# MS Govern Property Control Service Setup & Configuration

**Release 4.7.3 Version 1**

**Last Revision Update: 10/2/20**

**MS Govern**

**Property Control Service**

**Setup and Configuration**

**Version: 1.0**

**September 2012 - Release 4.7**

# Disclaimer

MS Govern Software applications are developed using Microsoft's .NET Technology. Applications are intended for use within the .NET Framework.

MS Governs' API Documentation is provided with the understanding that implementation of described methods and contracts will be with .NET Development tools and select programming languages. Users should note that with the exception of MS Governs' Methods or Contracts that are implemented with Visual Basic .NET and C#, MS Govern Software does not provide development guidance or support.

MS Govern has taken due care in preparing this manual. However, nothing contained herein modifies or alters in any way the standard terms and conditions of the purchase, lease, or license agreement by which the product was acquired, nor increases in any way the liability of MS Govern to the customer.

This page is intentionally blank

# Preface

Welcome to *Govern* for Windows, a comprehensive and fully integrated transaction-driven system written exclusively for local governments. *Govern* includes a wide variety of database modules:

## Computer-Assisted Mass Appraisal (CAMA)

- Appeals & Grievances
- Appraisals / Property Valuations
- Comparables Sales Management
- Financial Management
- Account Receivable
- Cash Collection

## Land Management

- Business & Individual Licenses
- Complaint Tracking
- Leasing
- Permit Tracking & Inspection Scheduling
- Planning
- Violations

## Revenue Management

- Aircraft & Boat Excise Tax
- Miscellaneous Billing
- Personal Property Tax Billing
- Real Property Tax Billing
- Self-Reported Tax Billing
- Special Assessments
- Tax Title / Tax Lien / Tax Sales
- Utility Billing

Since 1980, *MS Govern* has worked hand-in-hand with State and Local Governments to simplify the implementation of software solutions that automate the flow of information related to their properties.

This page is intentionally blank

# Table of Contents

# Setup of the Property Control System Web Service

**WARNING:** Always make a backup of your **web.config** file and the content of your **Resource Files** folder, as these files may be inadvertently overwritten during this procedure.

The following setup procedure is for installing the Property Control System Web Service.

## Users with the Web Service already installed

If you already have the Web service installed, and are performing an update, the only action required is to copy the new .DLL's into your already existing **bin** folder. *See the **Step 3 – Update your .DLL files** section below.*

## Content of Install Package

The *Property Control Web Service* installation files are distributed in a .zip archive package called WSPropertyControl4.X.X.X.zip. (The X's refer to version numbers). Contact MS Govern Technical Support for the location of the package.

The content of the package is as follows. There are three (3) folders titled as follows:

**1 - WebService IN IIS** (Web Service files that are to be copied into the **wwwroot** IIS folder)
**2 - Resource Files** (Resource files)
**3 - Test Application** (Test application and support files testing the Property Control)

## Installation

### Step 1 – Create the Web Service Folder

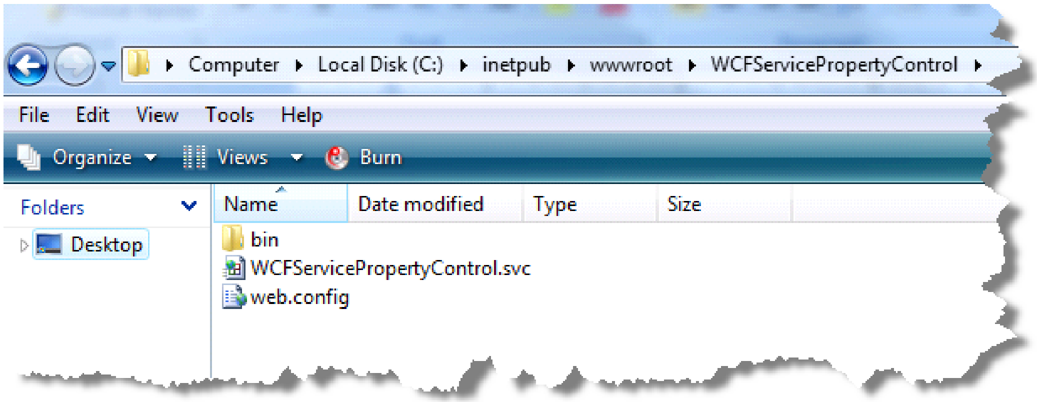
1. In the IIS folder, create a new *Web service* folder called **WCFServicePropertyControl**.
2. In a temporary folder outside of the IIS folder, unzip the install package contents.

**WARNING:** Make a backup of your **web.config** file, and the content of your **Resource Files** folder, as these files may be inadvertently overwritten during this procedure.

3. In the folder with the restored files, copy the contents of the ***"1 - WebService IN IIS"*** folder to the Web Service IIS folder; maintain the directory structure.

Your IIS folder with directory structure should now look like the following.

**bin** *(folder containing .DLL's)*
**WCFServicePC.svc** (service file)
**web.config** (Web config file)


Computer ▸ Local Disk (C:) ▸ inetpub ▸ wwwroot ▸ WCFServicePropertyControl ▸
File
Edit
View
Tools
Help
Organize
Views
Burn
Folders
Desktop
Name
Date modified
Type
Size
bin
WCFServicePropertyControl.svc
web.config

The **web.config** file that has been copied from the package contains the following lines:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.diagnostics>
    <sources>
      <!-- This section defines the logging configuration for My.Application.Log -->
      <source name="DefaultSource" switchName="DefaultSwitch">
        <listeners>
          <add name="FileLog"/>
          <!-- Uncomment the below section to write to the Application Event Log -->
          <!--<add name="EventLog"/>-->
        </listeners>
      </source>
    </sources>
    <switches>
      <add name="DefaultSwitch" value="Information" />
    </switches>
    <sharedListeners>
      <add name="FileLog"
           type="Microsoft.VisualBasic.Logging.FileLogTraceListener,
Microsoft.VisualBasic, Version=8.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a,
processorArchitecture=MSIL"
           initializeData="FileLogWriter"/>
      <!-- Uncomment the below section and replace APPLICATION_NAME with the name of your
application to write to the Application Event Log -->
      <!--<add      name="EventLog"      type="System.Diagnostics.EventLogTraceListener"
initializeData="APPLICATION_NAME"/> -->
    </sharedListeners>
  </system.diagnostics>

  <appSettings>
    <add key="MSGovern.GovernNetConfig" value="C:\MSGovern\GovernNetConfig.xml"/>
  </appSettings>

  <!-- Local host service url which will be available once start -->
  <system.serviceModel>
    <bindings>
      <basicHttpBinding>
        <binding name ="soapBinding">
          <security mode="None">
          </security>
        </binding>
      </basicHttpBinding>

      <webHttpBinding>
        <binding name="webBinding">
        </binding>
      </webHttpBinding>
    </bindings>

    <behaviors>

      <endpointBehaviors>
        <behavior name="jsonBehavior">
          <enableWebScript  />
        </behavior>
      </endpointBehaviors>

      <serviceBehaviors>
        <behavior name="MyServiceBehavior">
          <serviceMetadata httpGetEnabled="true" />
          <serviceDebug includeExceptionDetailInFaults="true" />
        </behavior>
      </serviceBehaviors>
    </behaviors>

    <services>
      <service                                  behaviorConfiguration="MyServiceBehavior"
name="MSGovern.WCFServicePC.SV_PropertyControl">
        <endpoint address="soap"
                  contract="MSGovern.WCFServicePC.IPropertyControlService"
                  binding="basicHttpBinding"
                  bindingConfiguration="soapBinding"
                  name="HttpEndPoint"  />

        <endpoint address="json"
                  contract="MSGovern.WCFServicePC.IPropertyControlService"
                  binding="webHttpBinding"
                  bindingConfiguration="webBinding"
                  behaviorConfiguration="jsonBehavior"
                  name="JsonEndPoint"/>

      </service>
    </services>
  </system.serviceModel>
</configuration>
```

### Step 2 – Modify the *web.config* file

With a text editor like ***Notepad.exe*** open the **web.config** file and go to the section with the **<appSettings>** XML tag:

```
<appSettings>
       <add key="MSGovern.GovernNetConfig" value="C:\MSGovern\GovernNetConfig.xml"/>
```

```
</appSettings>
```

Make sure the path to the *GovernNetConfig.xml* file is changed to reflect the current location.

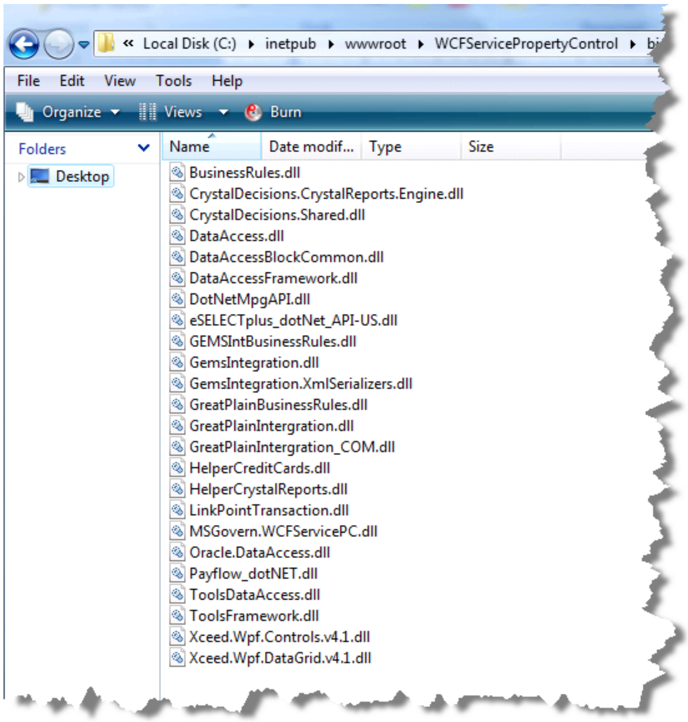
For example:

```
<add key="MSGovern.GovernNetConfig" value="\\Server\inetpub\wwwroot\GovernNetConfig.xml"/>
```

**NOTE:** If you already had a prior installation, and your **web.config** file that was backed up as recommended, copy it back, replacing the one in the package.

## Step 3 – Update your .DLL files

Locate your bin directory in your existing **IIS** folder that contains the service and delete all .DLL's.
Copy the new **.DLL** files from the "**1 - WebService IN IIS/bin**" sub folder of the install package and paste them into the *IIS* service **bin** folder.

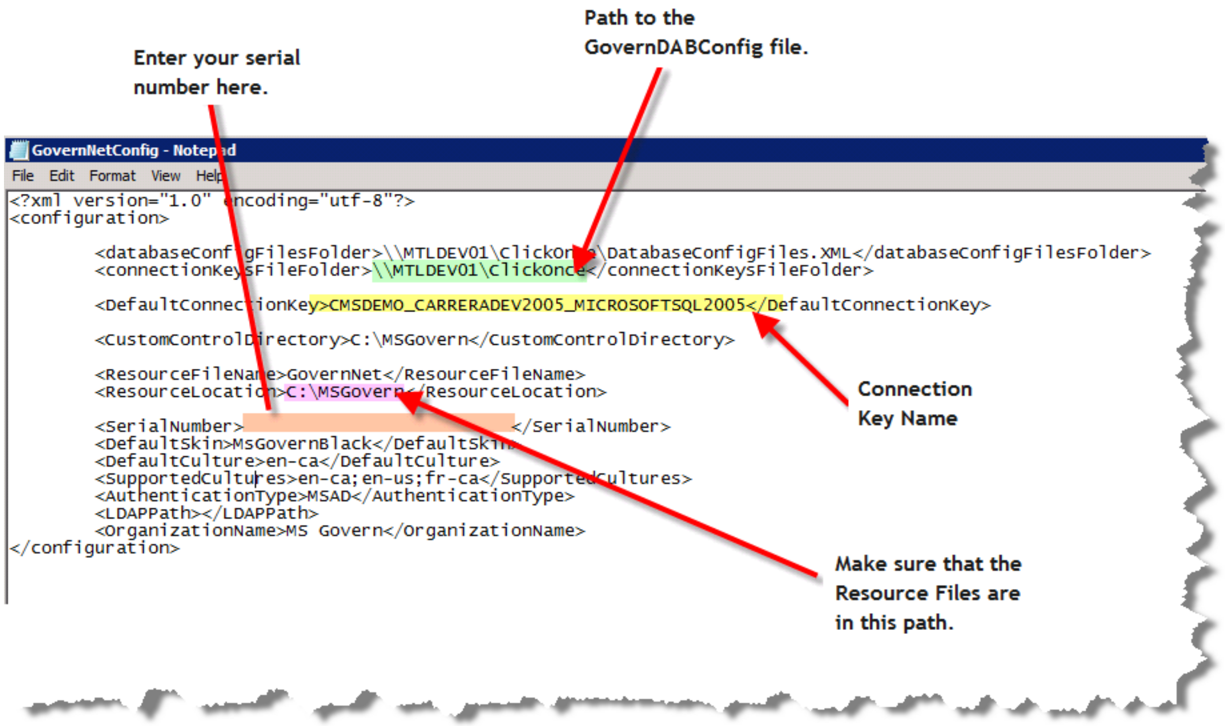


BusinessRules.dll
CrystalDecisions.CrystalReports.Engine.dll
CrystalDecisions.Shared.dll
DataAccess.dll
DataAccessBlockCommon.dll
DataAccessFramework.dll
DotNetMpgAPI.dll
eSELECTplus_dotNet_API-US.dll
GEMSIntBusinessRules.dll
GemsIntegration.dll
GemsIntegration.XmlSerializers.dll
GreatPlainBusinessRules.dll
GreatPlainIntergration.dll
GreatPlainIntergration_COM.dll
HelperCreditCards.dll
HelperCrystalReports.dll
LinkPointTransaction.dll
MSGovern.WCFServicePC.dll
Oracle.DataAccess.dll
Payflow_dotNET.dll
ToolsDataAccess.dll
ToolsFramework.dll
Xceed.Wpf.Controls.v4.1.dll
Xceed.Wpf.DataGrid.v4.1.dll

## Step 4 – Make Modifications to the GovernNetConfig.xml file

Open your **GovernNetConfig.xml** file with a text editor, and perform the following verification and modifications; this file may be located in the *Deployments* folder.

- Verify that the path to the **<databaseConfigFilesFolder>** is correct.
- Ensure that the path in the **<connectionKeysFileFolder>** is correct.
- Confirm that the name of your *Connection Key* is correct in the **<DefaultConnectionKey>** section.
- Verify that the path in the **<ResourceLocation>** section is correct; this is the location of your resource files.



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

        <databaseConfigFilesFolder>\\MTLDEV01\ClickOnce\DatabaseConfigFiles.XML</databaseConfigFilesFolder>
        <connectionKeysFileFolder>\\MTLDEV01\ClickOnce</connectionKeysFileFolder>

        <DefaultConnectionKey>CMSDEMO_CARRERADEV2005_MICROSOFTSQL2005</DefaultConnectionKey>

        <CustomControlDirectory>C:\MSGovern</CustomControlDirectory>

        <ResourceFileName>GovernNet</ResourceFileName>
        <ResourceLocation>C:\MSGovern</ResourceLocation>

        <SerialNumber>                         </SerialNumber>
        <DefaultSkin>MsGovernBlack</DefaultSkin>
        <DefaultCulture>en-ca</DefaultCulture>
        <SupportedCultures>en-ca;en-us;fr-ca</SupportedCultures>
        <AuthenticationType>MSAD</AuthenticationType>
        <LDAPPath></LDAPPath>
        <OrganizationName>MS Govern</OrganizationName>
</configuration>
```

