

Rover Client Programming Documentation

Version 1.8
Nov 15th, 2010

1	Overview	2
2	Source Code License	2
3	RoverClient.....	4
3.1	Setting up development system	4
3.1.1	Retrieving Source Code.....	4
3.1.2	Development Environment.....	4
3.1.3	Install Visual Studio Solution.....	4
3.1.4	Running Project from the IDE.....	4
3.1.5	Deploying Project (building CAB file for distribution)	5
3.2	Source Code Structure	5
3.2.1	Rover	5
3.2.2	GPS_Util.....	7
3.2.3	RoverMaster	7
3.2.4	ROVERClientCABbuild	7
3.2.5	RoverCamera.....	8
3.2.6	RoverRelease.....	8
3.3	Making GUI Changes.....	8
3.4	XML Site File Handling.....	12
3.4.1	TAGS.....	13
4	Rover Server	16
5	Source Code Contributions to Group	19
6	Appendices	20
6.1	Sitefile XML example	20

1 Overview

Overview Text

2 Source Code License

Below we include the text of our source code use license. We reserve the right to change the terms of this license at any time.

```
//  
// =====  
// Copyright (C) 2010 ROVER Development Partners Foundation.  
//  
// Redistribution and use in source and binary forms, with or without  
// modification, are permitted provided that the following conditions  
// are met:  
// 1. Redistributions of source code, or portions of this source code,  
// must retain the above copyright notice, this list of conditions  
// and the following disclaimer.  
// 2. Redistributions in binary form must reproduce the above copyright  
// notice, this list of conditions and the following disclaimer in  
// the documentation and/or other materials provided with the  
// distribution.  
// 3. All advertising materials mentioning features or use of this  
// software must display the following acknowledgment:  
// "This product includes software developed by ROVER Development  
// Partners Foundation (http://www.roverfound.com)."  
// 4. If the software is provided with, or as part of a commercial  
// product, or is used in other commercial software products the  
// customer must be informed that "This product includes software  
// developed by ROVER Development Partners Foundation  
// (http://www.roverfound.com)"  
// 5. The names "ROVER Development Partners Foundation" and "ROVER"  
// must not be used to endorse or promote products derived from  
// this software without prior written permission. For written  
// permission, please contact "info@roverfound.com".  
// 6. Products derived from this software may not be called "ROVER"  
// nor may "ROVER" appear in their names without prior written  
// permission of ROVER Development Partners Foundation.  
// 7. The following requirements apply to any modification to the  
// source code that, once compiled, is later installed on more than  
// one client or more than one server. Any such modification must be  
// submitted back to ROVER Development Partners Foundation or its  
// assignee, via its Subversion (SVN) revision control system, or other  
// repository identified by the ROVER Development Partners or its  
// assignee. Any such modification must be royalty free and becomes  
// covered by the present license, unless a different license is agreed  
// in writing by the contributor and the ROVER Development Partners. Any  
// such modification must be accompanied by documentation briefly  
// explaining the objective of the modification and demonstrating that the  
// modification works, meeting its objective without impairing the  
// functionality of the combined code.  
// 8. Redistributions of any form whatsoever must retain the following  
// acknowledgment:  
// "This product includes software developed by ROVER Development  
// Partners Foundation (http://www.roverfound.com/)."  
// THIS SOFTWARE IS PROVIDED BY ROVER DEVELOPMENT  
// PARTNERS FOUNDATION. "AS IS" AND ANY EXPRESSED OR IMPLIED  
// WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES  
// OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE  
// DISCLAIMED. IN NO EVENT SHALL ROVER DEVELOPMENT PARTNERS  
// FOUNDATION. OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,  
// INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES  
// (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR  
// SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
// HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
```

```
// STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
// ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
// OF THE POSSIBILITY OF SUCH DAMAGE.
// =====
// A current version of the software can be found at
// http://www.roverfound.com
// Bug reports and comments should be directed to
// ROVER Development Partners Foundation at info@roverfound.com
// =====
//
```

3 RoverClient

3.1 Setting up development system

How to retrieve source code from the internet, and how to setup a complete development environment is beyond the scope of this document. However, we provide starting points for the user here.

3.1.1 Retrieving Source Code

Rover Client is written using C#, which is Microsoft's proprietary version of the Java Programming Language.

The source code may be requested by contacting the Rover Development Partners at rdp@xxxxxxxx.org. Alternatively you may request a developer account at our web site www.ourweb.org. Once you have a login on the system, you will be able to directly access our software repository.

3.1.2 Development Environment

The ROVER Client was developed using Microsoft Visual Studio 2005 – Professional Edition Integrated Development Environment (IDE). Newer versions of MSVS Professional should work fine with this software. We have not tested with versions of MSVS prior to 2005, nor with versions lower than Professional. The free version of Visual Studio may work, but we have not verified that. There are some reports that the free versions may not include a software debugger.

Please let us know what IDE's work for you and we will update this documentation appropriately. Remember, this is a community effort; we need your help documenting functional development platforms.

3.1.3 Install Visual Studio Solution

Typically a VS Solution can be installed by placing the distribution folder inside of your *projects* folder under your Visual Studio Folder.

3.1.4 Running Project from the IDE

The main project in our Solution is the RoverMaster. To run the project manually, right click on the RoverMaster item in the SolutionExplorer, and select 'Debug' -> 'Start New Instance'. This will compile the program and install and run it on either the emulator or an actual phone connected to the PC via Activesync.

3.1.4.1 Selecting Emulator vs. Phone for testing.

Right click on the RoverMaster item in the SolutionExplorer, and select 'Properties'. The Property tab for RoverMaster should open in the center of the IDE. Click on 'Devices' on the left hand tab of the RoverMaster Property window. Under 'Deployment Options' ->

‘Target Device’ you will find many choices. The ones of interest are ‘Windows Mobile 5.0 Pocket PC Device’ or ‘Windows Mobile 5.0 Pocket PC Emulator’.

