль шрифта, задание шаблона надписи с подстановкой прикладных данных, динамическое вычисление положения и размера надписи в зависимости от масштаба карты и расположения других надписей, ….

# Особенности Win-приложения ГИС.

1. Векторный рендеринг в System.Drawing.Image на winforms средствами GDI.
2. Много инструментов карты.
3. Создание проекта карты, настройка проекций, добавление новых слоев.
4. Настройка слоев, карты, отчетов, их соединений и пр.
5. …

# Особенности Web-компонента ГИС.

## Клиентская часть.

1. Для отрисовки карты используется opensource-библиотека leaflet (JS).
2. Легко встраиваемый WebControl = JS + CSS + JQuery
3. Основные инструменты карты из Win-приложения
4. На данный момент проект карты и слои не настраиваются и не сохраняются. Используется только для отображения и построения отчетов. Вся настройка производится в Win-приложении.

## Серверная часть.

Серверная часть – это WCF-сервис (**MapInteractionService**), к которому обращается клиентский код с целью получения картографических данных, либо выполнения каких-либо операций с картой.

В состав WCF-сервиса входит WMS-сервис, к которому обращается leaflet для получения тайлов (растровые кусочки карты).

WMS-сервис соответствует основным стандартам 1.1.1 и 1.3.0, и поэтому он хорошо понимает запросы leaflet. WMS парсит GET-запросы вида *?getMap&location=...&zoom=...&...* , рендерит соответствующий кусок карты с помощью SharpMap (точно также, как это происходит в Win-приложении) и возвращает его как растровую картинку. По такому-же принципу (отрисовка растровыми кусочками - тайлами) реализованы Google.Maps и Яндекс.Карты.

Когда клиентскому коду нужны какие-либо данные, делается ajax-запрос к WCF-сервису.

WCF-сервис может быть развернут в IIS (рекомендуется на отдельном узле от основного сайта, на котором расположен веб-компонент карты, чтобы разделить нагрузку), а также есть возможность захостить его в Windows-службе.

Исходный код, солюшены и проекты.

Рассмотрим содержимое папки **$/ГИС/Main** в TFS проекта:

* **Lib** – наши сборки (IIS.\*), которые используют проекты ГИС: IIS.Business, IIS.Charting, IIS.Core, IIS.OLAP, IIS.WinUI и другие. А также сторонние библиотеки и утилиты в папке 3rdParty.
* **Source** – исходный код.
* **STORM.NET** – Caseberry, технологические сборки.
* **Скрипты** – набор скриптов для создания таблиц, в которых будут храниться различные настройки подсистемы. Работы карты без этих таблиц невозможна.

В папке **Source** располагаются 3 решения:

* **GIS(Windows Application).sln** – windows-приложение ГИС. Используется для настройки объектов ГИС и проектов карт. Может использоваться как пример подключения карты к прикладному windows-приложению.
* **GIS(ASP.NET Application).sln** – тестовое web-приложение ГИС. Используется при разработке web-компонента карты. Может использоваться как пример подключения карты к прикладному web-приложению.
* **GIS(Service).sln** – web-приложение с сервисами ГИС. Содержит в себе WCF-сервис (MapInteractionService) для доступа к данным карты.

Классы объектов и форм для настроек ГИС-подсистемы располагаются в пространстве имен **IIS.GIS** (папка \Source\IIS.GIS\). Они генерируются с помощью Caseberry.

Классы картографической части располагаются в пространствах имен **SharpMap** и **IIS.Map\*** (папка \Source\IIS.Map\).

**Проекты IIS.GIS**

Большая часть параметров конфигурации картографического компонента хранится в настройках GIS. Настройки GIS представляют собой объекты данных с префиксом “GIS\_” и генерируются из Caseberry. Объекты содержатся в сборке **IIS.GIS(Objects)**, сопутствующие им формы и бизнес-серверы – в сборках **IIS.GIS(Forms)** и **IIS.GIS(BusinessServers)** соответственно.

В настройках GIS активно используются общие настройки объектов приложения из сборок IIS.Core и распространение библиотек IIS.GIS без IIS.Core невозможно.

Также для некоторых классов настроек ГИС ведется аудит, поэтому существует зависимость от сборок IIS.Аудит.