Enter your serial number here.

Path to the GovernDABConfig file.

Connection Key Name

Make sure that the Resource Files are in this path.

## Step 5 – Copy Resource Files

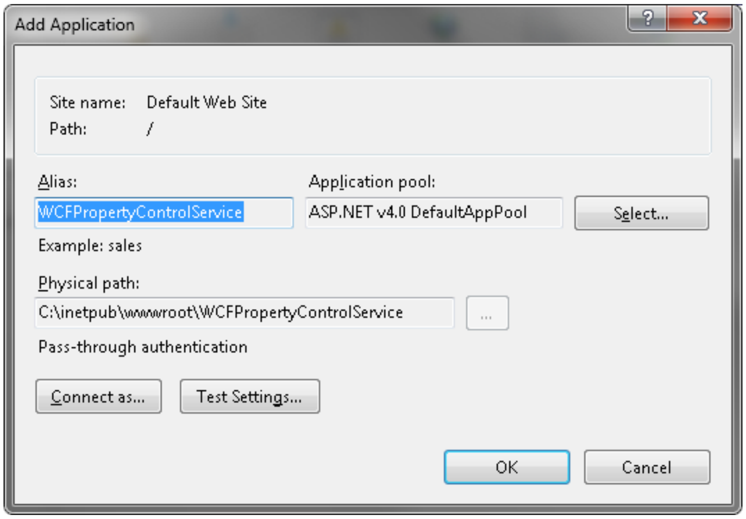
**WARNING:** The following step is for new installations only.

1. In the restored package files, copy the resource files that are in the”**2 – Resource Files**”.
2. Go to the location of your resource files; they are indicated in the path seen in **<ResourceLocation>** tag in the *GovernNetConfig.xml* file and paste them into the folder.

**NOTE:** Users with pre-existing installations, if you have overwritten you *resource files*, restore them with your backup files.

## Step 6 – Convert the Web Service folder into an Application

1. In the Control Panel, open the **Internet Information Services (IIS) Manager**.
2. Drill down to the location of the service folder.
3. Right click on the folder and select **Convert to Application**.



Site name: Default Web Site
Path: /

Alias: WCFPropertyControlService
Application pool: ASP.NET v4.0 DefaultAppPool
Example: sales

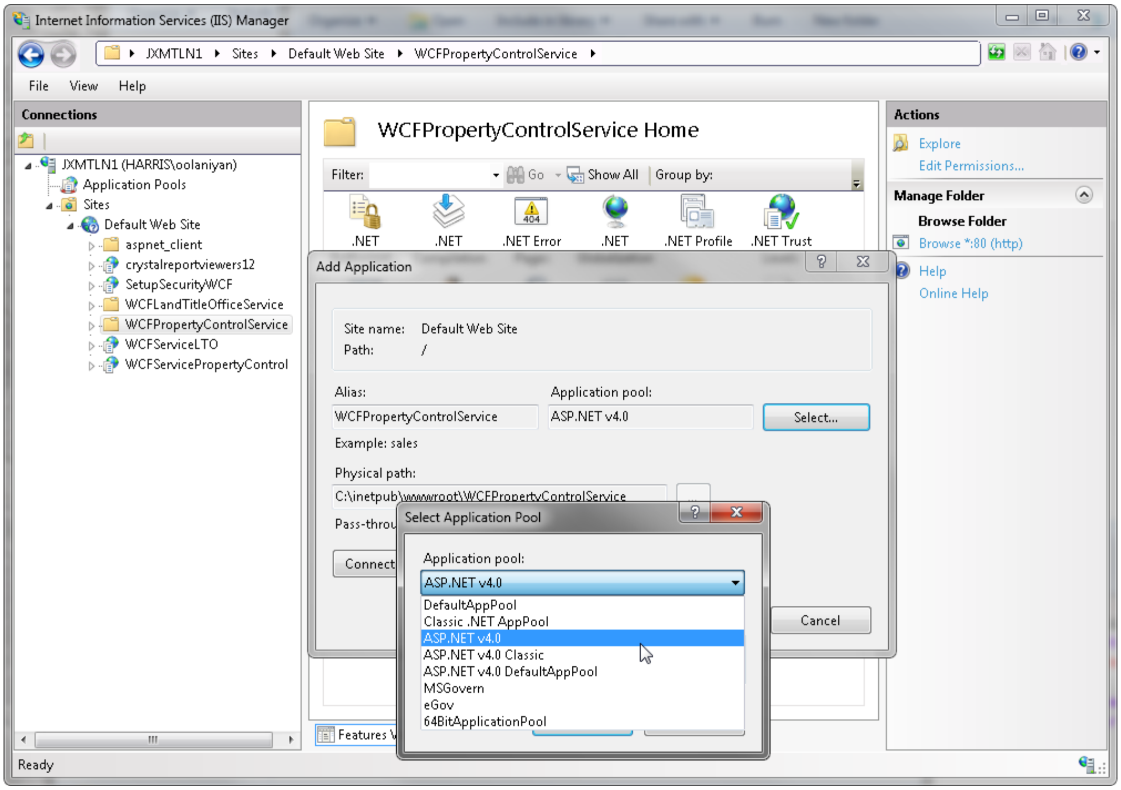
Physical path: C:\inetpub\wwwroot\WCFPropertyControlService

Pass-through authentication

4. In the *Add Application* window, verify that the Application pool is set to **ASP.NET v4.0**. If the application pool is not set to ASP.NET v4.0, do the following:

## Changing the Application Pool to ASP.NET v4.0

5. Click **Select...**.
6. In the **Select Application Pool** window, select ASP.NET v4.0 from the drop down list and Click **OK**.



Site name: Default Web Site
Path: /

Alias: WCFPropertyControlService
Application pool: ASP.NET v4.0
Example: sales

Physical path: C:\inetpub\wwwroot\WCFPropertyControlService

Pass-throu

Application pool: ASP.NET v4.0

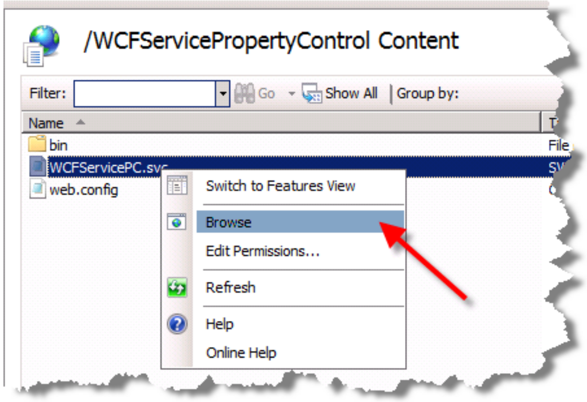
DefaultAppPool
Classic .NET AppPool
ASP.NET v4.0
ASP.NET v4.0 Classic
ASP.NET v4.0 DefaultAppPool
MSGovern
eGov
64BitApplicationPool

7. Click **OK** again to return to the IIS Manager.

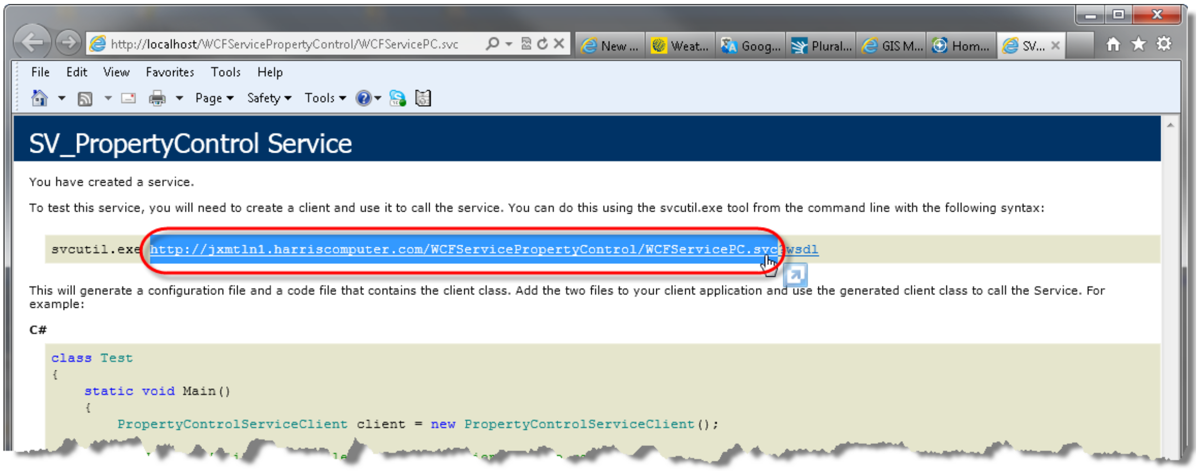
This completes the installation procedure for the Web service.

# Testing the service

1. In the *IIS Manager*, if you are not in the Content View, right click on the *WCFServicePropertyControl* icon and select **Content View** from the floating menu.
2. In the center pane, right click on *WCFServicePC.svc* and select **Browse** from the floating menu; this action will launch your Web browser.



3. When the service is successfully running, you will see the following in your browser window.



SV_PropertyControl Service

You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command line with the following syntax:

```
svcutil.exe http://jxmtln1.harriscomputer.com/WCFServicePropertyControl/WCFServicePC.svc?wsdl
```

This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the generated client class to call the Service. For example:

C#

```
class Test
{
    static void Main()
    {
        PropertyControlServiceClient client = new PropertyControlServiceClient();
```

4. In the browser window, copy the path without the "**?wsdl**", "**/soap**", or "**/json**" depending on the protocol that you are using.