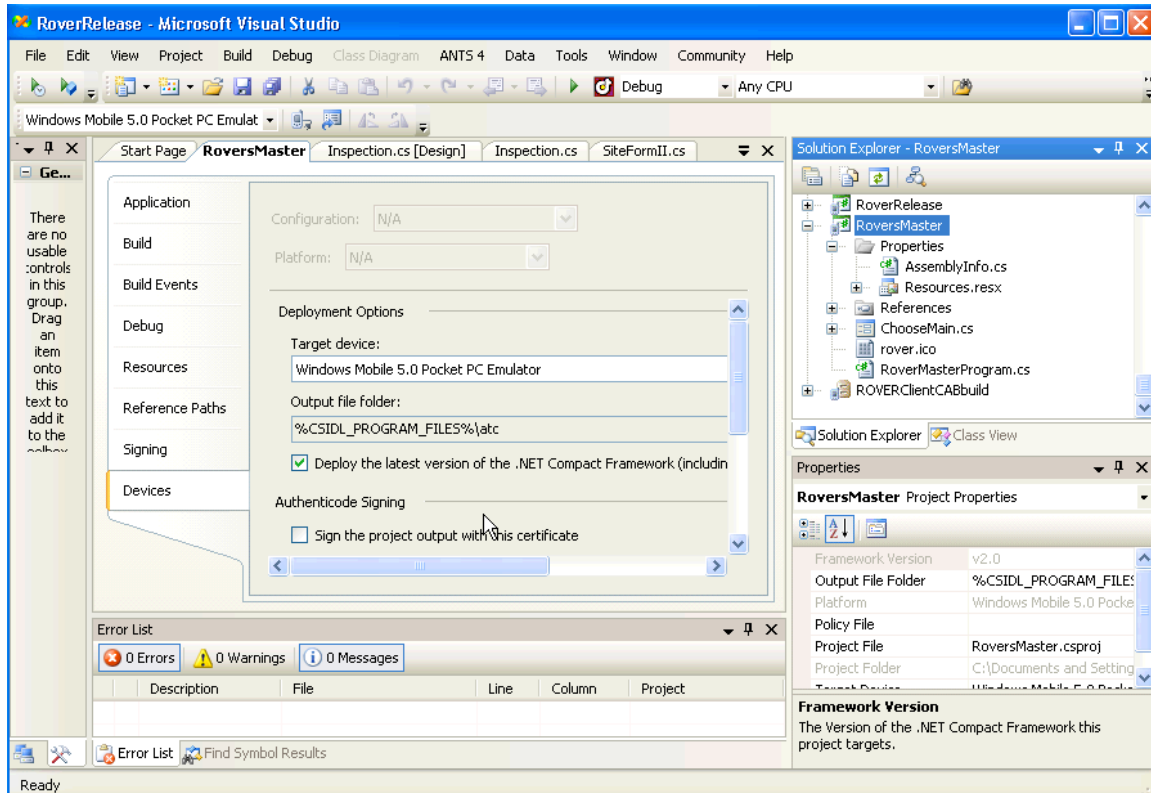


Figure 1: Setting the Target Device to the Emulator

3.1.5 Deploying Project (building CAB file for distribution)

For instructions on building and deploying the CAB file for this tool, please see section 3.2.4, ROVERClientCABbuild below.

3.2 Source Code Structure

The main Solution in Visual Studio for the ROVER Client is called RoverRelease. Within this Solution there are a number of inter-related projects. Below are the projects contained in this solution.

3.2.1 Rover

The Rover project holds most of the functionality of the project.

Most of the files held in the project are in current use. Some are reserved for future work.

As this is an open source project, any developer may modify any portion of this project. However, we anticipate that the developer may wish to modify certain portions of this tool, and for that reason we will detail some particular files found in the release that may be of particular interest to the programmer.

GPS.cs and GpsForm.cs

These objects handle all of the Bluetooth GPS interactions. Any modifications to GPS handling should be done in these files.

IstiHelp.cs

The help system is run from this class. By using this object rather than the built in help system, we may use standard HTML files with images as our help files.

Rotator.cs

This object rotates photos.

SiteFormII.cs

This file holds most of the tab items found in the GUI. Not all tabs that are created are always on display. Certain tabs are displayed or hidden as needed.

The method `initTabHolders()` creates 3 lists of `TabPage`s. It is from these `TabPage`s lists that the particular displays are created.

Most of the tabs are defined in the 'Design' pane for this file, but some of them are created manually. In `InitDetailedTabHolders()` we create individual `TabPage`s and add them to the `DetailedTabHolder` list. In the future, this is the preferred method of creating and defining new tabbed pages. An example of this method's power may be seen by reading it. In this method we currently create new `UserControl`s and add them to this list. These `UserControl` then simply appear in the tabbed window as needed. (See `Inspection.cs` as an example of a `UserControl` that can be inserted into the tabbed pan at run time).

We will also mention the camera at this point. The `Camera` function is called from the `SiteFormII`. We do not call the camera in the usual way, however. The camera's program is run from within a separate camera only program. There are some issues with camera memory being released after a photo is taken, so we have created a small stand alone camera program that is terminated after each photo is taken. If this stand alone program is missing, the standard method of using the embedded camera tool is used instead.

Site.cs

The main object of this tool is the `Site` Object. This object extends the `XmlDocument` base class and stores all the `Site` information. The public class held in this file is `ATC67Site`. The magic of this object is its ability to read `Site` information from the XML file, and to write it back to XML once the information in the object has been edited (by `SiteFormII.cs`).

SiteSelectForm.cs

This form presents the user with a list of all of the XML site files held on the phone. There are currently 4 subfolders under the main Sites folder. These folders are 'new', 'assigned', 'repository' and 'bad'. Site files that have been modified by the user on the phone, but not synchronized with the remote repository are stored in the 'new' folder. Site files that have been sent to the phone from the server, but have not yet been edited are stored in the 'assigned' folder. Site files that have been sent back to the server are stored in the 'repository' folder. If for any reason we can not parse a Site file, we store it in the 'bad' folder in order to deal with this file at some later time.

ScoreMatrix.cs and ScoreColumn.cs

These complex objects are where the logic and GUI elements for handling scoring are stored.

SyncForm.cs

This object handles all of the Server Synchronization. The local file system is examined to determine if any Site files must be sent to the server (and files in the 'new' folder). All user authentication is handled in this object and all interactions with the server's html are done in this object.

3.2.2 GPS_Util

This project encapsulates the GPS functions in a stand alone tool. This is useful for debugging Bluetooth connection issues, and is a convenient way to simply view your location information without launching ROVER.

3.2.3 RoverMaster

This is the main starting point for the ROVER Client. In this Project the user is presented with the startup screen and instructed to select the type of inspection desired. Depending on the answer (Pre-earthquake, Rapid Post-earthquake or Detailed Post-earthquake), the main program is then launched. This main program is defined in the Rover project (section 3.2.1). Other than this startup task, the RoverMaster project does not provide any additional functionality.