**Проекты IIS.Map**

* **IIS.Map.Common** – содержит типы данных, вспомогательные классы и классы-расширения использующиеся в windows-приложении и серверной части web-приложения. Проект независим и ссылается только на типы в системных пространствах имен.
* **IIS.Map.Common.UI** – содержит элементы windows-интерфейса, преимущественно на основе технологии WPF.
* **IIS.Map** – наряду с SharpMap, это основная библиотека картографической подсистемы. В ней содержится код, отвечающий за обработку данных и настройку картографических слоев, класс проекта карты, средства для построения отчетов на карте и прочее.
* **IIS.Map.WinUI** – формы и контролы, а также инструменты карты в windows-приложении.
* **IIS.Map.Reports(BusinessServers, Objects, Forms)**

**IIS.Map.Reports(BusinessServers)** содержит класс для генерации отчетов карты (соединение прикладных данных с данными картографического слоя) на основе настроек, определенных в **IIS.Map.Reports(Objects)**, заданных на формах из **IIS.Map.Reports(Forms)**.

* **IIS.Map.Services** – содержит сервисы для доступа к проекту карты и работы с данными ГИС. Содержит WCF-сервис MapInterationService для работы веб-компонента карты.
* **IIS.Map.Web** – содержит определения некоторых классов для работы карты в веб-окружении.
* **IIS.Map.Windows.Services** – содержит классы windows-служб для хостинга сервисов из IIS.Map.Services. Предназначается для того, чтобы развернуть сервисы ГИС вне IIS.

**Компонент SharpMap**

Это сторонний компонент с открытым исходным кодом (<http://sharpmap.codeplex.com/>).

За время разработки картографической подсистемы в код SharpMap было внесено много изменений, и на данный момент обновление библиотеки напрямую невозможно.

**SharpMap, SharpMap.UI** – основные проекты картографической подсистемы. SharpMap содержит в себе: компонент карты, wms-сервис, средства рендеринга карты, инструменты для работы с файлами слоев и их проекциями. В SharpMap.UI определен компонент карты для встраивания в windows-приложение.