**NOTE:** Take a note of the service URL, it will be needed when using the *Test Client* application

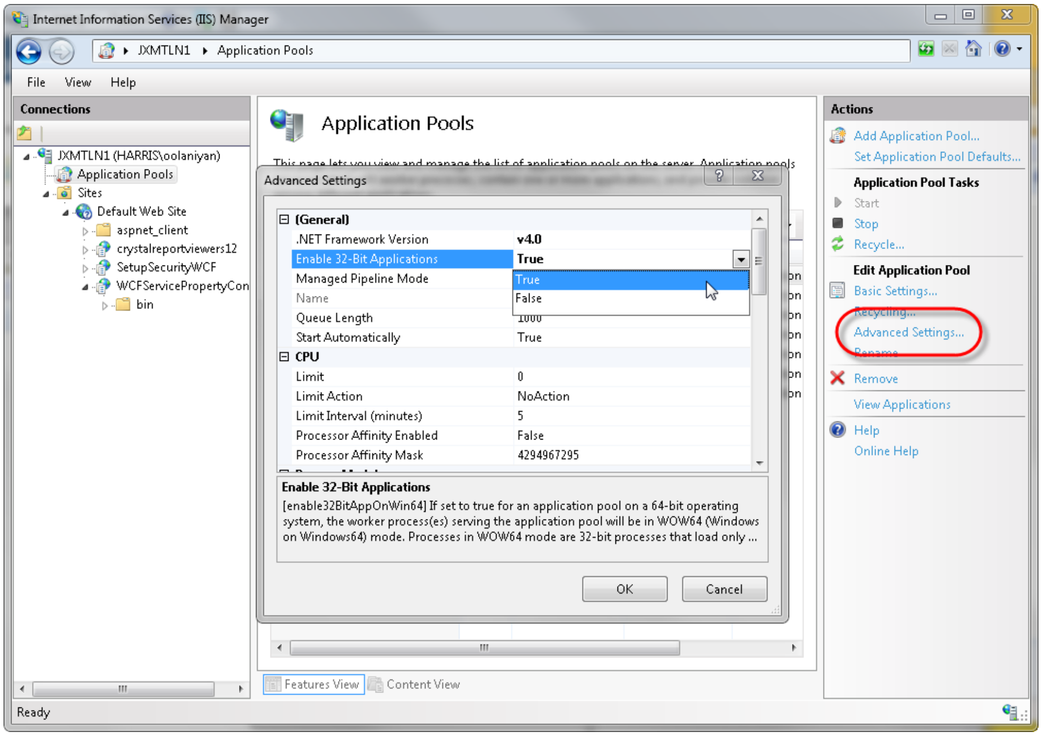
## A Note to Users of 64-bit versions of Windows

When using a 64-bit version of Windows, ensure that the *Application Pool* is enabled to allow for 32-bit applications.

To enable for 32-bit applications…

1. In the IIS Manager, select **Application Pools**.
2. In the middle pane, click to select the application pool that your service is running in
3. Under the *Actions* pane on the right hand side, click to select **Advanced Settings…**

4. In the Advanced Settings form, locate the **Enable 32-bit Applications** parameters and set it to **True**.



| | |
|---|---|
| **(General)** | |
| .NET Framework Version | **v4.0** |
| Enable 32-Bit Applications | **True**<br>True<br>False |
| Managed Pipeline Mode | |
| Name | |
| Queue Length | 1000 |
| Start Automatically | True |
| **CPU** | |
| Limit | 0 |
| Limit Action | NoAction |
| Limit Interval (minutes) | 5 |
| Processor Affinity Enabled | False |
| Processor Affinity Mask | 4294967295 |

**Enable 32-Bit Applications**
[enable32BitAppOnWin64] If set to true for an application pool on a 64-bit operating system, the worker process(es) serving the application pool will be in WOW64 (Windows on Windows64) mode. Processes in WOW64 mode are 32-bit processes that load only ...

5. Click **OK** to save the configuration.
6. Exit from the *IIS Manager*.

# Using the WCF Test Client Application

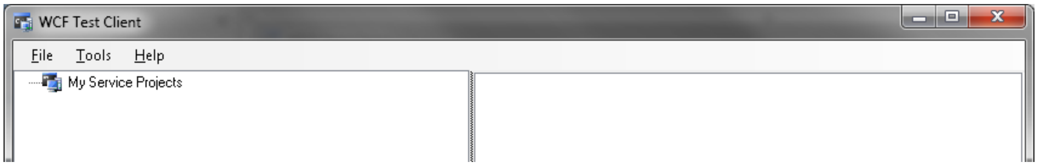
**NOTE:** Windows Communication Foundation (**WCF**) is an API in the .NET Framework. Although *MS Governs  WCF PC Web Services* supports *JavaScript Object Notation* (**JSON**) serialization to WCF, *MS Govern* does not provide developmental technical support for this type of implementation.

When any of the Web Services are running, you can use the *Test Client* application that is included in the download package to test the services. The test client is found in the “**3 – Test Application**” folder of the install package.

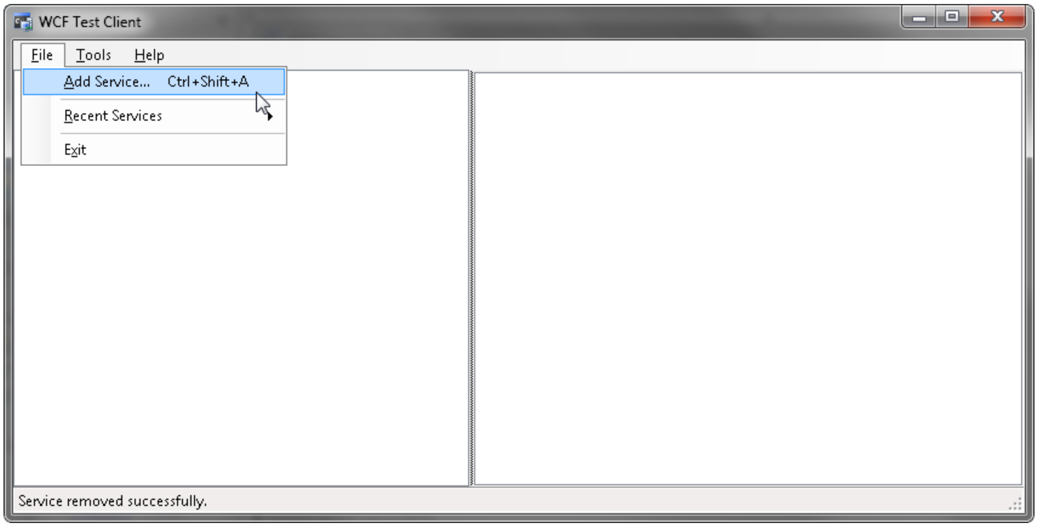
**NOTE**: Prior to starting the test application, locate the service URL that was recorded when testing the service. *Refer to the Testing the Service section of this document.*

To run the test application…

1. Locate the install package and look for the **3 – Test Application** folder.
2. Inside the folder, locate the **WcfTestClient.exe** application.
3. Double click to start the application.

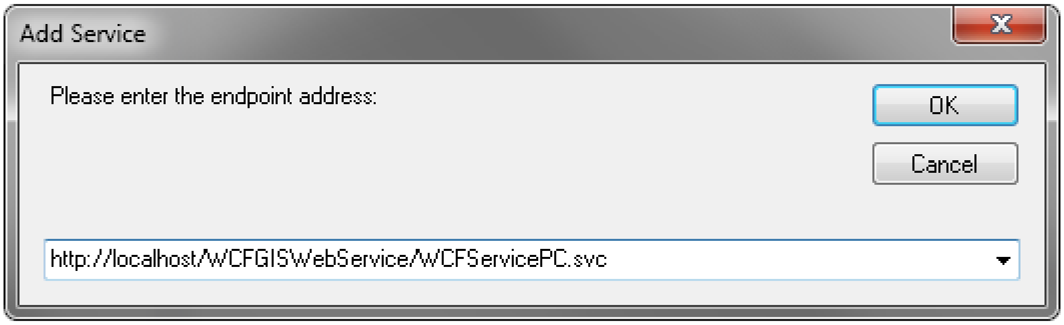


## Viewing the Contents of a Service



Service removed successfully.

4. In the open WCF Test Client, select *File* > **Add Service (Ctrl + Shift + A)** to add a service. In the **Add Service** screen, paste the URL of the service.



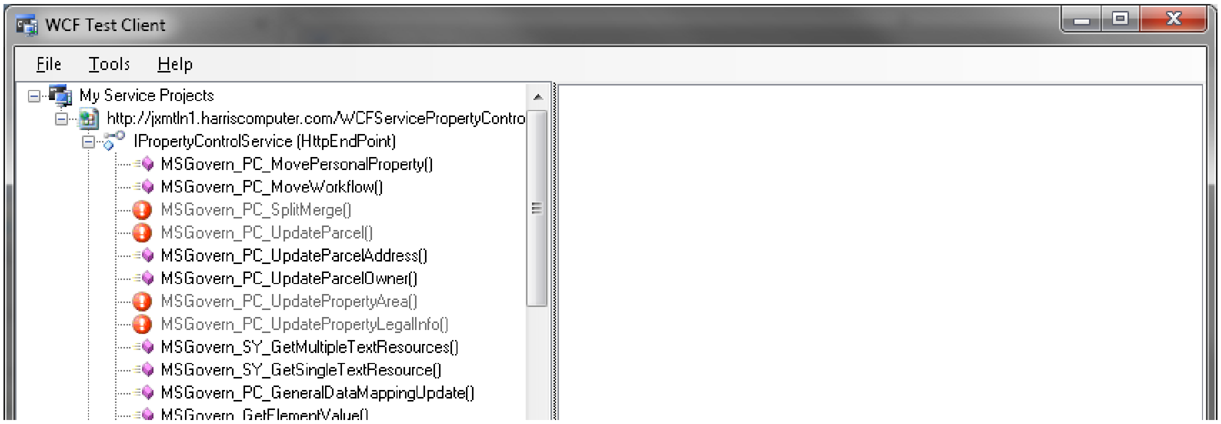
Please enter the endpoint address:

http://localhost/WCFGISWebService/WCFServicePC.svc

5. Click **OK**.

After the service has been successfully added, the WCF Test Client Screen is displayed with all the methods that are available in the service.

**NOTE:** Multiple services can be tested by selecting *File* > **Add Service (Ctrl + Shift + A)**.

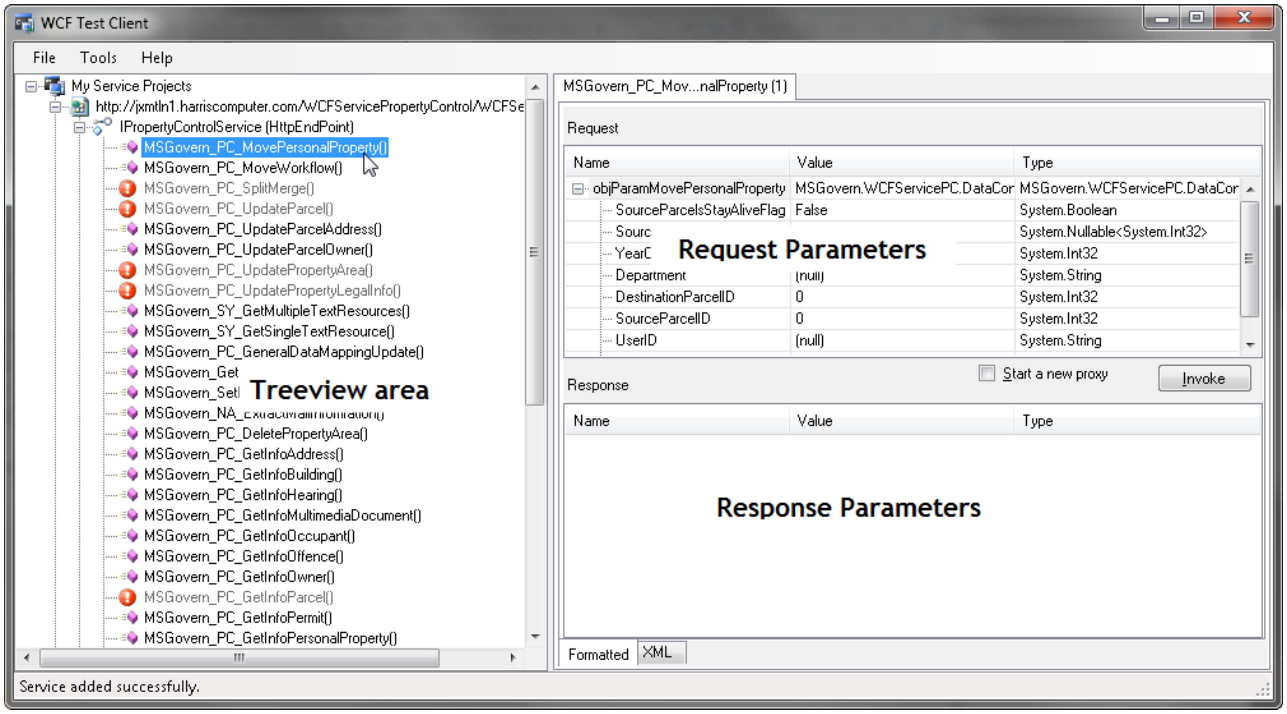


```
My Service Projects
  http://jxmtln1.harriscomputer.com/WCFServicePropertyContro
    IPropertyControlService (HttpEndPoint)
      MSGovern_PC_MovePersonalProperty()
      MSGovern_PC_MoveWorkflow()
      MSGovern_PC_SplitMerge()
      MSGovern_PC_UpdateParcel()
      MSGovern_PC_UpdateParcelAddress()
      MSGovern_PC_UpdateParcelOwner()
      MSGovern_PC_UpdatePropertyArea()
      MSGovern_PC_UpdatePropertyLegalInfo()
      MSGovern_SY_GetMultipleTextResources()
      MSGovern_SY_GetSingleTextResource()
      MSGovern_PC_GeneralDataMappingUpdate()
      MSGovern_GetElementValue()
```

## Viewing a Method

To view a method, double click on it in the treeview area. On the Right hand side, the parameters will be displayed in the Request Parameters area.