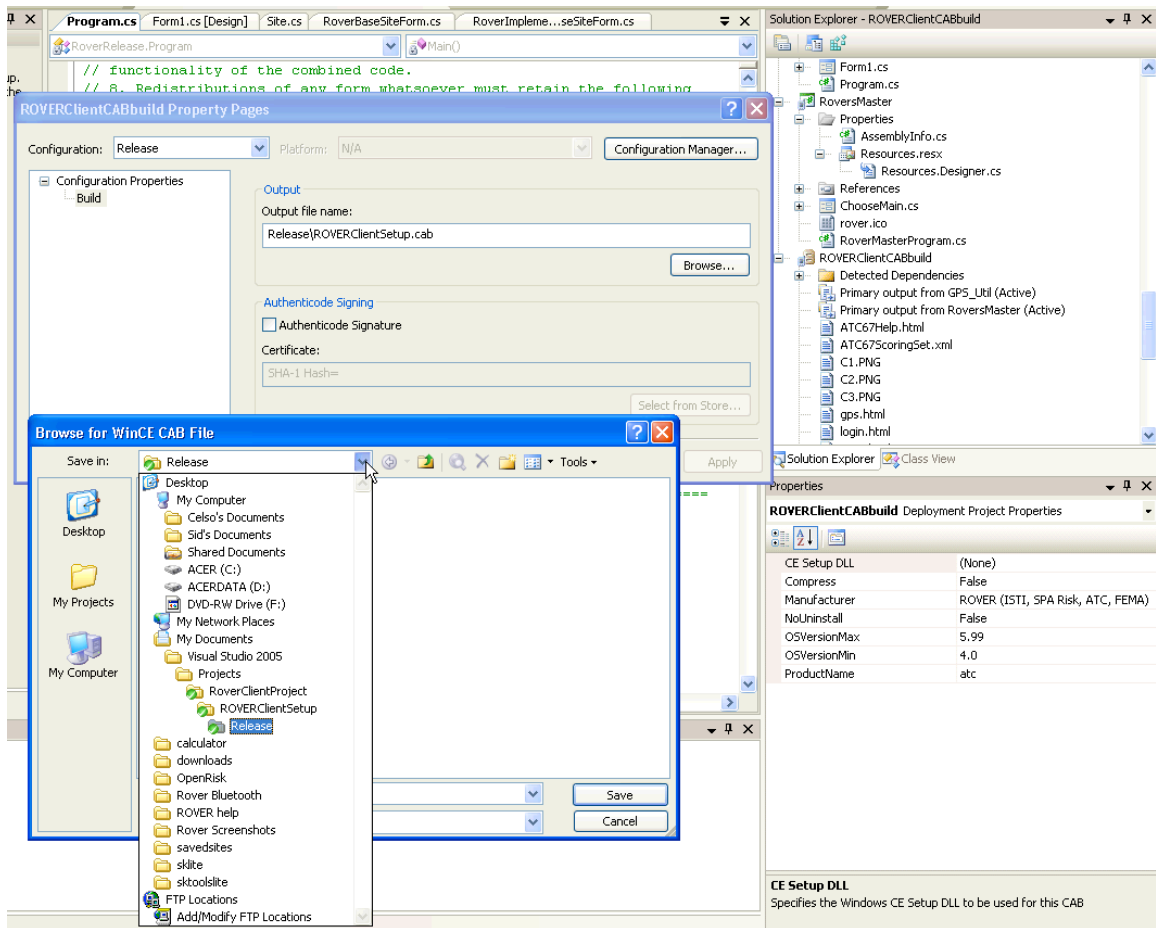
3.2.4 ROVERClientCABbuild

This project is used for the sole purpose of bundling the ROVER Client for release.

C:\Documents and Settings_____\My Documents\Visual Studio
2005\Projects\RoverClientProject\ROVERClientSetup\Release\ROVERClientSetup.cab

Right click on the project name and select "properties" to view/modify the name and location of the distribution cab file.

Installing the CAB file on a phone is covered by the ROVER End User Documentation and is beyond the scope of this document.



3.2.5 RoverCamera

This is the project that contains the Camera handling code. The Camera is now a stand alone module to avoid a memory leak in the built in camera libraries.

3.2.6 RoverRelease

This project is currently a placeholder and may be removed in future releases.

3.3 Making GUI Changes

In general there are 2 ways to modify the GUI in **any** program. You can edit the [Design] view of an object that extends Form. Or you may create/modify the GUI elements in the code directly. If you examine the SiteFormII.cs object, you will discover that there are 2 .cs files for this object. The SiteFormII.Designer.cs file is created automatically from the [Design] gui design object and in general should not be modified. The SiteFormII.cs file is where user modifications should be made as this file is not autogenerated.

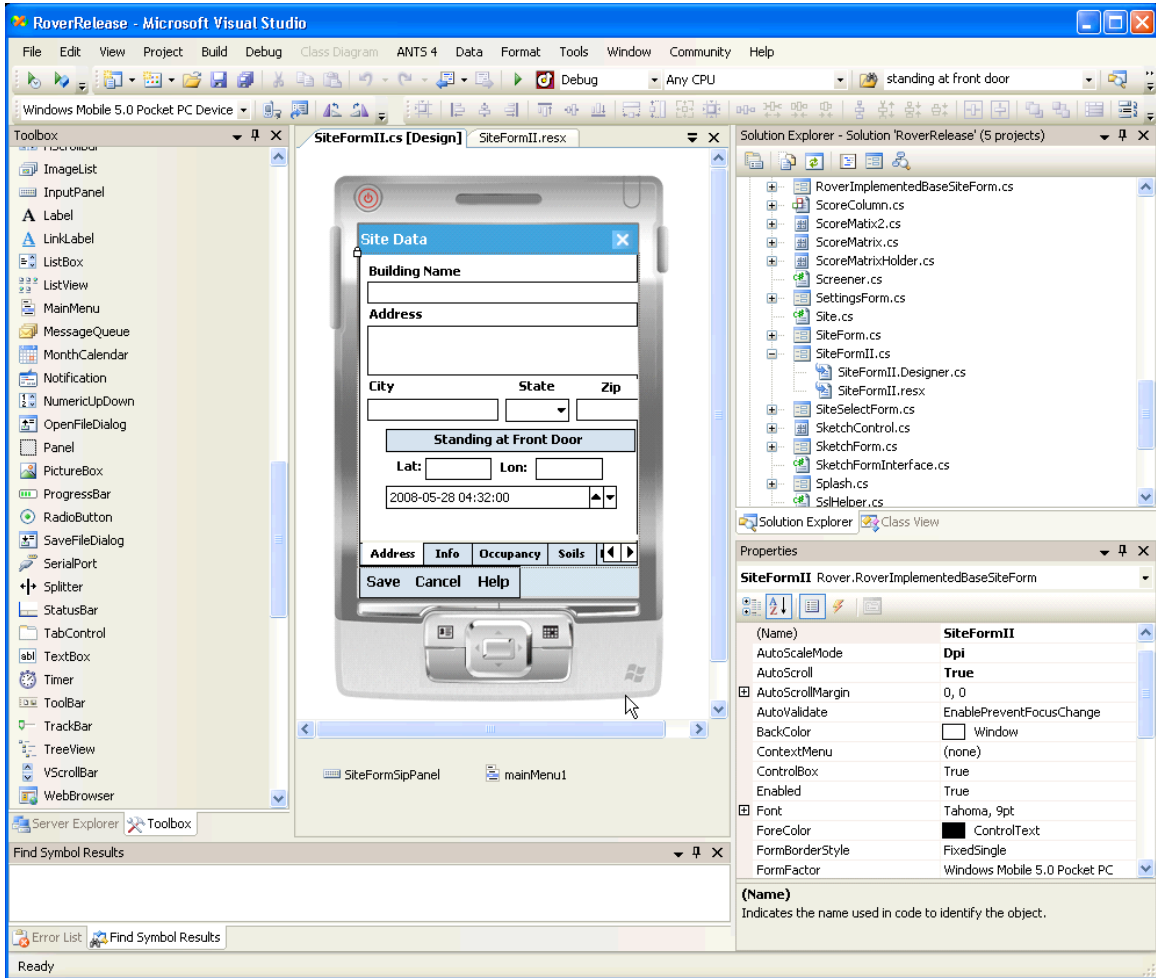


Figure 2: GUI editor for the [Design] of a class.