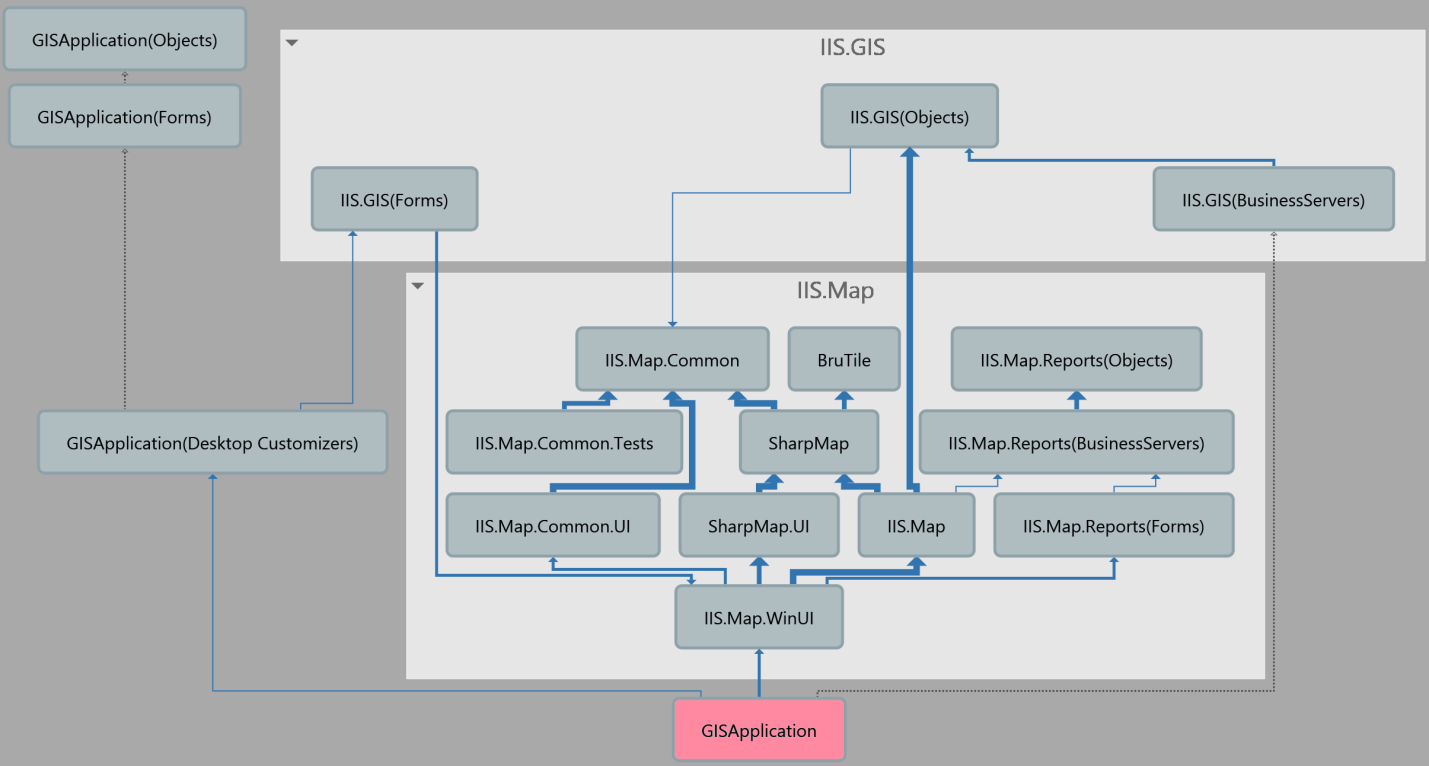


Диаграмма зависимостей проектов решения GIS(Windows Application).sln.