**NOTE**: Methods that are not supported by the WCF client are listed with a red exclamation mark beside them.



My Service Projects
- http://jxmtln1.harriscomputer.com/WCFServicePropertyControl/WCFSe
  - IPropertyControlService (HttpEndPoint)
    - MSGovern_PC_MovePersonalProperty()
    - MSGovern_PC_MoveWorkflow()
    - MSGovern_PC_SplitMerge()
    - MSGovern_PC_UpdateParcel()
    - MSGovern_PC_UpdateParcelAddress()
    - MSGovern_PC_UpdateParcelOwner()
    - MSGovern_PC_UpdatePropertyArea()
    - MSGovern_PC_UpdatePropertyLegalInfo()
    - MSGovern_SY_GetMultipleTextResources()
    - MSGovern_SY_GetSingleTextResource()
    - MSGovern_PC_GeneralDataMappingUpdate()
    - MSGovern_Get
    - MSGovern_Set
    - MSGovern_NA_ExtractMailInformation()
    - MSGovern_PC_DeletePropertyArea()
    - MSGovern_PC_GetInfoAddress()
    - MSGovern_PC_GetInfoBuilding()
    - MSGovern_PC_GetInfoHearing()
    - MSGovern_PC_GetInfoMultimediaDocument()
    - MSGovern_PC_GetInfoOccupant()
    - MSGovern_PC_GetInfoOffence()
    - MSGovern_PC_GetInfoOwner()
    - MSGovern_PC_GetInfoParcel()
    - MSGovern_PC_GetInfoPermit()
    - MSGovern_PC_GetInfoPersonalProperty()

MSGovern_PC_Mov...nalProperty (1)

Request

| Name | Value | Type |
|---|---|---|
| objParamMovePersonalProperty | MSGovern.WCFServicePC.DataCor | MSGovern.WCFServicePC.DataCor |
| SourceParcelsStayAliveFlag | False | System.Boolean |
| Sourc | | System.Nullable<System.Int32> |
| YearC | | System.Int32 |
| Department | (null) | System.String |
| DestinationParcelID | 0 | System.Int32 |
| SourceParcelID | 0 | System.Int32 |
| UserID | (null) | System.String |

Response

| Name | Value | Type |
|---|---|---|

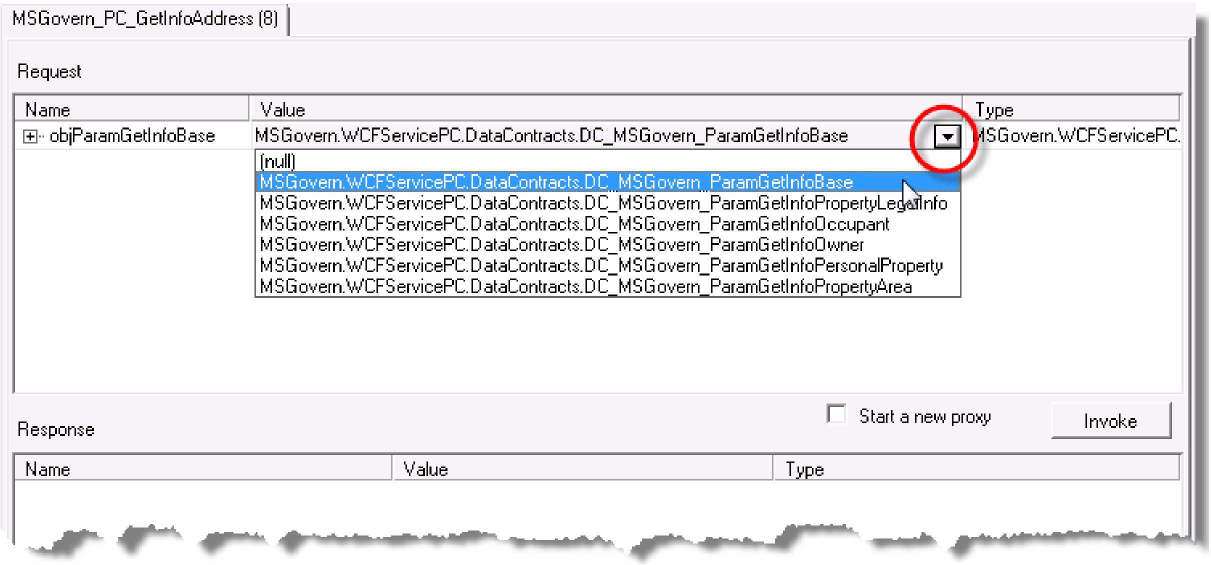
Service added successfully.

## Testing a Method

For this example we want to test the **MSGovern_PC_GetInfoAddress** method. This method returns information on the address of a given property. When used, a selection of address objects will be returned.

To test the method:

1. Double click on the **MSGovern_PC_GetInfoAddress** method to select it in the treeview area.
2. The objects for the method are displayed on the right hand side.
3. Under the **Value** column click in the column and select one of the parameters from the drop down menu; this is the information that will be sent to the client.



MSGovern_PC_GetInfoAddress (8)

Request

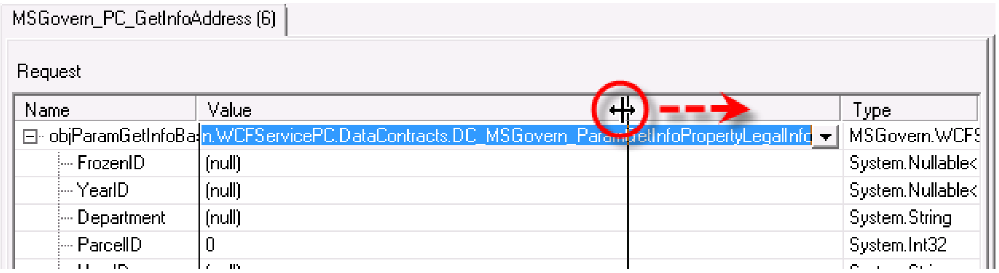
| Name | Value | Type |
|---|---|---|
| objParamGetInfoBase | MSGovern.WCFServicePC.DataContracts.DC_MSGovern_ParamGetInfoBase | MSGovern.WCFServicePC. |

(null)
MSGovern.WCFServicePC.DataContracts.DC_MSGovern_ParamGetInfoBase
MSGovern.WCFServicePC.DataContracts.DC_MSGovern_ParamGetInfoPropertyLegalInfo
MSGovern.WCFServicePC.DataContracts.DC_MSGovern_ParamGetInfoOccupant
MSGovern.WCFServicePC.DataContracts.DC_MSGovern_ParamGetInfoOwner
MSGovern.WCFServicePC.DataContracts.DC_MSGovern_ParamGetInfoPersonalProperty
MSGovern.WCFServicePC.DataContracts.DC_MSGovern_ParamGetInfoPropertyArea

Response

| Name | Value | Type |
|---|---|---|

**TIP**: The available items in the list may be difficult to see, but the column can be expanded manually or with a double click on the space between the column headings.

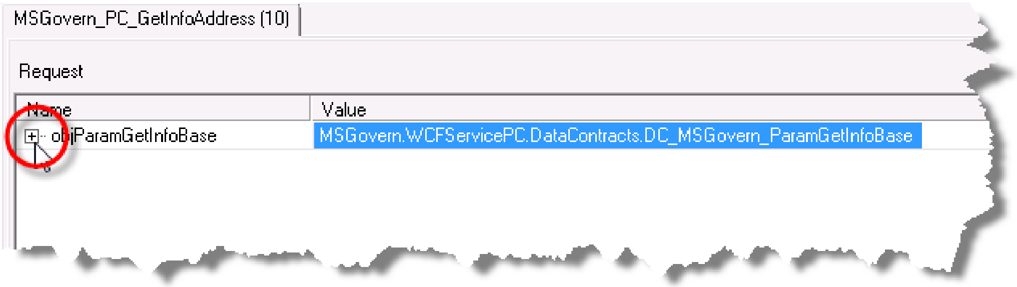


MSGovern_PC_GetInfoAddress (6)

Request

| Name | Value | Type |
|---|---|---|
| ⊟ objParamGetInfoBa | n.WCFServicePC.DataContracts.DC_MSGovern_Par[illegible]etInfoPropertyLegalInfo | MSGovern.WCFS |
| FrozenID | (null) | System.Nullable< |
| YearID | (null) | System.Nullable< |
| Department | (null) | System.String |
| ParcelID | 0 | System.Int32 |

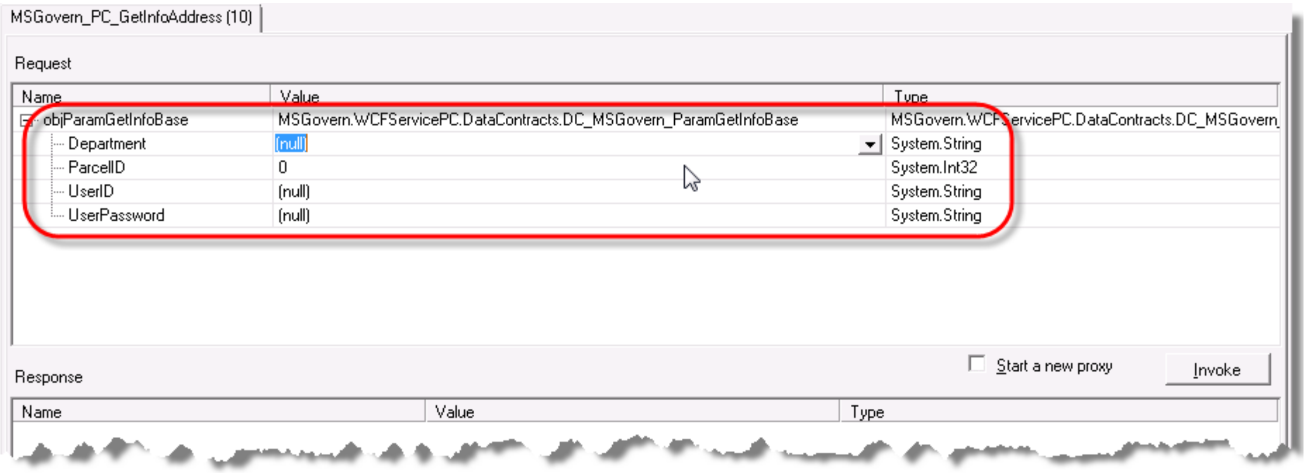
4. When one of the parameters is selected, you will see a "+" appear beside the name indicating that parameters will need to be provided.
5. Click on the plus to expand the list of fields that need to be populated with information, including a username and password.



MSGovern_PC_GetInfoAddress (10)

Request

| Name | Value |
|---|---|
| ⊞ objParamGetInfoBase | MSGovern.WCFServicePC.DataContracts.DC_MSGovern_ParamGetInfoBase |



MSGovern_PC_GetInfoAddress (10)

Request

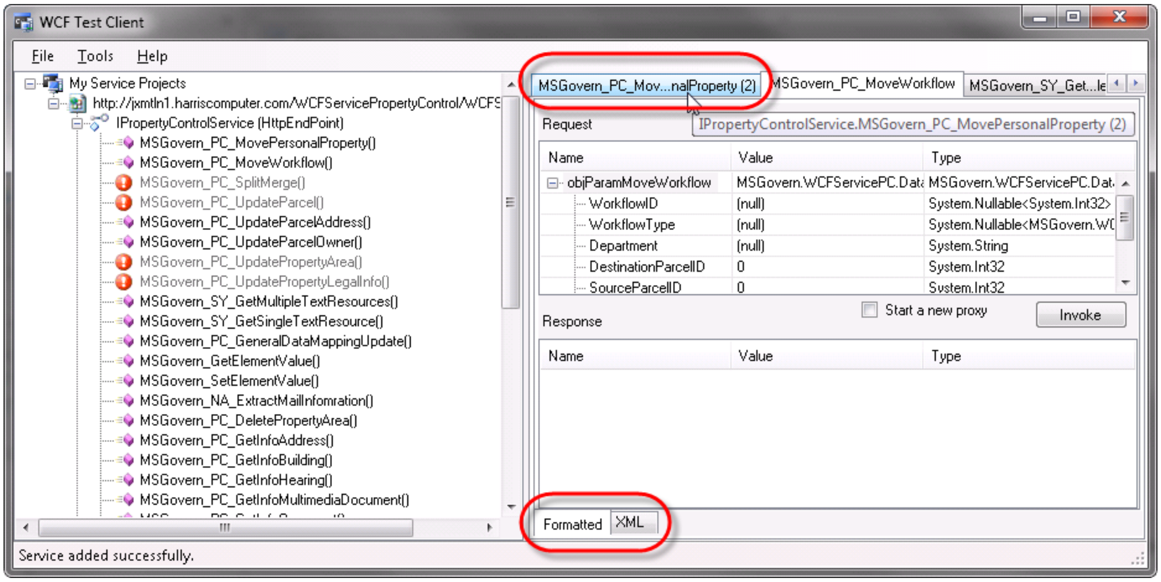
| Name | Value | Type |
|---|---|---|
| ⊟ objParamGetInfoBase | MSGovern.WCFServicePC.DataContracts.DC_MSGovern_ParamGetInfoBase | MSGovern.WCFServicePC.DataContracts.DC_MSGovern |
| Department | (null) | System.String |
| ParcelID | 0 | System.Int32 |
| UserID | (null) | System.String |
| UserPassword | (null) | System.String |