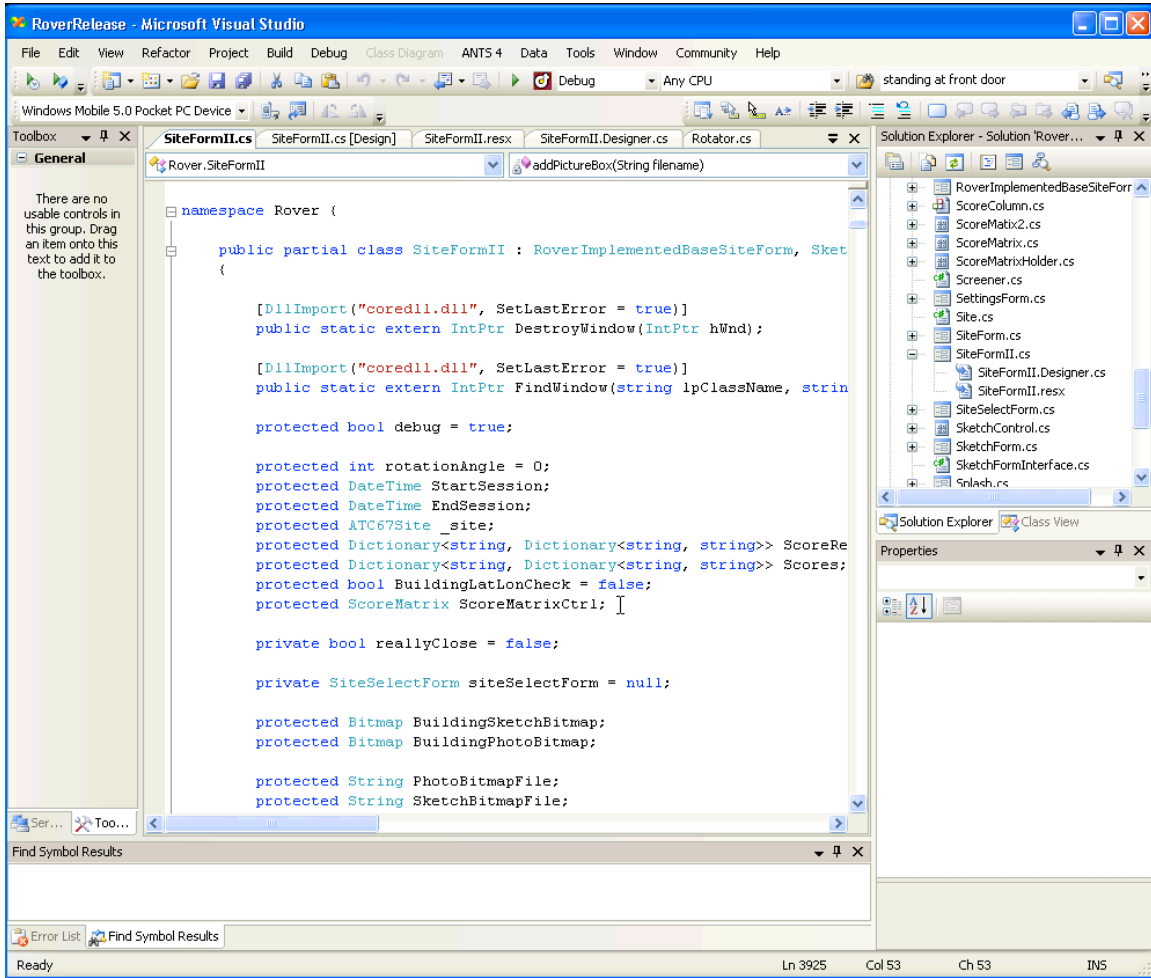


Figure 3: Manually editing the Form

```

/**
 * setup the ArrayList for the Detailed Post EQ survey tool.
 * To add new tabs to this view, Create new UserControls,
 * and add them to the list via the addTab() method.
 */
private void initDetailedTabsHolders(System.Collections.ArrayList tlist) {

    tlist.Add(this.atc20Address);

    UserControl uc = new Inspection(this);
    string ucName = "Inspection";
    tlist.Add(this.addTab(uc, ucName));

    finalPosting = new FinalPosting(Site);
    //finalPosting.Dock = DockStyle.Fill;
    this.finalPostingHeight = finalPosting.Height;
    ucName = "Posting";
    tlist.Add(this.addTab(finalPosting, ucName));

    uc = new BuildingInfo();
    ucName = "Building";
    tlist.Add(this.addTab(uc, ucName));

    uc = new Construction();
    ucName = "Construction";
    tlist.Add(this.addTab(uc, ucName));
}

```

Figure 4: code to add User Controls to the Detailed Post EQ Survey Tool

Inspection.cs [Design] Inspection.cs SiteFor

Standing at Building Now

Inspection Date and Time:

2008-05-28 04:32:00 ▲ ▼

Inspection Location:

Lat: Lon:

Figure 5: GUI design for the Inspection User Control

```
namespace Rover
{
    public partial class Inspection : UserControl
    {
        SiteFormII siteF = null;
        public NMEAPosition gpsCurrentPosition;
        public NMEAFixData gpsCurrentFixData;

        public Inspection(SiteFormII inSite)
        {
            siteF = inSite;
            InitializeComponent();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            DateTime syncTm = DateTime.Now;

            if (Program.RoverConfig.ActiveGPS == "true")
            {
                GetCurrentGPSData();
            }

            this.dateTimePicker1.Value = syncTm.ToLocalTime();
        }
    }
}
```

Figure 6: Adding functional code to the Inspection User Control.

3.4 XML Site File Handling

An example of the XML Site File format is shown in Appendix A.

The code to read and write this file, and in particular, make the connections between the SiteFormII's inputs and the storage in the Site File is handled by the Site.cs file. The important methods to review for reading and writing the items to/from the site file are UpdateSiteFromControls (to write the FORM info TO the XML file) and UpdateControlsFromSite (to populate the FORM info FROM the XML file).

This is important because if you create your own UserControl to get information from the user, you must modify these 2 methods to ensure that you store/parse the information correction in/from the Site file.

