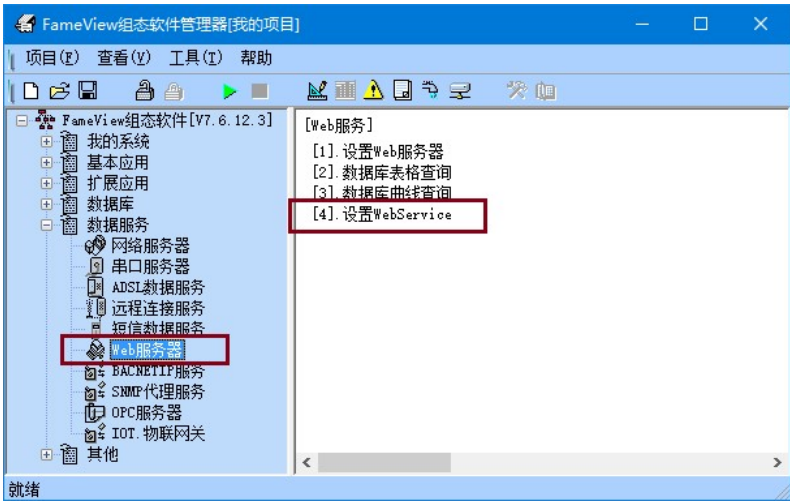


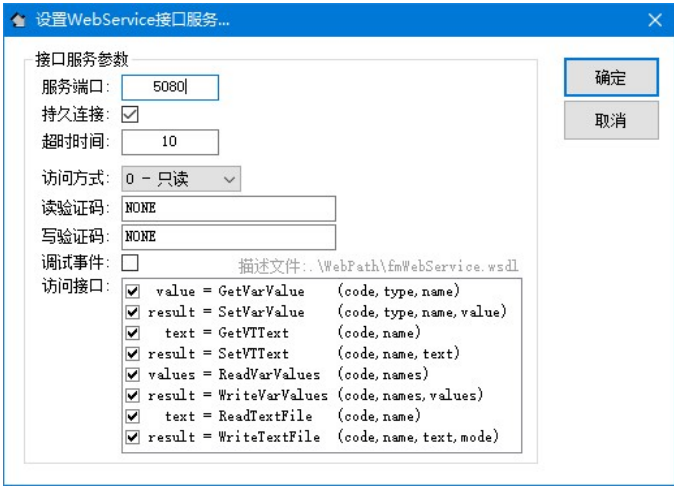
44. Webservice 接口服务

44.1 设置 Webservice 接口

□ 选择[Web 服务器]:



□ 执行[设置 Webservice]:

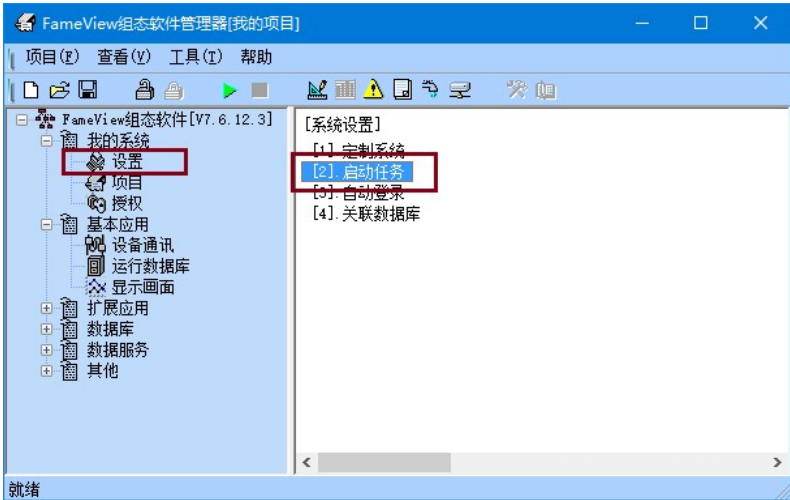


□ 参数说明:

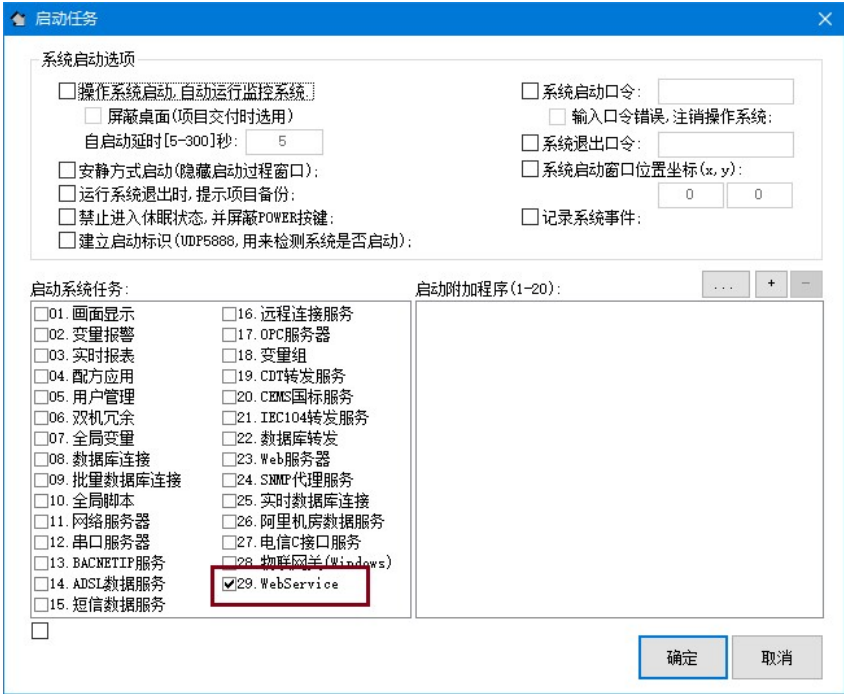
| 参数名称 | 参数描述 | 缺省值 |
|------|---|--|
| 服务端口 | Webservice 服务使用的 TCP 端口号, 范围:100~100000 | 5080 |
| 持久连接 | 是:以长连接方式连续处理多条请求直到超时关闭, 否:以短连接方式处理单条请求后关闭 | 是 |
| 访问方式 | 0:只读, 只支持读功能函数, 1=读写, 支持全部功能函数 | 0 |
| 读验证码 | 读功能函数执行, 需要检查读验证码参数, 验证成功则继续执行 | 最大长度:30 NONE |
| 写验证码 | 写功能函数执行, 需要检查写验证码参数, 验证成功则继续执行 | NONE:忽略检查验证码 NONE |
| 调试事件 | 生成调试事件, 并通过实用工具. 系统事件记录查看 | 否 |
| 访问接口 | 选择是否支持下列接口函数 | |
| | 获取某个数值变量值 | value = GetVarValue (code, type, name) 是 |
| | 设置某个数值变量值 | result = SetVarValue (code, type, name, value) 是 |
| | 获取某个文本变量值 | text = GetVTText (code, name) 是 |
| | 设置某个文本变量值 | result = SetVTText (code, name, text) 是 |
| | 读取某批数值和文本变量值 | values = ReadVarValues (code, name) 是 |
| | 修改某批数值和文本变量值 | result = WriteVarValue (code, name, values) 是 |
| | 读取 WebPath 目录下的某文本文件 | text = ReadTextFile (code, name) 是 |
| | 写入 WebPath 目录下的某文本文件 | result = WriteTextFile(code, name, text, mode) 是 |

44.2 启动 WebService 服务

□ 选择[我的系统. 设置]:

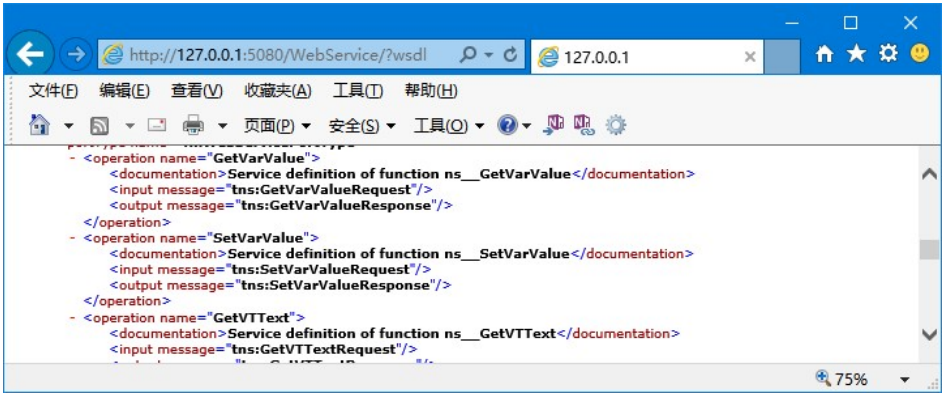


□ 执行[启动任务]:



□ 选择启动 WebService;

□ 通过 IE 输入 `http://127.0.0.1:5080/WebService/?wsdl`, 检查 WebService 是否启动:



Url 地址后附加“?”, 可获取 WSDL 文件;