Response

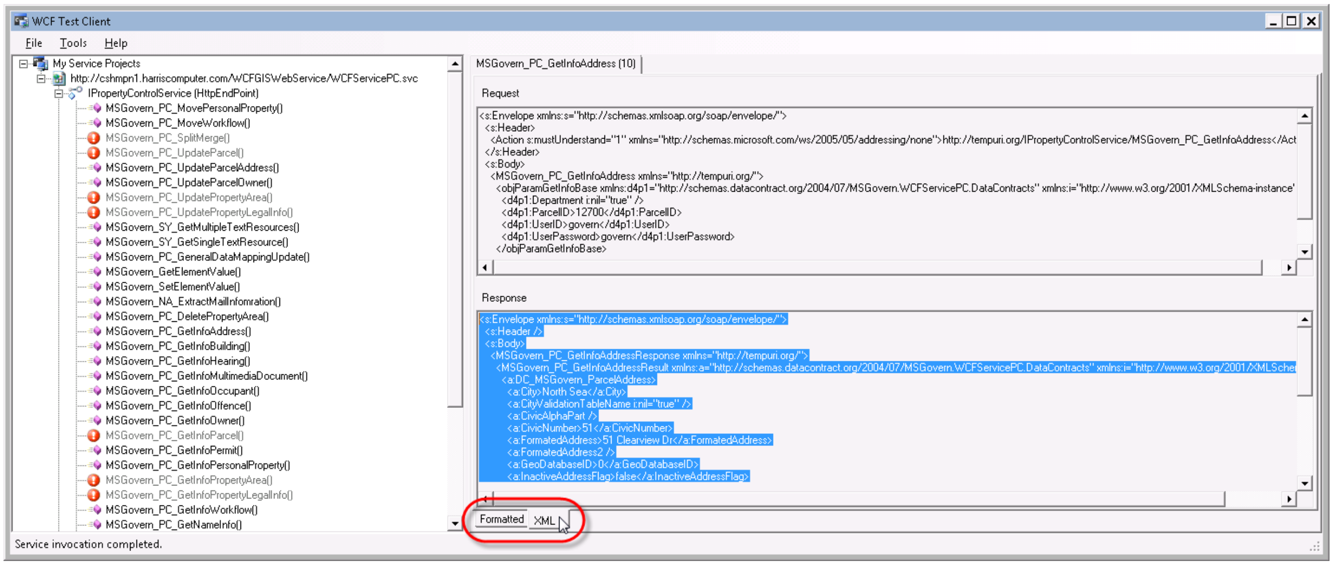
| Name | Value | Type |
|---|---|---|

6. When parameters are to be entered, click in the field to highlight the existing parameter, delete and enter the required values; for this method, values are required for Department, ParcelID, UserID, and a UserPassword.
7. When the fields have been completed, click **Invoke** to enter the request.
8. When the test is successful, a "Response" is sent back and displayed in the **Response Parameters** area in the lower right hand section of the client.

As a new method is selected for testing, a tab is displayed at the top of the Request Parameters area for easy selection. In addition the request and response parameters can be displayed in two (2) modes, **Formatted**, and **XML**. You can switch between the two views with a click on the corresponding tabs along the bottom of the *Request Parameters* pane.


WCF Test Client
File
Tools
Help
My Service Projects
http://jxmtln1.harriscomputer.com/WCFServicePropertyControl/WCFS
IPropertyControlService (HttpEndPoint)
MSGovern_PC_MovePersonalProperty()
MSGovern_PC_MoveWorkflow()
MSGovern_PC_SplitMerge()
MSGovern_PC_UpdateParcel()
MSGovern_PC_UpdateParcelAddress()
MSGovern_PC_UpdateParcelOwner()
MSGovern_PC_UpdatePropertyArea()
MSGovern_PC_UpdatePropertyLegalInfo()
MSGovern_SY_GetMultipleTextResources()
MSGovern_SY_GetSingleTextResource()
MSGovern_PC_GeneralDataMappingUpdate()
MSGovern_GetElementValue()
MSGovern_SetElementValue()
MSGovern_NA_ExtractMailInfomration()
MSGovern_PC_DeletePropertyArea()
MSGovern_PC_GetInfoAddress()
MSGovern_PC_GetInfoBuilding()
MSGovern_PC_GetInfoHearing()
MSGovern_PC_GetInfoMultimediaDocument()
MSGovern_PC_Mov...nalProperty (2)
MSGovern_PC_MoveWorkflow
MSGovern_SY_Get...le
Request
IPropertyControlService.MSGovern_PC_MovePersonalProperty (2)
Response
Start a new proxy
Invoke
Formatted
XML
Service added successfully.

| Name | Value | Type |
|---|---|---|
| objParamMoveWorkflow | MSGovern.WCFServicePC.Data | MSGovern.WCFServicePC.Dat |
| WorkflowID | (null) | System.Nullable<System.Int32> |
| WorkflowType | (null) | System.Nullable<MSGovern.WC |
| Department | (null) | System.String |
| DestinationParcelID | 0 | System.Int32 |
| SourceParcelID | 0 | System.Int32 |


WCF Test Client
File
Tools
Help
My Service Projects
http://cshmpn1.harriscomputer.com/WCFGISWebService/WCFServicePC.svc
IPropertyControlService (HttpEndPoint)
MSGovern_PC_MovePersonalProperty()
MSGovern_PC_MoveWorkflow()
MSGovern_PC_SplitMerge()
MSGovern_PC_UpdateParcel()
MSGovern_PC_UpdateParcelAddress()
MSGovern_PC_UpdateParcelOwner()
MSGovern_PC_UpdatePropertyArea()
MSGovern_PC_UpdatePropertyLegalInfo()
MSGovern_SY_GetMultipleTextResources()
MSGovern_SY_GetSingleTextResource()
MSGovern_PC_GeneralDataMappingUpdate()
MSGovern_GetElementValue()
MSGovern_SetElementValue()
MSGovern_NA_ExtractMailInfomration()
MSGovern_PC_DeletePropertyArea()
MSGovern_PC_GetInfoAddress()
MSGovern_PC_GetInfoBuilding()
MSGovern_PC_GetInfoHearing()
MSGovern_PC_GetInfoMultimediaDocument()
MSGovern_PC_GetInfoOccupant()
MSGovern_PC_GetInfoOffence()
MSGovern_PC_GetInfoOwner()
MSGovern_PC_GetInfoParcel()
MSGovern_PC_GetInfoPermit()
MSGovern_PC_GetInfoPersonalProperty()
MSGovern_PC_GetInfoPropertyArea()
MSGovern_PC_GetInfoPropertyLegalInfo()
MSGovern_PC_GetInfoWorkflow()
MSGovern_PC_GetNameInfo()
MSGovern_PC_GetInfoAddress (10)
Request
Response
Formatted
XML
Service invocation completed.

Request

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Header>
  <Action s:mustUnderstand="1" xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">http://tempuri.org/IPropertyControlService/MSGovern_PC_GetInfoAddress</Act
 </s:Header>
 <s:Body>
  <MSGovern_PC_GetInfoAddress xmlns="http://tempuri.org/">
   <objParamGetInfoBase xmlns:d4p1="http://schemas.datacontract.org/2004/07/MSGovern.WCFServicePC.DataContracts" xmlns:i="http://www.w3.org/2001/XMLSchema-instance'
    <d4p1:Department i:nil="true" />
    <d4p1:ParcelID>12700</d4p1:ParcelID>
    <d4p1:UserID>govern</d4p1:UserID>
    <d4p1:UserPassword>govern</d4p1:UserPassword>
   </objParamGetInfoBase>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Header />
 <s:Body>
  <MSGovern_PC_GetInfoAddressResponse xmlns="http://tempuri.org/">
   <MSGovern_PC_GetInfoAddressResult xmlns:a="http://schemas.datacontract.org/2004/07/MSGovern.WCFServicePC.DataContracts" xmlns:i="http://www.w3.org/2001/XMLSche
    <a:DC_MSGovern_ParcelAddress>
     <a:City>North Sea</a:City>
     <a:CityValidationTableName i:nil="true" />
     <a:CivicAlphaPart />
     <a:CivicNumber>51</a:CivicNumber>
     <a:FormatedAddress>51 Clearview Dr</a:FormatedAddress>
     <a:FormatedAddress2 />
     <a:GeoDatabaseID>0</a:GeoDatabaseID>
     <a:InactiveAddressFlag>false</a:InactiveAddressFlag>
```

**NOTE**: Methods that are not supported by the WCF client are listed with a red exclamation mark beside them.

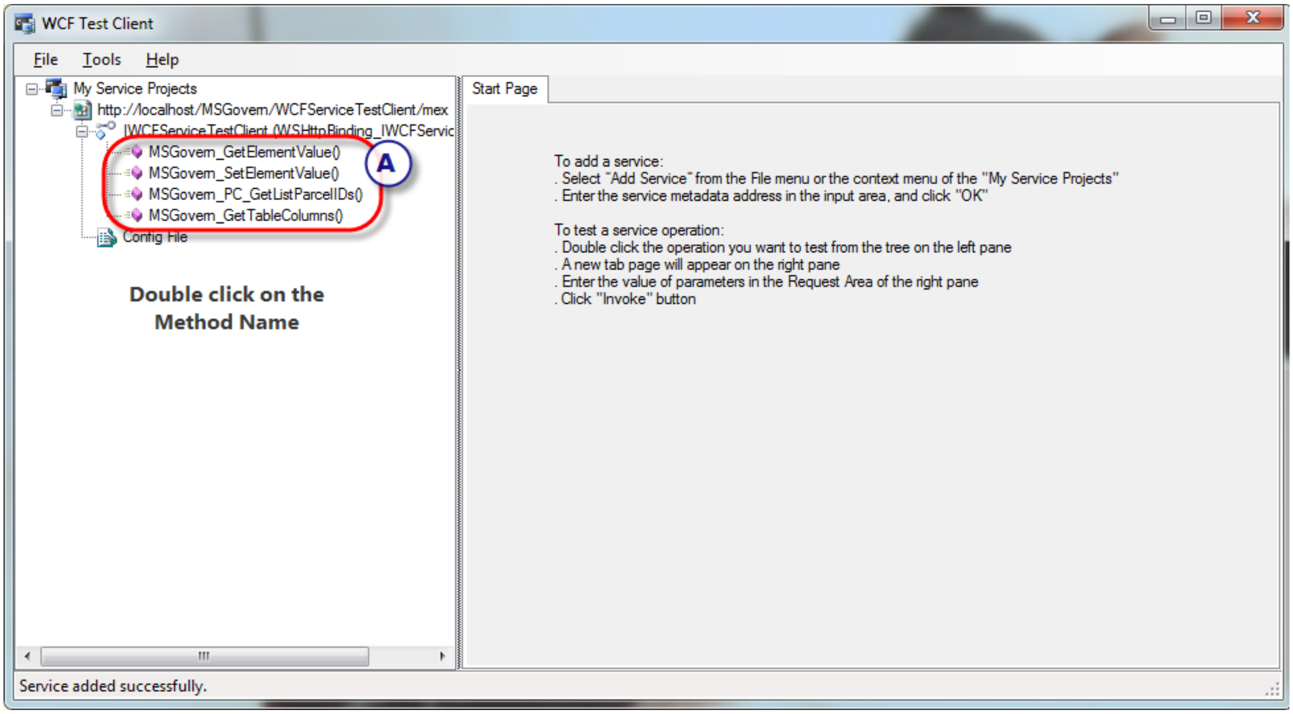
## Testing the GISGetElementValue / GISSetElementValue Methods

**NEW!** Two (2) new GIS related methods have been added to the Property Control Web Service. The methods are called *MSGovern_GetElementValue*, and *MSGovern_SetElementValue*. *Refer to the **MS Govern Property Control System Service API** document for details about these methods.*

When an external application, e.g. a GIS application, is required to communicate with the Govern system, it will need to understand how data is stored within the system. The **GISGetElementValue** will allow you *Retrieve* data, and the **GISSetElementValue** method will allow you to *Update*, or *Delete* data.

In order to be able to accomplish the aforementioned actions through the Web service, you must first map the required tables and columns within Govern's database, to unique *Data Element Names*. Refer to the **Data Mapping Editor** section of the *Govern Net Admin* **(GNA)** *release 4.7* user guide for details, steps for using the *WCF Test Client* application.

### Testing Get Element Value / Set Element Value / Get Table Column


WCF Test Client
File
Tools
Help
My Service Projects
http://localhost/MSGovern/WCFServiceTestClient/mex
IWCFServiceTestClient (WSHttpBinding_IWCFServic
MSGovern_GetElementValue()
MSGovern_SetElementValue()
MSGovern_PC_GetListParcelIDs()
MSGovern_GetTableColumns()
Config File
A
Start Page

Double click on the Method Name

To add a service:
. Select "Add Service" from the File menu or the context menu of the "My Service Projects"
. Enter the service metadata address in the input area, and click "OK"

To test a service operation:
. Double click the operation you want to test from the tree on the left pane
. A new tab page will appear on the right pane
. Enter the value of parameters in the Request Area of the right pane
. Click "Invoke" button

Service added successfully.

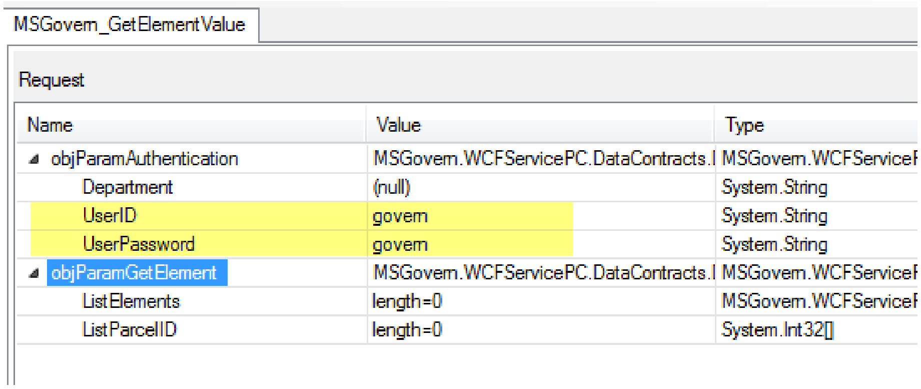
The following are the procedures for testing the *Get Element Value*, *Set Element Value*, and the Get *Table Column* methods (**A**) with the WCF Application.

### Get Element Value

In the WCF Test Client…

1. Double click on `MSGovern_GetElementValue`

2. Set `User_ID` and `Password` in `objParamAuthentication`



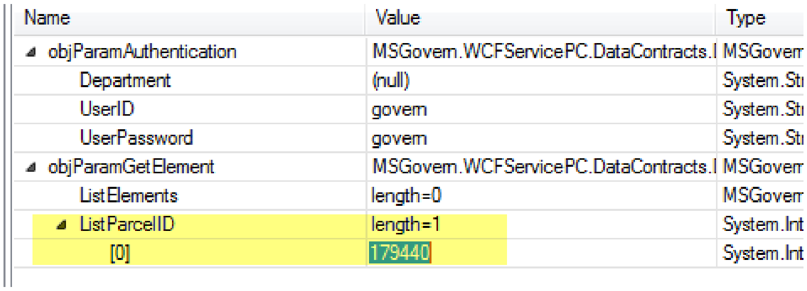
MSGovern_GetElementValue

Request

| Name | Value | Type |
|---|---|---|
| ◢ objParamAuthentication | MSGovern.WCFServicePC.DataContracts.I | MSGovern.WCFServiceP |
| Department | (null) | System.String |
| UserID | govern | System.String |
| UserPassword | govern | System.String |
| ◢ objParamGetElement | MSGovern.WCFServicePC.DataContracts.I | MSGovern.WCFServiceP |
| ListElements | length=0 | MSGovern.WCFServiceP |
| ListParcelID | length=0 | System.Int32[] |

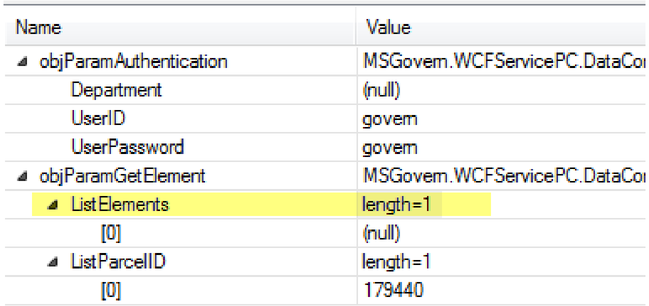
In the `objParamGetElement`, set the following:

3. Set `length=1` in `ListParcel` and set the `p_id`.



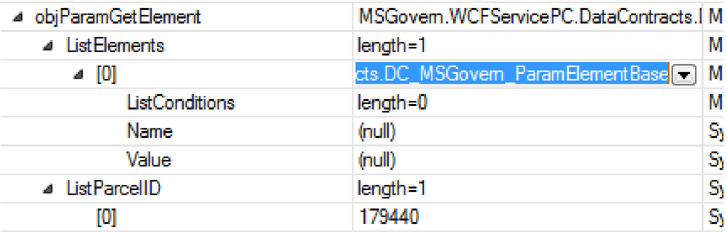
| Name | Value | Type |
|---|---|---|
| ◢ objParamAuthentication | MSGovern.WCFServicePC.DataContracts.I | MSGovern |
| Department | (null) | System.Str |
| UserID | govern | System.Str |
| UserPassword | govern | System.Str |
| ◢ objParamGetElement | MSGovern.WCFServicePC.DataContracts.I | MSGovern |
| ListElements | length=0 | MSGovern |
| ◢ ListParcelID | length=1 | System.Int |
| [0] | 179440 | System.Int |

4. Set the `length` in `ListElements` according to the number of elements, i.e. the code that you would like to search for. This value is set in the data mapping from the *Govern System Configuration Administrator* (**SCA**).



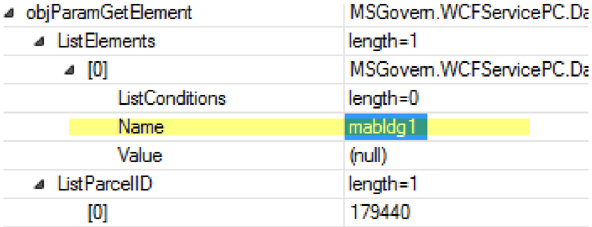
| Name | Value |
|---|---|
| ◢ objParamAuthentication | MSGovern.WCFServicePC.DataCo |
| Department | (null) |
| UserID | govern |
| UserPassword | govern |
| ◢ objParamGetElement | MSGovern.WCFServicePC.DataCo |
| ◢ ListElements | length=1 |
| [0] | (null) |
| ◢ ListParcelID | length=1 |
| [0] | 179440 |

5. Click on `ListElements`; in the drop down menu select the required object type.



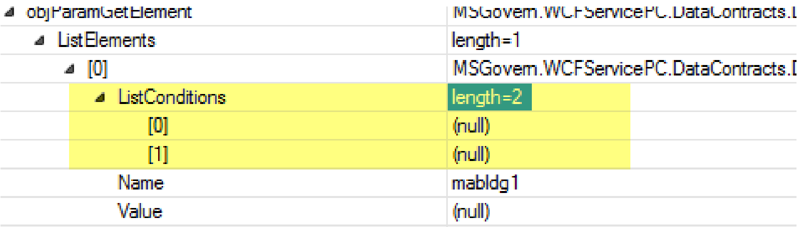
| | | |
|---|---|---|
| ◢ objParamGetElement | MSGovern.WCFServicePC.DataContracts.I | M |
| ◢ ListElements | length=1 | M |
| ◢ [0] | cts.DC_MSGovern_ParamElementBase ▾ | M |
| ListConditions | length=0 | M |
| Name | (null) | Sy |
| Value | (null) | Sy |
| ◢ ListParcelID | length=1 | Sy |
| [0] | 179440 | Sy |

6. Enter the element code in the Name field



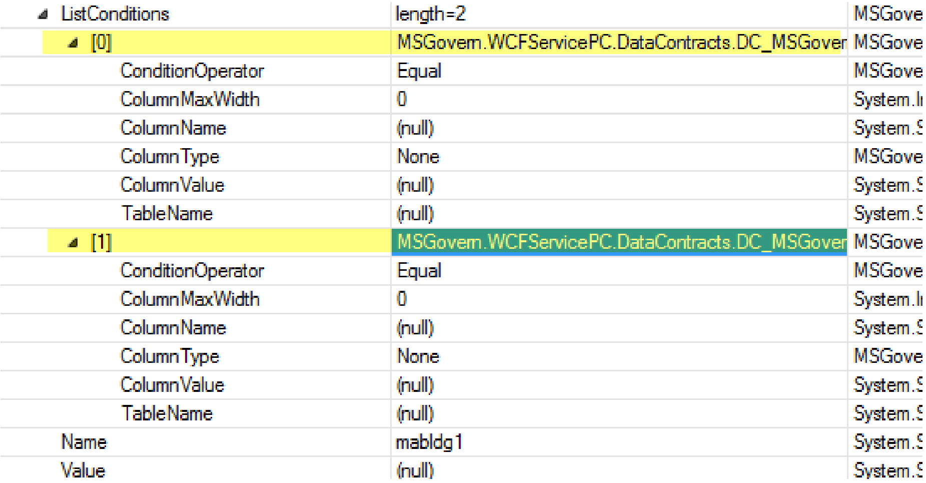
| | |
|---|---|
| ◢ objParamGetElement | MSGovern.WCFServicePC.Da |
| ◢ ListElements | length=1 |
| ◢ [0] | MSGovern.WCFServicePC.Da |
| ListConditions | length=0 |
| Name | mabldg1 |
| Value | (null) |
| ◢ ListParcelID | length=1 |
| [0] | 179440 |

7. Set the `length` value in `ListConditions` depending on the number of conditions you would like to add in your search.



| | |
|---|---|
| ◢ objParamGetElement | MSGovern.WCFServicePC.DataContracts.I |
| ◢ ListElements | length=1 |
| ◢ [0] | MSGovern.WCFServicePC.DataContracts.I |
| ◢ ListConditions | length=2 |
| [0] | (null) |
| [1] | (null) |
| Name | mabldg1 |
| Value | (null) |

8. Select the required type in the drop down menu.



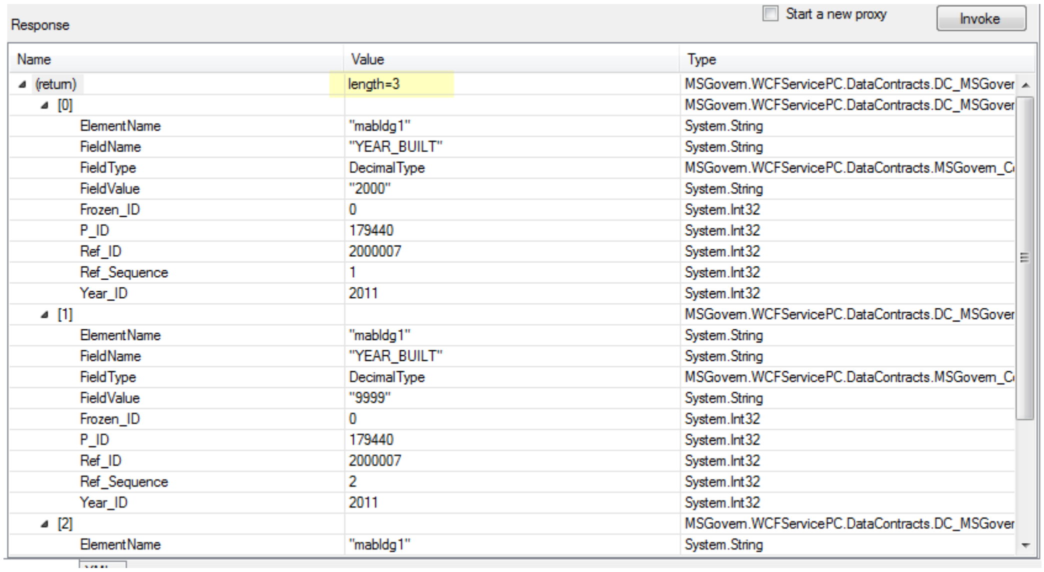
| | | |
|---|---|---|
| ◢ ListConditions | length=2 | MSGove |
| ◢ [0] | MSGovern.WCFServicePC.DataContracts.DC_MSGover | MSGove |
| ConditionOperator | Equal | MSGove |
| ColumnMaxWidth | 0 | System.I |
| ColumnName | (null) | System.S |
| ColumnType | None | MSGove |
| ColumnValue | (null) | System.S |
| TableName | (null) | System.S |
| ◢ [1] | MSGovern.WCFServicePC.DataContracts.DC_MSGover | MSGove |
| ConditionOperator | Equal | MSGove |
| ColumnMaxWidth | 0 | System.I |
| ColumnName | (null) | System.S |
| ColumnType | None | MSGove |
| ColumnValue | (null) | System.S |
| TableName | (null) | System.S |
| Name | mabldg1 | System.S |
| Value | (null) | System.S |

9. Enter `ColumnName`, `ConditionOperator` and `ColumnValue`; note that…

- `ColumnValue`: In order to be able to search for a null value in the database, this value can be set as null.

- `ConditionOperator`: The default operator is `EQUAL TO` (=)

10. Click on the Invoke button; three (3) records will be returned.



| Name | Value | Type |
|---|---|---|
| (return) | length=3 | MSGovern.WCFServicePC.DataContracts.DC_MSGover |
| [0] | | MSGovern.WCFServicePC.DataContracts.DC_MSGover |
| ElementName | "mabldg1" | System.String |
| FieldName | "YEAR_BUILT" | System.String |
| FieldType | DecimalType | MSGovern.WCFServicePC.DataContracts.MSGovern_C |
| FieldValue | "2000" | System.String |
| Frozen_ID | 0 | System.Int32 |
| P_ID | 179440 | System.Int32 |
| Ref_ID | 2000007 | System.Int32 |
| Ref_Sequence | 1 | System.Int32 |
| Year_ID | 2011 | System.Int32 |
| [1] | | MSGovern.WCFServicePC.DataContracts.DC_MSGover |
| ElementName | "mabldg1" | System.String |
| FieldName | "YEAR_BUILT" | System.String |
| FieldType | DecimalType | MSGovern.WCFServicePC.DataContracts.MSGovern_C |
| FieldValue | "9999" | System.String |
| Frozen_ID | 0 | System.Int32 |
| P_ID | 179440 | System.Int32 |
| Ref_ID | 2000007 | System.Int32 |
| Ref_Sequence | 2 | System.Int32 |
| Year_ID | 2011 | System.Int32 |
| [2] | | MSGovern.WCFServicePC.DataContracts.DC_MSGover |
| ElementName | "mabldg1" | System.String |

**NOTE**: You do not need to provide a condition in `ListConditions` when you want to search all records in the database.

The `Name` of `objParamGetElement` must exist in the `USR_DATA_MAPPING` table; this can be set up in the SCA.

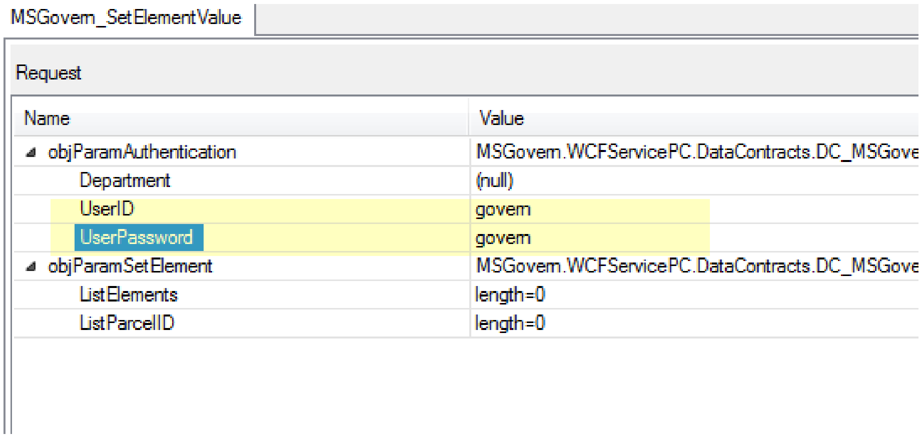
## Set Element Value *(Update)*

To test the Set Element Value method...

In the WCF Test Client...

1. Double click on `MSGovern_SetElementValue`

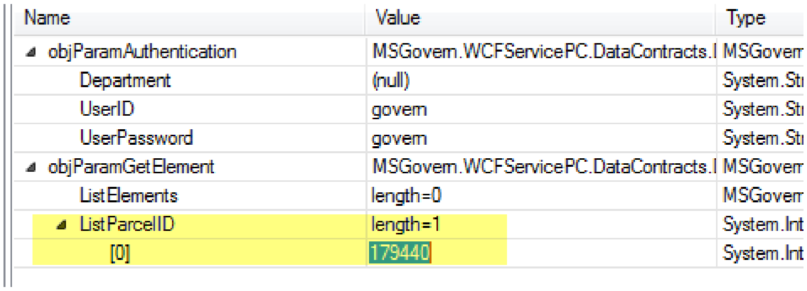
2. Set user ID and password in `objParamAuthentication.`



| Name | Value |
|---|---|
| objParamAuthentication | MSGovern.WCFServicePC.DataContracts.DC_MSGove |
| Department | (null) |
| UserID | govern |
| UserPassword | govern |
| objParamSetElement | MSGovern.WCFServicePC.DataContracts.DC_MSGove |
| ListElements | length=0 |
| ListParcelID | length=0 |

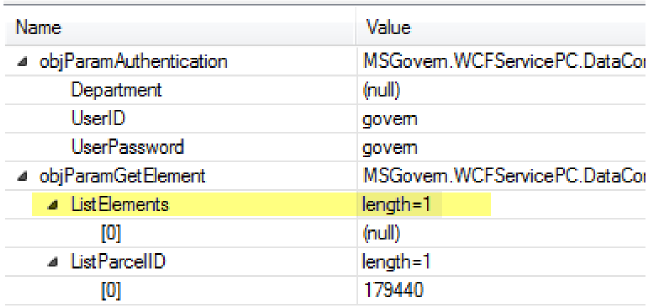
In the `objParamGetElement`, set the following:

3. Set `length=1` in `ListParcel` and set the `p_id`.



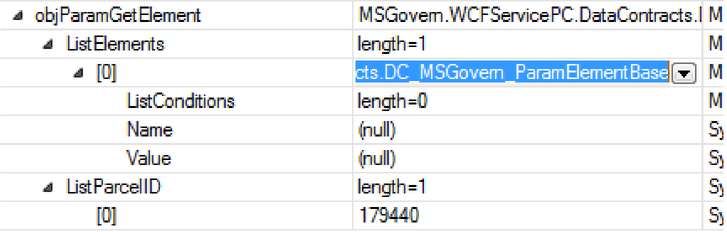
| Name | Value | Type |
|---|---|---|
| ◢ objParamAuthentication | MSGovern.WCFServicePC.DataContracts.I | MSGovern |
| Department | (null) | System.Str |
| UserID | govern | System.Str |
| UserPassword | govern | System.Str |
| ◢ objParamGetElement | MSGovern.WCFServicePC.DataContracts.I | MSGovern |
| ListElements | length=0 | MSGovern |
| ◢ ListParcelID | length=1 | System.Int |
| [0] | 179440 | System.Int |

4. Set the `length` in `ListElements` according to the number of elements, i.e. the code that you would like to search for. This value is set in the data mapping from the *Govern System Configuration Administrator* (**SCA**).



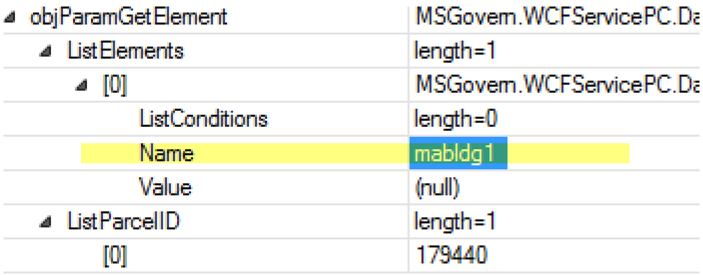
| Name | Value |
|---|---|
| ◢ objParamAuthentication | MSGovern.WCFServicePC.DataCo |
| Department | (null) |
| UserID | govern |
| UserPassword | govern |
| ◢ objParamGetElement | MSGovern.WCFServicePC.DataCo |
| ◢ ListElements | length=1 |
| [0] | (null) |
| ◢ ListParcelID | length=1 |
| [0] | 179440 |

5. Click on `ListElements`; in the drop down menu select the required object type.



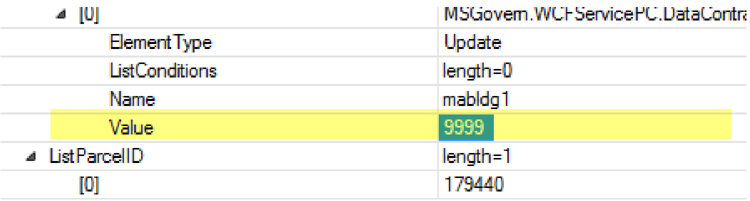
| | | |
|---|---|---|
| ◢ objParamGetElement | MSGovern.WCFServicePC.DataContracts.I | M |
| ◢ ListElements | length=1 | M |
| ◢ [0] | cts.DC_MSGovern_ParamElementBase | M |
| ListConditions | length=0 | M |
| Name | (null) | Sy |
| Value | (null) | Sy |
| ◢ ListParcelID | length=1 | Sy |
| [0] | 179440 | Sy |

6. Enter the element code in the Name field



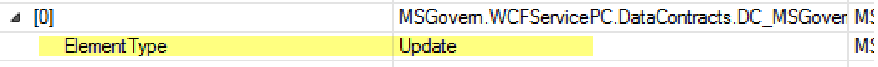
| | |
|---|---|
| ◢ objParamGetElement | MSGovern.WCFServicePC.Da |
| ◢ ListElements | length=1 |
| ◢ [0] | MSGovern.WCFServicePC.Da |
| ListConditions | length=0 |
| Name | mabldg1 |
| Value | (null) |
| ◢ ListParcelID | length=1 |
| [0] | 179440 |

7. Enter the element value to be set in the database.



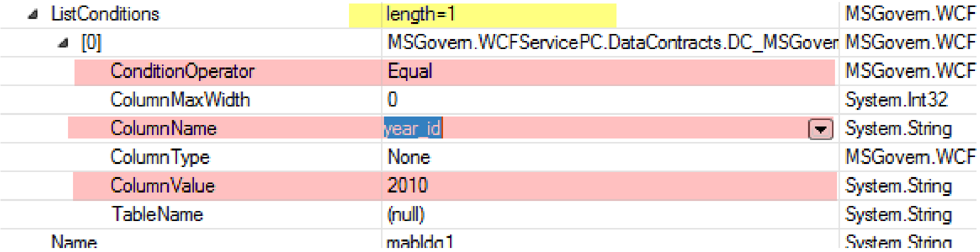
| | |
|---|---|
| ◢ [0] | MSGovern.WCFServicePC.DataContra |
| ElementType | Update |
| ListConditions | length=0 |
| Name | mabldg1 |
| Value | 9999 |
| ◢ ListParcelID | length=1 |
| [0] | 179440 |

8. Make sure the `ElementType` is set to `Update`.



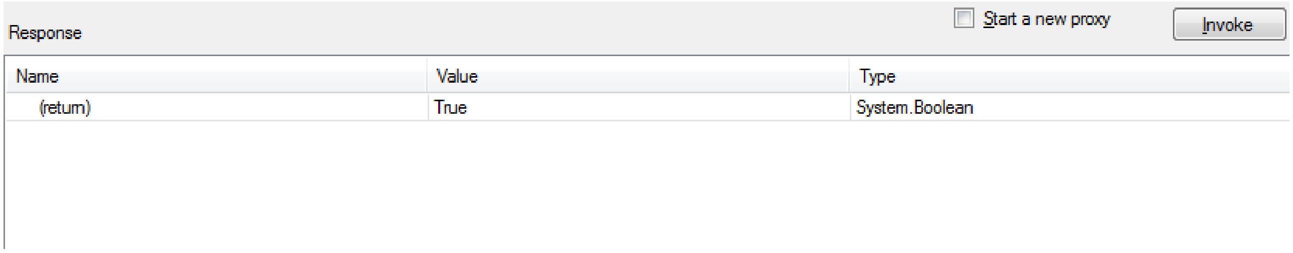
| | | |
|---|---|---|
| ◢ [0] | MSGovern.WCFServicePC.DataContracts.DC_MSGover | MS |
| ElementType | Update | MS |

9. Depending on the number of conditions you want to add in your search, set the `length` under the `ListConditions`.



| | | |
|---|---|---|
| ◢ ListConditions | length=1 | MSGovern.WCF |
| ◢ [0] | MSGovern.WCFServicePC.DataContracts.DC_MSGover | MSGovern.WCF |
| ConditionOperator | Equal | MSGovern.WCF |
| ColumnMaxWidth | 0 | System.Int32 |
| ColumnName | year_id | System.String |
| ColumnType | None | MSGovern.WCF |
| ColumnValue | 2010 | System.String |
| TableName | (null) | System.String |
| Name | mabldg1 | System.String |

10. Click **Invoke**; the record will be updated in the database.



| Name | Value | Type |
|---|---|---|
| (return) | True | System.Boolean |

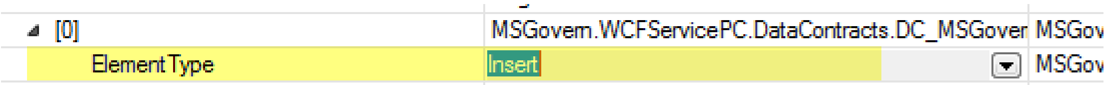
**NOTE**:
When you want to search all records in the database. You do not need to provide a condition in `ListConditions.`

The `Name` of `objParamGetElement` must exist in the `USR_DATA_MAPPING` table; this can be set up in the SCA.

## Set Element Value *(Insert)*

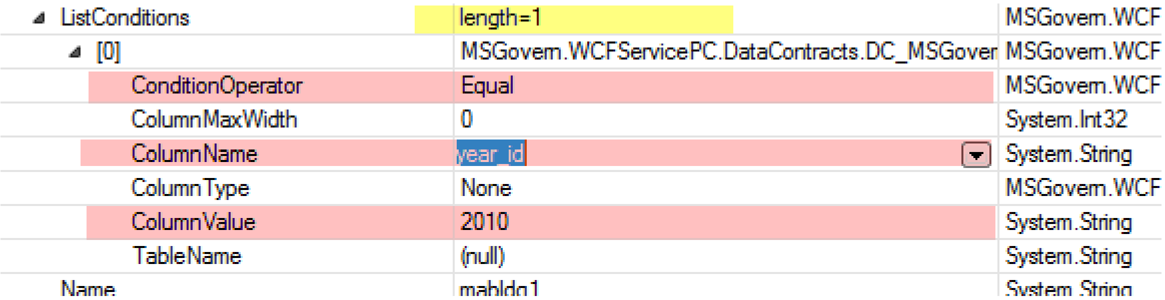
To test the `Set Element Value` method...

1. Repeat steps 1, and 3-7 from Set Element Value (Update)
2. Select ElementType is Insert



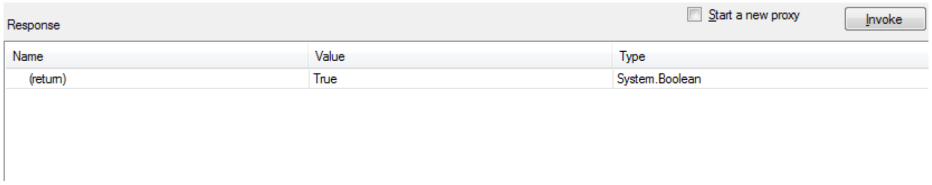
| | | |
|---|---|---|
| ◢ [0] | MSGovern.WCFServicePC.DataContracts.DC_MSGover | MSGov |
| ElementType | Insert | MSGov |

3. Set `length` in `ListConditions` depending on the number of condition you want to add in your search. All the objects added in this list will be inserting in the record. You will need to add all the primary keys in this list.



| | | |
|---|---|---|
| ListConditions | length=1 | MSGovern.WCF |
| [0] | MSGovern.WCFServicePC.DataContracts.DC_MSGover | MSGovern.WCF |
| ConditionOperator | Equal | MSGovern.WCF |
| ColumnMaxWidth | 0 | System.Int32 |
| ColumnName | year_id | System.String |
| ColumnType | None | MSGovern.WCF |
| ColumnValue | 2010 | System.String |
| TableName | (null) | System.String |
| Name | mabldg1 | System.String |

4. Click **Invoke**; the record will be inserted in the database.



Response

| Name | Value | Type |
|---|---|---|
| (return) | True | System.Boolean |

**NOTE**: If a table has primary keys and you want to insert more values in the database, you will need to provide a condition in `ListConditions.`

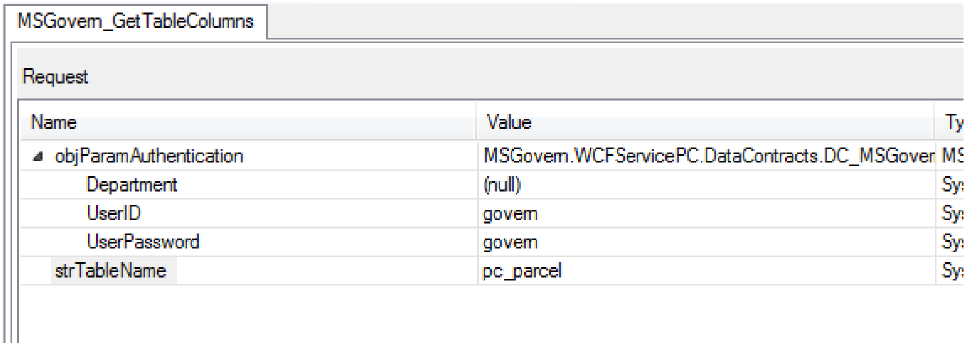
The `Name` of `objParamGetElement` must exist in the `USR_DATA_MAPPING` table; this can be set up in the SCA.

You can use the `MsGovern_GetTableColumns` method to retrieve all fields from the given table which can then be added in the `ListConditions`.

## Get Table Column

To test the Get Table Column method...

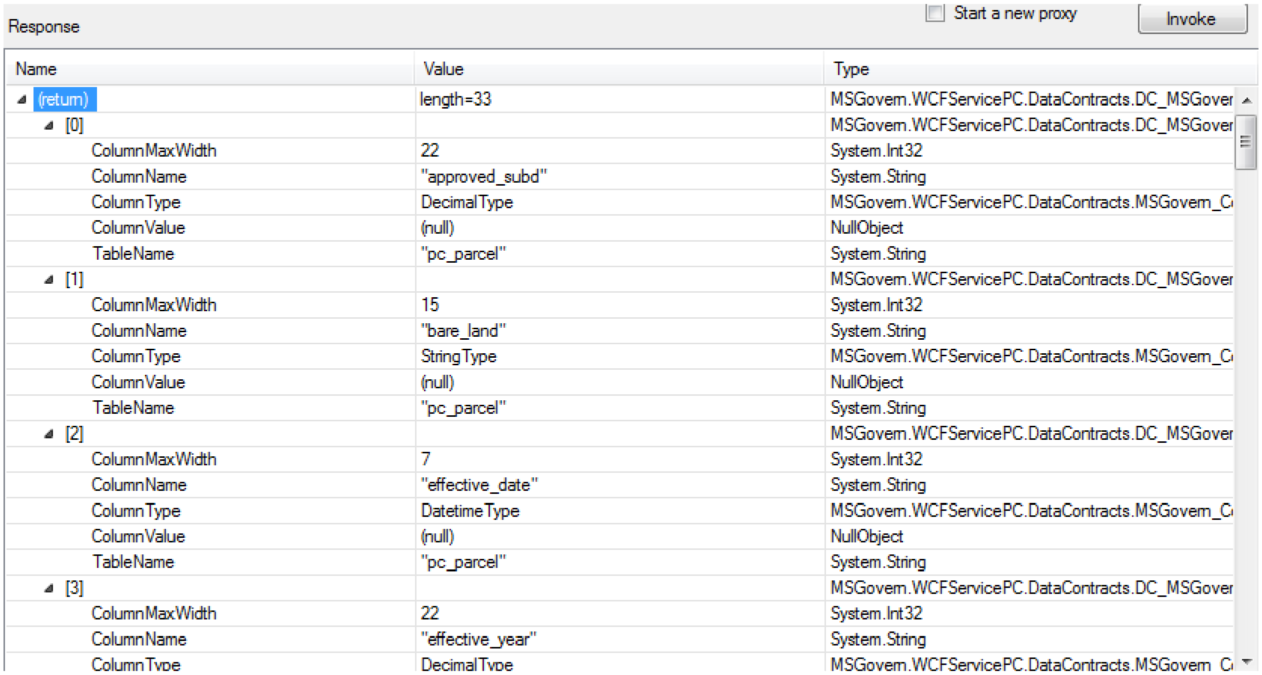
1. Enter the User ID, Password, and Table Name



Request

| Name | Value | Ty |
|---|---|---|
| objParamAuthentication | MSGovern.WCFServicePC.DataContracts.DC_MSGover | MS |
| Department | (null) | Sy |
| UserID | govern | Sy |
| UserPassword | govern | Sy |
| strTableName | pc_parcel | Sy |

2. Click **Invoke**.



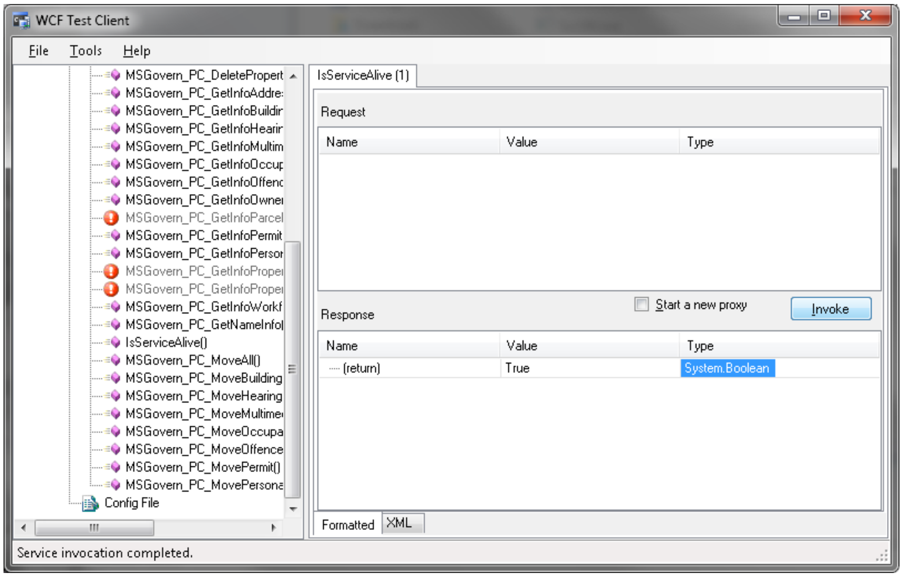
| Name | Value | Type |
|---|---|---|
| ◢ (return) | length=33 | MSGovern.WCFServicePC.DataContracts.DC_MSGover |
| ◢ [0] | | MSGovern.WCFServicePC.DataContracts.DC_MSGover |
| ColumnMaxWidth | 22 | System.Int32 |
| ColumnName | "approved_subd" | System.String |
| ColumnType | DecimalType | MSGovern.WCFServicePC.DataContracts.MSGovern_C |
| ColumnValue | (null) | NullObject |
| TableName | "pc_parcel" | System.String |
| ◢ [1] | | MSGovern.WCFServicePC.DataContracts.DC_MSGover |
| ColumnMaxWidth | 15 | System.Int32 |
| ColumnName | "bare_land" | System.String |
| ColumnType | StringType | MSGovern.WCFServicePC.DataContracts.MSGovern_C |
| ColumnValue | (null) | NullObject |
| TableName | "pc_parcel" | System.String |
| ◢ [2] | | MSGovern.WCFServicePC.DataContracts.DC_MSGover |
| ColumnMaxWidth | 7 | System.Int32 |
| ColumnName | "effective_date" | System.String |
| ColumnType | DatetimeType | MSGovern.WCFServicePC.DataContracts.MSGovern_C |
| ColumnValue | (null) | NullObject |
| TableName | "pc_parcel" | System.String |
| ◢ [3] | | MSGovern.WCFServicePC.DataContracts.DC_MSGover |
| ColumnMaxWidth | 22 | System.Int32 |
| ColumnName | "effective_year" | System.String |
| ColumnType | DecimalType | MSGovern.WCFServicePC.DataContracts.MSGovern_C |

# TROUBLE SHOOTING

## Connections

In instances when there are issues with the connection, there is a method win the service that can be used to assist in trouble shooting.

1. In the WCF Test Client, double click to select the **IsServiceAlive()** method.
2. Click **Invoke**.



MSGovern_PC_DeletePropert
MSGovern_PC_GetInfoAddre:
MSGovern_PC_GetInfoBuildir
MSGovern_PC_GetInfoHearir
MSGovern_PC_GetInfoMultim
MSGovern_PC_GetInfoOccup
MSGovern_PC_GetInfoOffenc
MSGovern_PC_GetInfoOwnei
MSGovern_PC_GetInfoParcel
MSGovern_PC_GetInfoPermit
MSGovern_PC_GetInfoPersor
MSGovern_PC_GetInfoPropei
MSGovern_PC_GetInfoPropei
MSGovern_PC_GetInfoWorkf
MSGovern_PC_GetNameInfo
IsServiceAlive()
MSGovern_PC_MoveAll()
MSGovern_PC_MoveBuilding
MSGovern_PC_MoveHearing
MSGovern_PC_MoveMultime
MSGovern_PC_MoveOccupa
MSGovern_PC_MoveOffence
MSGovern_PC_MovePermit()
MSGovern_PC_MovePersona

| Name | Value | Type |
|---|---|---|

| Name | Value | Type |
|---|---|---|
| (return) | True | System.Boolean |

Service invocation completed.

The *IsServiceAlive()* method will test the validity of the connection key and the connection to the database. When the connection is successful, the method will return as **TRUE**. If the service fails, make a note of the error message and verify the validity of your connection key and your connection to the database. The error message that is recorded can assist in the resolution process should the problem need to be escalated.