For example, the UserControl DateTimePicker has the following definition in UpdateControlsFromSite()

```

case "DateTimePicker":
    String syncTmStr = null;
    try
    {
        syncTmStr = Site.GetField(ctrl.Tag.ToString());
    }
    catch
    {
        syncTmStr = null;
    }
    finally
    {
        Cursor.Current = Cursors.Default;
    }
    if (syncTmStr.Length > 0 && !(syncTmStr.Equals("None")))
    {
        DateTime syncTm = DateTime.Parse(syncTmStr);
        (ctrl as DateTimePicker).Value = syncTm.ToLocalTime();
    }
    else
    {
        (ctrl as DateTimePicker).Value = DateTime.Now;
    }
    break;

```

Figure 7: Updating the Form from the Site XML file.

In many cases your UserControl will consist of multiple Controls, some of which you will create and some of which will be standard Controls. The Inspection.cs object is a good example of this. In the Inspection.cs object, we have a UserControl DateTimePicker as well as two standard Control TextBox items. When populating and saving the SiteFormII, no modifications need to be made to any code to handle the 2 TextBoxes. But we did need to create special code to handle the DateTimePicker in both UpdateSiteFromControls() and UpdateControlsFromSite ().

3.4.1 TAGS

The most important aspect of the Controls with respect to reading and writing the XML files are the TAG elements of each Control. The TAG element stores the XPATH representation of the XML node. The SiteFormII's reading and writing method uses this TAG to know where to read/write the critical data. In the 2 examples below, we are editing the Inspection.cs [Design] object. In the first picture we have selected the DateTimePicker. In the lower right box, you can see that we are editing the **dateTimePicker1**, and the **TAG** value is **atc67site/building/synctime**. In the second image we are editing the **atc20lat** TextBox, and the **TAG** value is **atc67site/building/latitude**.

If these TAGS are not set, then no information will pass back and forth from the XML file to the SiteForm.