44.3 功能函数

- 获取运行数据库某个数值变量值

| | | | |
|---|-------|-------|--|
| double varValue=GetVarValue(char* readCode, int varType, char* varName) | | | |
| readCode | 读验证码 | 字符串 | 长度 1-30, 服务被设置 NONE 时忽略 |
| varType | 变量类型 | 整型 | 1=AI, 2=A0, 3=AR, 4=DI, 5=DO, 6=DR, 7=VA, 8=VD |
| varName | 变量名称 | 字符串 | 长度 1-20 |
| varValue | 返回变量值 | 双精度浮点 | 变量类型 (DI DO DR VD) 时, 内容总为 0 1 |
| 例: double varValue=GetVarValue("test", 1, "AI1") | | | |

- 设置运行数据库某个数值变量值

| | | | |
|--|-------|-------|------------------------------------|
| int result=SetVarValue(char* writeCode, int varType, char* varName, double varValue) | | | |
| writeCode | 写验证码 | 字符串 | 长度 1-30, 服务被设置 NONE 时忽略 |
| varType | 变量类型 | 整型 | 2=A0, 3=AR, 5=DO, 6=DR, 7=VA, 8=VD |
| varName | 变量名称 | 字符串 | 长度 1-20 |
| varValue | 变量设置值 | 双精度浮点 | 变量类型 (DO DR VD) 时, 内容总为 0 1 |
| result | 返回结果值 | 整型 | 0:成功 |
| 例: int result=SetVarValue("test", 7, "VA1", 123.456) | | | |

- 获取运行数据库某个文本变量值

| | | | |
|---|-------|-----|-------------------------|
| char* varValue=GetVTText(char* readCode, char* varName) | | | |
| readCode | 读验证码 | 字符串 | 长度 1-30, 服务被设置 NONE 时忽略 |
| varName | 变量名称 | 字符串 | 长度 1-20 |
| varValue | 返回变量值 | 字符串 | 最大长度 240 |
| 例: char* varValue=GetVTText("test", "VT1") | | | |

- 设置运行数据库某个文本变量值

| | | | |
|--|-------|-----|-------------------------|
| int result=SetVTText(char* writeCode, char* varName, char* varValue) | | | |
| writeCode | 写验证码 | 字符串 | 长度 1-30, 服务被设置 NONE 时忽略 |
| varName | 变量名称 | 字符串 | 长度 1-20 |
| varValue | 变量设置值 | 字符串 | 最大长度 240 |
| result | 返回结果值 | 整型 | 0:成功 |
| 例: int result=SetVTText("test", "VT1", "abcd1234") | | | |

- 获取运行数据库某批数值和文本变量值

| | | | |
|---|--------|-----|---------------------------|
| char* varValues=ReadVarValues(char* readCode, char* varNames) | | | |
| readCode | 读验证码 | 字符串 | 长度 1-30, 服务被设置 NONE 时忽略 |
| varNames | 变量名称包 | 字符串 | 最大长度 32K, 多个变量名间 ' ' 分隔 |
| varValues | 返回变量值包 | 字符串 | 最大长度 32K, 多个变量值间 ' ' 分隔 |
| 例 1: char* varValues=ReadVarValues("test", "AI. AI1 DI. DI1 VT. VT1") | | | |
| 例 2: //从. \WebPath\VarPackage1.txt 读取内容作为变量名包 | | | |
| char* varValues=ReadVarValues("test", "[varPackage1]") | | | |

- 设置运行数据库某批数值和文本变量值

| | | | |
|---|-------|-----|---------------------------|
| Int result=WriteVarValues(char* writeCode, char* varNames, char* varValues) | | | |
| writeCode | 写验证码 | 字符串 | 长度 1-30, 服务被设置 NONE 时忽略 |
| varNames | 变量名称包 | 字符串 | 最大长度 32K, 多个变量名间 ' ' 分隔 |
| varValues | 变量值包 | 字符串 | 最大长度 32K, 多个变量值间 ' ' 分隔 |
| 例 1: int result=WriteVarValues("test", "AR. AR1 DR. DR1 VT. VT1", "123. 45 1 ABCDabcd") | | | |
| 例 2: //从. \WebPath\VarPackage1.txt 读取内容作为变量名包 | | | |
| Int result=WriteVarValues("test", "[varPackage1]", "123. 45 1 ABCDabcd") | | | |

- 读取 WebPath 目录下的某文本文件内容

| | | | |
|--|------|-----|-------------------------|
| char* fileText=ReadTextFile(char* readCode,char* fileName) | | | |
| readCode | 读验证码 | 字符串 | 长度 1-30, 服务被设置 NONE 时忽略 |
| fileNames | 文件名称 | 字符串 | 文件类型 Txt, 无需后缀自动附加 |
| fileText | 文件内容 | 字符串 | 最大长度 1024K |
| 例: char* fileText=ReadFileText("test","file1") | | | |