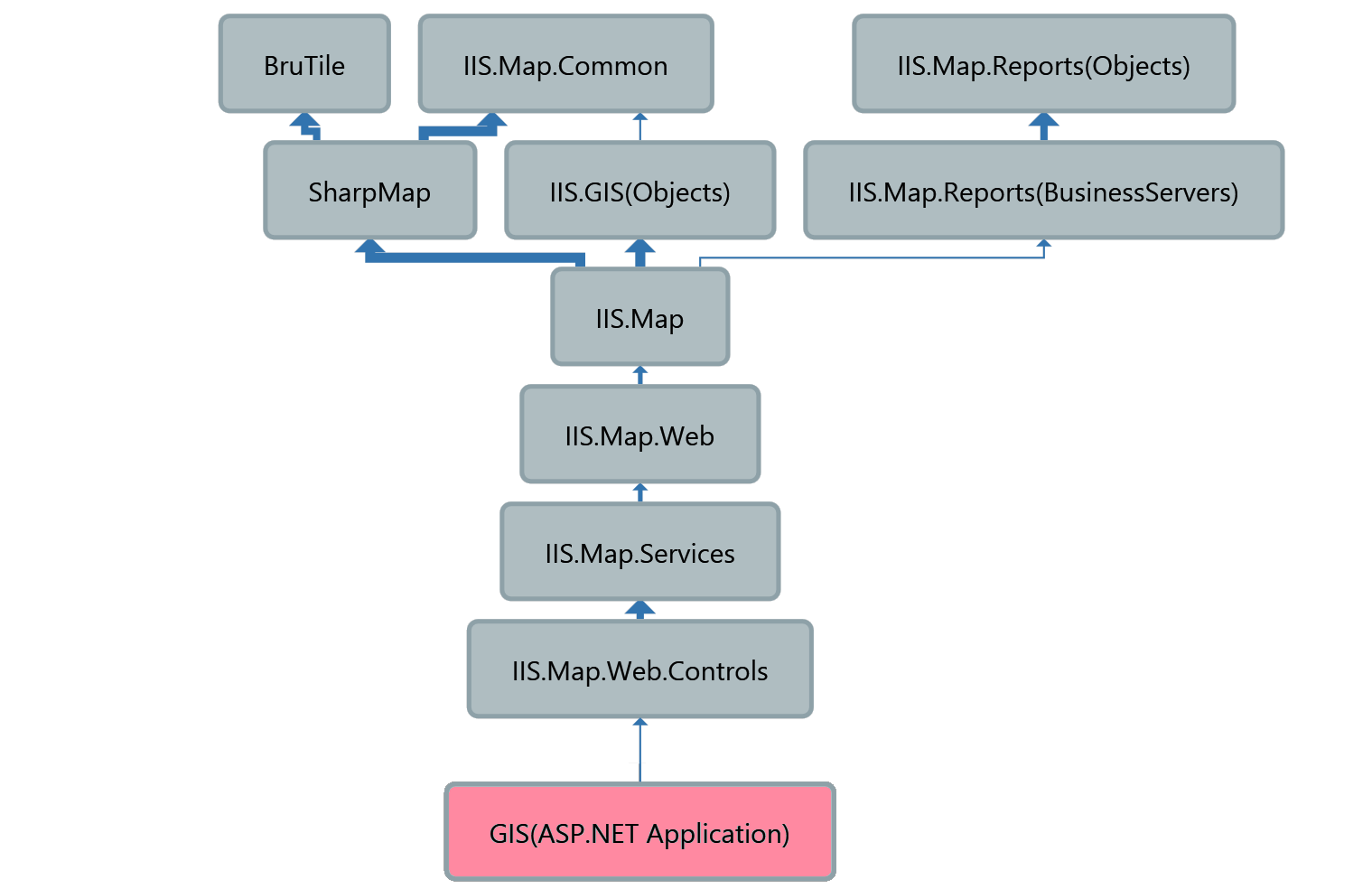


Диаграмма зависимостей проектов решения GIS(ASP.NET Application).sln.

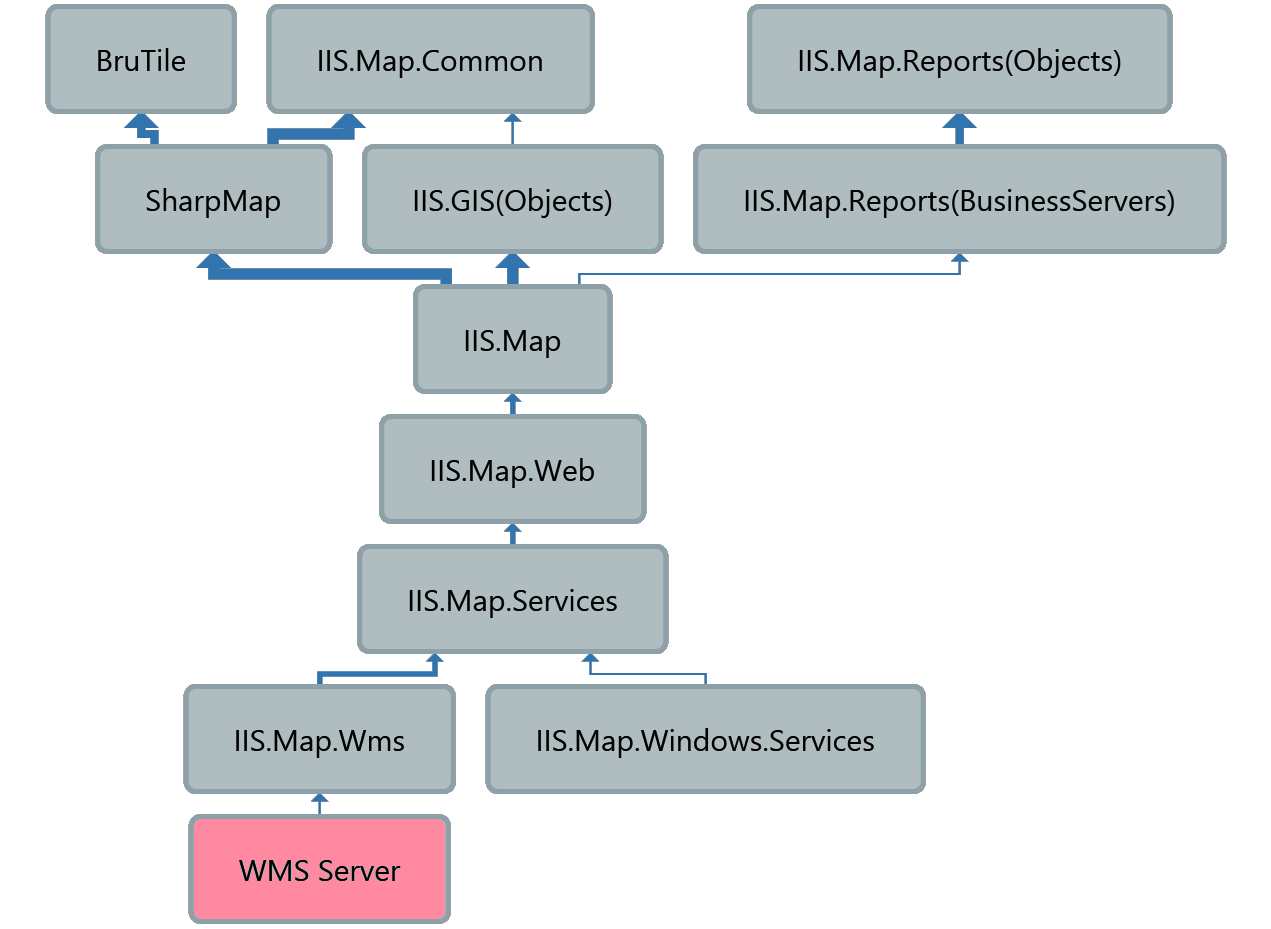


Диаграмма зависимостей проектов решения GIS(Service).sln.

WMS Server – устаревшее название, сейчас проект именуется GIS(Service).

Проект IIS.Map.Wms устаревший, сейчас он поглощен проектом IIS.Map.Services.

Работа с проектами, отладка.

Для каждого солюшена определены серверные билды: GIS(Windows Application), GIS(ASP.NET Application), GIS(Service) в папки [\\RU\Builds\GIS(Windows Application)](file:///\\RU\Builds\GIS(Windows%20Application)), [\\RU\Builds\GIS(ASP.NET Application)](file:///\\RU\Builds\GIS(ASP.NET%20Application)) и [\\RU\Builds\GIS(Service)](file:///\\RU\Builds\GIS(Service)) соответственно. Запускаются вручную (после чекина они не запускаются автоматически).

Если были внесены изменения в код проектов IIS.Map, IIS.GIS или SharpMap, то после чекина нужно обязательно запустить все 3 серверных билда и убедиться, что они завершатся успешно.

Все 3 решения (win-приложение, web-приложение и web-сервис) настроены на работу с БД GIS\_TEST на сервере STORM\SQL2008R2.

Тестовое web-приложение доступно по адресу ru:5200. Логин – root, пароль – 1. Сервисы разворачиваются на ru:5201. После успешного серверного билда узел обновляется автоматически.

Открытие карты в вин-приложении: Карты – Проекты карт – Выбрать карту из списка – кнопка Запуск (либо двойной щелчок, либо Enter).

Открытие карты в веб-приложении: Залогиниться (root – 1), Карты – Проекты карт – Щелкнуть по записи в списке.

GIS(ASP.NET Application) локально запускается на порту 21295, GIS(Service) – на 21296.

~~Веб-компонент карты по умолчанию настроен так, что сервисы он будет искать по адресу, прибавив 1 к текущему порту. Можно указать адрес сервисов вручную в компоненте HtmlMap на asp.net-странице, либо дописав в url “~~*~~&interactionService=http://[host:port]/Services/MapInteractionService.svc&wmsService=http://[host:port]/Services/MapInteractionService.svc/wms”~~*~~.~~

~~Это бывает полезно, если требуется отладить только код в проекте GIS(ASP.NET Application), либо только сервисы – т.е. нет надобности запускать оба проекта у себя на компе.~~

Для локальной отладки нужно в проекте GIS(ASP.NET Application) в web.config указать адрес конечной точки сервиса localhost:21296 вместо ru:5201 (system.serviceModel\client\endpoint address=” http://localhost:21296/Services/MapInteractionService.svc”).

Если при построении какого-либо проекта возникает окошко, сообщающее о незарегистрированной копии компонента ComponentOne:

* нажать OK и проект сбилдится без каких-либо ограничений. Но при каждом построении это окошко будет вам надоедать.
* взять лицензионный ключ в [\\storm\Storm2008\Distribute\Required NET Components\C1\Official purchased](file:///\\storm\Storm2008\Distribute\Required%20NET%20Components\C1\Official%20purchased).