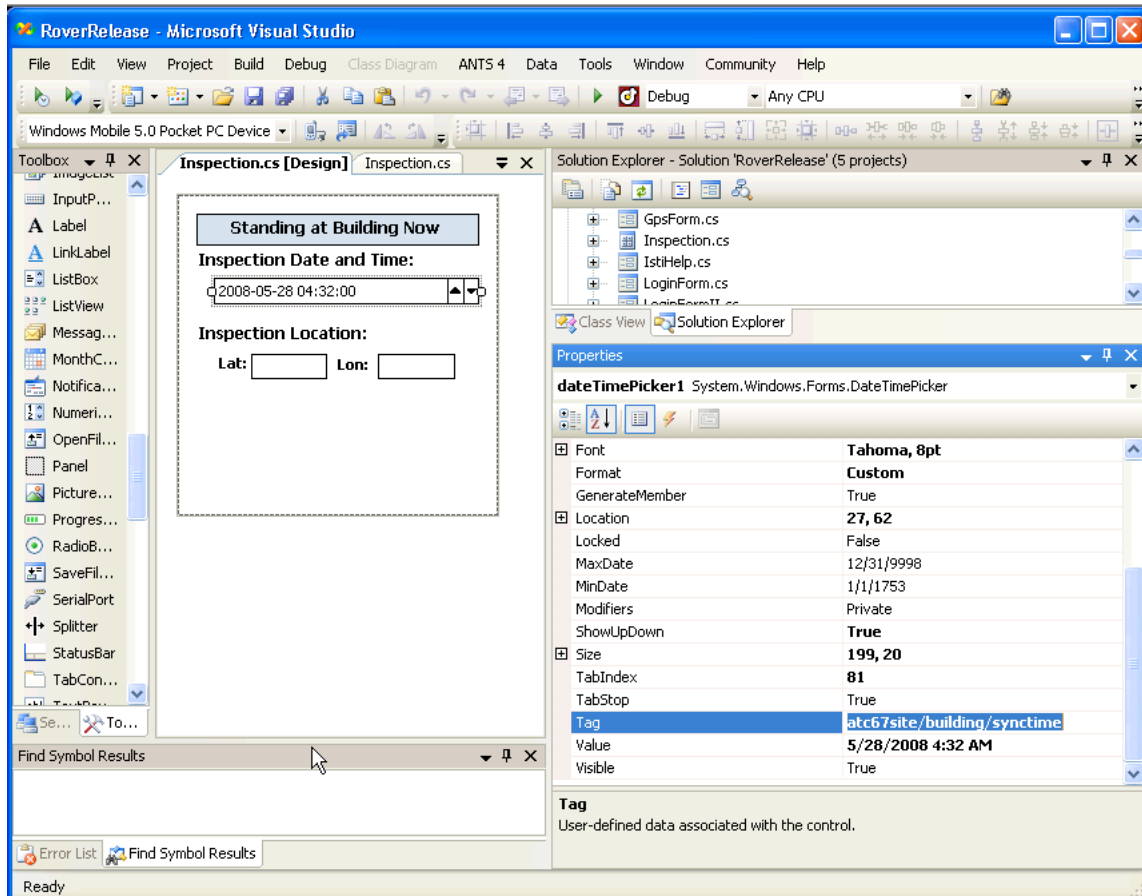


Figure 8: Setting the TAG for the DateTimePicker

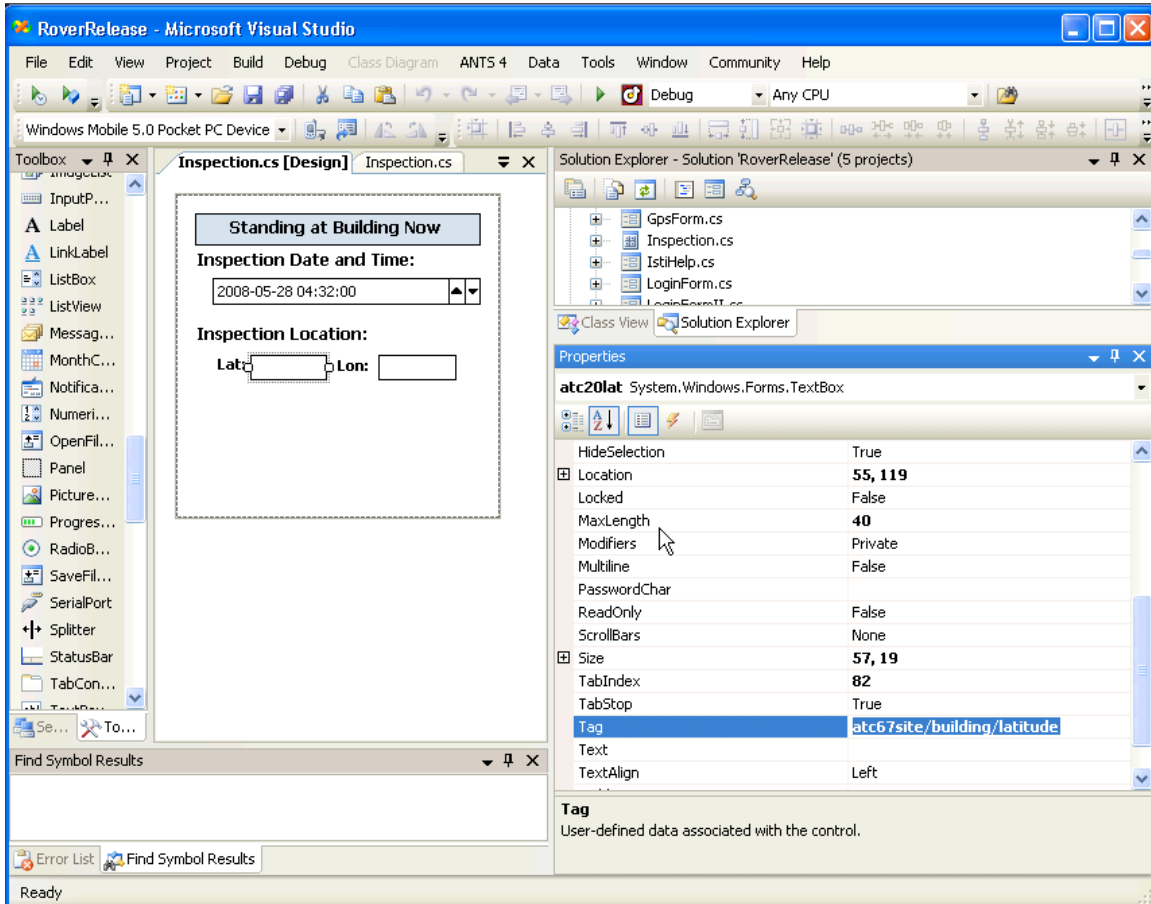


Figure 9: Setting the TAG for a TextBox

4 Rover Server

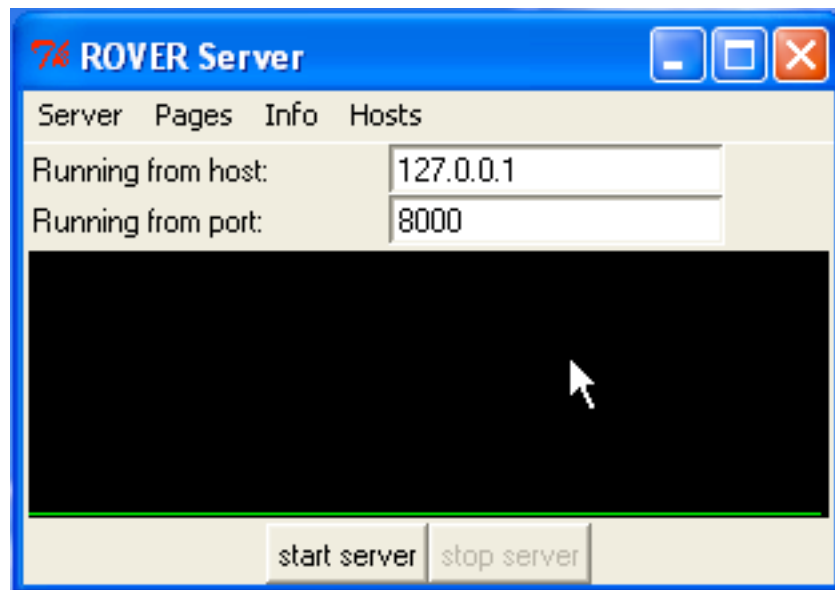
Program Organization

The server is implemented within the web2py framework; please refer to details at <http://www.web2py.com>. This is a framework for building database-backed web applications, written in Python. The source code is included within the application, and thus can be directly edited to make modifications; all files referred to in this document will be within the `Rover` folder of the `applications` folder of an installed server.

The GUI Widget

The GUI widget presented with when starting RoverServer in Windows is slightly different from the standard one included with web2py in the following ways:

- There is an extra menu, Hosts, with IP addresses that the program has determined are available for the server to use. One can fill the host and port fields by choosing from this list, or by the more traditional route of typing into the text boxes.
- The "Choose a password" field is missing, or rather moved: once you start the server, and you have selected "Enable Administrator Access" from the Server menu, a dialog asking for this password will appear.



RoverServer's GUI Widget

Administrator Access

If enabled, this provides an interface to the database and the source code behind the application, as well as "tickets" generated when errors occur. It is not necessary to use this interface to access these things; any text editor can be used to manipulate the source files, and a utility for accessing the database (in this case, an SQLite database) may be employed.

For security reasons, this interface is not only password-protected, but remote access is not allowed.

Database Organization

The specification of the database schema is in the file `models/db.py`. This is checked against every access to the application; thus, any modifications will be applied the next time a request is made of RoverServer, and no restarting is required.

Person Table

Each record specifies a user

Building Table

Each record specifies a building; this is information that is consistent across all forms associated with that building.

Worksheet Table

Each record corresponds to either a FEMA154 worksheet or those components in common with an ATC20 worksheet; in the latter case, the `atc20` field will identify the correct ATC20 worksheet.

ATC20 Table

Each record corresponds to an ATC20 worksheet.

RawImage Table

Each record corresponds to an image file

Image Table

Each record corresponds to an image, which may have 2 image files associated with it (the original, and the version with rotation, scaling and/or watermarking applied), and explicitly represents th???

Adding a field

Adding a field to a worksheet involves the following:

- Add it to the definition of the worksheet table in `models/db.py`.

- Modify the appropriate controller (in the `controllers` folder) and view (in the `views` folder) to have the field appear and work in the proper form
- Modify `controllers/api.py` and/or `controllers/client.py` to handle importing and exporting of the field

For example, the comments field of a pre-earthquake worksheet:

- `SQLField('comments', 'text')` is included in the definition of the worksheet table
 - Because no special processing of this field is needed, there is nothing specific about this field in `controllers/worksheet.py`
 - In `views/worksheet.html`, there is a specification of this field in the form:


```
<textarea type="text" class="text"
name="comments" id="worksheet_comments"
rows="2"
cols="40">{{=data.comments}}</textarea>
```
 - Note:
 - The name attribute is the name of the field
 - The id attribute is a combination of the table and field names
 - The class attribute specifies the datatype of the field; this is distinct from the type attribute, which informs the browser what kind of control to use. Thus, fields for both integers and doubles would have different class attributes, but both would have a type of text.
 - The `{{=data.comments}}` is a template that gets replaced with the value of `data.comments`, which is the comments field of a dictionary named `data` which is passed by the controller for this page
- The web2py framework actually provides functions to handle much of this for you, allowing for some customization, particularly for standard presentations.
- Code to encode this field into XML for export is in `controllers/api.py`
 - Code to decode this field from XML for import is in `controllers/client.py`

5 Source Code Contributions to Group

As we refine the process by which we manage code modifications, that process will be documented in this space.