- 写入 WebPath 目录下的某文本文件内容

| | | | |
|---|-------|-----|-------------------------|
| int result=WriteTextFile(char* writeCode,char* filename,char* fileText,int writeMode) | | | |
| writeCode | 写验证码 | 字符串 | 长度 1-30, 服务被设置 NONE 时忽略 |
| fileNames | 文件名称 | 字符串 | 文件类型 Txt, 无需后缀自动附加 |
| fileText | 文件内容 | 字符串 | 最大长度 1024K |
| writeMode | 写入方式 | 整型 | 0:新建, 1:追加 |
| result | 返回结果值 | 整型 | 0:成功 |
| 例: int result=ReadFileText("test","file1","123456abcdef",0) | | | |

- 获取最后执行功能错误码

| | | | |
|--------------------------------|--------|----|------|
| int errCode=GetLastErrorCode() | | | |
| errCode | 返回错误代码 | 整型 | 0:成功 |
| 例: int n=GetLastErrorCode() | | | |

- 持久连接方式, 访问服务完成, 通知服务器尽快关闭连接, 否则等待超时后关闭连接

| | | | |
|-------------------------|-------|----|------|
| int result=EndConnect() | | | |
| result | 返回结果值 | 整型 | 0:成功 |
| 例: int n=EndConnect() | | | |

- 例(VBScript):

```
Set soapClient = CreateObject("MSSOAP.SoapClient30")
```

```
If Err.Number = 0 Then
```

```
    soapClient.ClientProperty("ServerHTTPRequest") = True
```

```
    soapClient.mssoapinit "http://192.168.1.206:5080/WebService/?wsdl", "", "", ""
```

```
    x1=soapClient.GetVarValue("", 7, "%VA1")
```

```
    x2=soapClient.GetVarValue("", 7, "%VA2")
```

```
    y=x1+x2
```

```
    soapClient.SetVarValue "", 7, "%VA3", y
```

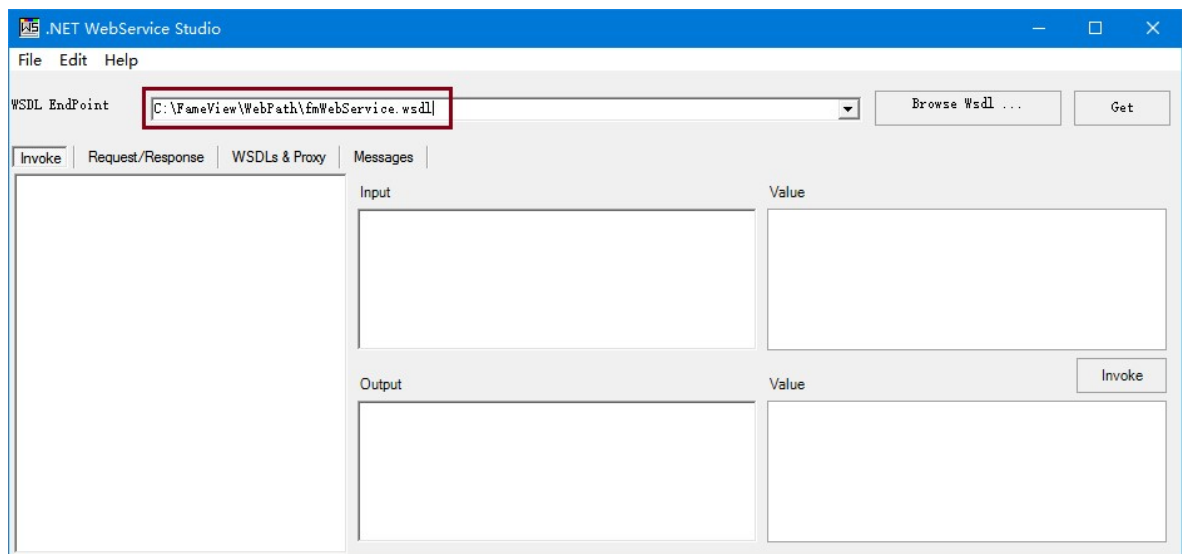
```
    soapClient. EndConnect
```

```
End If
```