6 Appendices

6.1 Sitefile XML example

```
<atc67site>
<version>0.0.1</version>
<worksheetversion>2</worksheetversion>
<screeener>
  <screeenerid>
  </screeenerid>
  <name>sid</name>
  <sessions>
    <session>3/23/10 8:07:39 PM,3/23/10 8:08:17 PM</session>
  </sessions>
  <affiliation>
  </affiliation>
</screeener>
<seismicity>MODERATE</seismicity>
<gps>
</gps>
<building>
  <buildingID>1</buildingID>
  <name>temple</name>
  <address>509 Broadway</address>
  <city>Saratoga Springs</city>
  <state>NY</state>
  <zipcode>12866</zipcode>
  <use>
  </use>
  <numstories>2</numstories>
  <yearconstruction>1880</yearconstruction>
  <area>6000</area>
  <otheridentifiers>
  </otheridentifiers>
  <synctime>2010-03-23 20:07:54Z</synctime>
  <latitude>NA</latitude>
  <longitude>NA</longitude>
  <comments>
  </comments>
  <soiltype_a>>false</soiltype_a>
  <soiltype_b>>false</soiltype_b>
  <soiltype_c>>false</soiltype_c>
  <soiltype_d>>true</soiltype_d>
  <soiltype_e>>false</soiltype_e>
  <soiltype_f>>false</soiltype_f>
  <occupancyclasses>
    <assembly>>false</assembly>
    <commercial>>false</commercial>
    <emergservices>>false</emergservices>
    <government>>false</government>
    <historic>>false</historic>
    <industrial>>false</industrial>
    <office>>false</office>
    <residential>>false</residential>
    <school>>false</school>
    <otherres>>false</otherres>
    <other>
    </other>
  </occupancyclasses>
  <occupancyload>
  </occupancyload>
  <fallinghazards>
    <unreinforcedchimneys>>false</unreinforcedchimneys>
    <parapets>>false</parapets>
    <cladding>>false</cladding>
    <other>>false</other>
    <othertext>
    </othertext>
  </fallinghazards>
  <scores>
    <score bldgtype="W1">
      <basicscore>
      </basicscore>
      <midrise>
      </midrise>
      <highrise>
      </highrise>
      <verticalirregularity>
      </verticalirregularity>
      <planirregularity>
      </planirregularity>
      <precode>
      </precode>
      <postbenchmark>
      </postbenchmark>
      <soiltypec>
      </soiltypec>
    </score>
  </scores>
</building>
</atc67site>
```

```
<soiltyped>
</soiltyped>
<soiltypee>
</soiltypee>
</score>
<score bldgtype="W2">
  <basicscore>
  </basicscore>
  <midrise>
  </midrise>
  <highrise>
  </highrise>
  <verticalirregularity>
  </verticalirregularity>
  <planirregularity>
  </planirregularity>
  <precode>
  </precode>
  <postbenchmark>
  </postbenchmark>
  <soiltypec>
  </soiltypec>
  <soiltyped>
  </soiltyped>
  <soiltypee>
  </soiltypee>
</score>
<score bldgtype="S1(MRF)">
  <basicscore>3.6</basicscore>
  <midrise>
  </midrise>
  <highrise>
  </highrise>
  <verticalirregularity>
  </verticalirregularity>
  <planirregularity>
  </planirregularity>
  <precode>
  </precode>
  <postbenchmark>
  </postbenchmark>
  <soiltypec>
  </soiltypec>
  <soiltyped>-1</soiltyped>
  <soiltypee>
  </soiltypee>
</score>
<score bldgtype="S2(BR)">
  <basicscore>3.6</basicscore>
  <midrise>
  </midrise>
  <highrise>
  </highrise>
  <verticalirregularity>
  </verticalirregularity>
  <planirregularity>
  </planirregularity>
  <precode>-0.4</precode>
  <postbenchmark>
  </postbenchmark>
  <soiltypec>
  </soiltypec>
  <soiltyped>-1.2</soiltyped>
  <soiltypee>
  </soiltypee>
</score>
<score bldgtype="S3(LM)">
  <basicscore>
  </basicscore>
  <midrise>
  </midrise>
  <highrise>
  </highrise>
  <verticalirregularity>
  </verticalirregularity>
  <planirregularity>
  </planirregularity>
  <precode>
  </precode>
  <postbenchmark>
  </postbenchmark>
  <soiltypec>
  </soiltypec>
  <soiltyped>
  </soiltyped>
  <soiltypee>
  </soiltypee>
</score>
<score bldgtype="S4(RC SW)">
  <basicscore>
  </basicscore>
  <midrise>
  </midrise>
  <highrise>
  </highrise>
  <verticalirregularity>
  </verticalirregularity>
```

```
<planirregularity>
</planirregularity>
<precode>
</precode>
<postbenchmark>
</postbenchmark>
<soiltypec>
</soiltypec>
<soiltyped>
</soiltyped>
<soiltypee>
</soiltypee>
</score>
<score bldgtype="S5(URM INF)">
  <basicscore>
  </basicscore>
  <midrise>
  </midrise>
  <highrise>
  </highrise>
  <verticalirregularity>
  </verticalirregularity>
  <planirregularity>
  </planirregularity>
  <precode>
  </precode>
  <postbenchmark>
  </postbenchmark>
  <soiltypec>
  </soiltypec>
  <soiltyped>
  </soiltyped>
  <soiltypee>
  </soiltypee>
</score>
<score bldgtype="C1(MRF)">
  <basicscore>
  </basicscore>
  <midrise>
  </midrise>
  <highrise>
  </highrise>
  <verticalirregularity>
  </verticalirregularity>
  <planirregularity>
  </planirregularity>
  <precode>
  </precode>
  <postbenchmark>
  </postbenchmark>
  <soiltypec>
  </soiltypec>
  <soiltyped>
  </soiltyped>
  <soiltypee>
  </soiltypee>
</score>
<score bldgtype="C2(SW)">
  <basicscore>
  </basicscore>
  <midrise>
  </midrise>
  <highrise>
  </highrise>
  <verticalirregularity>
  </verticalirregularity>
  <planirregularity>
  </planirregularity>
  <precode>
  </precode>
  <postbenchmark>
  </postbenchmark>
  <soiltypec>
  </soiltypec>
  <soiltyped>
  </soiltyped>
  <soiltypee>
  </soiltypee>
</score>
<score bldgtype="C3(URM INF)">
  <basicscore>
  </basicscore>
  <midrise>
  </midrise>
  <highrise>
  </highrise>
  <verticalirregularity>
  </verticalirregularity>
  <planirregularity>
  </planirregularity>
  <precode>
  </precode>
  <postbenchmark>
  </postbenchmark>
  <soiltypec>
  </soiltypec>
  <soiltyped>
  </soiltyped>
```

```
</soiltyped>
<soiltypee>
</soiltypee>
</score>
<score bldgtype="PC1(TU)">
  <basicscore>
  </basicscore>
  <midrise>
  </midrise>
  <highrise>
  </highrise>
  <verticalirregularity>
  </verticalirregularity>
  <planirregularity>
  </planirregularity>
  <precode>
  </precode>
  <postbenchmark>
  </postbenchmark>
  <soiltypec>
  </soiltypec>
  <soiltyped>
  </soiltyped>
  <soiltypee>
  </soiltypee>
</score>
<score bldgtype="PC2">
  <basicscore>
  </basicscore>
  <midrise>
  </midrise>
  <highrise>
  </highrise>
  <verticalirregularity>
  </verticalirregularity>
  <planirregularity>
  </planirregularity>
  <precode>
  </precode>
  <postbenchmark>
  </postbenchmark>
  <soiltypec>
  </soiltypec>
  <soiltyped>
  </soiltyped>
  <soiltypee>
  </soiltypee>
</score>
<score bldgtype="RM1(FD)">
  <basicscore>
  </basicscore>
  <midrise>
  </midrise>
  <highrise>
  </highrise>
  <verticalirregularity>
  </verticalirregularity>
  <planirregularity>
  </planirregularity>
  <precode>
  </precode>
  <postbenchmark>
  </postbenchmark>
  <soiltypec>
  </soiltypec>
  <soiltyped>
  </soiltyped>
  <soiltypee>
  </soiltypee>
</score>
<score bldgtype="RM2(RD)">
  <basicscore>
  </basicscore>
  <midrise>
  </midrise>
  <highrise>
  </highrise>
  <verticalirregularity>
  </verticalirregularity>
  <planirregularity>
  </planirregularity>
  <precode>
  </precode>
  <postbenchmark>
  </postbenchmark>
  <soiltypec>
  </soiltypec>
  <soiltyped>
  </soiltyped>
  <soiltypee>
  </soiltypee>
</score>
<score bldgtype="URM">
  <basicscore>
  </basicscore>
  <midrise>
  </midrise>
```

```

<highrise>
</highrise>
<verticalirregularity>
</verticalirregularity>
<planirregularity>
</planirregularity>
<precode>
</precode>
<postbenchmark>
</postbenchmark>
<soiltypec>
</soiltypec>
<soiltyped>
</soiltyped>
<soiltypee>
</soiltypee>
</score>
</scores>
<imagery>
<sketch>
  <timestamp>
</timestamp>
  <latitude>
</latitude>
  <longitude>
</longitude>
  <ID>
</ID>
  <position>
</position>
  <imagerotation>
</imagerotation>
  <imagecomment>
</imagecomment>
  <scale>
</scale>
  <image>
</image>
</sketch>
<photo>
  <timestamp>
</timestamp>
  <latitude>
</latitude>
  <longitude>
</longitude>
  <ID>
</ID>
  <position>
</position>
  <imagerotation>0</imagerotation>
  <imagecomment>Comment?</imagecomment>
  <scale>
</scale>
  <image>
</image>
IMAGE REMOVED!!!!
</image>
  <name>\My Documents\My Pictures\047.jpg</name>
</photo>
</imagery>
<detailedevaluation>>false</detailedevaluation>
<numundergroundstories>0</numundergroundstories>
</building>
<atc20i>
<edited>>false</edited>
<actions>
  <comments>
</comments>
  <otherreccomendation>
</otherreccomendation>
  <barricades>
</barricades>
  <eval>
    <structural>>false</structural>
    <geotech>>false</geotech>
    <other>
</other>
  </eval>
</actions>
<posting>
  <green>>false</green>
  <yellow>>false</yellow>
  <red>>false</red>
  <restrictions>
</restrictions>
</posting>
<evaluation>
  <comments>
</comments>
  <othertext>
</othertext>
  <damage>
    <none>>true</none>
    <p0>>false</p0>
    <p1>>false</p1>
    <p10>>false</p10>
    <p30>>false</p30>

```

```
<p60>false</p60>
<p100>false</p100>
</damage>
<collapse>
</collapse>
<leaning>
</leaning>
<racking>
</racking>
<falling>
</falling>
<ground>
</ground>
<other>
</other>
</evaluation>
<description>
<constructiontype>
  <wood>false</wood>
  <steel>false</steel>
  <tilt>false</tilt>
  <concrete>false</concrete>
  <cshear>false</cshear>
  <unreinforcemason>false</unreinforcemason>
  <reinforcemason>false</reinforcemason>
  <other>
  </other>
</constructiontype>
<numbadunits>
</numbadunits>
<numunits>
</numunits>
<phone>
</phone>
<contact>
</contact>
</description>
<inspection>
  <interior>
  </interior>
  <exterior>true</exterior>
  <datetime>
  </datetime>
</inspection>
<shakemap_id>
</shakemap_id>
<shakemap_vsn>
</shakemap_vsn>
<detailed>
  <evaluation>
    <generalcomments>
    </generalcomments>
    <overall>
      <collapse>
        <level>
        </level>
        <comment>
        </comment>
      </collapse>
      <leaning>
        <level>
        </level>
        <comment>
        </comment>
      </leaning>
      <other>
        <level>
        </level>
        <comment>
        </comment>
        <label>
        </label>
      </other>
    </overall>
  </structural>
  <foundations>
    <level>
    </level>
    <comment>
    </comment>
  </foundations>
  <roofs>
    <level>
    </level>
    <comment>
    </comment>
  </roofs>
  <columns>
    <level>
    </level>
    <comment>
    </comment>
  </columns>
  <diaphragms>
    <level>
    </level>
    <comment>
    </comment>
  </diaphragms>
```

```
</comment>
</diaphragms>
<walls>
  <level>
  </level>
  <comment>
  </comment>
</walls>
<precast>
  <level>
  </level>
  <comment>
  </comment>
</precast>
<other>
  <level>
  </level>
  <comment>
  </comment>
  <label>
  </label>
</other>
</structural>
<nonstructural>
  <parapets>
    <level>
    </level>
    <comment>
    </comment>
  </parapets>
  <cladding>
    <level>
    </level>
    <comment>
    </comment>
  </cladding>
  <ceilings>
    <level>
    </level>
    <comment>
    </comment>
  </ceilings>
  <interior>
    <level>
    </level>
    <comment>
    </comment>
  </interior>
  <elevators>
    <level>
    </level>
    <comment>
    </comment>
  </elevators>
  <stairs>
    <level>
    </level>
    <comment>
    </comment>
  </stairs>
  <electric_gas>
    <level>
    </level>
    <comment>
    </comment>
  </electric_gas>
  <other>
    <level>
    </level>
    <comment>
    </comment>
    <label>
    </label>
  </other>
</nonstructural>
<geotech>
  <slopefailure>
    <level>
    </level>
    <comment>
    </comment>
  </slopefailure>
  <ground_movement>
    <level>
    </level>
    <comment>
    </comment>
  </ground_movement>
  <other>
    <level>
    </level>
    <comment>
    </comment>
    <label>
    </label>
  </other>
</geotech>
```

```
<sketch>
  <timestamp>
</timestamp>
  <ID>
</ID>
  <imagerotation>
</imagerotation>
  <imagecomment>
</imagecomment>
  <scale>
</scale>
  <image>
</image>
</sketch>
</evaluation>
<posting>
  <green>>false</green>
  <yellow>>false</yellow>
  <red>>false</red>
</posting>
</detailed>
</atc20i>
</atc67site>
```