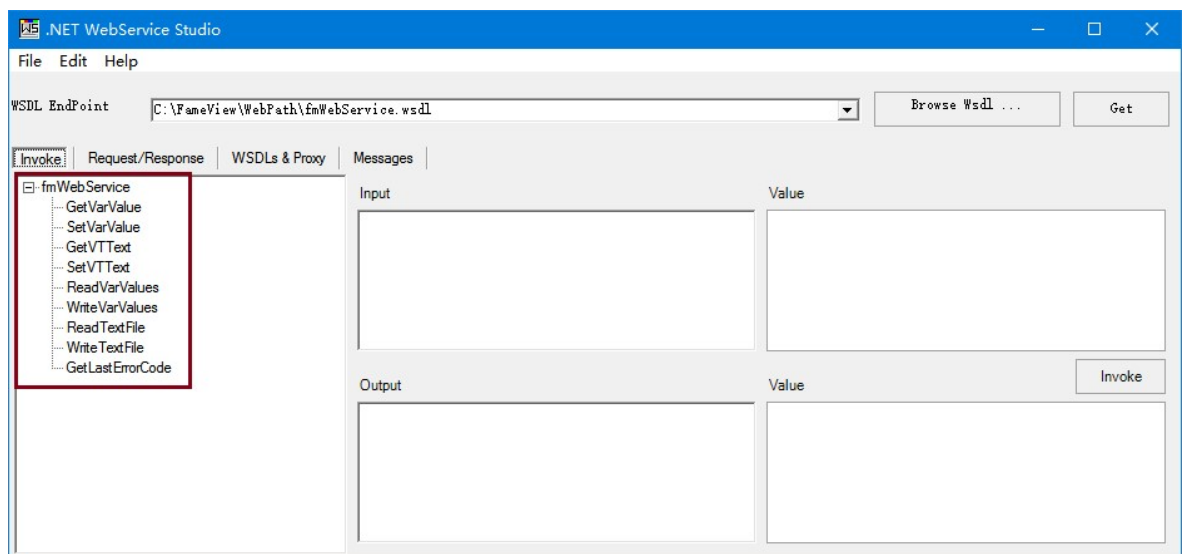
```
Set soapClient = Nothing
```

44.4 测试 WebService 接口

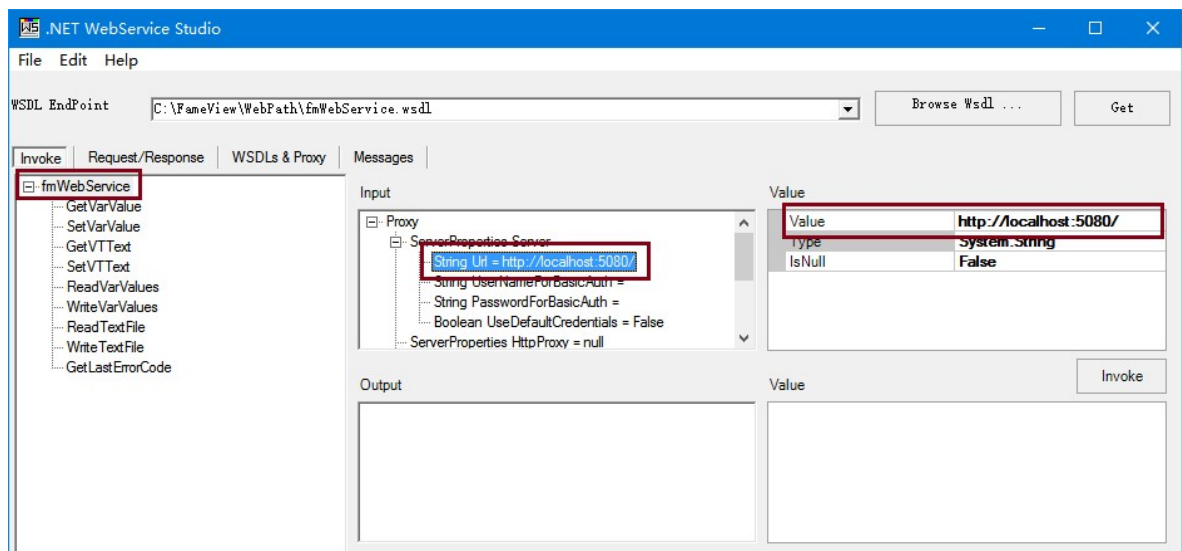
- 执行 WebPath\WebServiceStudio.exe 测试工具程序：



- 通过[Browse WSDL]按钮选择 WebPath\fmWebService.wsdl 文件；
- 执行[Get]按钮, 获取功能接口函数：

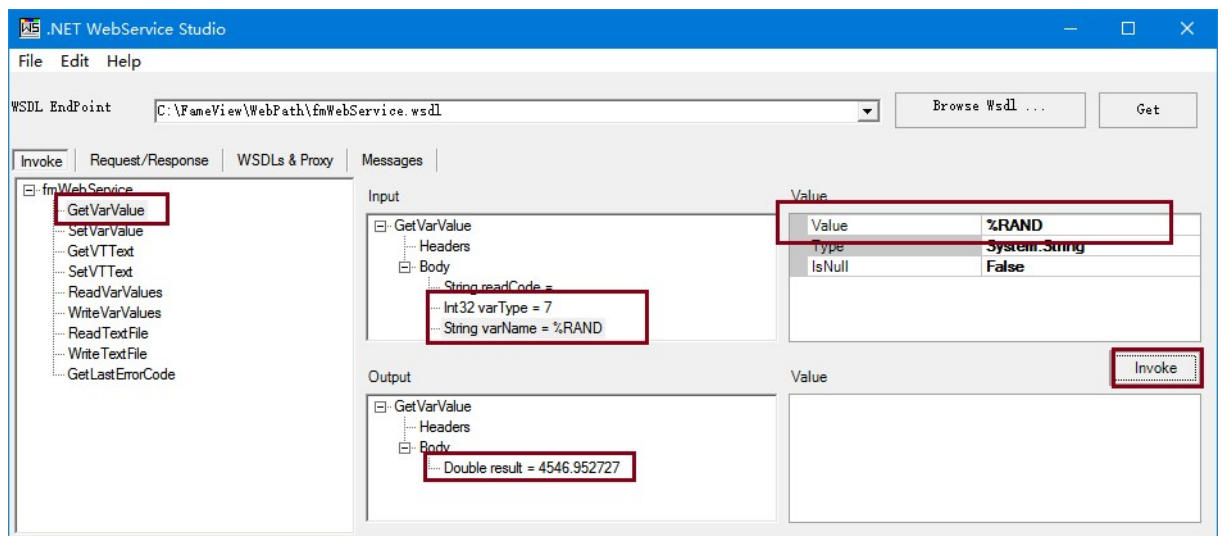


- 设置服务地址及端口, 例如:http://localhost:5080, http://192.168.1.100:5080

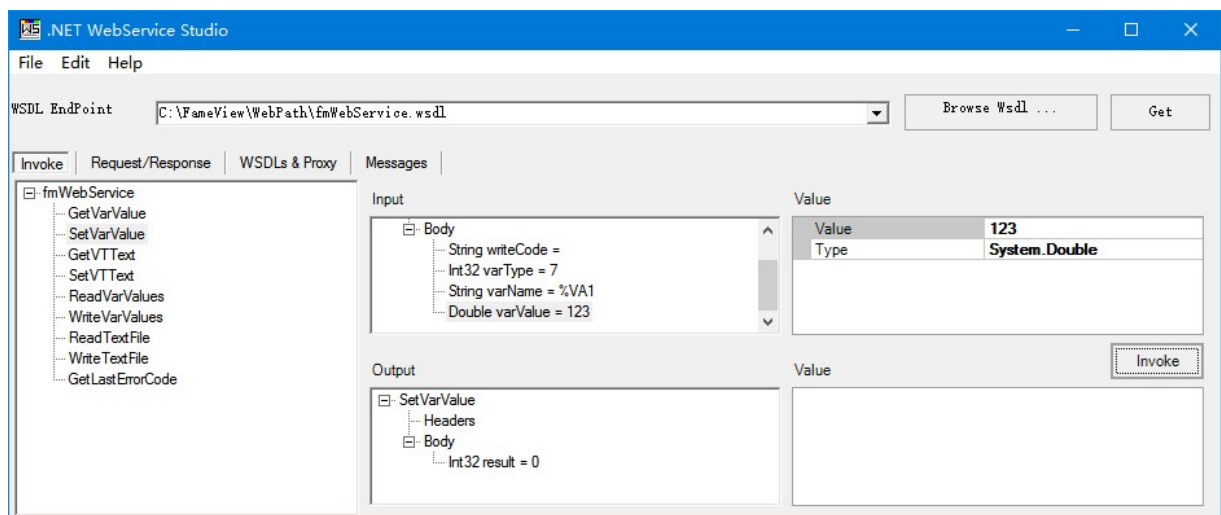


- 选择函数, 输入参数, 执行 [Invoke] 按钮执行函数, 并获取返回内容:

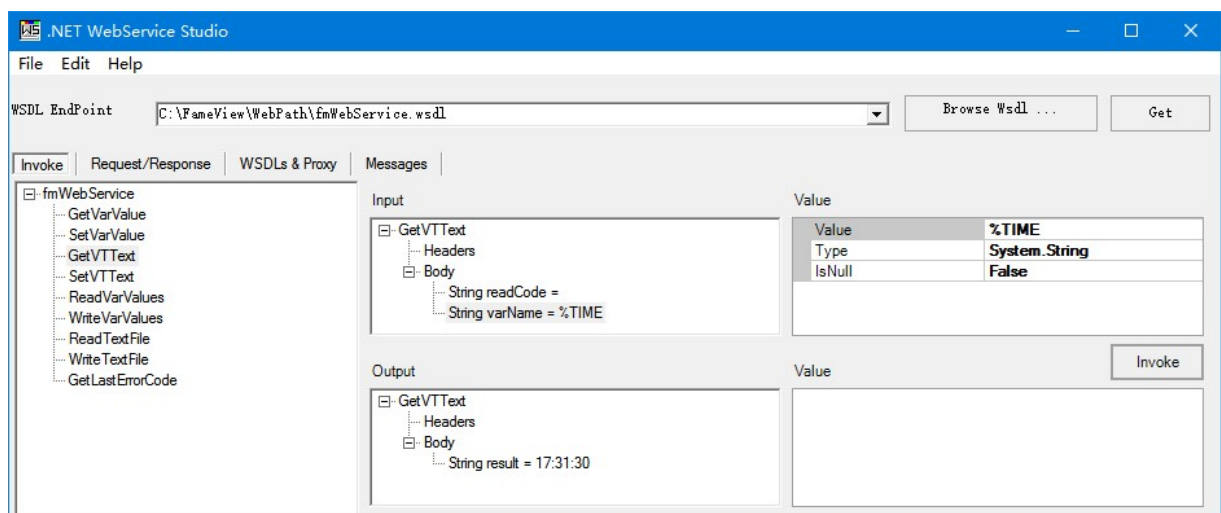
测试 GetVarValue:



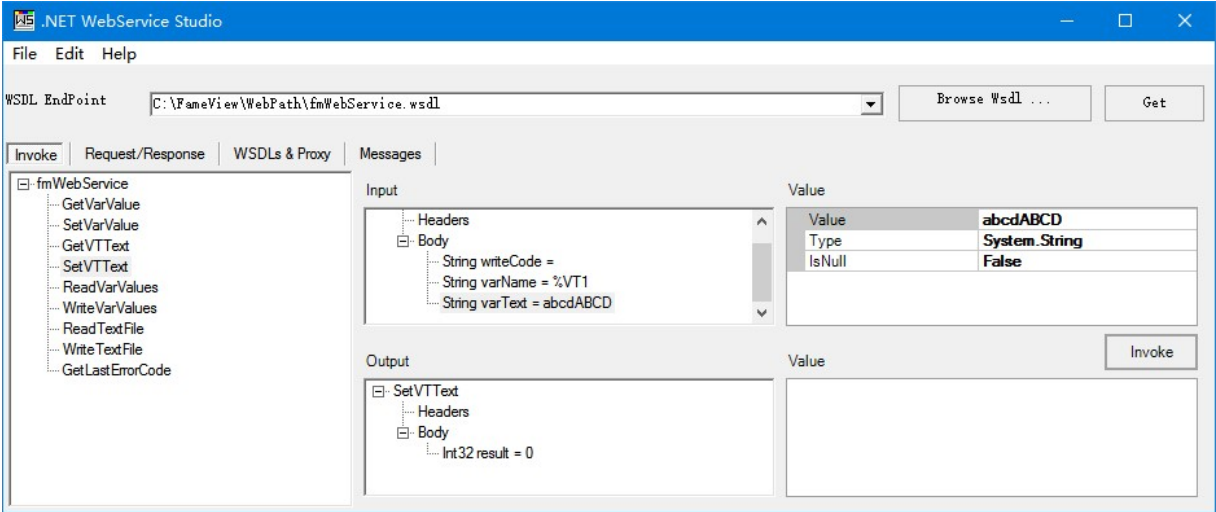
测试 SetVarValue:



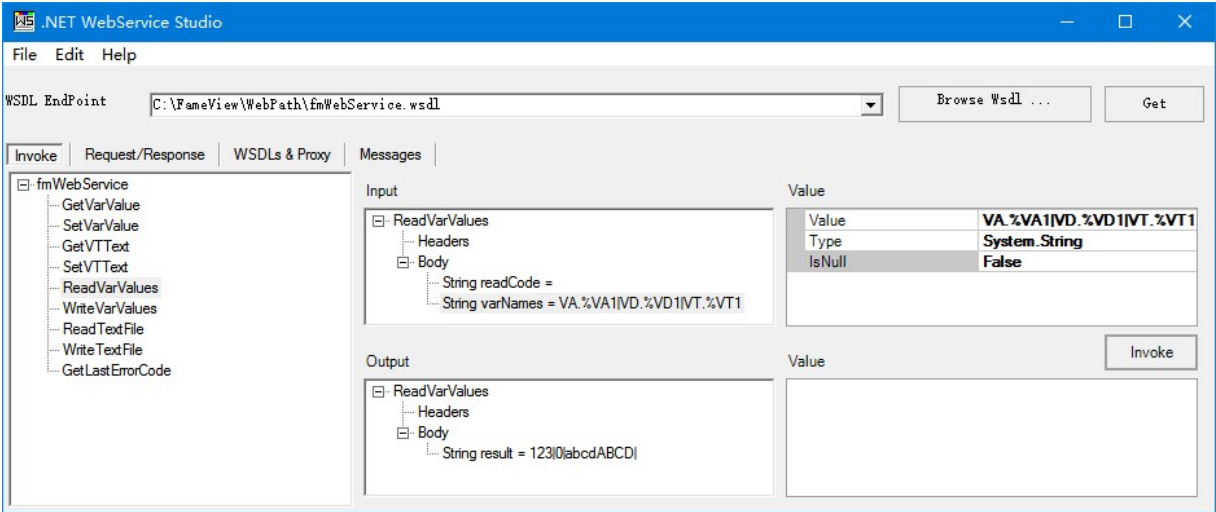
测试 GetVTText:



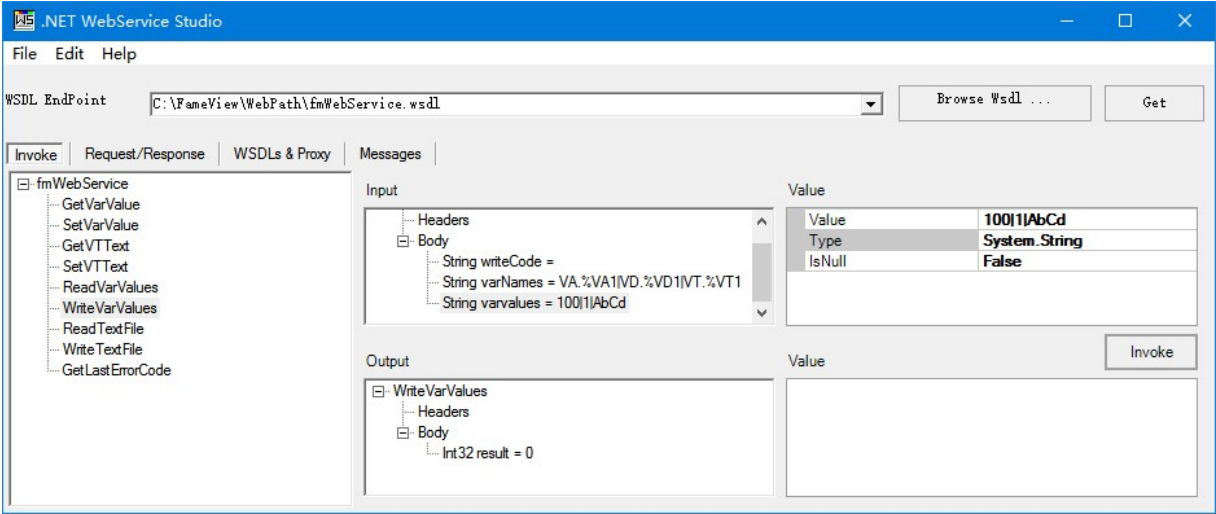
测试 SetVTText:



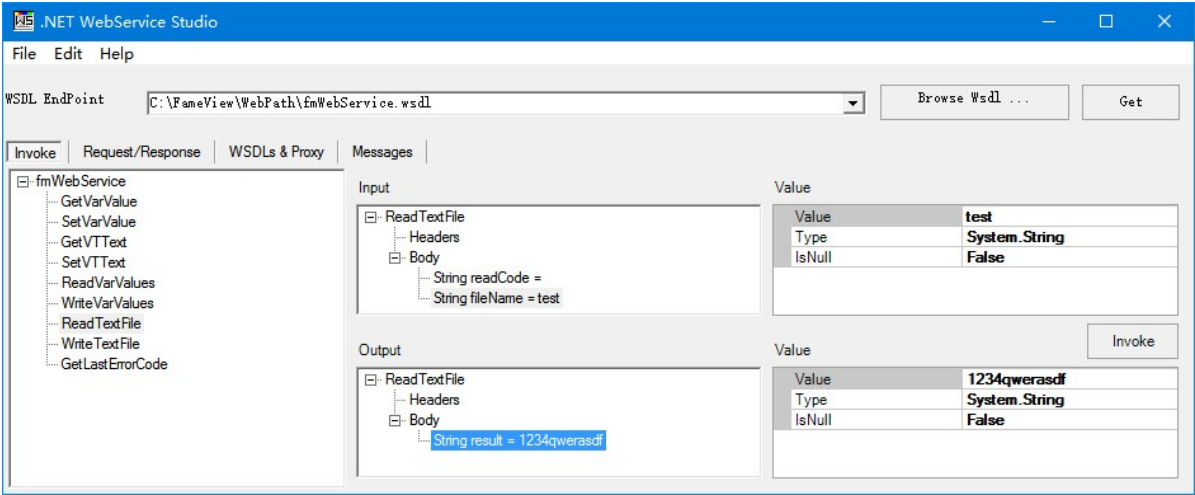
测试 ReadVarValues:



测试 WriteVarValues:



测试 ReadTextFile:



测试写 TextFile:

