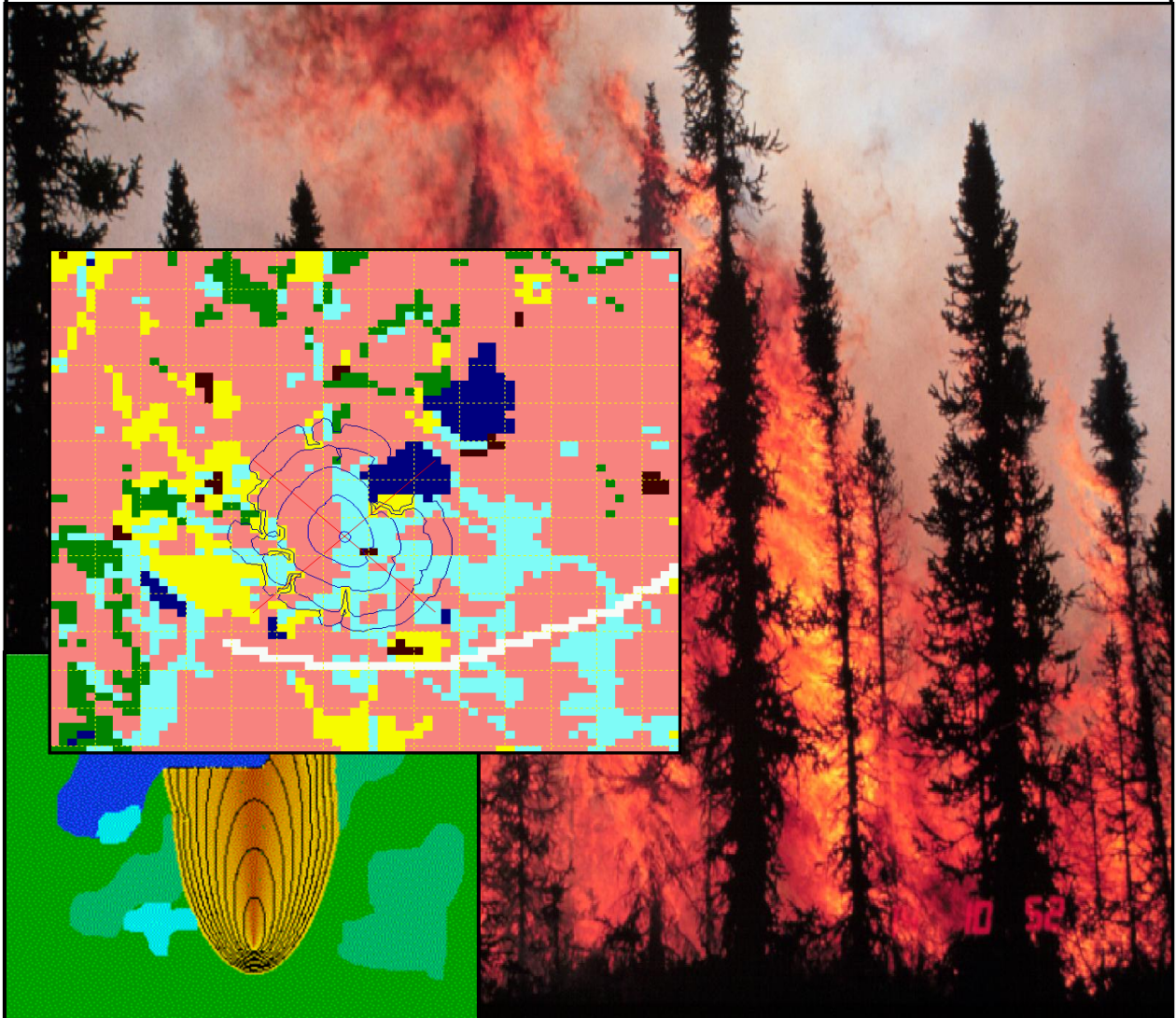




# *Prometheus* COM Programmer's Manual

CWFGM Project Steering Committee  
May 2009



## 1. Why do we need this COM?

COM (Common Object Model) is a standard method for modern Microsoft operating systems for sharing binary code across different applications and languages without sharing the actual source code. COM is a language independent object model that defines a binary interface standard. This standard specifies that the modules must be compiled using a specific structure, and that the COM object interfaces must be organized in a specific manner. The binary standard allows the DLLs to be used by most programming languages.

A component is a reusable piece of software in binary form that can, with relatively little effort, be linked into other components. The Prometheus COM can therefore be integrated into other applications such as Pandora, Burn-P3 or SFMS.

The important principles of COM include:

- Reuseability of the component objects
- Interoperability of the binary standard
- Allowance for distributed capabilities

Prometheus is engineered using 5 separate object-oriented COMs. These are called the low-level interfaces. They include:

- FuelCOM
- FWICom
- GridCom
- FireEngine
- WeatherCom

The Prometheus COM combines all of the low-level COMs to provide a high-level procedural interface that is easier for programmers to learn, understand and use. The COM is intended to be used by agencies interested in developing their own Fire Growth Model with a different user interface, or integrating the Prometheus Fire Growth Model into other applications. Additionally, the Microsoft *Interop* technology now allows modern .Net applications to use the high-level COM interface.

## 2. About the COM setup

The Prometheus application and COM setup are integrated into one install program. Both the Prometheus application and Prometheus COM are now automatically installed. Previously, users were prompted to enter a Prometheus COM license when installing the Prometheus COM; this is no longer the case.

If a very old version of Prometheus (or PrometheusCOM) is currently installed, it is recommended to first uninstall before loading a current version.

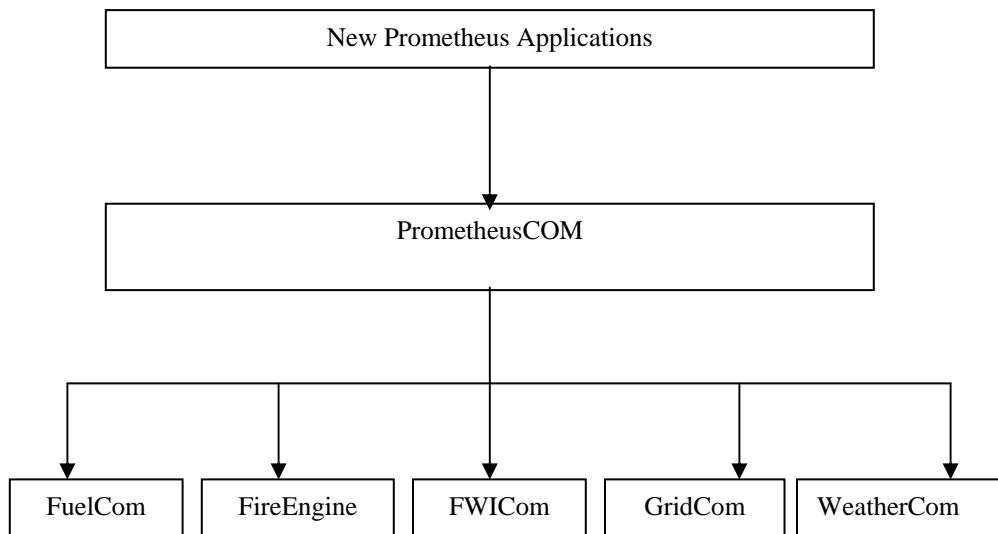
### 3. About the COM license

A license (software key) is no longer required to use the Prometheus COM. Any previously installed Prometheus COM license is stored in the computer registry in the following folder:

HKEY\_CURRENT\_USER/Software/Prometheus COM/Certificate key

This entry is no longer used and can be removed, if desired.

### 4. COM Architecture



Note: The arrows illustrate the dependencies.

This diagram shows the relationship of the Prometheus COM and the low-level COMs. The aim of the Prometheus COM is to present a set of simple interfaces that are easy to use. Details of the interfaces are included later in this document.

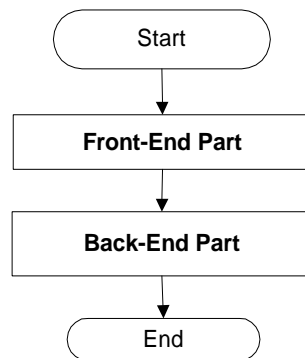
### 5. How does it work?

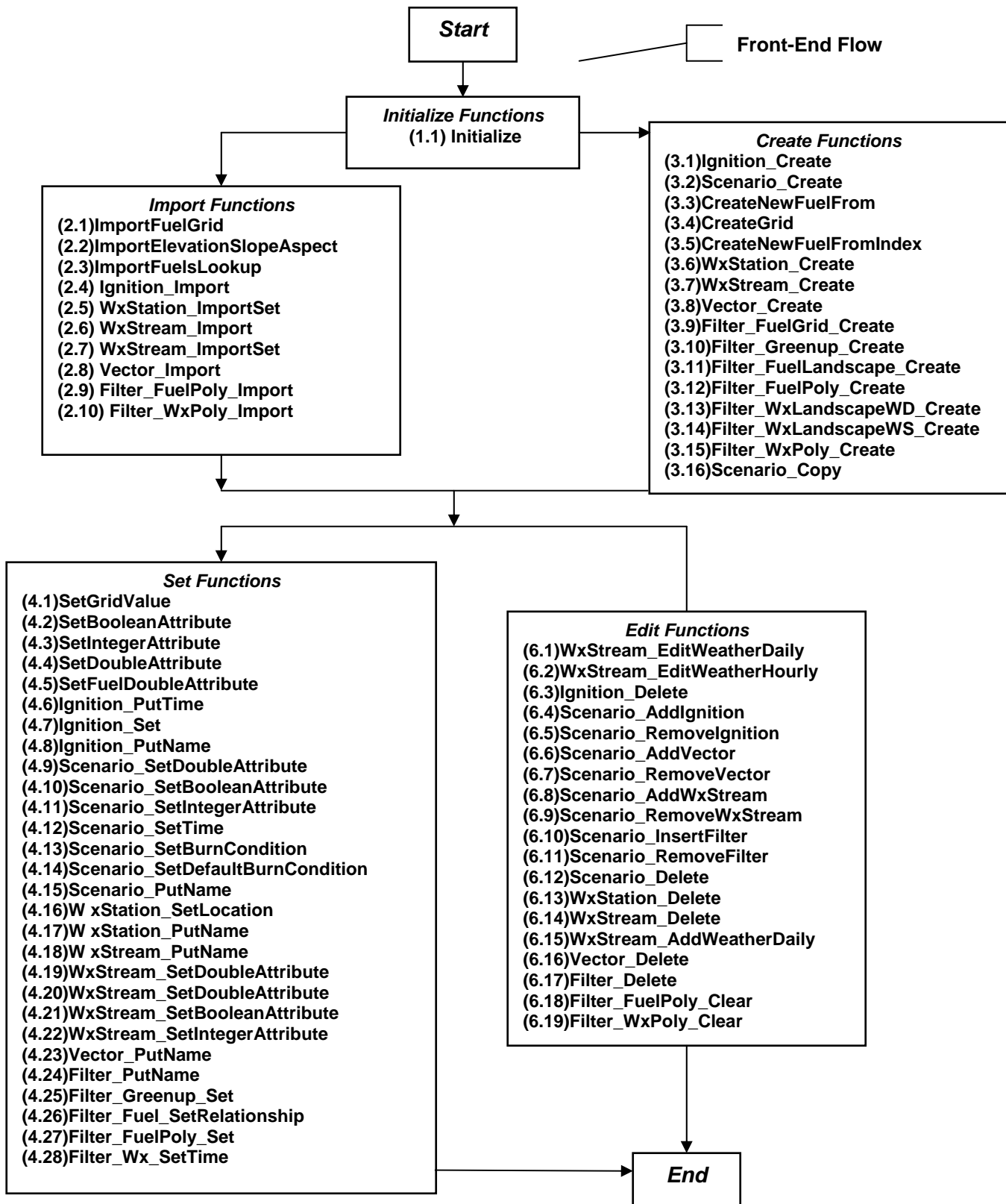
Prometheus is supported by the low-level COMs. When one of the methods in the Prometheus COM interface is called, the Prometheus COM will actually eventually perform one or many calls to the corresponding interface(s) of low-level COM. Prometheus COM also contains collection classes to manage the objects created by the low-level COM DLLs. Programmers therefore do not need to know any specific details about the low-level COMs. They only need to know how to make the Prometheus COM work properly.

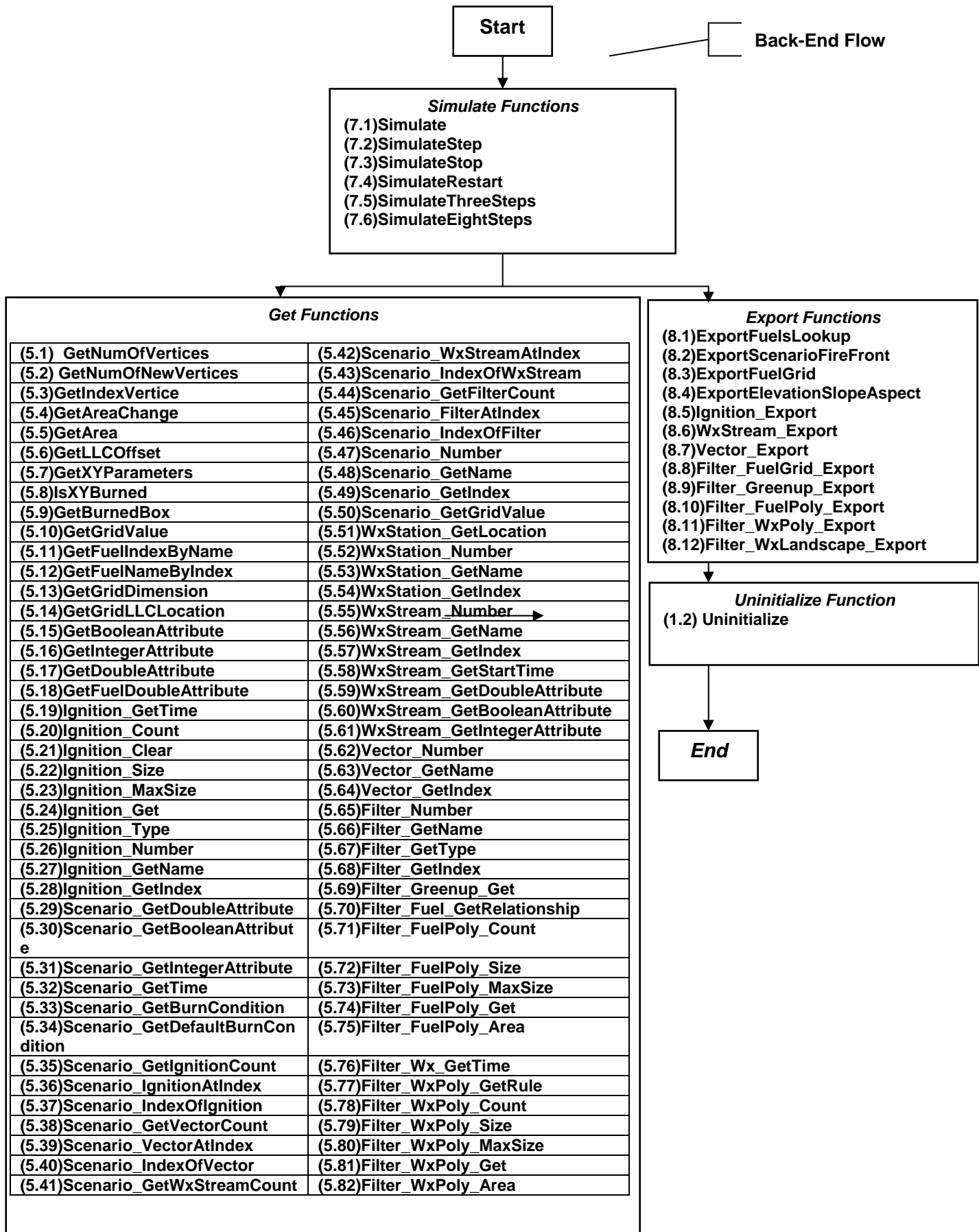
The Prometheus COM includes 8 categories of functions: Initialize Functions, Import Functions, Create Functions, Set Functions, Simulate Functions, Get Functions, Edit Functions and Export Functions.

- The Initialize Functions should be called before and after calling other interfaces.
- The Import Functions are used to import data from files. You can call this set of interfaces after calling the Initialize interface.
- The Create Functions are used to create entities such as weather stations, weather streams, simulation scenarios, and so on. They are useful when you don't have the necessary input data files, but would like to create the data in the program.
- The Set Functions are used to modify parameters of objects that already exist. For example, if you already have a fire and a scenario, and want to associate the fire with the scenario, you will call the corresponding set interface.
- The Simulate Functions are used to simulate the process of a fire growth in a scenario.
- The Get Functions are used to obtain some data from the COM or any of the objects it is managing.
- The Export Functions are used to export the simulation results to a data file.

The interfaces can usually be called in the sequence described in the following control flow diagram. The flow is divided into a Front-End part and a Back-End part.







## 6. Necessary information for programming

### 6.1 File Name

PrometheusCOM.dll

### 6.2 CLSID

```
const CLSID CLSID_PromeApp =  
{0xF985F261,0xCFA5,0x11D5,{0xB4,0xAC,0x00,0x02,0xA5,0x36,0x81,0x38}};
```

### 6.3 Name space available in C++ program

PROMETHEUSCOMLib

*Note:* The Name space is case sensitive. So when used in C++ (only useful in C++ environment), it should be exactly as above.

### 6.4 ProgramID

PrometheusCOM.PromeApp

*Note:* ProgramID is also case sensitive. Just like the name space above, it is useful only in the C++ environment, so please pay attention to the above strings and make sure they are typed correctly.

### 6.5 Users

Prometheus COM users are programmers interested in developing applications or completing fire growth model tests on Microsoft Windows/NT/2000 platform with Visual Basic, Visual C++, Delphi, C++ Builder, etc.

## 7. Duplicate functions no longer supported:

The functions listed below are replaced with corresponding Simulate... functions. They are no longer supported.

1. SimulateSetActiveScenario
2. SimulateActive
3. SimulateRestartActive
4. SimulateStepActive
5. SimulateThreeStepsActive
6. SimulateEightStepsActive

The functions listed below (i.e. 7 to 17) are replaced with GetIndexVertice and GetNumOfVertices functions. They are no longer supported.

7. GetFirstVertice
8. GetNextVertice
9. GetLastVertice
10. GetPreviousVertice
11. GetFirstVerticeActive
12. GetNextVerticeActive
13. GetPreviousVerticeActive
14. GetLastVerticeActive
15. GetNumOfVerticesActive
16. GetNumOfNewVerticesActive
17. GetIndexVerticeActive

The functions listed below (i.e. 18 to 20) are replaced by calling Ignition\_Create, Ignition\_PutTime and Ignition\_Set, respectively.

18. CreateIgnitionPoint
19. CreateIgnitionLine
20. CreateIgnitionPolygon

Funtion 21 has been replaced by Scenario\_Create.

21. CreateScenario

The functions listed below (i.e. 22 to 26) are replaced by calling either Scenario\_SetBooleanAttribute, or Scenario\_SetIntegerAttribute and then passing the correct parameters. Functions 27 and 28 can be replaced by calling Scenario\_SetBurnCondition and Scenario\_SetDefaultBurnCondition respectively.

22. SetScenarioFBPOptions
23. SetScenarioIntervals
24. SetScenarioBoundaryStopOption
25. SetScenarioAngleDistanceThreshold
26. SetScenarioTerrainEffective
27. SetScenarioBurnCondition
28. SetScenarioDefaultBurnCondition
29. SetLLCGridLocation

Functions 29-31 are replaced by using Ignition\_PutTime and Ignition\_Set.

30. EditIgnitionPoint
31. EditIgnitionLine
32. EditIgnitionPolygon

Functions 33-35 are replaced by Ignition\_Clear, Ignition\_Import and Ignition\_Export respectively

33. DeleteIgnition
34. ImportIgnition
35. ExportIgnition

Functions 36 – 95 are replaced with various get and set attribute functions of either Boolean, integer and double attributes. The other functions can be replaced by a function with a similar name. For example ImportWeatherStream() can be replaced by WxStream\_Import().

36. GetIndexStreamName
37. ImportProjection
38. SetScenarioWeatherGrid
39. SetFuelParam
40. SetScenarioGrid
41. ExportGenerate
42. SetDaylightSaving
43. GetResolution
44. SetInitialFFMC
45. SetInitialDMC
46. SetInitialDC
47. SetInitialTemp
48. SetInitialWS
49. SetStreamTempAlpha
50. SetStreamTempBeta
51. SetStreamTempGamma
52. SetStreamWindAlpha
53. SetStreamWindBeta
54. SetStreamWindGamma
55. SetStreamDaylightSaving
56. SetStreamFFMCMethod
57. ImportFuelPatch()
58. SetPatchSourceAndDest()
59. CreateLandscapeFuelPatch()
60. GetPatchSourceFuel()
61. GetPatchDestinationFuel()
62. GetPatchCurrentFuel()
63. ReplaceWithFuelPatch()
64. RestorePatchToOriginal()
65. ImportWeatherGrid()
66. SetWeatherGridTime()
67. SetGrassFuelParameters()
68. SetMixedFuelParameters()
69. SetInitialRain()
70. SetGridDefaultElevation()
71. SetGridMedianElevation()
72. ImportWeatherStations()
73. ImportWeatherStreams()

74. ImportVector()
75. ImportWeatherStream()
76. CreateFuelBreak()
77. CreateWeatherStation()
78. CreateWeatherStream()
79. SetScenarioFire()
80. SetScenarioStream()
81. SetScenarioTime()
82. SetScenarioFuelBreak()
83. GetNumOfWeatherStations()
84. GetIndexStationName()
85. GetNumOfWeatherStreams()
86. GetPatchArea()
87. AddDailyConditionToStream()
88. RemoveScenarioIgnition()
89. RemoveScenarioWeatherStream()
90. RemoveScenarioFuelBreak()
91. DeleteWeatherStream()
92. DeleteScenario()
93. DeleteVector()
94. ExportVector()
95. ExportWeatherStream()

In summary, all the functions related to active scenario are not be supported. The GetIndexVertice function provides all the required functionality to access vertices.

## 8. Usage

### 8.1 When Used in Visual Basic 6.0

#### 8.1.1 Simulate and export fire perimeter

```

Set ob = New PromeApp
ob.Initialize 51.7, -115.4, 3, 23
ob.ImportProjection txtPrjPath.Text
ob.ImportFuelGrid "Test", txtAscPath.Text
ob.CreateWeatherStation "First", 52, -115, 500
ob.CreateWeatherStream "First", "FirstStream", 85#, 15, 25#, 200#, "20/06/2000:00:00:00"
ob.SetInitialFFMC "First", "FirstStream", 85, 15
ob.SetStreamFFMCMethod "First", "FirstStream", 0
ob.AddDailyConditionToStream "First", "FirstStream", 15, 25, 15, 25, 20, 20, 90
ob.CreateScenario "FirstScenario"
ob.SetScenarioGrid "FirstScenario", "Test"
ob.SetScenarioTime "FirstScenario", "20/06/2000:10:00:00", "20/06/2000:20:00:00"
ob.CreateIgnitionPoint "FirstFire", "20/06/2000:9:00:00", 2485, 2900
ob.SetScenarioFire "FirstScenario", "FirstFire"
ob.SetScenarioStream "FirstScenario", "First", "FirstStream"
ob.SimulateRestart "FirstScenario"

```

```
Dim stepResult As Long
ob.SimulateStep "FirstScenario", stepResult
```

```
ob.ExportScenarioFireFront "FirstScenario", pathHead + "1.shp"
b.SimulateStep "FirstScenario", stepResult
ob.ExportScenarioFireFront "FirstScenario", pathHead + "2.shp"
ob.SimulateStep "FirstScenario", stepResult
ob.ExportScenarioFireFront "FirstScenario", pathHead + "3.shp"
ob.SimulateStop "FirstScenario"
ob.UnInitialize
```

### 8.1.2 Simulate with terrain and export fire perimeter

```
Set ob = New PromeApp
ob.Initialize 51.7, -115.4, 3, 23
ob.ImportProjection txtPrjPath.Text
ob.ImportFuelGrid "Test", txtAscPath.Text
ob.ImportElevationSlopeAspect "Test", txtElevationPath.Text, txtSlopePath.Text,
txtAspectPath.Text
ob.CreateWeatherStation "First", 52, -115, 1479
ob.ImportWeatherStream "First", "FirstStream", txtStreamPath
ob.SetInitialFFMC "First", "FirstStream", 94, 17
ob.SetStreamFFMCMethod "First", "FirstStream", 0
ob.SetInitialDC "First", "FirstStream", 482#
ob.SetInitialDMC "First", "FirstStream", 58#
```

```
ob.CreateScenario "FirstScenario"
ob.SetScenarioGrid "FirstScenario", "Test"
ob.SetScenarioTime "FirstScenario", "16/10/2001:13:00:00", "16/10/2001:22:00:00"
ob.CreateIgnitionPoint "FirstFire", "16/10/2001:13:00:00", 979#, 1168#
ob.SetScenarioFire "FirstScenario", "FirstFire"
ob.SetScenarioStream "FirstScenario", "First", "FirstStream"
ob.SimulateRestart "FirstScenario"
Dim stepResult As Long
ob.SimulateStep "FirstScenario", stepResult
```

```
ob.ExportScenarioFireFront "FirstScenario", pathHead + "1.shp"
ob.SimulateStep "FirstScenario", stepResult
ob.ExportScenarioFireFront "FirstScenario", pathHead + "2.shp"
ob.SimulateStep "FirstScenario", stepResult
ob.ExportScenarioFireFront "FirstScenario", pathHead + "3.shp"
ob.SimulateStop "FirstScenario"
ob.UnInitialize
```

### 8.1.3 Simulate and draw with internal coordinates

```
Set ob = New PromeApp
ob.Initialize 51.7, -115.4, 3, 23
ob.ImportProjection txtPrjPath.Text
ob.ImportFuelGrid "Test", txtAscPath.Text
ob.CreateWeatherStation "First", 52, -115, 500
```

```

ob.CreateWeatherStream "First", "FirstStream", 85#, 15, 25#, 200#, "20/06/2000:00:00:00"
ob.SetInitialFFMC "First", "FirstStream", 85, 15
ob.SetStreamFFMCMethod "First", "FirstStream", 0
ob.AddDailyConditionToStream "First", "FirstStream", 15, 25, 15, 25, 20, 20, 90
ob.CreateScenario "FirstScenario"
ob.SetScenarioGrid "FirstScenario", "Test"
ob.SetScenarioTime "FirstScenario", "20/06/2000:10:00:00", "20/06/2000:20:00:00"
ob.CreateIgnitionPoint "FirstFire", "20/06/2000:9:00:00", 2485.36, 2900
ob.SetScenarioFire "FirstScenario", "FirstFire"
ob.SetScenarioStream "FirstScenario", "First", "FirstStream"
ob.SimulateRestart "FirstScenario"
Dim stepResult As Long
Dim count As Long
Dim picWidth As Long
Dim picHeight As Long
Dim xDim As Long
Dim yDim As Long
Dim xCapacity As Double
Dim yCapacity As Double
Dim LastIndex As Long
picWidth = picResult.Width
picHeight = picResult.Height

resolution = ob.GetResolution()
ob.GetGridDimension "Test", xDim, yDim
xCapacity = xDim * resolution / picWidth
yCapacity = yDim * resolution / picHeight
LastIndex = 0
For i = 1 To 3
    Dim LastX As Long
    Dim LastY As Long
    Dim lX, lY As Long
    ob.SimulateStep "FirstScenario", stepResult
    count = ob.GetNumOfNewVertices("FirstScenario")
    For j = 0 To count - 1

        ob.GetIndexVertice "FirstScenario", LastIndex + j, X, Y
        If X <> -1 And Y <> -1 Then
            X = (X - 50) * 10
            Y = (Y - 50) * 10
            lX = X
            lY = Y
            If j = 0 Then
                LastX = X
                LastY = Y
            End If
            picResult.Line (LastX, LastY)-(lX, lY), RGB(255, 0, 0)
            LastX = X
            LastY = Y
        End If
    Next j

```

```
LastIndex = LastIndex + count  
Next i  
ob.SimulateStop "FirstScenario"  
ob.UnInitialize
```

### 8.1.4 Simulate with terrain and actual weather, draw with internal coordinates

```
Dim ob As PromeApp
Dim offX As Double
Dim offY As Double
Dim resolution As Double
  Dim X As Double
  Dim Y As Double

Set ob = New PromeApp
ob.Initialize 51.7, -115.4, 3, 23
ob.ImportProjection txtPrjPath.Text
ob.ImportFuelGrid "Test", txtAscPath.Text
ob.ImportElevationSlopeAspect "Test", txtElevationPath.Text, txtSlopePath.Text,
txtAspectPath.Text
ob.CreateWeatherStation "First", 52, -115, 1479
ob.ImportWeatherStream "First", "FirstStream", txtStreamPath
ob.SetInitialFFMC "First", "FirstStream", 94, 17
ob.SetStreamFFMCMethod "First", "FirstStream", 0
ob.SetInitialDC "First", "FirstStream", 482#
ob.SetInitialDMC "First", "FirstStream", 58#

ob.CreateScenario "FirstScenario"
ob.SetScenarioGrid "FirstScenario", "Test"
ob.SetScenarioTime "FirstScenario", "16/10/2001:13:00:00", "16/10/2001:22:00:00"
ob.CreateIgnitionPoint "FirstFire", "16/10/2001:13:00:00", 979#, 1168#
ob.SetScenarioFire "FirstScenario", "FirstFire"
ob.SetScenarioStream "FirstScenario", "First", "FirstStream"
ob.SimulateRestart "FirstScenario"
Dim stepResult As Long
Dim count As Long
Dim picWidth As Long
Dim picHeight As Long
Dim xDim As Long
Dim yDim As Long
Dim xCapacity As Double
Dim yCapacity As Double
Dim lastIndex As Long
picWidth = picResult.Width
picHeight = picResult.Height

resolution = ob.GetResolution()
ob.GetGridDimension "Test", xDim, yDim
xCapacity = xDim * resolution / picWidth
yCapacity = yDim * resolution / picHeight
lastIndex = 0
For i = 1 To 3
  Dim lastX As Long
  Dim lastY As Long
  Dim iX, iY As Long
  ob.SimulateStep "FirstScenario", stepResult
```

```

count = ob.GetNumOfNewVertices("FirstScenario")
For j = 0 To count - 1

    ob.GetIndexVertice "FirstScenario", LastIndex + j, X, Y
    If X <> -1 And Y <> -1 Then
        X = (X - 30) * 20
        Y = (Y - 30) * 20
        IX = X
        IY = Y
        IY = picHeight - IY
        If j = 0 Then
            LastX = IX
            LastY = IY
        End If
        picResult.Line (LastX, LastY)-(IX, IY), RGB(255, 0, 0)
        LastX = IX
        LastY = IY
    End If
Next j
LastIndex = LastIndex + count
Next i
ob.SimulateStop "FirstScenario"
ob.UnInitialize

```

## 8.2 When used in C/C++

One technique to use the COM DLL inside a C++ application is to import the COM dll file first, the statement could be like this:

```
#import "C:\\Program Files\\Prometheus\\PrometheusCOM.dll"
```

The path provided in the above statement should be replaced with the correct one for your system.

There must be the following declaration of CLSID;

```

const CLSID CLSID_PromeApp =
{0xF985F261,0xCFA5,0x11D5,{0xB4,0xAC,0x00,0x02,0xA5,0x36,0x81,0x38}};
.....

```

### 8.2.1 Simulate with manual weather stream, export fire perimeter

```

PROMETHEUSCOMLib::IPromeAppPtr pApp;
CoInitialize(NULL);
pApp.CreateInstance(CLSID_PromeApp);
HRESULT hr=pApp->Initialize(51.7, -115.4, 3, 23);
if(FAILED(hr))
    return;

```

```

CString csScenarioStart("20/06/2000:10:00:00"),
        csScenarioEnd("20/06/2000:20:00:00");
BSTR bScenarioStart, bScenarioEnd;
BSTR bPrjPath, bGridPath;
bScenarioStart=csScenarioStart.AllocSysString();
bScenarioEnd=csScenarioEnd.AllocSysString();
bPrjPath=m_csPrjPath.AllocSysString();
bGridPath=m_csGridPath.AllocSysString();
pApp->ImportProjection(bPrjPath);
pApp->ImportFuelGrid("Test",bGridPath);

pApp->CreateWeatherStation("First", 52, -115, 500);
pApp->CreateWeatherStream("First", "FirstStream", 85, 15, 25, 200,
"20/06/2000:00:00:00");
pApp->SetInitialFFMC( "First", "FirstStream", 85, 15);
pApp->SetStreamFFMCMethod( "First", "FirstStream", 0 );
pApp->AddDailyConditionToStream("First", "FirstStream", 15, 25, 15, 25, 20, 20, 90);

pApp->CreateScenario("FirstScenario");
pApp->SetScenarioGrid("FirstScenario", "Test");
pApp->SetScenarioTime("FirstScenario", bScenarioStart,bScenarioEnd);
pApp->CreateIgnitionPoint("FirstFire", "20/06/2000:9:00:00", 2485.36, 2900);
pApp->SetScenarioFire( "FirstScenario", "FirstFire");
pApp->SetScenarioStream("FirstScenario", "First", "FirstStream");

CString first,second, third;
first=pathHead+"1.shp";
second=pathHead+"2.shp";
third=pathHead+"3.shp";
BSTR bFirst,bSecond,bThird;
bFirst=first.AllocSysString();
bSecond=second.AllocSysString();
bThird=third.AllocSysString();

long stepResult;
stepResult=pApp->SimulateRestart( "FirstScenario" );
pApp->SimulateStep( "FirstScenario", &stepResult);

pApp->ExportScenarioFireFront( "FirstScenario", bFirst);
pApp->SimulateStep( "FirstScenario", &stepResult);
pApp->ExportScenarioFireFront( "FirstScenario", bSecond);
pApp->SimulateStep( "FirstScenario", &stepResult);
pApp->ExportScenarioFireFront( "FirstScenario", bThird);

MessageBox("Done");
pApp->UnInitialize();
pApp.Release();
CoUninitialize();

```

```

::SysFreeString(bFirst);
::SysFreeString(bSecond);
::SysFreeString(bThird);
SysFreeString( bScenarioStart);
SysFreeString( bScenarioEnd);
SysFreeString(bPrjPath);
SysFreeString(bGridPath);

```

## 8.2.2 Simulate with imported weather stream, export fire perimeter

```

CoInitialize(NULL);
PROMETHEUSCOMLib::IPromeAppPtr pApp;
pApp.CreateInstance(CLSID_PromeApp);
HRESULT hr=pApp->Initialize(51.7, -115.4, 3, 23);
if(FAILED(hr))
    return;
CString csScenarioStart("16/10/2001:13:00:00"),
        csScenarioEnd("16/10/2001:22:00:00");
BSTR bScenarioStart, bScenarioEnd;
BSTR bPrjPath, bGridPath;
bScenarioStart=csScenarioStart.AllocSysString();
bScenarioEnd=csScenarioEnd.AllocSysString();
bPrjPath=m_csPrjPath.AllocSysString();
bGridPath=m_csGridPath.AllocSysString();

BSTR bElevPath,bSlopePath,bAspPath, bWeatherPath;
bElevPath=m_csElevPath.AllocSysString();
bSlopePath=m_csSlopePath.AllocSysString();
bAspPath=m_csAspPath.AllocSysString();
bWeatherPath=m_csWeatherPath.AllocSysString();

pApp->ImportProjection(bPrjPath);
pApp->ImportFuelGrid("Test",bGridPath);
pApp->ImportElevationSlopeAspect("Test",bElevPath,bSlopePath,bAspPath);

pApp->CreateWeatherStation("First", 52, -115, 500);
pApp->ImportWeatherStream("First", "FirstStream", bWeatherPath);
pApp-> SetInitialFFMC( "First", "FirstStream", 94, 17);
pApp->SetStreamFFMCMethod( "First", "FirstStream", 0 );
pApp->SetInitialDC("First","FirstStream",482);
pApp->SetInitialDMC("First","FirstStream",58);

pApp->CreateScenario("FirstScenario");
pApp->SetScenarioGrid("FirstScenario","Test");
pApp->SetScenarioTime("FirstScenario", bScenarioStart,bScenarioEnd);
pApp->CreateIgnitionPoint("FirstFire", "16/10/2001:13:00:00", 979, 1168);
pApp->SetScenarioFire( "FirstScenario", "FirstFire");
pApp->SetScenarioStream("FirstScenario", "First", "FirstStream");

CString first,second, third;
first=pathHead+"1.shp";

```

```

second=pathHead+"2.shp";
third=pathHead+"3.shp";
BSTR bFirst,bSecond,bThird;
bFirst=first.AllocSysString();
bSecond=second.AllocSysString();
bThird=third.AllocSysString();

long stepResult;
stepResult=pApp->SimulateRestart( "FirstScenario" );
pApp->SimulateStep( "FirstScenario", &stepResult);

pApp->ExportScenarioFireFront( "FirstScenario", bFirst);
pApp->SimulateStep( "FirstScenario", &stepResult);
pApp->ExportScenarioFireFront( "FirstScenario", bSecond);
pApp->SimulateStep( "FirstScenario", &stepResult);
pApp->ExportScenarioFireFront( "FirstScenario", bThird);

MessageBox("Done");
pApp->UnInitialize();
pApp.Release();
CoUninitialize();

::SysFreeString(bFirst);
::SysFreeString(bSecond);
::SysFreeString(bThird);
SysFreeString( bScenarioStart);
SysFreeString( bScenarioEnd);
SysFreeString(bPrjPath);
SysFreeString(bGridPath);

```

### 8.2.3 Simulate with manual weather stream, draw with internal coordinates

```

m_arVertices.RemoveAll();
CoInitialize(NULL);
PROMETHEUSCOMLib::IPromeAppPtr pApp;
pApp.CreateInstance(CLSID_PromeApp);
HRESULT hr=pApp->Initialize(51.7, -115.4, 3, 23);
if(FAILED(hr))
    return;
CString csScenarioStart("20/06/2000:10:00:00"),
        csScenarioEnd("20/06/2000:20:00:00");
BSTR bScenarioStart, bScenarioEnd;
BSTR bPrjPath, bGridPath;
bScenarioStart=csScenarioStart.AllocSysString();
bScenarioEnd=csScenarioEnd.AllocSysString();
bPrjPath=m_csPrjPath.AllocSysString();
bGridPath=m_csGridPath.AllocSysString();
pApp->ImportProjection(bPrjPath);
pApp->ImportFuelGrid("Test",bGridPath);

```

```

pApp->CreateWeatherStation("First", 52, -115, 500);
pApp->CreateWeatherStream("First", "FirstStream", 85, 15, 25, 200,
"20/06/2000:00:00:00");
pApp-> SetInitialFFMC( "First", "FirstStream", 85, 15);
pApp->SetStreamFFMCMethod( "First", "FirstStream", 0 );
pApp->AddDailyConditionToStream("First", "FirstStream", 15, 25, 15, 25, 20, 20, 90);

pApp->CreateScenario("FirstScenario");
pApp->SetScenarioGrid("FirstScenario","Test");
pApp->SetScenarioTime("FirstScenario", bScenarioStart,bScenarioEnd);
pApp->CreateIgnitionPoint("FirstFire", "20/06/2000:9:00:00", 2485.36, 2900);
pApp->SetScenarioFire( "FirstScenario", "FirstFire");
pApp->SetScenarioStream("FirstScenario", "First", "FirstStream");

pApp->GetGridDimension("Test",&m_xDim,&m_yDim);
m_dblResolution=pApp->GetResolution();

long stepResult;
int LastIndex=0;
stepResult=pApp->SimulateRestart( "FirstScenario" );
for(int i=0;i<3;i++)
{
    pApp->SimulateStep( "FirstScenario", &stepResult);
    long NumPoints;
    NumPoints = pApp->GetNumOfNewVertices("FirstScenario");

    for(int j = 0;j<NumPoints; j++)
    {
        XY_Point temp;
        pApp->GetIndexVertice("FirstScenario", j + LastIndex, &temp.x,
&temp.y);
        double ros,fi,raz,sfc,cfb,cfc,tfc;
        pApp->GetXYParameters(
            "FirstScenario",
            temp.x,temp.y,&ros,
            &fi,&raz,&sfc,&cfb,
            &cfc,&tfc);
        m_arVertices.Add(temp);
    }
    LastIndex += NumPoints;
    Invalidate();
}

MessageBox("Done");
pApp->UnInitialize();
pApp.Release();
CoUninitialize();

```

```

SysFreeString( bScenarioStart);
SysFreeString( bScenarioEnd);
SysFreeString(bPrjPath);
SysFreeString(bGridPath);

```

## 8.2.4 Simulate with imported weather stream, draw with internal coordinates

```

m_arVertices.RemoveAll();
CoInitialize(NULL);
PROMETHEUSCOMLib::IPromeAppPtr pApp;
pApp.CreateInstance(CLSID_PromeApp);
HRESULT hr=pApp->Initialize(51.7, -115.4, 3, 23);
if(FAILED(hr))
    return;
CString csScenarioStart("16/10/2001:13:00:00"),
        csScenarioEnd("16/10/2001:22:00:00");
BSTR bScenarioStart, bScenarioEnd;
BSTR bPrjPath, bGridPath;
bScenarioStart=csScenarioStart.AllocSysString();
bScenarioEnd=csScenarioEnd.AllocSysString();
bPrjPath=m_csPrjPath.AllocSysString();
bGridPath=m_csGridPath.AllocSysString();

BSTR bElevPath,bSlopePath,bAspPath, bWeatherPath;
bElevPath=m_csElevPath.AllocSysString();
bSlopePath=m_csSlopePath.AllocSysString();
bAspPath=m_csAspPath.AllocSysString();
bWeatherPath=m_csWeatherPath.AllocSysString();

pApp->ImportProjection(bPrjPath);
pApp->ImportFuelGrid("Test",bGridPath);
pApp->ImportElevationSlopeAspect("Test",bElevPath,bSlopePath,bAspPath);

pApp->CreateWeatherStation("First", 52, -115, 500);
pApp->ImportWeatherStream("First", "FirstStream", bWeatherPath);
pApp-> SetInitialFFMC( "First", "FirstStream", 94, 17);
pApp->SetStreamFFMCMethod( "First", "FirstStream", 0 );
pApp->SetInitialDC("First","FirstStream",482);
pApp->SetInitialDMC("First","FirstStream",58);

pApp->CreateScenario("FirstScenario");
pApp->SetScenarioGrid("FirstScenario","Test");
pApp->SetScenarioTime("FirstScenario", bScenarioStart,bScenarioEnd);
pApp->CreateIgnitionPoint("FirstFire", "16/10/2001:13:00:00", 979, 1168);
pApp->SetScenarioFire( "FirstScenario", "FirstFire");
pApp->SetScenarioStream("FirstScenario", "First", "FirstStream");

pApp->GetGridDimension("Test",&m_xDim,&m_yDim);
m_dblResolution=pApp->GetResolution();

long stepResult;

```

```

int LastIndex=0;
stepResult=pApp->SimulateRestart( "FirstScenario" );
for(int i=0;i<3;i++)
{
    pApp->SimulateStep( "FirstScenario", &stepResult);
    long NumPoints;
    NumPoints = pApp->GetNumOfNewVertices("FirstScenario");

    for(int j = 0;j<NumPoints; j++)
    {
        XY_Point temp;
        pApp->GetIndexVertice("FirstScenario", j + LastIndex, &temp.x,
        &temp.y);
        m_arVertices.Add(temp);
    }
    LastIndex += NumPoints;
    Invalidate();
}

MessageBox("Done");
pApp->UnInitialize();
pApp.Release();
CoUninitialize();

SysFreeString( bScenarioStart);
SysFreeString( bScenarioEnd);
SysFreeString(bPrjPath);
SysFreeString(bGridPath);
SysFreeString(bSlopePath);
SysFreeString(bAspPath);

```

### 8.3 Alternate method in C++

```

CoInitialize(NULL);
PROMETHEUSCOMLib::IPromeAppPtr pApp;
_GUID guidPromeApp;
CString str;
BSTR bst;
str="PrometheusCOM.PromeApp";
bst=str.AllocSysString();
::CLSIDFromProgID(bst,&guidPromeApp);
pApp.CreateInstance(guidPromeApp);
CString csScenarioStart("20/06/2000:10:00:00"),
HRESULT hr=pApp->Initialize(51.7, -115.4, 3, 23);
if(FAILED(hr))
    return;
CString csScenarioStart("20/06/2000:10:00:00"),
        csScenarioEnd("20/06/2000:20:00:00");
BSTR bScenarioStart, bScenarioEnd;
BSTR bPrjPath, bGridPath;

```

```

bScenarioStart=csScenarioStart.AllocSysString();
bScenarioEnd=csScenarioEnd.AllocSysString();
bPrjPath=m_csPrjPath.AllocSysString();
bGridPath=m_csGridPath.AllocSysString();
pApp->ImportProjection(bPrjPath);
pApp->ImportFuelGrid("Test",bGridPath);

pApp->CreateWeatherStation("First", 52, -115, 500);
pApp->CreateWeatherStream("First", "FirstStream", 85, 15, 25, 200,
"20/06/2000:00:00:00");
pApp-> SetInitialFFMC( "First", "FirstStream", 85, 15);
pApp->SetStreamFFMCMethod( "First", "FirstStream", 0 );
pApp->AddDailyConditionToStream("First", "FirstStream", 15, 25, 15, 25, 20, 20, 90);

pApp->CreateScenario("FirstScenario");
pApp->SetScenarioGrid("FirstScenario","Test");
pApp->SetScenarioTime("FirstScenario", bScenarioStart,bScenarioEnd);
pApp->CreateIgnitionPoint("FirstFire", "20/06/2000:9:00:00", 2485.36, 2900);
pApp->SetScenarioFire( "FirstScenario", "FirstFire");
pApp->SetScenarioStream("FirstScenario", "First", "FirstStream");

CString first,second, third;
first=pathHead+"1.shp";
second=pathHead+"2.shp";
third=pathHead+"3.shp";
BSTR bFirst,bSecond,bThird;
bFirst=first.AllocSysString();
bSecond=second.AllocSysString();
bThird=third.AllocSysString();

long stepResult;
stepResult=pApp->SimulateRestart( "FirstScenario" );
pApp->SimulateStep( "FirstScenario", &stepResult);

pApp->ExportScenarioFireFront( "FirstScenario", bFirst);
pApp->SimulateStep( "FirstScenario", &stepResult);
pApp->ExportScenarioFireFront( "FirstScenario", bSecond);
pApp->SimulateStep( "FirstScenario", &stepResult);
pApp->ExportScenarioFireFront( "FirstScenario", bThird);

MessageBox("Done");
pApp->UnInitialize();
pApp.Release();
CoUninitialize();

::SysFreeString(bFirst);
::SysFreeString(bSecond);
::SysFreeString(bThird);
SysFreeString( bScenarioStart);

```

```

SysFreeString( bScenarioEnd);
SysFreeString(bPrjPath);
SysFreeString(bGridPath);

```

## 8.4 Access COM interface in C

To access the COM interfaces in C, you have to access the virtual function table directly, and not via the stub functions provided in the DLL. This technique can also be done in C++ and will provide two advantages: a slight increase in performance, and no need to surround calls to PrometheusCOM with a C++ “try” block, described later.

There is a pointer accessing the virtual function table in the COM which is named "lpVtbl". By accessing the virtual function table directly, and then accessing the function you want to call, you can call any of the interfaces presented by the COM. All return values (HRESULT's) should be inspected after any call to this interface.

```

IPromeApp* g_pDP = NULL;
...
CoInitialize( NULL );
...
hr = CoCreateInstance(CLSID_PromeApp, NULL, CLSCTX_INPROC_SERVER,
                    IID_IPromeApp, (LPVOID*) &g_pDP );
...
g_pDP->lpVtbl->Initialize( 52,-115,3,23);
...
g_pDP->lpVtbl->ImportProjection("E:\\DataSet\\projections.prj ")
...
g_pDP->lpVtbl->UnInitialize();

```

### Comments:

1. Calling the Create Functions can be substituted by using the Import Functions. In other words, if you don't want to create these in your program, you can import them from files.
2. When using the stub functions (i.e., using the #import keyword), note that an error condition will throw a C++ exception. The above example source codes, for purposes of completeness, should place all calls to PrometheusCOM inside a C++ “try” block, as follows:

```

HRESULT stepResult;
try {
    stepResult = pApp->SimulateRestart( "Scenario1" );
} catch (_com_error &ce) {
    CString str1 = ce.Description();
    HRESULT err = ce.Error();

```

```
        stepResult = err;  
        CString str2 = ce.ErrorMessage();  
    }
```

Failing to include any call to PrometheusCOM, via the stub functions, inside a similar “try” block, will cause the application to fail upon any error condition.

3. Scenario settings cannot be modified when a simulation is in progress. Some settings can (currently) be modified immediately after a simulation has been reset, but this behaviour may change in the future. This rule is imposed to ensure that a simulation output can be reproduced.
4. All date/time values are specified in the same manner. The standard COM date/time structure is *not* used. Rather, any date/time is specified in a character string which is parsed. The format for the date/time string is specifically “DD/MM/YYYY:HH:MM:SS”, so the client application may have to perform translations from any locale-specific (or other) formatting.
5. All ‘Get’ functions are described using their default template. If one is using a language such as Visual Basic, or using the proxy/stub functions which are available in C++, it is entirely possible (likely) that the usage of the function will change to hide the HRESULT return value and instead return the value which is passed as a VAR parameter.

## 8.4 Access COM interface in .Net Languages: C#, Visual Basic

This approach requires the use of Microsoft’s *Interop* technology. Please contact support for specific instructions for this approach.

## Appendix I: Prometheus COM Interface

---

### The high-level interfaces:

When programming, the import and set interfaces should be called before calling the simulate interface. After (or sometimes during) simulating, call the get and export interfaces to obtain the final results.

*Note: Any error response you encounter that is not listed in this document (for a given function) should be considered as an internal error condition, and reported to the PrometheusCOM development team.*

### Sections / Prefaces:

- *Intialize~*      *Set up the project.*
- *Import~*        *Get data from file.*
- *Create~*        *Create a new object.*
- *Set~*            *Set the relationships between some objects.*
- *Get~*            *Get some data in memory. Usually coordinates.*
- *Edit~*           *Change the parameters of some objects that have already existed.*
- *Simulate~*     *Run the Model*
- *Export~*        *Output some of the data in program(COM) to a file.*
- *Active~*        *Operations for the active scenario.*

## (1) INITIALIZE Functions

### (1.1) *Initialize()*

**Description:** Initialize the PrometheusCOM DLL for one application use. Various initial parameters and structures are prepared for usage. This function must be called before any other functions.

The rules imposed by the COM architecture allow you to instantiate and initialize many PrometheusCOM objects simultaneously. However, the current design is limited to at most one PrometheusCOM object per application. This is a known design limitation.

Previously, this method required parameters to specify (e.g.) latitude, longitude. Now, this data is retrieved from the provided projection files during an import operation.

**Returns:**

S_OK	Successful.
ERROR_INVALID_ACCESS   ERROR_SEVERITY_WARNING	Internal error to be reported to COM development team.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
TRUE	Operation succeeded.
REGDB_E_CLASSNOTREG	Internal error to be reported to COM development team.
CLASS_E_NOAGGREGATION	Internal error to be reported to COM development team.
E_NOINTERFACE	Internal error to be reported to COM development team.
ERROR_SCENARIO_SIMULATION_RUNNING	Value cannot be changed as it is being used in a currently running scenario.
E_INVALIDARG	Invalid parameters.
E_FAIL	A general failure occurred.

**Updates:** This function no longer takes any parameters. The longitude and latitude will be imported with the ASCII file and the time zone is set with a different function. Use the SetIntegerAttribute function and the CWFGM\_GRID\_ATTRIBUTE\_TIMEZONE key.

### (1.2) *UnInitialize*

**Description:** Performs necessary clean-up when simulations are finished. De-allocates all related objects, etc. This function ends the current COM session. No further COM functions can be called unless this function is re-initialized.

This method should be used with care. Its functionality is redundant to simply deleting the PrometheusCOM object.

**Returns:**

S_OK	Successful.
------	-------------

## (2) IMPORT Functions

### (2.1) *ImportFuelGrid(BSTR projFileName, BSTR fileName)*

**Description:** Imports a fuel grid and its associated projection file.

Currently, all grids use the same projection and LUT information, and most of PrometheusCOM now supports the concept of multiple grids. However, at this time, this function does not require a name for this default initial grid.

**Parameters:**

<code>projFileName</code>	String identifying the name and path to the projection file.
<code>fileName</code>	Character string identifying the name and path to the ASCII Grid file.

**Returns:**

<code>S_OK</code>	Successful.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete the operation.
<code>ERROR_GRID_UNINITIALIZED</code>	The grid is not initialized properly.
<code>ERROR_FILE_NOT_FOUND</code>	The file cannot be found in the filesystem.
<code>ERROR_ACCESS_DENIED</code>	The file cannot be opened.
<code>ERROR_READ_FAULT</code>   <code>ERROR_SEVERITY_WARNING</code>	The file format is unrecognized, or if there is an error in the file format.
<code>ERROR_GRID_LOCATION_OUT_OF_RANGE</code>	If the import files do not appear to match data that has already been imported.
<code>ERROR_FUELS_FUEL_UNKNOWN</code>	The grid file contains a fuel index that is not defined in the fuel look-up table.
<code>E_POINTER</code>	Address provided is invalid.
<code>E_INVALIDARG</code>	Invalid parameters.
<code>ERROR_SCENARIO_SIMLUATION_RUNNING</code>	Value cannot be changed while running a simulation.
<code>E_FAIL</code>	A general failure occurred.
<code>ERROR_TOO_MANY_OPEN_FILES</code>	System error; current application has too many files open.
<code>SUCCESS_GRID_DATA_UPDATED</code>	The grid data was successfully updated.
<code>ERROR_INVALID_HANDLE</code>	The handle has been invalidated, or closed.
<code>ERROR_FILE_EXISTS</code>	The file exists.
<code>ERROR_INVALID_PARAMETER</code>	The parameter is incorrect.
<code>ERROR_HANDLE_DISK_FULL</code>	The disk is full.
<code>ERROR_SUCCESS</code>	Success.

**(2.2) *ImportElevationSlopeAspect(BSTR elevationFile, BSTR slopeFile, BSTR aspectFile)***

**Description:** Import the elevation, slope, and aspect data.

Before calling this function, you should call the ImportFuelGrid() function first.

None of the parameters should contain an empty string; all elevation data must be provided.

Currently, all grids use the same projection and LUT information. Most of PrometheusCOM does not support the concept of multiple grids.

**Parameters:**

elevationFile	Character string identifying the name and path to the ASCII elevation file name.
slopeFile	Character string identifying the name and path to the ASCII slope grid file name.
aspectFile	Character string identifying the name and path to the ASCII aspect grid file name.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_GRID_UNINITIALIZED	The grid is not initialized properly.
ERROR_FILE_NOT_FOUND	One of the files cannot be found in the filesystem.
ERROR_ACCESS_DENIED	One of the files cannot be opened.
ERROR_READ_FAULT   ERROR_SEVERITY_WARNING	The file format is unrecognized, or if there is an error in the file format.
ERROR_GRID_LOCATION_OUT_OF_RANGE	The import files do not appear to match data that has already been imported.
E_POINTER	Address provided is invalid.
E_INVALIDARG	Invalid parameters.
ERROR_SCENARIO_SIMULATION_RUNNING	Value cannot be changed as it is being used in a currently running scenario.
SUCCESS_GRID_IMPORT_CONTAINED_NODATA	Successful operation, but the elevation grid contained NODATA entries.
SUCCESS_GRID_DATA_UPDATED	The original elevation grid map has been successfully replaced.
ERROR_TOO_MANY_OPEN_FILES	System error; current application has too many files open.
E_FAIL	A valid geographic projection transformation could not be created.
ERROR_SUCCESS	Success.
ERROR_FILE_EXISTS	The file exists.
ERROR_INVALID_PARAMETER	The parameter is incorrect.

<a href="#">ERROR_FILE_NOT_FOUND</a>	The specified file cannot be found.
<a href="#">ERROR_HANDLE_DISK_FULL</a>	The disk is full.

### (2.3) *ImportFuelsLookup(BSTR fileName)*

**Description:** Input the Fuels Lookup table. This function is optional but should be called before `ImportFuelGrid()`. If no lookup table is specified, then the default Prometheus table is used.

Please refer to the Prometheus User Manual and Prometheus I/O Standard Document for this file format specification.

**Parameters:**

<a href="#">fileName</a>	Character string identifying the name and path to the fuels lookup file.
--------------------------	--

**Returns:**

<a href="#">S_OK</a>	Successful.
<a href="#">ERROR_SEVERITY_WARNING</a>	Failed initialization or a non-specific I/O error.
<a href="#">E_OUTOFMEMORY</a>	There is not enough memory to complete the operation.
<a href="#">ERROR_FILE_FORMAT_INVALID</a>	If the format of the file does not conform to expected specifications.
<a href="#">ERROR_GRID_INITIALIZED</a>	The assignment of the same fuel map is tried more than once.
<a href="#">ERROR_FILE_NOT_FOUND</a>   <a href="#">ERROR_SEVERITY_WARNING</a>	The specified file cannot be found.
<a href="#">E_FAIL</a>	A general failure occurred.

### (2.4) *Ignition\_Import(BSTR IgnitionName, BSTR filename)*

**Description:** Import an ignition from the given file path and filename. A variety of file formats are now supported. Please refer to the Prometheus User Manual and Prometheus I/O Standard Document for this file format specification.

**Parameters:**

<a href="#">IgnitionName</a>	Unique identifier for the new ignition.
<a href="#">filename</a>	Character string identifying the name and path to a file containing the ignition data.

**Returns:**

<a href="#">S_OK</a>	Successful.
<a href="#">ERROR_SEVERITY_WARNING</a>	Failed initialization or a non-specific I/O error.
<a href="#">E_OUTOFMEMORY</a>	There is not enough memory to complete the operation.
<a href="#">ERROR_SCENARIO_FIRE_UNKNOWN</a>	The ignition does not exist.
<a href="#">E_POINTER</a>	Address provided is invalid.
<a href="#">ERROR_FILE_NOT_FOUND</a>	The specified file cannot be found.

<a href="#">ERROR_TOO_MANY_OPEN_FILES</a>	System error; current application has too many files open.
<a href="#">ERROR_ACCESS_DENIED</a>	Access denied.
<a href="#">ERROR_INVALID_HANDLE</a>	The handle has been invalidated, or closed.
<a href="#">ERROR_FILE_EXISTS</a>	The file exists.
<a href="#">ERROR_HANDLE_DISK_FULL</a>	The disk is full.
<a href="#">E_FAIL</a>	A general failure occurred.

## (2.5) *WxStation\_ImportSet(BSTR FileName)*

**Description:** Import weather stations.  
 BSTR is a binary (basic) string type.  
 Whether in Visual Basic or in C++ (using stub functions), you can just input a constant string as the actual parameter, such as "c:\Test.dat" (or "c:\\Test.dat"). This applies to all functions that use the BSTR.

In the weather station file, there may be several weather stations. Each line of the file should include the following information:

- ID: Weather Station ID
- Latitude
- Longitude
- Elevation
- Weather Station Type (I / P, where I is interpolated and P is permanent)

Additional information is available in the Prometheus user manual.

### Parameters:

<a href="#">FileName</a>	Character string identifying the name and path to a file containing information regarding weather stations (ID, location, elevation, etc.)
--------------------------	--

### Returns:

<a href="#">S_OK</a>	Successful.
<a href="#">ERROR_SEVERITY_WARNING</a>	Failed initialization or a non-specific I/O error.
<a href="#">E_OUTOFMEMORY</a>	There is not enough memory to complete the operation.
<a href="#">ERROR_ACCESS_DENIED</a>	Access denied.
<a href="#">ERROR_INVALID_HANDLE</a>	The handle has been invalidated, or closed.
<a href="#">ERROR_FILE_EXISTS</a>	The file exists.
<a href="#">ERROR_INVALID_PARAMETER</a>	The parameter is incorrect.
<a href="#">ERROR_TOO_MANY_OPEN_FILES</a>	System error; current application has too many files open.
<a href="#">ERROR_FILE_NOT_FOUND</a>	The specified file could not be found.
<a href="#">ERROR_HANDLE_DISK_FULL</a>	The disk is full.
<a href="#">ERROR_FILE_FORMAT_INVALID</a>	The file format is not valid.

## (2.6) *WxStream\_Import*(*BSTR WeatherStationID*, *BSTR WeatherStreamID*, *BSTR FileName*)

**Description:** Attempts to import a weather stream for the specified weather station from the specified file.

**Parameters:**

<code>FileName</code>	The file name and path of the file where the weather stream data is.
<code>WeatherStationID</code>	Unique identifier for a weather station.
<code>WeatherStreamID</code>	Unique identifier for a weather stream.

**Returns:**

<code>S_OK</code>	Successful.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete the operation.
<code>ERROR_WEATHER_STATION_UNKNOWN</code>	There is no weather station with this id.
<code>ERROR_WEATHER_STREAM_ALREADY_ADDED</code>	There is already a weather stream with this id.

## (2.7) *WxStream\_ImportSet*(*BSTR FileName*)

**Description:** Import weather streams.

By default, the weather stream uses the Van Wagner technique for calculating FFMC, and uses the Canadian default values for the diurnal equations by Beck and Trevitt, 1989.

The weather stream file should contain all of the required information. Details are provided in the Prometheus user manual. The following information should also be added:

- ◆ `WeatherStationID`: the format is `W1#K5#`
- ◆ `WeatherStreamID`: this field will keep the name of each separate weather stream.
- ◆ `WeatherStationTYPE`:
  - `IA`( Interpolated, Actual)
  - `PA`(Permanent Actual)
  - `PF`(Forecast)
  - `IF`(Interpolated, Forecast)
- ◆ When the type is with `I` flag, there should be at least two station ID in the ID field.
- ◆ Example: `ImportWeatherStreams("C:\WeatherStream.dat")`
- ◆ The file format should be:

```
StationID      xxxx
StationType   xxxx
StreamID      xxxx
stream data part
end
stationID     xxxx
```

```

stationtype xxxx
streamID xxxx
.....
end

```

**Parameters:**

FileName	Character string identifying the name and path to a file containing information regarding this weather stream (ID, location, elevation, etc.)
----------	---

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_ACCESS_DENIED	Access denied.
ERROR_INVALID_HANDLE	The handle has been invalidated, or closed.
ERROR_FILE_EXISTS	The file exists.
ERROR_INVALID_PARAMETERS	The parameter is incorrect.
ERROR_TOO_MANY_OPEN_FILES	System error; current application has too many files open.
ERROR_FILE_NOT_FOUND	The specified file could not be found.
ERROR_HANDLE_DISK_FULL	The disk is full.
ERROR_FILE_FORMAT_INVALID	The file format is not valid.
E_FAIL	A general failure occurred.

**(2.8) Vector\_Import(BSTR filename, BSTR VectorName)**

**Description:** This function attempts to import a fuel break vector from a file.

**Parameters:**

FileName	The file name and path of the file where the vector data is.
VectorName	Unique identifier for the vector.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_VECTORENGI NE_KNOWN	The vector data object does not exist.
E_POINTER	Address provided is invalid.
ERROR_GRID_UNINITIALIZED	The grid is not initialized properly.
E_FAIL	A general failure occurred.
S_FALSE	False.

E_INVALIDARG	Invalid parameters.
--------------	---------------------

**(2.9) *Filter\_FuelPoly\_Import(BSTR FilterName, BSTR fileName)***

**Description:** This function attempts to import a fuel polygon patch from a file.

**Parameters:**

FileName	The file name and path of the file where the vector data is.
FilterName	Unique identifier for the fuel polygon patch.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_GRID_FILTER_UNKNOWN	There is no grid filter with that name.
E_POINTER	Address provided is invalid.
ERROR_GRID_UNINITIALIZED	The grid is not initialized properly.
S_FALSE	False.
E_INVALIDARG	Invalid parameters.
ERROR_FILE_NOT_FOUND	The specified file could not be found.
ERROR_TOO_MANY_OPEN_FILES	System error; current application has too many files open.
ERROR_ACCESS_DENIED	Access denied.
ERROR_INVALID_HANDLE	The handle has been invalidated, or closed.
ERROR_FILE_EXISTS	The file exists.
ERROR_HANDLE_DISK_FULL	The disk is full.
E_FAIL	A general failure occurred.

**(2.10) *Filter\_WxPoly\_Import(BSTR FilterName, BSTR fileName)***

**Description:** This function attempts to import a weather polygon patch from a file.

**Parameters:**

FileName	The file name and path of the file where the vector data is.
FilterName	Unique identifier for the weather polygon patch.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete

	the operation.
ERROR_GRID_FILTER_UNKNOWN	There is no grid filter with that name.
E_POINTER	Address provided is invalid.
ERROR_GRID_UNINITIALIZED	The grid is not initialized properly.
S_FALSE	False.
E_INVALIDARG	Invalid parameters.
ERROR_FILE_NOT_FOUND	The specified file could not be found.
ERROR_TOO_MANY_OPEN_FILES	System error; current application has too many files open.
ERROR_ACCESS_DENIED	Access denied.
ERROR_INVALID_HANDLE	The handle has been invalidated, or closed.
ERROR_FILE_EXISTS	The file exists.
ERROR_HANDLE_DISK_FULL	The disk is full.
E_FAIL	A general failure occurred.

### (3) CREATE Functions

#### (3.1) *Ignition\_Create(BSTR IgnitionName)*

**Description:** Creates a new, empty ignition object, given a name. Note that the name should be unique among existing fire ignitions.

**Parameters:**

IgnitionName	Unique identifier for the new ignition point.
--------------	---

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_FIRE_ALREADY_ADDED	An ignition with this name has already been added.

#### (3.2) *Scenario\_Create(BSTR ScenarioName)*

**Description:** Creates a new, empty scenario object. A scenario is an object which associates together all spatial and temporal data required for a simulation.

This scenario is assigned the default FBP and scenario options. These can be modified by calling Scenario\_SetBooleanAttribute, Scenario\_SetIntegerAttribute and passing the correct parameters.

This function requires further modification for verification of input parameters, such as uniqueness of the ignition name.

**Parameters:**

ScenarioName	Unique identifier for the new scenario.
--------------	---

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_GRID_UNINITIALIZED	The grid object is not initialized.

**(3.3) CreateNewFuelFrom(BSTR OldFuelName, BSTR NewFuelName, int \*fuel\_index)**

**Description:** Creates a new fuel type from an old fuectype. The COM first tries to locate a match via the FBP fuel name (from the LUT file), then the agency name, then will finally try to find a first “close match” to copy from. The new fuel name is assigned as the new fuels FBP name. The new fuel is assigned the same import and export indices.

This function requires modification, because it does not verify that the provided name for the new fuel is unique.

**Parameters:**

OldFuelName	Name of the fuel to copy from.
NewFuelName	Unique FBP name of the newly created fuel.
fuel_index	The index of the new fuel if the function is successful, otherwise the value will be set to 0.

**Returns:**

S_OK	Successful.
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_NOINTERFACE	Internal error. Please report this to the PrometheusCOM development team.
ERROR_FUELS_FILEINDEX_ALREADY_ASSIGNED	The fuel to add has assigned a non-unique file identifier that is not -1.
SUCCESS_FUELS_FUEL_ALREADY_ADDED	The fuel to be added is already in the FuelMap; a particular fuel can occur in a FuelMap more than once.
ERROR_FUELS_FUEL_UNKNOWN	There is no fuel with that name.

**(3.4) CreateGrid(int xsize, int ysize, double latitude, double longitude, double xllcorner, double yllcorner, double resolution, int fuel\_index)**

**Description:** Creates a uniform xsize by ysize grid with xllcorner and yllcorner as the lower-left coordinate in world coordinate system.

Currently all grids use the same projection and LUT information, and most of PrometheusCOM does not support the concept of multiple grids. Also, this function does not verify that the provided name for the new grid is unique (if multiple grids are to be continued to be supported).

**Parameters:**

xsize	X dimension of the grid.
ysize	Y dimension of the grid.
latitude	The latitude the grid is based at.
longitude	The longitude the grid is based at.
xllcorner	X-coordinate of the lower-left corner of the grid.
yllcorner	Y-coordinate of the lower-left corner of the grid.
resolution	Resolution of the grid (measured as number of metres per grid unit).
fuel_index	Default fuel for the grid -- it corresponds to a fuel index value used in the lookup table (if users are using the standard lookup table, the fuel_index should be the standard value for the corresponding fuel).

**Returns:**

S_OK	Successful
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_NOINTERFACE	Internal error. Please report this to the PrometheusCOM development team.
ERROR_GRID_INITIALIZED	Attempting to set the same fuel map object twice.
ERROR_GRID_UNINITIALIZED	The grid object is not initialized
E_INVALIDARG	The latitude/longitude used to set the world location are invalid, or if the arguments used to create the grid (e.g. xllcorner) are invalid.
ERROR_FUELS_FUEL_UNKNOWN	The fuel referenced by fuel_index is unknown.
ERROR_SCENARIO_SIMULATION_RUNNING	Value cannot be changed as it is being used by a currently running scenario.
TRUE	Operation succeeded.

**(3.5) CreateNewFuelFromIndex(int source\_index, BSTR NewFuelName, int \*NewFuelIndex)**

**Description:** This function is similar to the CreateNewFuelFrom function. The difference is this function requires an index to specify the source fuel to create new fuel from.

If successfully executed, the fuel\_index will contain the index for the new fuel, otherwise, fuel\_index should be 0.

**Parameters:**

source_index	Index of the source fuel.
NewFuelName	Unique FBP name of the newly created fuel.
NewFuelIndex	The index of the new fuel if the function is successful, otherwise the value will be set to 0.

**Returns:**

S_OK	Successful.
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_FUELS_FUEL_UNKNOWN	The fuel referenced by fuel_index is unknown.

### (3.6) *WxStation\_Create(BSTR WeatherStationID)*

**Description:** This method attempts to create a new weather station object with the id given it.

**Parameters:**

WeatherStationID	Unique identifier for the new weather station.
------------------	--

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
ERROR_WEATHER_STATION_ALREADY_PRESENT	There is already a weather station with this identifier.
E_OUTOFMEMORY	There is not enough memory to complete the operation.

### (3.7) *WxStream\_Create(BSTR WeatherStationID, BSTR WeatherStreamID, BSTR StartTime)*

**Description:** This method attempts to create a new weather stream object with the given name, station and start time.

**Parameters:**

WeatherStationID	Unique identifier for the weather station.
WeatherStreamID	Unique identifier for a weather stream.
StartTime	Start time for the weather stream.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
ERROR_WEATHER_STATION_ALR EADY_PRESENT	There is already a weather station with this identifier.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_WEATHER_STATION_UNK NOWN	The weather station specified can not be found.
E_FAIL	A general failure occurred.
E_INVALIDARG	Invalid parameters.

### (3.8) *Vector\_Create(BSTR FuelBreakName, BSTR xString, BSTR yString)*

**Description:** This method creates a new vector fuel break identified by name and formed by the strings of x and y values.

**Parameters:**

FuelBreakName	Unique identifier for the vector fuelbreak.
xString	String of x coordinates.
yString	String of y coordinates.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
ERROR_SCENARIO_VECTORENGI NE_KNOWN	A vector object with this name already exists.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_INVALIDARG	Invalid parameters.

### (3.9) *Filter\_FuelGrid\_Create(BSTR FilterName, BSTR FileName)*

**Description:** This method creates a new fuel grid filter.

**Parameters:**

FilterName	Unique identifier for the filter.
FileName	Name of the file.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.

### (3.10) *Filter\_Greenup\_Create(BSTR FilterName, BSTR FileName)*

**Description:** This method creates a new green-up grid filter.

**Parameters:**

FilterName	Unique identifier for the filter.
FileName	Name of the file.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.

**(3.11) Filter\_FuelLandscape\_Create(BSTR FilterName)**

**Description:** This method creates a new fuel landscape grid filter.

**Parameters:**

FilterName	Unique identifier for the filter.
------------	-----------------------------------

**Returns:**

S_OK	Successful.
E_OUTOFMEMORY	There is not enough memory to complete the operation.

**(3.12) Filter\_FuelPoly\_Create(BSTR FilterName)**

**Description:** This method creates a new fuel polygon grid filter.

**Parameters:**

FilterName	Unique identifier for the filter.
------------	-----------------------------------

**Returns:**

S_OK	Successful.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
ERROR_GRIDFILTER_UNKNOWN	The grid filter does not exist.

**(3.13) Filter\_WxLandscapeWD\_Create(BSTR FilterName, BSTR FileName)**

**Description:** This method creates a new directional weather grid from a file.

**Parameters:**

FilterName	Unique identifier for the filter.
FileName	Path and filename of the data to be imported.

**Returns:**

S_OK	Successful.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.

### (3.14) *Filter\_WxLandscapeWS\_Create(BSTR FilterName, BSTR FileName)*

**Description:** This method creates a new weather speed grid from a file.

**Parameters:**

<code>FilterName</code>	Unique identifier for the filter.
<code>FileName</code>	Path and filename of the data to be imported.

**Returns:**

<code>S_OK</code>	Successful.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete the operation.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.

### (3.15) *Filter\_WxPoly\_Create(BSTR FilterName)*

**Description:** This method creates a new polygon weather patch.

**Parameters:**

<code>FilterName</code>	Unique identifier for the filter.
-------------------------	-----------------------------------

**Returns:**

<code>S_OK</code>	Successful.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete the operation.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.

### (3.16) *Scenario\_Copy(BSTR fromScenarioName, BSTR newScenarioName)*

**Description:** This method copies an already existing scenario.

**Parameters:**

<code>fromScenarioName</code>	Unique identifier of the scenario to be copied.
<code>newScenarioName</code>	Unique identifier of the new scenario created from the old one.

**Returns:**

<code>S_OK</code>	Successful.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete the operation.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.
<code>ERROR_SCENARIO_UNKOWN</code>	No scenario exists with that name.

## (4) SET Functions

### (4.1) *SetGridValue(int X, int Y, int fuel\_index)*

**Description:** Sets a specific cell in a fuel grid to a fuel index value.

This function allows you to modify the originally imported data, which is typically assumed to be unmodifiable. There are other, safer means available to override grid contents

**Parameters:**

X	X-coordinate of the cell in question.
Y	Y-coordinate of the cell in question.
fuel_index	Value in the fuel lookup table.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_GRID_UNINITIALIZED	The grid object is not properly initialized.
ERROR_GRID_LOCATION_OUT_OF_RANGE	The x/y coordinate is out of range.
ERROR_FUELS_FUEL_UNKNOWN	Fuel_index does not resolve to any known fuel.
ERROR_SCENARIO_SIMULATION_RUNNING	Value cannot be changed as it is being used in a currently running scenario.
S_OK	successful

### (4.2) *SetBooleanAttribute(int attribute\_key, int value)*

**Description:** Set the Boolean value for the specified attribute.

**Parameters:**

attribute_key	Valid key is CWFGM_GRID_ATTRIBUTE_DAYLIGHT_SAVINGS
value	The Boolean value to be used.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_INVALIDARG	The elevation parameter has an invalid value.

#### (4.3) *SetIntegerAttribute(int attribute\_key, int value)*

**Description:** Set the integer value for the specified attribute.

**Parameters:**

<code>attribute_key</code>	Valid key is CWFGM_GRID_ATTRIBUTE_TIMEZONE
<code>value</code>	The integer value to be used.

**Returns:**

<code>S_OK</code>	Successful.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete the operation.
<code>ERROR_SCENARIO_BAD_STATE</code>	The scenario is in a bad state.
<code>E_INVALIDARG</code>	Invalid parameters.
<code>ERROR_SCENARIO_BAD_TIMES</code>	The scenario has invalid times.
<code>ERROR_SCENARIO_BAD_TIMES_TEPS</code>	The scenario has an invalid time step.
<code>ERROR_SCENARIO_OPTION_INVALID</code>	Unrecognized key code.

#### (4.4) *SetDoubleAttribute(int attribute\_key, double value)*

**Description:** Set the double value for the specified attribute.

**Parameters:**

<code>attribute_key</code>	Valid key is CWFGM_GRID_ATTRIBUTE_DEFAULT_ELEVATION
<code>value</code>	The double value to be used.

**Returns:**

<code>S_OK</code>	Successful.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete the operation.
<code>ERROR_SCENARIO_BAD_STATE</code>	The scenario is in a bad state.
<code>E_INVALIDARG</code>	Invalid parameters.
<code>ERROR_SCENARIO_BAD_TIMES</code>	The scenario has invalid times.
<code>ERROR_SCENARIO_BAD_TIMES_TEPS</code>	The scenario has an invalid time step.
<code>ERROR_SCENARIO_OPTION_INVALID</code>	Unrecognized key code.

#### (4.5) *SetFuelDoubleAttribute(int, fuel\_index, int attribute\_key, double value)*

**Description:** Set the double value for the specified fuel attribute.  
**Parameters:**

<code>fuel_index</code>	The index number of the fuel to set the attribute for.
<code>attribute_key</code>	Valid keys are: FUELCOM_ATTRIBUTE_A FUELCOM_ATTRIBUTE_B FUELCOM_ATTRIBUTE_C FUELCOM_ATTRIBUTE_Q FUELCOM_ATTRIBUTE_MAXBE FUELCOM_ATTRIBUTE_BUI0 FUELCOM_ATTRIBUTE_CBH FUELCOM_ATTRIBUTE_CFL FUELCOM_ATTRIBUTE_PC FUELCOM_ATTRIBUTE_PDF FUELCOM_ATTRIBUTE_CURINGDEGREE FUELCOM_ATTRIBUTE_M3M4_C2_A FUELCOM_ATTRIBUTE_M3M4_C2_B FUELCOM_ATTRIBUTE_M3M4_C2_C FUELCOM_ATTRIBUTE_M4_D1_A FUELCOM_ATTRIBUTE_M4_D1_B FUELCOM_ATTRIBUTE_M4_D1_C FUELCOM_ATTRIBUTE_EQ35_THRESHO LD FUELCOM_ATTRIBUTE_EQ35A_MULT1 FUELCOM_ATTRIBUTE_EQ35A_MULT2 FUELCOM_ATTRIBUTE_EQ35B_ADDER FUELCOM_ATTRIBUTE_EQ35B_MULT1
<code>value</code>	The double value to be used.

**Returns:**

<code>S_OK</code>	Successful.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete the operation.
<code>ERROR_SCENARIO_BAD_STATE</code>	The scenario is in a bad state.
<code>E_INVALIDARG</code>	Invalid parameters.
<code>ERROR_SCENARIO_BAD_TIMES</code>	The scenario has invalid times.
<code>ERROR_SCENARIO_BAD_TIMES TEPS</code>	The scenario has an invalid time step.
<code>ERROR_SCENARIO_OPTION_IN VALID</code>	Unrecognized key code.
<code>E_NOINTERFACE</code>	No interface.
<code>E_POINTER</code>	Address provided is invalid.
<code>ERROR_FUELS_FUEL_UNKOWN</code>	There is no fuel with this index.

**(4.6) *Ignition\_PutTime(BSTR IgnitionName, BSTR StartTime)***

**Description:** Gives an ignition a start time, given an ignition name.

**Parameters:**

IgnitionName	Unique identifier for the ignition point.
StartTime	The start time of the ignition point.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_FIRE_UNKNO WN	There is no fire with this name.
E_INVALIDARG	The time entered is in a format that cannot be parsed.
ERROR_SCENARIO_SIMULATION _RUNNING	A scenario is running and needs to be stopped before this action takes place.

**(4.7) Ignition\_Set(BSTR IgnitionName, int index, int ignition\_type, BSTR xString, BSTR yString)**

**Description:** Sets the x and y coordinates for all types of ignitions.

**Parameters:**

IgnitionName	Unique identifier for the ignition point.
index	The ignition index identifier.
ignition_type	Whether the ignition is of the type point, line, or polygon.
xString	An x coordinate, or group of x coordinates.
yString	A y coordinate, or group of y coordinates.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_FIRE_UNKNO WN	There is no fire with this name.
E_INVALIDARG	Invalid parameters.
ERROR_SCENARIO_SIMULATION _RUNNING	A scenario is running and needs to be stopped before this action takes place.
ERROR_FIRE_IGNITION_TYPE_U NKNOWN	The fire ignition type is unknown.
E_POINTER	Address provided is invalid.
FALSE	False.

**(4.8) Ignition\_PutName(int index, BSTR IgnitionName)**

**Description:** Set the name for an ignition based on its index.

**Parameters:**

IgnitionName	Unique identifier for the ignition point.
index	The index of the ignition.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_FIRE_UNKNO WN	There is no fire with this name.
E_INVALIDARG	Invalid parameters.
ERROR_SCENARIO_SIMULATION _RUNNING	A scenario is running and needs to be stopped before this action takes place.
ERROR_FIRE_IGNITION_TYPE_U NKNOWN	The fire ignition type is not known.
E_POINTER	Address provided is invalid.
FALSE	False.

**(4.9) Scenario\_SetDoubleAttribute(BSTR ScenarioName, int attribute\_key, double value)**

**Description:** Set the scenario attribute determined by the attribute key to a double value.

**Parameters:**

ScenarioName	Name of the scenario to be modified
attribute_key	Valid attribute keys are: CWFGM_SCENARIO_OPTION_PERIMETER_RESOLUTION CWFGM_SCENARIO_OPTION_SPATIAL_THRESHOLD CWFGM_SCENARIO_OPTION_SMOOTHING_WEIGHT CWFGM_SCENARIO_OPTION_SPECIFIED_FMC CWFGM_SCENARIO_OPTION_IGNITION_SIZE
value	The value to which the attribute key will be set.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
ERROR_SCENARIO_BAD_STATE	The scenario is in a bad state.
E_INVALIDARG	Invalid parameters.
ERROR_SCENARIO_BAD_TIMES	The scenario has invalid times.
ERROR_SCENARIO_BAD_TIMEST EPS	The scenario has invalid time steps.
TRUE	Operation succeeded.

FALSE	False.
E_FAIL	A general failure occurred.

**(4.10) Scenario\_SetBooleanAttribute(BSTR ScenarioName, int attribute\_key, double value)**

**Description:** Set the scenario attribute determined by the attribute key to a Boolean value.

**Parameters:**

ScenarioName	Name of the scenario to be modified
attribute_key	Valid attribute keys are: CWFGM_SCENARIO_OPTION_TOPOGRAPHY: CWFGM_SCENARIO_OPTION_ACCEL: CWFGM_SCENARIO_OPTION_BUI: CWFGM_SCENARIO_OPTION_GREENUP: CWFGM_SCENARIO_OPTION_FMC_TERRAIN: CWFGM_SCENARIO_OPTION_WIND: CWFGM_SCENARIO_OPTION_EXTINGUISHMENT: CWFGM_SCENARIO_OPTION_USE_2DGROWTH: CWFGM_SCENARIO_OPTION_BOUNDARY_STOP: CWFGM_SCENARIO_OPTION_SPOTTING: CWFGM_SCENARIO_OPTION_BREACHING: CWFGM_SCENARIO_OPTION_SINGLETHREADING: CWFGM_SCENARIO_OPTION_NONFUELS_AS_VECTOR_BREAKS: CWFGM_SCENARIO_OPTION_WEATHER_INTERPOLATE_TEMPORAL: CWFGM_SCENARIO_OPTION_WEATHER_INTERPOLATE_SPATIAL: CWFGM_SCENARIO_OPTION_ACCURATE_FMC_LOCATION:
value	The value to which the attribute key will be set.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
ERROR_SCENARIO_BAD_STATE	The scenario is in a bad state.
E_INVALIDARG	Invalid parameters.
ERROR_SCENARIO_BAD_TIMES	The scenario has invalid times.
ERROR_SCENARIO_BAD_TIMESTEPS	The scenario has an invalid time step.
TRUE	Operation succeeded.
FALSE	False.
E_FAIL	A general failure occurred.

**(4.11) Scenario\_SetIntegerAttribute(BSTR ScenarioName, int attribute\_key, double value)**

**Description:** Set the scenario attribute determined by the attribute key to an integer value.

**Parameters:**

<code>ScenarioName</code>	Name of the scenario to be modified
<code>attribute_key</code>	Valid attribute keys are: CWFGM_SCENARIO_OPTION_STARTING_POINTS CWFGM_SCENARIO_OPTION_MULTITHREADING CWFGM_SCENARIO_OPTION_TEMPORAL_THRESH OLD CWFGM_SCENARIO_OPTION_TEMPORAL_THRESH OLD_ACCEL CWFGM_SCENARIO_OPTION_DISPLAY_INTERVAL
<code>value</code>	The value to which the attribute key will be set.

**Returns:**

<code>S_OK</code>	Successful.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete the operation.
<code>ERROR_SCENARIO_UNKNOWN</code>	There is no scenario with this name.
<code>ERROR_SCENARIO_BAD_STATE</code>	The scenario is in a bad state.
<code>E_INVALIDARG</code>	Invalid parameters.
<code>ERROR_SCENARIO_BAD_TIMES</code>	The scenario has invalid times.
<code>ERROR_SCENARIO_BAD_TIMESTEPS</code>	The scenario has an invalid time step.
<code>TRUE</code>	Operation succeeded.
<code>FALSE</code>	False.
<code>E_FAIL</code>	A general failure occurred.

**(4.12) Scenario\_SetTime(BSTR ScenarioID, BSTR StartTime, BSTR EndTime)**

**Description:** Set the scenario start and end times.

**Parameters:**

<code>ScenarioID</code>	ID of the scenario to be modified
<code>StartTime</code>	The start time of the scenario.
<code>EndTime</code>	The end time of the scenario.

**Returns:**

<code>S_OK</code>	Successful.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete the operation.
<code>ERROR_SCENARIO_UNKNOWN</code>	There is no scenario with this name.
<code>E_INVALIDARG</code>	Invalid parameters.
<code>ERROR_SCENARIO_BAD_TIMES</code>	The scenario has invalid times.
<code>ERROR_SCENARIO_BAD_TIMESTEPS</code>	The scenario has an invalid time step.

EPS	
TRUE	Operation succeeded.
FALSE	False.
E_FAIL	A general failure occurred.
ERROR_SCENARIO_BAD_STATE	The scenario is in a bad state.

**(4.13) Scenario\_SetBurnCondition(BSTR ScenarioName, int Day, int Effective, int StartHour, int EndHour, double WSThreshold, double RHThreshold)**

**Description:** Sets the burn conditions for the scenario.

**Parameters:**

ScenarioName	Name of the scenario to be modified
Day	The integer representing the day.
Effective	Whether the burn condition is applied.
StartHour	The start hour.
EndHour	The end hour.
WSThreshold	The wind speed threshold value.
RHThreshold	The relative humidity threshold value.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_INVALIDARG	Invalid parameters.
ERROR_SCENARIO_BAD_STATE	The scenario is in a bad state.

**(4.14) Scenario\_SetDefaultBurnCondition(BSTR ScenarioName, int Effective, int StartHour, int EndHour, double WSThreshold, double RHThreshold)**

**Description:** Sets the default burn conditions for the scenario.

**Parameters:**

ScenarioName	Name of the scenario to be modified
Effective	Whether the burn condition is applied.
StartHour	The start hour.
EndHour	The end hour.
WSThreshold	The wind speed threshold value.
RHThreshold	The relative humidity threshold value.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete

	the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
ERROR_SCENARIO_BAD_STATE	The scenario is in a bad state.

**(4.15) Scenario\_PutName(int index, BSTR ScenarioName)**

**Description:** Sets the scenario name for the given scenario.

**Parameters:**

ScenarioName	Name of the scenario.
index	Index of the scenario.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.

**(4.16) WxStation\_SetLocation(BSTR WeatherStationID, double latitude, double longitude, double elevation)**

**Description:** Sets the latitude, longitude and elevation for a weather station identified by weather station ID.

**Parameters:**

WeatherStationID	Unique identifier for a weather station.
latitude	Latitude of weather station.
longitude	Longitude of weather station.
elevation	Elevation of weather station.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_BAD_STATE	There is no scenario with this name.
FALSE	False.
E_FAIL	A general failure occurred.
E_INVALIDARG	Invalid parameters.
ERROR_SCENARIO_BAD_TIMES	The scenario has invalid times.
ERROR_SCENARIO_BAD_TIMESTEP	The scenario has an invalid time step.
E_NOINTERFACE	No interface.
ERROR_SCENARIO_OPTION_INVALID	Internal error, please contact the software development team when this occurs.

**(4.17) *WxStation\_PutName(int index, BSTR StationName)***

**Description:** Sets the weather station at an index to the name passed to it.

**Parameters:**

<code>index</code>	Index of a weather station.
<code>StationName</code>	Unique identifier of a weather station.

**Returns:**

<code>S_OK</code>	Successful.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete the operation.
<code>ERROR_WEATHER_STATION_UNKNOW</code>	There is no weather station at this index.

**(4.18) *WxStream\_PutName(BSTR StationID, int index, BSTR StreamName)***

**Description:** Sets name of the weather stream specified by station and index.

**Parameters:**

<code>index</code>	Index of a weather station.
<code>StationID</code>	Unique identifier of a weather station.
<code>StreamName</code>	Unique identifier of a weather stream.

**Returns:**

<code>S_OK</code>	Successful.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete the operation.
<code>ERROR_WEATHER_STATION_UNKNOW</code>	There is no weather station with this name.
<code>ERROR_WEATHER_STREAM_UNKNOWN</code>	There is no weather stream at this index.

**(4.19) *WxStream\_SetDoubleAttribute(BSTR StationName, BSTR StreamName, int attribute\_key, double value)***

**Description:** Set station and stream double attributes.

**Parameters:**

<code>StationID</code>	Unique identifier for the station.
<code>attribute_key</code>	Used to identify the attribute to be changed.
<code>StreamName</code>	Unique identifier for the stream.
<code>value</code>	Value to set the attribute to.

**Returns:**

<code>S_OK</code>	Successful.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O

	error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_POINTER	Address provided is invalid.
ERROR_WEATHER_STATION_UNKOWN	There is no weather station with this name.
ERROR_WEATHER_STREAM_UNKOWN	There is no weather stream at this index.
ERROR_SCENARIO_SIMULATION_RUNNING	Can't change a weather stream while simulating.
FALSE	False.
E_FAIL	A general failure occurred.
E_INVALIDARG	Invalid parameters.
ERROR_WEATHER_OPTION_INVALIDID	Unrecognized key code.

**(4.20) *WxStream\_SetDoubleAttribute(BSTR StationName, BSTR StreamName, int attribute\_key, double value)***

**Description:** Set station and stream double attributes.

**Parameters:**

StationID	Unique identifier for the station.
attribute_key	Used to identify the attribute to be changed.
StreamName	Unique identifier for the stream.
value	Value to set the attribute to.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_POINTER	Address provided is invalid.
ERROR_WEATHER_STATION_UNKOWN	There is no weather station with this name.
ERROR_WEATHER_STREAM_UNKOWN	There is no weather stream at this index.
ERROR_SCENARIO_SIMULATION_RUNNING	Can't change a weather stream while simulating.
FALSE	False.
E_FAIL	A general failure occurred.
E_INVALIDARG	Invalid parameters.
ERROR_WEATHER_OPTION_INVALIDID	Unrecognized key code.

**(4.21) *WxStream\_SetBooleanAttribute(BSTR StationName, BSTR StreamName, int attribute\_key, double value)***

**Description:** Set station and stream attributes.

**Parameters:**

StationID	Unique identifier for the station.
attribute_key	Used to identify the attribute to be changed.
StreamName	Unique identifier for the stream.
value	Value to set the attribute to.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_POINTER	Address provided is invalid.
ERROR_WEATHER_STATION_UKN OWN	There is no weather station with this name.
ERROR_WEATHER_STREAM_UNKN OWN	There is no weather stream at this index.
ERROR_SCENARIO_SIMULATION _RUNNING	Can't change a weather stream while simulating.
FALSE	False.
E_FAIL	A general failure occurred.
E_INVALIDARG	Invalid parameters.
ERROR_WEATHER_OPTION_INVA LID	Unrecognized key code.

**(4.22) WxStream\_SetIntegerAttribute(BSTR StationName, BSTR StreamName, int attribute\_key, int value)****Description:** Set station and stream integer attributes.**Parameters:**

StationID	Unique identifier for the station.
attribute_key	Valid attribute keys are: CWFGM_WEATHER_OPTION_INITIAL_FFMCTIME CWFGM_WEATHER_OPTION_INITIAL_TEMPTIME CWFGM_WEATHER_OPTION_INITIAL_WSTIME
StreamName	Unique identifier for the stream.
value	Value to set the attribute to.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_POINTER	Address provided is invalid.
ERROR_WEATHER_STATION_UKN OWN	There is no weather station with this name.
ERROR_WEATHER_STREAM_UNKN OWN	There is no weather stream at this index.
ERROR_SCENARIO_SIMULATION _RUNNING	Can't change a weather stream while

	simulating.
FALSE	False.
E_FAIL	A general failure occurred.
E_INVALIDARG	Invalid parameters.
ERROR_WEATHER_OPTION_INVALID_LID	Unrecognized key code.

**(4.23) *Vector\_PutName(int index, BSTR VectorName)***

**Description:** Set the name of the vector at the index specified.

**Parameters:**

<code>index</code>	Index of the vector.
<code>VectorName</code>	Unique identifier for the vector.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_VECTORENGINE_UNKNOWN	Unrecognized key code.

**(4.24) *Filter\_PutName(int index, BSTR FilterName)***

**Description:** Set the name of the filter at the index specified.

**Parameters:**

<code>index</code>	Index of the filter.
<code>VectorName</code>	Unique identifier for the filter.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_GRIDFILTER_UNKNOWN	There is no grid filter at this index.

**(4.25) *Filter\_Greenup\_Set(BSTR FilterName, int X, int Y, int greenup\_on)***

**Description:** Set the name, coordinates and whether or not the green-up grid is used.

**Parameters:**

<code>FilterName</code>	Unique identifier of the filter.
<code>x</code>	An x coordinate.
<code>y</code>	A y coordinate.
<code>greenup_on</code>	Whether the green-up grid applies.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O

	error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_GRIDFILTER_UNKNOWN	There is no grid filter at this index.

**(4.26) *Filter\_Fuel\_SetRelationship(BSTR FilterName, int source\_fuel\_index, int dest\_fuel\_index)***

**Description:** Sets the source fuel name and destination fuel name for a patch filter.

**Parameters:**

FilterName	Unique identifier of the filter.
source_fuel_index	Source fuel index, or -1 for all fuels in the area.
dest_fuel_index	Destination fuel index, or -1 to replace no fuels in the area.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_FUELS_FUEL_UNKNOWN	No such fuel exists.
E_POINTER	Address provided is invalid.
ERROR_GRIDFILTER_UNKNOWN	There is no grid filter with this name.

**(4.27) *Filter\_FuelPoly\_Set(BSTR FilterName, BSTR xString, BSTR yString, int \*index)***

**Description:** Set the name, coordinates and index of a fuel polygon filter.

**Parameters:**

FilterName	Unique identifier of the filter.
xString	List of x coordinates.
yString	List of y coordinates.
index	Is set to the filter's index upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_INVALIDARG	Invalid parameters.
E_POINTER	Address provided is invalid.
ERROR_GRIDFILTER_UNKNOWN	There is no grid filter with this name.

**(4.28) *Filter\_Wx\_SetTime(BSTR FilterName, BSTR StartTime, BSTR EndTime)***

**Description:** Set the start and end times of the weather patch identified by filter name.

**Parameters:**

FilterName	Unique identifier of the filter.
StartTime	The starting time.
EndTime	The ending time.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_INVALIDARG	The start or end time format is incorrect.
ERROR_GRIDFILTER_UNKNOWN	No weather patch with this name exists.
E_FAIL	A general failure occurred.
NO_INTERFACE	No interface.

**(4.29) Filter\_WxPoly\_SetRule(BSTR FilterName, int wd\_operation, double wd\_val, int ws\_operation, double ws\_val)**

**Description:** Sets a weather polygon patch such that it will perform wind direction and/or wind speed changes and also sets the value for each.

**Parameters:**

FilterName	Unique identifier of the filter.
wd_operation	Whether or not directional operations will be performed.
wd_val	The value of the directional operation.
ws_operation	Whether or not speed operations will be performed.
ws_val	The value of the speed operation.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_GRIDFILTER_UNKNOWN	No weather patch with this name exists.

**(4.30) Filter\_WxPoly\_Set (BSTR FilterName, BSTR xString, BSTR yString, int \*index)**

**Description:** Set the name, coordinates and index of a polygon weather filter.

**Parameters:**

FilterName	Unique identifier of the filter.
xString	List of x coordinates.
yString	List of y coordinates.
index	Is set to the index of the filter upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O

	error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_GRIDFILTER_UNKNOWN	No weather patch with this name exists.

## (5) GET Functions

### (5.1) *GetNumOfVertices(ScenarioID, long \*Num)*

**Description:** Retrieves the number of the vertices produced by the simulation for all simulation display steps for all fires. This includes active and inactive vertices, plus marker points to separate the fire fronts (from different ignitions or times).

*Note: this replaces the depreciated GetNumOfVerticesActive function.*

**Parameters:**

ScenarioID	Unique identifier for a scenario.
Num	Set to the number of vertices on successful return.

**Returns:**

S_OK	Successful.
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_ SCENARIO_UNKNOWN	ScenarioID does not resolve to a known scenario.
E_FAIL	A general failure occurred.

### (5.2) *GetNumOfNewVertices(ScenarioID, long \*ResultNum)*

**Description:** Retrieves the number of the latest vertices produced by the last step. This value includes the marker vertices separating outputs from different ignitions.

*Note: this replaces the depreciated GetNumOfNewVerticesActive function.*

**Parameters:**

ScenarioID	Unique identifier for a scenario.
ResultNum	Set to the number of vertices on successful return.

**Returns:**

S_OK	Successful.
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to

	complete the operation.
<a href="#">ERROR_ SCENARIO_UNKNOWN</a>	ScenarioID does not resolve to a known scenario.
<a href="#">E_FAIL</a>	A general failure occurred.

**(5.3) *GetIndexVertice(BSTR ScenarioID, int index, double \*X, double \*Y)***

**Description:** Given a scenario identifier and an index, retrieves the coordinates of a specific point/vertex. X and Y will be set to these coordinates on successful return.

This function returns the coordinates in the grid cell units. If users need the distance to the LLC corner, they can multiply X and Y by the resolution (using the GetResolution function) to get the distance. The distance is measured in meters.

Note that (-1, -1) is a special vertex used for separating vertices produced by different fire ignitions or time steps in the simulation.

Note that this function replaces several old functions (GetFirstVertice, GetNextVertice, GetLastVertice, GetPreviousVertice, GetFirstVerticeActive, GetNextVerticeActive, GetPreviousVerticeActive, GetLastVerticeActive and GetIndexVerticeActive).

The current implementation of this function (and structures supporting it) significantly increases the memory overhead and operating inefficiencies associated with any PrometheusCOM application and therefore requires modification.

**Parameters:**

<a href="#">ScenarioID</a>	Unique identifier for a scenario.
<a href="#">Index</a>	Index of the point to retrieve.
<a href="#">X</a>	X coordinate on successful return (grid coordinate units).
<a href="#">Y</a>	Y coordinate on successful return (grid coordinate units).

**Returns:**

<a href="#">S_OK</a>	Successful.
<a href="#">E_POINTER</a>	Address provided is invalid.
<a href="#">ERROR_SEVERITY_WARNING</a>	Failed initialization or a non-specific I/O error.
<a href="#">E_OUTOFMEMORY</a>	There is not enough memory to complete the operation.
<a href="#">ERROR_ SCENARIO_UNKNOWN</a>	ScenarioID does not resolve to a known scenario

**(5.4) *GetAreaChange(BSTR ScenarioName, double \*areaChange)***

**Description:** Retrieves the change of area (in square metres) covered by fire of scenario referenced by ScenarioName. This change of area includes all fires.

**Parameters:**

ScenarioName	Unique identifier for a scenario.
areaChange	Set to the change of area on successful return (m <sup>2</sup> ).

**Returns:**

S_OK	Successful
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error; or if there is a problem with any of the fires attached to this scenario (e.g., the pointer to any fire in the list being NULL)
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	ScenarioName does not resolve to a known scenario.
E_FAIL	A general failure occurred.
ERROR_NO_DATA   ERROR_SEVERITY_WARNING	Nothing has been initialized yet.
ERROR_FIRE_INVALID_TIME	If the time is invalid.
ERROR_FIRE_SCENARIO_UNKNOWN	This fire object is not attached to the scenario specified by the scenario parameter.
SUCCESS_FIRE_NOT_STARTED	Fire not yet started (stats is set to 0).
ERROR_FIRE_STAT_UNKNOWN	If stat does not resolve to a known statistic.
E_INVALIDARG	Invalid parameters.
ERROR_POINT_NOT_IN_FIRE	The point is not in the fire area.

**(5.5) GetArea(BSTR ScenarioName, double \*area)**

**Description:** Retrieves the area (in square metres) covered by fire of scenario which is named as ScenarioName. This area includes all fires.

**Parameters:**

ScenarioName	Unique identifier for a scenario.
area	Set to the area on successful return (m <sup>2</sup> ).

**Returns:**

S_OK	Successful.
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error; or if there is a problem with any of the fires attached to this scenario (e.g., the pointer to any fire in the list being NULL)
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	ScenarioName does not resolve to a

	known scenario.
ERROR_NO_DATA   ERROR_SEVERITY_WARNING	Nothing has been initialized yet.
ERROR_FIRE_INVALID_TIME	If the time is invalid.
ERROR_FIRE_SCENARIO_UNKNOWN	This fire object is not attached to the scenario specified by the scenario parameter.
SUCCESS_FIRE_NOT_STARTED	Fire not yet started (stats is set to 0).
ERROR_FIRE_STAT_UNKNOWN	If stat does not resolve to a known statistic.
E_INVALIDARG	Invalid parameters.
ERROR_POINT_NOT_IN_FIRE	The point is not in the fire area.

**(5.6) GetLLCOffset(double \*xllcorner, double \*yllcorner)**

**Description:** Gets the offsets of the lower lefthand corner of any grid (*since multiple grids share the same projection*). These values are retrieved from the original projection file used when importing the ASCII grid layers.

**Parameters:**

xllcorner	Set to the x-offset on successful return.
yllcorner	Set to the y-offset on successful return.

**Returns:**

S_OK	Successful
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_GRID_UNINITIALIZED	The Grid object is not properly initialized.
S_FALSE	If asking for offset values where the appropriate array has not been initialized.
E_FAIL	A general failure occurred.
E_INVALIDARG	The attribute/option index was invalid/unknown for this object.
ERROR_GRID_LOCATION_OUT_OF_RANGE	The location is outside the grid.
ERROR_GRID_NO_DATA	The point is in a no data area.
E_UNEXPECTED	Internal error. Please contact the software development team.

**(5.7) GetXYParameters(BSTR ScenarioName, double X, double Y, double \*ros, double \*fi, double \*raz, double \*sfc, double \*cfb, double \*cfc, double \*tfc)**

**Description:** Given a specific scenario being simulated, and an (X/Y) location in the grid, this function will determine at what calculation time step a fire reached the point as

well as the fire conditions at that time. Statistical values for the fire point closest to the location requested are returned.

**Parameters:**

ScenarioName	Unique identifier for a scenario.
X	X coordinate in question.
Y	Y coordinate in question.
ros	Set to the actual rate of spread (m/min) on successful return.
fi	Set to the actual fire intensity (kW/m) on successful return.
raz	Set to the actual spreading direction (radians clockwise from north) on successful return.
sfc	Set to the surface fuel consumption (kg/m <sup>2</sup> ) on successful return.
cfb	Set to the crown fraction burned (per cent) on successful return.
cfc	Set to the crown fuel consumption (kg/m <sup>2</sup> ) on successful return.
tfc	Set to the total fuel consumption (kg/m <sup>2</sup> ) on successful return.

**Returns:**

S_OK	Successful
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	ScenarioName does not resolve to a known scenario.
ERROR_POINT_NOT_IN_FIRE	The x/y parameter passed in is not in a valid region (has not yet been burned).
ERROR_NO_DATA   ERROR_SEVERITY_WARNING	Nothing has been initialized yet.
ERROR_FIRE_INVALID_TIME	If the time is invalid.
ERROR_FIRE_SCENARIO_UNKNOWN	This fire object is not attached to the scenario specified by the scenario parameter.
SUCCESS_FIRE_NOT_STARTED	Fire not yet started (stats is set to 0).
ERROR_FIRE_STAT_UNKNOWN	If stat does not resolve to a known statistic.
ERROR_SCENARIO_BAD_STATE	If the function is run without a running scenario.
ERROR_GRID_UNINITIALIZED	The grid is not initialized properly.
ERROR_SCENARIO_NO_FIRES	There are no ignitions in the scenario.
ERROR_SCENARIO_BAD_TIMES	The scenario has invalid times.
ERROR_GRID_WEATHER_INVALID_DATES	The scenario has an invalid weather grid dates.

ERROR_GRID_WEATHER_NOT_IMPL EMENTED	Internal error. Please contact the software development team.
ERROR_WEATHER_STREAM_NOT_AS SIGNED	The scenario does not have a weather stream assigned.
ERROR_GRID_PRIMARY_STREAM_U NSPECIFIED	The scenario does not have a primary weather stream assigned.
ERROR_GRID_WEATHERSTATIONS_ TOO_CLOSE	The weather stations used are too close.
ERROR_GRID_WEATHERSTREAM_TI ME_OVERLAPS	The weather streams used overlap and are from the same station.
ERROR_SCENARIO_OPTION_INVAL ID	Internal error. Please contact the software development team.

**(5.8) *IsXYBurned(BSTR ScenarioName, double X, double Y, int \*burned)***

**Description:** Retrieves whether the location at (X, Y) was burned in a specific simulation.

This provides the same functionality as the function `GetXYParameters()`.

**Parameters:**

ScenarioName	Unique identifier for a scenario.
X	X-coordinate of interest.
Y	Y-coordinate of interest.
Burned	On successful return, this will be set to 1 if burned and 0 if not burned.

**Returns:**

S_OK	Successful.
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	ScenarioName does not resolve to a known scenario.
ERROR_SCENARIO_BAD_STATE	If the function is run while a scenario is running.
ERROR_NODATA   ERROR_SEVERIT Y_WARNING	Nothing initialized yet.
SUCCESS_FIRE_NOT_STARTED	Either no fires or the request predates the start time of the simulation.

**(5.9) *GetBurnedBox (BSTR ScenarioName, int \*left, int \*right, int \*bottom, int \*top)***

**Description:** Retrieves the rectangular area encompassing all burned areas of all fires at a specific time during a specific the simulation. If no fires have started burning yet, then return values are all set to -1.

**Parameters:**

ScenarioName	Unique identifier for a scenario.
Left	Set to the left bound of the rectangle on successful return (grid coordinate units).
Right	Set to the right bound of the rectangle on successful return (grid coordinate units).
Bottom	Set to the bottom bound of the rectangle on successful return (grid coordinate units).
Top	Set to the top bound of the rectangle on successful return (grid coordinate units).

**Returns:**

S_OK	Successful.
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	ScenarioName does not resolve to a known scenario.
ERROR_SCENARIO_BAD_STATE	If the function is run while a scenario is running.
SUCCESS_FIRE_NOT_STARTED	Either no fires or the request predates the start time of the simulation.

**(5.10) GetGridValue (int X, int Y, int \*fuel\_index)**

**Description:** Retrieves the fuel import index from the grid at the coordinate specified by X/Y.

**Parameters:**

X	X-coordinate in question.
Y	Y-coordinate in question.
fuel_index	Set to the fuel index on successful return - this value will correspond to one on the fuel lookup table. If fuel_index is -1, then there is no data at the specified coordinate.

**Returns:**

S_OK	Successful
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_GRID_UNINITIALIZED	The grid object associated with GridName is not properly initialized.
ERROR_GRID_LOCATION_OUT_OF_RANGE	The x/y coordinates passed in are out of range.
ERROR_FUELS_FUEL_UNKNOWN	There is no valid fuel index associated

	with the x/y coordinate.
--	--------------------------

**(5.11) *GetFuelIndexByName(BSTR FuelName, int \*fuel\_index)***

**Description:** Returns the fuel import index based on the fuel based on FuelName in the "FBP fuel type" column in the lookup table. If more than one import index is associated with the given fuel, then the first import index is returned.

**Parameters:**

FuelName	Unique identifier for a fuel.
fuel_index	Set to the import fuel index on successful return.

**Returns:**

S_OK	Successful.
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_FUELS_FUEL_UNKNOWN	FuelName does not resolve to a known fuel.

**(5.12) *GetFuelNameByIndex (int fuel\_index, BSTR \*FuelName)***

**Description:** This function returns the name of a fuel based on its index (fuel\_index). Please note that the caller has to free the memory returned via "StationName" after use.

**Parameters:**

fuel_index	Unique identifier for a fuel.
FuelName	Set to the fuel name on successful return.

**Returns:**

S_OK	Successful
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_FUELS_FUEL_UNKNOWN	'fuel_index' does not resolve to a known fuel.

**(5.13) *GetGridDimension (int \*xDim, int \*yDim)***

**Description:** Returns the X and Y dimensions of the grid referenced by gridName.

Currently all grids use the same projection and LUT information, and most of PrometheusCOM does not support the concept of multiple grids.

**Parameters:**

xDim	Set to the x dimension on successful return (grid coordinate units).
yDim	Set to the y dimension on successful return (grid coordinate units).

**Returns:**

S_OK	Successful
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_GRID_UNINITIALIZED	gridName does not resolve to a known grid, or if it is not properly initialized.

**(5.14) GetGridLLCLocation (double \*lat, double \*lon)**

**Description:** Returns the location on the planet for the grid.

Currently all grids use the same projection and LUT information, and most of PrometheusCOM does not support the concept of multiple grids.

**Parameters:**

GridName	Unique identifier for a grid.
Lat	Set to the latitude on successful return (degrees).
Long	Set to the longitude on successful return (degrees).

**Returns:**

S_OK	Successful.
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.

**(5.15) GetBooleanAttribute(int attribute\_key, int \*value)**

**Description:** Return the Boolean value of the attribute specified in the function call.

**Parameters:**

attribute_key	Valid keys are CWFGM_GRID_ATTRIBUTE_DAYLI GHT_SAVINGS, CWFGM_GRID_ATTRIBUTE_DEM_P RESENT, CWFGM_GRID_ATTRIBUTE_DSAM_
---------------	---

	PRESENT
value	Set to the value of the key upon return.

**Returns:**

S_OK	Successful.
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.

**(5.16)GetIntegerAttribute(int attribute\_key, int \*value)**

**Description:** Return the integer value of the attribute specified in the function call.

**Parameters:**

attribute_key	Valid key is CWFGM_GRID_ATTRIBUTE_TIMEZ ONE
value	Set to the value of the key upon return.

**Returns:**

S_OK	Successful.
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_INVALIDARG	Invalid parameters.

**(5.17)GetDoubleAttribute(int attribute\_key, double \*value)**

**Description:** Return the double value of the attribute specified in the function call.

**Parameters:**

attribute_key	Valid keys are CWFGM_GRID_ATTRIBUTE_DEFAULT_ELEVATION, CWFGM_GRID_ATTRIBUTE_MEDIAN_ELEVATION, CWFGM_GRID_ATTRIBUTE_PLOT_RESOLUTION
value	Set to the value of the key upon return.

**Returns:**

S_OK	Successful.
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.

E_INVALIDARG	Invalid parameters.
--------------	---------------------

**(5.18) *GetFuelDoubleAttribute(int fuel\_index, int attribute\_key, double \*value)***

**Description:** Return the double value of the fuel attribute specified in the function call.

**Parameters:**

fuel_index	
attribute_key	Valid keys are: FUELCOM_ATTRIBUTE_A FUELCOM_ATTRIBUTE_B FUELCOM_ATTRIBUTE_C FUELCOM_ATTRIBUTE_Q FUELCOM_ATTRIBUTE_MAXBE FUELCOM_ATTRIBUTE_BUI0 FUELCOM_ATTRIBUTE_CBH FUELCOM_ATTRIBUTE_CFL FUELCOM_ATTRIBUTE_PC FUELCOM_ATTRIBUTE_PDF FUELCOM_ATTRIBUTE_CURINGDEGREE FUELCOM_ATTRIBUTE_M3M4_C2_A FUELCOM_ATTRIBUTE_M3M4_C2_B FUELCOM_ATTRIBUTE_M3M4_C2_C FUELCOM_ATTRIBUTE_M4_D1_A FUELCOM_ATTRIBUTE_M4_D1_B FUELCOM_ATTRIBUTE_M4_D1_C FUELCOM_ATTRIBUTE_EQ35_THRESHOLD FUELCOM_ATTRIBUTE_EQ35A_MULT1 FUELCOM_ATTRIBUTE_EQ35A_MULT2 FUELCOM_ATTRIBUTE_EQ35B_ADDER FUELCOM_ATTRIBUTE_EQ35B_MULT1
value	Set to the value of the key upon return.

**Returns:**

S_OK	Successful.
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_INVALIDARG	Invalid parameters.
ERROR_FUELS_FUEL_UNKNOWN	There is no fuel with this index.

**(5.19) *Ignition\_GetTime(BSTR IgnitionName, BSTR\* StartTime)***

**Description:** Given an ignition name returns the start time.

**Parameters:**

IgnitionName	Unique identifier for the ignition point.
StartTime	Set to the start time of the ignition point upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_FIRE_UNKNO WN	There is no fire with this name.
E_POINTER	Address provided is invalid.

**(5.20) Ignition\_Count(BSTR IgnitionName, int \*count)**

**Description:** Returns the number of ignitions within a project.

**Parameters:**

IgnitionName	Unique identifier for the ignition point.
Count	Set to the number of ignitions within a project.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_FIRE_UNKNO WN	There is no fire with this name.
E_POINTER	Address provided is invalid.

**(5.21) Ignition\_Clear(BSTR IgnitionName)**

**Description:** Clears an ignition with multiple points.

**Parameters:**

IgnitionName	Unique identifier for the ignition point.
--------------	---

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_FIRE_UNKNO WN	There is no fire with this name.

**(5.22) Ignition\_Size(BSTR IgnitionName, int index, int \*count)**

**Description:** Gives the number of points that make up each sub-ignition within a single ignition file.

**Parameters:**

IgnitionName	Unique identifier for the ignition point.
Index	Index of the ignition.
Count	Set to the number of ignitions upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_FIRE_UNKNO WN	There is no fire with this name.
E_POINTER	Address provided is invalid.

**(5.23) Ignition\_MaxSize(BSTR IgnitionName, int index, int \*count)**

**Description:** Gives the number of points that make up the sub-ignition with the largest amount of points within a single ignition file.

**Parameters:**

IgnitionName	Unique identifier for the ignition point.
Index	Index of the ignition.
Count	Set to the ignition number upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_FIRE_UNKNO WN	There is no fire with this name.
E_POINTER	Address provided is invalid.

**(5.24) Ignition\_Get(BSTR IgnitionName, int index, int \*ignition\_type, int \*ignition\_size, BSTR \*xString, BSTR \*yString)**

**Description:** Gives information regarding the ignition such as the type, size and location.

**Parameters:**

IgnitionName	Unique identifier for the ignition point.
Index	Index of the ignition.
Ignition_type	The ignition type.
Ignition_size	The ignition size.
xString	List of x coordinates.
yString	List of y coordinates.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O

	error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_FIRE_UNKNO WN	There is no fire with this name.
E_POINTER	Address provided is invalid.

**(5.25) Ignition\_Type(BSTR IgnitionName, int index, int \*type)**

**Description:** Given an ignition name, gives the type.

**Parameters:**

IgnitionName	Unique identifier for the ignition point.
Index	Index of ignition.
Type	Set to the type of ignition upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_FIRE_UNKNO WN	There is no fire with this name.
E_POINTER	Address provided is invalid.

**(5.26) Ignition\_Number(int \*count)**

**Description:** Gives the number of ignitions in an FGM project.

**Parameters:**

Count	Set to the number of ignitions in the project.
-------	--

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_POINTER	Address provided is invalid.

**(5.27) Ignition\_GetName(int index, BSTR \*IgnitionName)**

**Description:** Given an ignition index, gives the ignition's name.

**Parameters:**

Index	Index of the ignition.
IgnitionName	Set to the name of the ignition upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_FIRE_UNKNO WN	There is no fire with this name.
E_POINTER	Address provided is invalid.

**(5.28) Ignition\_GetIndex(BSTR IgnitionName, int \*index)**

**Description:** Given an ignition name, gives the ignition's index.

**Parameters:**

IgnitionName	Uniquely identifies the ignition.
index	Set to the index of the ignition upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_FIRE_UNKNO WN	There is no fire with this name.
E_POINTER	Address provided is invalid.

**(5.29) Scenario\_GetDoubleAttribute(BSTR ScenarioName, int attribute\_key, double \*value)**

**Description:** Get the scenario attribute determined by the attribute key as a double value.

**Parameters:**

ScenarioName	Name of the scenario to be modified
attribute_key	Valid attribute keys are: CWFGM_SCENARIO_OPTION_PERIMETER_RESOLUTION CWFGM_SCENARIO_OPTION_SPATIAL_THRESHOLD CWFGM_SCENARIO_OPTION_SMOOTHING_WEIGHT CWFGM_SCENARIO_OPTION_SPECIFIED_FMC CWFGM_SCENARIO_OPTION_IGNITION_SIZE
value	Set to the value of the key upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.

ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_POINTER	Address provided is invalid.
ERROR_SCENARIO_OPTION_INVALID	This attribute key does not exist.
E_FAIL	A general failure occurred.

**(5.30) Scenario\_GetBooleanAttribute(BSTR ScenarioName, int attribute\_key, int \*value)**

**Description:** Get the scenario attribute determined by the attribute key as a boolean value.

**Parameters:**

ScenarioName	Name of the scenario to be modified
attribute_key	Valid attribute keys are: CWFGM_SCENARIO_OPTION_TOPOGRAPHY: CWFGM_SCENARIO_OPTION_ACCEL: CWFGM_SCENARIO_OPTION_BUI: CWFGM_SCENARIO_OPTION_GREENUP: CWFGM_SCENARIO_OPTION_FMC_TERRAIN: CWFGM_SCENARIO_OPTION_WIND: CWFGM_SCENARIO_OPTION_EXTINGUISHMENT: CWFGM_SCENARIO_OPTION_USE_2DGROWTH: CWFGM_SCENARIO_OPTION_BOUNDARY_STOP: CWFGM_SCENARIO_OPTION_SPOTTING: CWFGM_SCENARIO_OPTION_BREACHING: CWFGM_SCENARIO_OPTION_SINGLETHREADING:  CWFGM_SCENARIO_OPTION_NONFUELS_AS_VECTOR_BREAKS: CWFGM_SCENARIO_OPTION_WEATHER_INTERPOLATE_TEMPORAL: CWFGM_SCENARIO_OPTION_WEATHER_INTERPOLATE_SPATIAL: CWFGM_SCENARIO_OPTION_ACCURATE_FMC_LOCATION:
value	Set to the value of the key upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_POINTER	Address provided is invalid.
ERROR_SCENARIO_OPTION_INVALID	This attribute key does not exist.
TRUE	Operation succeeded.
FALSE	False.
E_FAIL	A general failure occurred.

**(5.31) Scenario\_GetIntegerAttribute(BSTR ScenarioName, int attribute\_key, int \*value)**

**Description:** Get the scenario attribute determined by the attribute key as an integer value.

**Parameters:**

ScenarioName	Name of the scenario to be modified
attribute_key	Valid attribute keys are: CWFGM_SCENARIO_OPTION_STARTING_POINTS CWFGM_SCENARIO_OPTION_MULTITHREADING CWFGM_SCENARIO_OPTION_TEMPORAL_THRESH OLD CWFGM_SCENARIO_OPTION_TEMPORAL_THRESH OLD_ACCEL CWFGM_SCENARIO_OPTION_DISPLAY_INTERVAL
value	Set to the value of the key upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_POINTER	Address provided is invalid.
ERROR_SCENARIO_OPTION_INV ALID	This attribute key does not exist.
E_FAIL	A general failure occurred.

**(5.32) Scenario\_GetTime(BSTR ScenarioID, BSTR \*StartTime, BSTR \*EndTime)**

**Description:** Get the scenario start and end times.

**Parameters:**

ScenarioName	ID of the scenario.
StartTime	Set to the start time upon return.
EndTime	Set to the end time upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.

**(5.33) Scenario\_GetBurnCondition(BSTR ScenarioName, int Day, int \*Effective, int \*StartHour, int \*EndHour, double \*WSThreshold, double \*RHThreshold)**

**Description:** Get the scenario burn condition for this specific day.

**Parameters:**

ScenarioName	Name of the scenario to look in.
Day	The Day in which we are interested.
Effective	Whether the conditions are applied.
StartHour	Set to the start hour upon return.
EndHour	Set to the end hour upon return.
WSThreshold	Set to the wind speed threshold upon return.
RHThreshold	Set to the relative humidity threshold upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_POINTER	Address provided is invalid.

**(5.34) Scenario\_GetDefaultBurnCondition(BSTR ScenarioName, int \*Effective, int \*StartHour, int \*EndHour, double \*WSThreshold, double \*RHThreshold)**

**Description:** Get the default scenario burn condition.

**Parameters:**

ScenarioName	Name of the scenario to look in.
Effective	Whether the conditions are applied.
StartHour	Set to the start hour upon return.
EndHour	Set to the end hour upon return.
WSThreshold	Set to the wind speed threshold upon return.
RHThreshold	Set to the relative humidity threshold upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_POINTER	Address provided is invalid.

**(5.35) Scenario\_GetIgnitionCount(BSTR ScenarioName, int \*count)**

**Description:** Get number of ignitions in a certain scenario.

**Parameters:**

ScenarioName	Name of the scenario to look in.
count	Set to the number of ignitions upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_POINTER	Address provided is invalid.

**(5.36) Scenario\_IgnitionAtIndex(BSTR ScenarioID, int index, BSTR \*FireID)**

**Description:** Get the ignition specified by the index.

**Parameters:**

ScenarioID	Unique identifier for the scenario.
index	Index number of ignition.
FireID	Unique identifier for the ignition.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_POINTER	Address provided is invalid.
ERROR_SCENARIO_FIRE_UNKNOWN	There is no ignition with this name.

**(5.37) Scenario\_IndexOfIgnition(BSTR ScenarioID, BSTR FireID, int \*index)**

**Description:** Get the index of the specified ignition.

**Parameters:**

ScenarioID	Unique identifier for the scenario.
FireId	Unique identifier for the ignition.
index	Set to the index of the ignition upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_POINTER	Address provided is invalid.
ERROR_SCENARIO_FIRE_UNKNOWN	There is no ignition with this name.

**(5.38) Scenario\_GetVectorCount(BSTR ScenarioName, int \*count)**

**Description:** Gets the number of vectors in the specified scenario.

**Parameters:**

<code>ScenarioName</code>	Unique identifier for the scenario.
<code>count</code>	Set to the number of vectors upon return.

**Returns:**

<code>S_OK</code>	Successful.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete the operation.
<code>ERROR_SCENARIO_UNKNOWN</code>	There is no scenario with this name.
<code>E_POINTER</code>	Address provided is invalid.

**(5.39) *Scenario\_VectorAtIndex(BSTR ScenarioName, int index, BSTR \*VectorID)***

**Description:** Gets the vector in the specified scenario by index.

**Parameters:**

<code>ScenarioName</code>	Unique identifier for the scenario.
<code>index</code>	Index of vector in the scenario.
<code>VectorID</code>	Unique identifier for the vector.

**Returns:**

<code>S_OK</code>	Successful.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete the operation.
<code>ERROR_SCENARIO_UNKNOWN</code>	There is no scenario with this name.
<code>E_POINTER</code>	Address provided is invalid.
<code>ERROR_SCENARIO_VECTORENGI VE_UNKNOWN</code>	There is no vector with this name.

**(5.40) *Scenario\_IndexOfVector(BSTR ScenarioID, BSTR VectorID, int \*index)***

**Description:** Gets the index in the specified scenario by index.

**Parameters:**

<code>ScenarioID</code>	Unique identifier for the scenario.
<code>VectorID</code>	Unique identifier for the vector.
<code>index</code>	Set to the index of the vector upon return.

**Returns:**

<code>S_OK</code>	Successful.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete the operation.
<code>ERROR_SCENARIO_UNKNOWN</code>	There is no scenario with this name.
<code>E_POINTER</code>	Address provided is invalid.
<code>ERROR_SCENARIO_VECTORENGI</code>	There is no vector with this name.

VE_UNKNOWN	
------------	--

**(5.41) Scenario\_GetWxStreamCount(BSTR ScenarioID, int \*count)**

**Description:** Gets the number of weather streams in the specified scenario.

**Parameters:**

ScenarioID	Unique identifier for the scenario.
count	Set to the number of weather streams upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_POINTER	Address provided is invalid.

**(5.42) Scenario\_WxStreamAtIndex(BSTR ScenarioName, int index, BSTR \*Station, BSTR \*Stream)**

**Description:** Gets the weather stream and station in the specified scenario by index.

**Parameters:**

ScenarioName	Unique identifier for the scenario.
index	Index of the weather stream within the scenario.
Station	Unique identifier for the station.
Stream	Unique identifier for the stream.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_POINTER	Address provided is invalid.
ERROR_WEATHER_STREAM_UNKNOWN	There is no weather stream with this name.

**(5.43) Scenario\_IndexOfWxStream(BSTR ScenarioID, BSTR StationID, BSTR StreamID, int \*index)**

**Description:** Gets the index in the specified scenario by weather stream and station.

**Parameters:**

ScenarioID	Unique identifier for the scenario.
StationID	Unique identifier for the weather station.
StreamID	Unique identifier for the weather stream.
index	Set to the index of the weather stream.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_POINTER	Address provided is invalid.
ERROR_WEATHER_STREAM_UNKNOWN	There is no weather stream with this name.
ERROR_WEATHER_STATION_UNKNOWN	There is no weather station with this name.

**(5.44) Scenario\_GetFilterCount(BSTR ScenarioID, int \*count)**

**Description:** Gets the number of filters in the scenario.

**Parameters:**

ScenarioID	Unique identifier for the scenario.
count	Set to the number of filters upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_POINTER	Address provided is invalid.

**(5.45) Scenario\_FilterAtIndex(BSTR ScenarioID, int index, BSTR \*FilterID)**

**Description:** Gets filter in the scenario by index.

**Parameters:**

ScenarioID	Unique identifier for the scenario.
index	Index of wanted filter.
FilterID	Set to the name of the filter upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_POINTER	Address provided is invalid.
ERROR_SCENARIO_WEATHERGRID_UNKNOWN	There is not weather grid with this name.

**(5.46) Scenario\_IndexOfFilter(BSTR ScenarioID, BSTR FilterID, int \*index)**

**Description:** Gets index of the filter in the scenario.

**Parameters:**

ScenarioID	Unique identifier for the scenario.
FilterID	Unique identifier for the filter.
index	Set to the index of the filter upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_POINTER	Address provided is invalid.
ERROR_SCENARIO_WEATHERGRID_UNKNOWN	There is not weather grid with this name.

**(5.47) Scenario\_Number(int \*count)**

**Description:** Gets the number of scenarios in the FGM.

**Parameters:**

count	Set to the number of scenarios upon return.
-------	---

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_POINTER	Address provided is invalid.

**(5.48) Scenario\_GetName(int index, BSTR \*ScenarioName)**

**Description:** Gets the name of the scenario identified by index.

**Parameters:**

index	Index of a scenario.
-------	----------------------

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_POINTER	Address provided is invalid.

**(5.49) Scenario\_GetIndex(BSTR ScenarioName, int \*index)**

**Description:** Gets the index of a scenario identified by name.

**Parameters:**

ScenarioName	Unique identifier of a scenario.
index	Set to the index of the scenario upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_POINTER	Address provided is invalid.

**(5.50) Scenario\_GetGridValue(BSTR ScenarioName, int x, int y, int \*fuel\_index)**

**Description:** Gets the fuel grid value at coordinates x, y in a specified scenario.

**Parameters:**

ScenarioName	Unique identifier of a scenario.
x	X coordinate
y	Y coordinate

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_POINTER	Address provided is invalid.
ERROR_GRID_UNINITIALIZED	The grid is not initialized properly.
ERROR_GRID_LOCATION_OUT_OF_RANGE	The location is outside the grid.
ERROR_FUELS_FUEL_UNKNOWN	There is no fuel with that name.

**(5.51) WxStation\_GetLocation(BSTR WeatherStationID, double \*Latitude, double \*Longitude, double \*Elevation)**

**Description:** Gets the latitude, longitude and elevation of a weather station specified by weather station id.

**Parameters:**

WeatherStationID	Unique identifier for a weather station.
latitude	Sets this to latitude on return.
longitude	Sets this to longitude on return.
elevation	Sets this to elevation on return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_POINTER	Address provided is invalid.

<code>ERROR_WEATHER_STATION_UKN OWN</code>	No weather station exists with this id.
--	---

**(5.52) *WxStation\_Number(int \*NumOfStations)***

**Description:** Gets the number of weather stations.

**Parameters:**

<code>NumOfStations</code>	Sets this to the number of weather stations on return.
----------------------------	--

**Returns:**

<code>S_OK</code>	Successful.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.
<code>E_POINTER</code>	Address provided is invalid.

**(5.53) *WxStation\_GetName(int index, BSTR\* StationName)***

**Description:** Retrieves the station name of a weather station given an index.

**Parameters:**

<code>StationName</code>	Unique identifier for the station.
<code>index</code>	Index of weather station.

**Returns:**

<code>S_OK</code>	Successful.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete the operation.
<code>E_POINTER</code>	Address provided is invalid.
<code>ERROR_WEATHER_STATION_UKN OWN</code>	There is no weather station at this index.

**(5.54) *WxStation\_GetIndex(BSTR StationName, int \*index)***

**Description:** Retrieves the index of a weather station given a name.

**Parameters:**

<code>StationName</code>	Unique identifier for the station.
<code>index</code>	Index of weather station.

**Returns:**

<code>S_OK</code>	Successful.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete the operation.
<code>E_POINTER</code>	Address provided is invalid.
<code>ERROR_WEATHER_STATION_UKN OWN</code>	There is no weather station with this name.

**(5.55) *WxStream\_Number(BSTR StationID, int \*NumOfStreams)***

**Description:** Retrieves the number of weathers streams for a certain station.

**Parameters:**

StationID	Unique identifier for the station.
NumOfStreams	Set to the number of streams upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_POINTER	Address provided is invalid.
ERROR_WEATHER_STATION_UKN OWN	There is no weather station with this name.

**(5.56) WxStream\_GetName(BSTR StationID, int index, BSTR \*StreamName)**

**Description:** Retrieves the name of a weather stream for a certain station and index.

**Parameters:**

StationID	Unique identifier for the station.
index	Index of weather station.
StreamName	Unique identifier for the stream.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_POINTER	Address provided is invalid.
ERROR_WEATHER_STATION_UKN OWN	There is no weather station with this name.
ERROR_WEATHER_STREAM_UNKN OWN	There is no weather stream at this index.

**(5.57) WxStream\_GetIndex(BSTR StationID, BSTR StreamName, int \*index)**

**Description:** Retrieves the index of a weather stream for a certain station and stream name.

**Parameters:**

StationID	Unique identifier for the station.
index	Index of weather station.
StreamName	Unique identifier for the stream.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_POINTER	Address provided is invalid.
ERROR_WEATHER_STATION_UKN	There is no weather station with this name.

OWN	
ERROR_WEATHER_STREAM_UNKN OWN	There is no weather stream at this index.

**(5.58) *WxStream\_GetStartTime(BSTR StationID, BSTR StreamName, BSTR \*StartTime)***

**Description:** Retrieves the start time of a stream.

**Parameters:**

StationID	Unique identifier for the station.
StartTime	Set to the start time upon return.
StreamName	Unique identifier for the stream.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_POINTER	Address provided is invalid.
ERROR_WEATHER_STATION_UKN OWN	There is no weather station with this name.
ERROR_WEATHER_STREAM_UNKN OWN	There is no weather stream at this index.

**(5.59) *WxStream\_GetDoubleAttribute(BSTR StationName, BSTR StreamName, int attribute\_key, double \*value)***

**Description:** Retrieves the value of a certain attribute of a station, or stream.

**Parameters:**

StationID	Unique identifier for the station.
attribute_key	Valid attributes are: CWFGM_WEATHER_OPTION_TEMP_ALPHA: CWFGM_WEATHER_OPTION_TEMP_BETA: CWFGM_WEATHER_OPTION_TEMP_GAMMA: CWFGM_WEATHER_OPTION_WIND_ALPHA: CWFGM_WEATHER_OPTION_WIND_BETA: CWFGM_WEATHER_OPTION_WIND_GAMMA: CWFGM_WEATHER_OPTION_INITIAL_FFMC: CWFGM_WEATHER_OPTION_INITIAL_TEMP: CWFGM_WEATHER_OPTION_INITIAL_WS: CWFGM_WEATHER_OPTION_INITIAL_DC: CWFGM_WEATHER_OPTION_INITIAL_DMC: CWFGM_WEATHER_OPTION_INITIAL_BUI: CWFGM_WEATHER_OPTION_INITIAL_RAIN:
StreamName	Unique identifier for the stream.
value	Set to the correct value upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.

E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_POINTER	Address provided is invalid.
ERROR_WEATHER_STATION_UKN OWN	There is no weather station with this name.
ERROR_WEATHER_STREAM_UNKN OWN	There is no weather stream at this index.
E_FAIL	A general failure occurred.

**(5.60) *WxStream\_GetBooleanAttribute(BSTR StationName, BSTR StreamName, int attribute\_key, int \*value)***

**Description:** Retrieves the value of a certain attribute of a station, or stream.

**Parameters:**

StationID	Unique identifier for the station.
attribute_key	Valid attributes are: CWFGM_WEATHER_OPTION_FFMC_VANWAGNER CWFGM_WEATHER_OPTION_FFMC_HYBRID CWFGM_WEATHER_OPTION_FFMC_LAWSON CWFGM_WEATHER_OPTION_USE_TEMP CWFGM_WEATHER_OPTION_USE_WS CWFGM_WEATHER_OPTION_ORIGIN_FILE
StreamName	Unique identifier for the stream.
value	Set to the correct value upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
E_POINTER	Address provided is invalid.
ERROR_WEATHER_STATION_UKN OWN	There is no weather station with this name.
ERROR_WEATHER_STREAM_UNKN OWN	There is no weather stream at this index.
E_FAIL	A general failure occurred.

**(5.61) *WxStream\_GetIntegerAttribute(BSTR StationName, BSTR StreamName, int attribute\_key, int \*value)***

**Description:** Retrieves the value of a certain attribute of a station, or stream.

**Parameters:**

StationID	Unique identifier for the station.
attribute_key	Valid attributes are: CWFGM_WEATHER_OPTION_INITIAL_FFMC CWFGM_WEATHER_OPTION_INITIAL_TEMPTIME CWFGM_WEATHER_OPTION_INITIAL_WSTIME
StreamName	Unique identifier for the stream.
value	Set to the correct value upon return.

**Returns:**

<a href="#">S_OK</a>	Successful.
<a href="#">ERROR_SEVERITY_WARNING</a>	Failed initialization or a non-specific I/O error.
<a href="#">E_OUTOFMEMORY</a>	There is not enough memory to complete the operation.
<a href="#">E_POINTER</a>	Address provided is invalid.
<a href="#">ERROR_WEATHER_STATION_UNKOWN</a>	There is no weather station with this name.
<a href="#">ERROR_WEATHER_STREAM_UNKOWN</a>	There is no weather stream at this index.
<a href="#">E_FAIL</a>	A general failure occurred.

**(5.62) *Vector\_Number(int \*count)***

**Description:** Retrieves the number of vectors in a project.

**Parameters:**

<a href="#">count</a>	Sets this to the number of vectors upon return.
-----------------------	---

**Returns:**

<a href="#">S_OK</a>	Successful.
<a href="#">ERROR_SEVERITY_WARNING</a>	Failed initialization or a non-specific I/O error.
<a href="#">E_POINTER</a>	Address provided is invalid.

**(5.63) *Vector\_GetName(int index, BSTR \*VectorName)***

**Description:** Retrieves the name of a vector given its index.

**Parameters:**

<a href="#">index</a>	Index of a vector.
<a href="#">VectorName</a>	Is set to the vector name upon return.

**Returns:**

<a href="#">S_OK</a>	Successful.
<a href="#">ERROR_SEVERITY_WARNING</a>	Failed initialization or a non-specific I/O error.
<a href="#">E_POINTER</a>	Address provided is invalid.
<a href="#">ERROR_SCENARIO_VECTORENGINE_UNKNOWN</a>	

**(5.64) *Vector\_GetIndex(BSTR VectorName, int \*index)***

**Description:** Retrieves the index of a vector given its name.

**Parameters:**

<a href="#">index</a>	Set to the index of the vector upon return.
<a href="#">VectorName</a>	Unique identifier for the vector.

**Returns:**

<a href="#">S_OK</a>	Successful.
<a href="#">ERROR_SEVERITY_WARNING</a>	Failed initialization or a non-specific I/O error.

	error.
E_POINTER	Address provided is invalid.
ERROR_SCENARIO_VECTORENGI NE_UNKNOWN	Unknown object.

**(5.65) *Filter\_Number(int \*count)***

**Description:** Retrieves the number of filters in a project.

**Parameters:**

count	Sets this to the number of filters upon return.
-------	---

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_POINTER	Address provided is invalid.
E_OUTOFMEMORY	There is not enough memory to complete the operation.

**(5.66) *Filter\_GetName(int index, BSTR \*FilterName)***

**Description:** Retrieves the name of a filter given its index.

**Parameters:**

index	Index of a filter.
FilterName	Is set to the filter name upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_POINTER	Address provided is invalid.
ERROR_GRIDFILTER_UNKNOWN	There is no grid filter at this index.
E_OUTOFMEMORY	There is not enough memory to complete this operation.

**(5.67) *Filter\_GetType(int index, BSTR \*FilterName)***

**Description:** Retrieves the name of a filter given the index.

**Parameters:**

index	Index of a filter.
FilterName	Name of the filter's type (returned).

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_POINTER	Address provided is invalid.
ERROR_GRIDFILTER_UNKNOWN	There is no grid filter at this index.
E_OUTOFMEMORY	There is not enough memory to complete this operation.

**(5.68) *Filter\_GetIndex(BSTR FilterName, int \*index)***

**Description:** Retrieves the index of a filter given the name.

**Parameters:**

<code>index</code>	Set to the index upon return.
<code>FilterName</code>	Unique identifier for the filter.

**Returns:**

<code>S_OK</code>	Successful.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.
<code>E_POINTER</code>	Address provided is invalid.
<code>ERROR_GRIDFILTER_UNKNOWN</code>	There is no grid filter at this index.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete this operation.

**(5.69) *Filter\_Greenup\_Get(BSTR FilterName, int X, int Y, int \*green\_up)***

**Description:** Retrieves a green up grid given its name and coordinates.

**Parameters:**

<code>FilterName</code>	Unique identifier for the filter.
<code>X</code>	An x coordinate
<code>Y</code>	A y coordinate
<code>greenup_on</code>	Whether the green-up grid is applied or not.

**Returns:**

<code>S_OK</code>	Successful.
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.
<code>E_POINTER</code>	Address provided is invalid.
<code>ERROR_GRIDFILTER_UNKNOWN</code>	There is no grid filter at this index.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete this operation.

**(5.70) *Filter\_Fuel\_GetRelationship(BSTR FilterName, int \*source\_fuel\_index, int \*dest\_fuel\_index)***

**Description:** Retrieves the file import index of the source and destination fuels of a filter patch referenced by PatchName.

**Parameters:**

<code>FilterName</code>	Unique identifier for the filter.
<code>source_fuel_index</code>	Set to the file import index of the source fuel on successful return. If FuelFileIndex is -1 indicates failure to identify the fuel patch.
<code>dest_fuel_index</code>	Set to the import index of the destination fuel on successful return. If FuelFileIndex is -1 indicates failure to identify the fuel patch.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_POINTER	Address provided is invalid.
E_OUTOFMEMORY	There is not enough memory to complete this operation.
E_NOINTERFACE	No interface.

**(5.71) *Filter\_FuelPoly\_Count(BSTR FilterName, int \*count)***

**Description:** Retrieve the number of fuel polygons within a project.

**Parameters:**

FilterName	Unique identifier for the filter.
count	Is set to the count number upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_POINTER	Address provided is invalid.
E_OUTOFMEMORY	There is not enough memory to complete this operation.
ERROR_GRIDFILTER_UNKNOWN	No grid filter with this name exists.

**(5.72) *Filter\_FuelPoly\_Size(BSTR FilterName, int index, int \*count)***

**Description:** Return the size of a polygon given the name and index of it.

**Parameters:**

FilterName	Unique identifier for the filter.
count	Is set to the count number upon return.
index	Index of the fuel polygon filter.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_POINTER	Address provided is invalid.
E_OUTOFMEMORY	There is not enough memory to complete this operation.
ERROR_GRIDFILTER_UNKNOWN	No grid filter with this name exists.

**(5.73) *Filter\_FuelPoly\_MaxSize(BSTR FilterName, int \*count)***

**Description:** Return the maximum size of a polygon given the name.

**Parameters:**

FilterName	Unique identifier for the filter.
count	Is set to the count number upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_POINTER	Address provided is invalid.
E_OUTOFMEMORY	There is not enough memory to complete this operation.
ERROR_GRIDFILTER_UNKNOWN	No grid filter with this name exists.

**(5.74) *Filter\_FuelPoly\_Get(BSTR FilterName, int index, int \*ignition\_size, BSTR \*xString, BSTR \*yString)***

**Description:** Returns the polygon data defining the polygon fuel patch.

**Parameters:**

FilterName	Unique identifier for the filter.
index	Is set to the count number upon return.
ignition_size	The size of the ignition.
xString	The list of X coordinates.
yString	The list of Y coordinates.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_POINTER	Address provided is invalid.
E_OUTOFMEMORY	There is not enough memory to complete this operation.
ERROR_GRIDFILTER_UNKNOWN	No grid filter with this name exists.

**(5.75) *Filter\_FuelPoly\_Area(BSTR PatchName, double \*area)***

**Description:** Calculates the area of the specified fuel patch and passes it into area.

**Parameters:**

PatchName	Unique identifier for the filter.
area	Is set to the area upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_POINTER	Address provided is invalid.
E_OUTOFMEMORY	There is not enough memory to complete this operation.
ERROR_GRIDFILTER_UNKNOWN	No grid filter with this name exists.

**(5.76) *Filter\_Wx\_GetTime(BSTR FilterName, BSTR \*StartTime, BSTR \*EndTime)***

**Description:** Retrieve the start and end time for a specified weather polygon patch.

**Parameters:**

FilterName	Unique identifier for the filter.
StartTime	Is set to the start time upon return.
EndTime	Is set to the end time upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_POINTER	Address provided is invalid.
E_OUTOFMEMORY	There is not enough memory to complete this operation.
ERROR_GRIDFILTER_UNKNOWN	No grid filter with this name exists.
TRUE	Operation succeeded.
FALSE	False.

**(5.77) Filter\_WxPoly\_GetRule(BSTR FilterName, int \*wd\_operation, double \*wd\_val, int \*ws\_operation, double \*ws\_val)**

**Description:** Retrieve the rules for the weather patch specified. The rules are whether or not speed and direction are being set to certain values.

**Parameters:**

FilterName	Unique identifier for the filter.
wd_operation	Is set to whether or not direction is used upon return.
wd_val	Is set to the direction upon return.
ws_operation	Is set to whether or not speed is used upon return.
ws_val	Is set to the speed upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_POINTER	Address provided is invalid.
E_OUTOFMEMORY	There is not enough memory to complete this operation.
ERROR_GRIDFILTER_UNKNOWN	No grid filter with this name exists.

**(5.78) Filter\_WxPoly\_Count(BSTR FilterName, int \*count)**

**Description:** Retrieve the number of weather polygons in a project.

**Parameters:**

FilterName	Unique identifier for the filter.
count	Is set to the number of weather polygon patches upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_POINTER	Address provided is invalid.

E_OUTOFMEMORY	There is not enough memory to complete this operation.
ERROR_GRIDFILTER_UNKNOWN	No grid filter with this name exists.

**(5.79) *Filter\_WxPoly\_Size(BSTR FilterName, int index, int \*count)***

**Description:** Returns the number of vertices of the polygon at 'index'.

**Parameters:**

FilterName	Unique identifier for the filter.
count	The weather polygon patch count.
index	Index of the weather polygon.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_POINTER	Address provided is invalid.
E_OUTOFMEMORY	There is not enough memory to complete this operation.
ERROR_GRIDFILTER_UNKNOWN	No grid filter with this name exists.

**(5.80) *Filter\_WxPoly\_MaxSize(BSTR FilterName, int \*count)***

**Description:** Returns the maximum number of vertices of any polygon in this weather patch.

**Parameters:**

FilterName	Unique identifier for the filter.
count	The weather polygon patch count.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_POINTER	Address provided is invalid.
E_OUTOFMEMORY	There is not enough memory to complete this operation.
ERROR_GRIDFILTER_UNKNOWN	No grid filter with this name exists.

**(5.81) *Filter\_WxPoly\_Get(BSTR FilterName, int index, int size, BSTR xString, BSTR, yString)***

**Description:** Given a weather polygons name and index returns its size and coordinates.

**Parameters:**

FilterName	Unique identifier for the filter.
index	Unique index for the filter.
size	Size of the polygon.
xString	List of x coordinates.
yString	List of y coordinates.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_POINTER	Address provided is invalid.
E_OUTOFMEMORY	There is not enough memory to complete this operation.
ERROR_GRIDFILTER_UNKNOWN	No grid filter with this name exists.

**(5.82) *Filter\_WxPoly\_Area(BSTR PatchName, double \*area)***

**Description:** Calculates the area of the specified patch and passes the value to area.

**Parameters:**

FilterName	Unique identifier for the filter.
area	Is set to the area upon return.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_POINTER	Address provided is invalid.
E_OUTOFMEMORY	There is not enough memory to complete this operation.
ERROR_GRIDFILTER_UNKNOWN	No grid filter with this name exists.

<b>(6) EDIT Functions</b>
---------------------------

**(6.1) *WxStream\_EditWeatherDaily( BSTR WeatherStationID, BSTR WeatherStreamID, BSTR Day, double MinTemp, double MaxTemp, double MinWS, double MaxWS, double MinRH, double Preci, double WD)***

**Description:** Edits a specific weather stream's daily observation values for a specific day.

This day must already exist in the weather stream, or be adjacent to an existing day.

Judy Beck's diurnal curve equations are used to calculate hourly data from these daily observations.

**Parameters:**

WeatherStationID	Unique identifier for a weather station.
WeatherStreamID	Unique identifier for a weather stream.
Day	Identifies the day to modify (or add) in the weather stream.
MinTemp	Minimum observed temperature (°C) for the day.
MaxTemp	Maximum observed temperature (°C) for the day.
MinWS	Minimum observed wind speed

	(km/hr) for the day.
MaxWS	Maximum observed wind speed (km/hr) for the day.
MinRH	Minimum observed relative humidity (%) for the day.
Preci	Total observed precipitation (mm) for the day.
WD	Mean observed wind direction (degrees clockwise from north) for the day.

**Returns:**

S_OK	Successful
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error; or if the operation of making daily observations failed (e.g. because the requested day is not present or adjacent to existing data).
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_WEATHER_STATION_UNKNOWN	WeatherStationID does not correspond to a known weather station.
ERROR_WEATHER_STREAM_UNKNOWN	WeatherStreamID does not correspond to a known weather stream.
ERROR_SCENARIO_SIMULATION_RUNNING	If trying to set a new start time when the fire is attached to a running simulation.
E_POINTER	Address provided is invalid.

**(6.2) *WxStream\_EditWeatherHourly(BSTR WeatherStationID, BSTR WeatherStreamID, BSTR Day, int Hour, double Temp, double RH, double WS, double Preci, double WD)***

**Description:** Edits a specific weather stream's hourly readings for a specific hour of a specific day. This day must already exist in the weather stream, or be adjacent to an existing day.

**Parameters:**

WeatherStationID	uniquely identifies a weather station
WeatherStreamID	uniquely identifies a weather stream
Day	Identifies the day to modify (or add) in the weather stream.
Hour	Hour (0 to 23).
Temp	Temperature for the hour (°C).
RH	Relative humidity for the hour (%).
WS	Wind speed for the hour (km/hr).
Preci	Precipitation for the hour (mm).

WD	Wind direction for the hour (degrees clockwise from north).
----	---

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error; or if the operation of making hourly readings failed (e.g. because the requested day is not present or adjacent to existing data).
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_WEATHER_STATION_UNKNOWN	WeatherStationID does not correspond to a known weather station.
ERROR_WEATHER_STREAM_UNKNOWN	WeatherStreamID does not correspond to a known weather stream.
ERROR_SCENARIO_BAD_TIMES	Day is not in a valid date format.
E_POINTER	Address provided is invalid.
ERROR_SCENARIO_SIMULATION_RUNNING	If trying to set a new start time when the fire is attached to a running simulation.

**(6.3) Ignition\_Delete(BSTR IgnitionName)**

**Description:** Deletes the ignition denoted by IgnitionName and removes it from all related scenarios.

**Parameters:**

IgnitionName	Unique identifier for an ignition.
--------------	------------------------------------

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error; or if there is a problem with the scenario indexing.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_FIRE_UNKNOWN	The ignition of that name does not exist.

**(6.4) Scenario\_AddIgnition(BSTR ScenarioID, BSTR FireId)**

**Description:** Adds the specified ignition to the specified scenario.

**Parameters:**

ScenarioID	ID of the scenario to be modified.
FireID	ID of the ignition to be added.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O

	error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
ERROR_SCENARIO_FIRE_UNKNOWN	There is no ignition with this name.
E_FAIL	A general failure occurred.

**(6.5) Scenario\_RemoveIgnition(BSTR ScenarioName, BSTR IgnitionName)**

**Description:** Removes the specified ignition from the specified scenario.

**Parameters:**

IgnitionName	Unique identifier for an ignition.
ScenarioName	Unique identifier for a scenario.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error; or if there is a problem with the scenario indexing.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_FIRE_UNKNOWN	The ignition of that name does not exist.
ERROR_SCENARIO_UNKNOWN	The scenario of that name does not exist.

**(6.6) Scenario\_AddVector(BSTR ScenarioID, BSTR FuelBreakID)**

**Description:** Adds the specified vector to the specified scenario.

**Parameters:**

ScenarioID	ID of the scenario to be modified.
FuelBreakID	ID of the fuel break vector to be added.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_FAIL	A general failure occurred.
ERROR_SCENARIO_VECTORENGINE_UNKNOWN	There is no vector with this name.

**(6.7) Scenario\_RemoveVector(BSTR ScenarioName, BSTR FuelBreakName)**

**Description:** Removes the specified fuel break vector from the specified scenario.

**Parameters:**

FuelBreakName	Unique identifier for a fuel break.
---------------	-------------------------------------

**Returns:**

ScenarioName	Unique identifier for a scenario.
S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error; or if there is a problem with the scenario indexing.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_VECTORENGINE_UNKNOWN	The ignition of that name does not exist.
ERROR_SCENARIO_UNKNOWN	The scenario of that name does not exist.
E_FAIL	A general failure occurred.

**(6.8) Scenario\_AddWxStream(BSTR ScenarioID, BSTR StationID, BSTR StreamID)****Description:** Adds the specified weather stream to the specified scenario.**Parameters:**

ScenarioID	ID of the scenario to be modified.
StationID	ID of the weather station to be used.
StreamID	ID of the weather stream to be added.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_FAIL	A general failure occurred.
ERROR_SCENARIO_WEATHER_STATION_UNKNOWN	There is no weather station with this name.
ERROR_SCENARIO_WEATHER_STREAM_UNKNOWN	There is no weather stream with this name.

**(6.9) Scenario\_RemoveWxStream(BSTR ScenarioID, BSTR StationID, BSTR StreamID)****Description:** Removes the specified weather stream to the specified scenario.**Parameters:**

ScenarioID	ID of the scenario to be modified.
StationID	ID of the weather station to be used.
StreamID	ID of the weather stream to be added.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete

	the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_FAIL	A general failure occurred.
ERROR_SCENARIO_WEATHER_STATION_UNKNOWN	There is no weather station with this name.
ERROR_SCENARIO_WEATHER_STREAM_UNKNOWN	There is no weather stream with this name.
E_UNEXPECTED	
E_POINTER	Address provided is invalid.
ERROR_SCENARIO_SIMULATION_RUNNING	Weather stream cannot be removed while the scenario is running.

**(6.10) Scenario\_InsertFilter(BSTR ScenarioID, BSTR FilterID, int index)**

**Description:** Adds the specified filter to the specified scenario.

**Parameters:**

ScenarioID	ID of the scenario to be modified.
FilterID	ID of the filter to be added.
index	Index of where the filter is to be added.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
ERROR_SCENARIO_WEATHERGRID_UNKNOWN	There is no weather grid with this name.
E_FAIL	A general failure occurred.
E_POINTER	Address provided is invalid.

**(6.11) Scenario\_RemoveFilter(BSTR ScenarioID, BSTR FilterID)**

**Description:** Removes the specified filter to the specified scenario.

**Parameters:**

ScenarioID	ID of the scenario to be modified.
FilterID	ID of the filter to be added.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
ERROR_SCENARIO_WEATHERGRID_UNKNOWN	There is no weather grid with this name.

**(6.12) Scenario\_Delete(BSTR ScenarioName)**

**Description:** Deletes the scenario.

**Parameters:**

ScenarioName	Unique identifier for the scenario.
--------------	-------------------------------------

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
E_POINTER	Address provided is invalid.
ERROR_SCENARIO_BAD_STATE	The scenario is in a bad state.

**(6.13) WxStation\_Delete(BSTR StationName)**

**Description:** Deletes a weather station.

**Parameters:**

StationName	Unique identifier for the station.
-------------	------------------------------------

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	There is no scenario with this name.
ERROR_SCENARIO_BAD_STATE	The scenario is in a bad state.
ERROR_WEATHER_STATION_UNKNOWN	No weather station with this name can be found.
ERROR_SCENARIO_SIMULATION_RUNNING	Action cannot be performed while scenario is running.
SUCCESS_SCENARIO_SIMULATION_COMPLETE	Action cannot be performed while scenario is in complete state.

**(6.14) WxStream\_Delete(BSTR StationID, BSTR StreamID)**

**Description:** Deletes a weather stream.

**Parameters:**

StationID	Unique identifier for the station.
StreamID	Unique identifier for the stream.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.

ERROR_WEATHER_STATION_UNKNOWN	No weather station with this name can be found.
ERROR_WEATHER_STREAM_UNKNOWN	No weather stream with this name can be found.
E_UNEXPECTED	Internal error. Please report to the software development team.
E_POINTER	Address provided is invalid.
ERROR_SCENARIO_SIMULATION_RUNNING	The weather stream cannot be delete while a scenario is running.

**(6.15) *WxStream\_AddWeatherDaily*(*BSTR szStationName, BSTR szStreamName, double min\_temp, double max\_temp, double min\_ws, double max\_ws, double rh, double precip, double wd*)**

**Description:** Adds daily weather to the weather stream specified by station name and stream name.

**Parameters:**

szStationName	Unique identifier for the station.
szStreamName	Unique identifier for the stream.
min_temp	Minimum temperature.
max_temp	Maximum temperature.
min_ws	Minimum windspeed.
max_ws	Maximum windspeed.
rh	Relative humidity.
precip	Precipitation.
wd	Wind direction.

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_WEATHER_STATION_UNKNOWN	No weather station with this name can be found.
ERROR_WEATHER_STREAM_UNKNOWN	No weather stream with this name can be found.
E_UNEXPECTED	Internal error. Please report to the software development team.
E_POINTER	Address provided is invalid.
ERROR_SCENARIO_SIMULATION_RUNNING	Cannot add weather while simulation is running.

**(6.16) *Vector\_Delete*(*BSTR VectorName*)**

**Description:** Deletes a vector identified by its name.

**Parameters:**

VectorName	Unique identifier for the vector.
------------	-----------------------------------

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_VECTORENGI VE_UNKNOWN	No object with the provided name.

**(6.17) *Filter\_Delete(BSTR FilterName)***

**Description:** Deletes a filter identified by its name.

**Parameters:**

FilterName	Unique identifier for the filter.
------------	-----------------------------------

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_GRIDFILTER_UNKNOWN	No grid filter by this name exists.

**(6.18) *Filter\_FuelPoly\_Clear(BSTR FilterName)***

**Description:** Removes all polygons associated with this fuel patch.

**Parameters:**

FilterName	Unique identifier for the filter.
------------	-----------------------------------

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_GRIDFILTER_UNKNOWN	No grid filter by this name exists.

**(6.19) *Filter\_WxPoly\_Clear(BSTR FilterName)***

**Description:** Removes all polygons associated with this weather patch.

**Parameters:**

FilterName	Unique identifier for the filter.
------------	-----------------------------------

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_GRIDFILTER_UNKNOWN	No grid filter by this name exists.

## (7) SIMULATE Functions

### (7.1) *Simulate(BSTR ScenarioID)*

**Description:** Simulates the a scenario completely, from start (reset) to finish.

*This replaces the deprecated SimulateActive function.*

**Parameters:**

ScenarioID	Unique identifier for a scenario.
------------	-----------------------------------

**Returns:**

SUCCESS_SCENARIO_SIMULATION_COMPLETE	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_FIRE_SCENARIO_UNKNOWN	ScenarioID does not correspond to a known scenario.
ERROR_SCENARIO_NO_FIRES	The scenario contains no fires.
ERROR_SCENARIO_BAD_TIMES	Start or end time is invalid, or if the internal or output calculation intervals are invalid.
ERROR_SCENARIO_BAD_TIMESTEPS	The scenario's time steps are invalid.
ERROR_GRID_WEATHER_NOT_IMPLEMENTED	No weather data is currently associated with this scenario.
ERROR_GRID_UNINITIALIZED	Some grid information is incomplete, or unspecified, for example if that object doesn't have an initialized latitude, longitude, or time zone.
ERROR_SCENARIO_UNKNOWN	ScenarioID is unknown or invalid.
S_OK	Scenario ready to execute, but has not begun.
ERROR_SCENARIO_BAD_STATE	If this function is run while the scenario is running.
ERROR_GRID_WEATHER_INVALID_DATES	The weather grid has invalid dates.
S_FALSE	False.
E_INVALIDARG	Invalid parameters.
E_FAIL	A general failure occurred.
ERROR_WEATHER_STREAM_NOT_ASSIGNED	The scenario has no weather stream assigned to it.
E_POINTER	Address provided is invalid.
SUCCESS_STATE_OBJECT_UNLOCKED	Internal error. Please report to the software development team.
SUCCESS_STATE_OBJECT_LOCKED_WRITE	Internal error. Please report to the software development team.
SUCCESS_STATE_OBJECT_LOCKED_SCENARIO	Internal error. Please report to the software development team.

SUCCESS_STATE_OBJECT_LOCKED_READ	Internal error. Please report to the software development team.
----------------------------------	---

### (7.2) *SimulateStep(BSTR ScenarioID)*

**Description:** Simulates the fire growth in steps.

*This replaces the depreciated SimulateStepActive function.*

**Parameters:**

ScenarioID	Unique identifier for a scenario.
StepResult	0: operation completed successfully -1: operation failed

**Returns:**

SUCCESS_SCENARIO_SIMULATION_COMPLETE	Successful.
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	ScenarioID does not correspond to a known scenario.
ERROR_SCENARIO_NO_FIRES	The scenario contains no fires.
ERROR_SCENARIO_BAD_TIMES	Start or end time is invalid, or if the internal or output calculation intervals are invalid.
ERROR_SCENARIO_BAD_TIMESTEPS	The time steps are invalid.
ERROR_GRID_WEATHER_NOT_IMPLEMENTED	No weather data is currently associated with this scenario.
ERROR_GRID_UNINITIALIZED	Some grid information is incomplete, or unspecified, for example if that object doesn't have an initialized latitude, longitude, or time zone.
ERROR_SCENARIO_BAD_STATE	Scenario is not running.
SUCCESS_SCENARIO_SIMULATION_COMPLETE	The scenario has successfully simulated.

### (7.3) *SimulateStop(BSTR ScenarioID)*

**Description:** Stops the simulation that is running.

**Parameters:**

ScenarioID	Unique identifier for a scenario.
------------	-----------------------------------

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.

E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	ScenarioID does not correspond to a known scenario.
ERROR_SCENARIO_BAD_STATE	If the simulation wasn't in a reset, running, or completed state.

#### (7.4) *SimulateRestart(BSTR ScenarioID)*

**Description:** Restarts the simulation. This operation must take place before any calls to SimulateStep().

*This replaces the deprecated SimulateRestartActive function.*

**Parameters:**

ScenarioID	Unique identifier for a scenario.
------------	-----------------------------------

**Returns:**

S_OK	Successful.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	ScenarioID does not correspond to a known scenario.
ERROR_SCENARIO_NO_FIRES	The scenario contains no fires.
ERROR_SCENARIO_BAD_TIMES	Start or end time is invalid, or if the internal or output calculation intervals are invalid.
ERROR_SCENARIO_BAD_TIMESTEPS	The time steps are invalid.
ERROR_GRID_WEATHER_NOT_IMPLEMENTED	No weather data is currently associated with this scenario.
ERROR_GRID_UNINITIALIZED	Some grid information is incomplete, or unspecified, for example if that object doesn't have an initialized latitude, longitude, or time zone.
ERROR_SCENARIO_BAD_STATE	If this function is run while the scenario is running.
ERROR_GRID_WEATHER_INVALID_DATES	The weather grid has invalid dates.
S_FALSE	False.
E_INVALIDARG	Invalid parameters.
E_FAIL	A general failure occurred.
ERROR_WEATHER_STREAM_NOT_ASSIGNED	The scenario has no weather stream assigned to it.
E_POINTER	Address provided is invalid.
SUCCESS_STATE_OBJECT_UNLOCKED	Internal error. Please report to the software development team.
SUCCESS_STATE_OBJECT_LOCKED_WRITE	Internal error. Please report to the software development team.

SUCCESS_STATE_OBJECT_LOCKED_SCENARIO	Internal error. Please report to the software development team.
SUCCESS_STATE_OBJECT_LOCKED_READ	Internal error. Please report to the software development team.
ERROR_GRID_PRIMARY_STREAM_UNSPECIFIED	The scenario has no primary stream assigned to it.
ERROR_GRID_WEATHERSTATIONS_TOO_CLOSE	The weather stations in the scenario are too close together.
ERROR_GRID_WEATHERSTREAM_TIME_OVERLAPS	At least two streams from the station have overlapping times.
ERROR_GRID_LOCATION_OUT_OF_RANGE	The location is off the grid.

### (7.5) *SimulateThreeSteps(BSTR ScenarioID, int \*StepResult)*

**Description:** Simulates the fire growth three steps forward. According to the value of StepResult, a decision can be made about whether the simulate operation should be stopped.

It is equivalent to calling SimulateStep() three times.

*This replaces the deprecated SimulateThreeStepsActive function.*

This functionality is similar to that provided by the function SimulateStep().

**Parameters:**

ScenarioID	Unique identifier for a scenario.
StepResult	0: operation completed successfully -1: operation failed

**Returns:**

S_OK	Successful.
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error; or if no weather streams are attached to this scenario.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	ScenarioID does not correspond to a known scenario.
ERROR_SCENARIO_BAD_STATE	Scenario is not running.
SUCCESS_SCENARIO_SIMULATION_COMPLETE	This step completes simulation, i.e. the “current time” has reached the end time.
ERROR_GRID_UNINITIALIZED	No ICWFGM_GridEngine object has been specified, or that object doesn’t have an initialized latitude, longitude, or time zone.
E_OUTOFMEMORY	Insufficient memory.

**(7.6) *SimulateEightSteps(BSTR ScenarioID, int \*StepResult)***

**Description:** Simulates the fire growth eight steps forward. According to the value of StepResult, a decision can be made about whether the simulate operation should be stopped.

It is equivalent to calling SimulateStep() eight times.

*This replaces the deprecated SimulateEightStepsActive function.*

This functionality is similar to that provided by the function SimulateStep().

**Parameters:**

ScenarioID	Unique identifier for a scenario.
StepResult	0: operation completed successfully -1: operation failed

**Returns:**

S_OK	Successful.
E_POINTER	Address provided is invalid.
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error; or if no weather streams are attached to this scenario.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_UNKNOWN	ScenarioID does not correspond to a known scenario.
ERROR_SCENARIO_BAD_STATE	Scenario is not running.
SUCCESS_SCENARIO_SIMULATION_COMPLETE	This step completes simulation, i.e. the “current time” has reached the end time.
ERROR_GRID_UNINITIALIZED	No ICWFGM_GridEngine object has been specified, or that object doesn’t have an initialized latitude, longitude, or time zone.
E_OUTOFMEMORY	Insufficient memory.

## (8) EXPORT Functions

### (8.1) *ExportFuelsLookup(BSTR fileName)*

**Description:** Exports the fuel lookup table (LUT) being used in the COM model. This file is an ASCII text file and is compatible with Prometheus.

**Parameters:**

<code>filename</code>	Character string identifying the name and path of the export file.
-----------------------	--

**Returns:**

<code>S_OK</code>	Successful
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error.
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete the operation.

### (8.2) *ExportScenarioFireFront(BSTR ScenarioName, BSTR FileName, int only\_active, int only\_exterior, int combine\_all)*

**Description:** Exports the fire front of the specified scenario. The scenario must contain at least one step's simulation data, otherwise this function will fail.

**Parameters:**

<code>ScenarioName</code>	Unique identifier for a scenario.
<code>FileName</code>	Character string identifying the name and path of the export file.
<code>only_active</code>	Option to export only active parts of the fire front.
<code>only_exterior</code>	Option to export only exterior parts of the fire front.
<code>combine_all</code>	Option to export all parts of the fire front.

**Returns:**

<code>S_OK</code>	Successful
<code>ERROR_SEVERITY_WARNING</code>	Failed initialization or a non-specific I/O error; or for an undetermined error in exporting
<code>E_OUTOFMEMORY</code>	There is not enough memory to complete the operation.
<code>ERROR_SCENARIO_UNKNOWN</code>	ScenarioName does not resolve to a known scenario.
<code>E_NOTIMPL</code>	Method has not been implemented yet.
<code>E_POINTER</code>	Address provided is invalid.
<code>ERROR_FIRE_INVALID_TIME</code>	The time specified is invalid.
<code>E_INVALIDARG</code>	Invalid parameters.
<code>ERROR_NO_DATA</code>	There is no data.

E_FAIL	A general failure occurred.
ERROR_SCENARIO_BAD_STATE	The scenario is in a bad state.

**(8.3) *ExportFuelGrid(BSTR projFileName, BSTR filename)***

**Description:** Exports the project's default (original) fuel grid along with the corresponding projection file.

**Parameters:**

filename	Character string identifying the name and path of the export file.
projFileName	String identifying the name and path of the projection file.

**Returns:**

S_OK	Successful
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_ACCESS_DENIED	Access denied.
ERROR_INVALID_HANDLE	The handle has been invalidated, or closed.
ERROR_FILE_EXISTS	The file exists.
ERROR_INVALID_PARAMETER	The parameter is incorrect.
ERROR_TOO_MANY_OPEN_FILES	System error; current application has too many files open.
ERROR_FILE_NOT_FOUND	The specified file could not be found.
ERROR_HANDLE_DISK_FULL	The disk is full.
E_POINTER	Address provided is invalid.
E_INVALIDARG	Invalid parameters.

**(8.4) *ExportElevationSlopeAspect(BSTR projFileName, BSTR elevationFile, BSTR slopeFile, BSTR aspectFile)***

**Description:** Exports the project elevation, slope, or aspect file and the corresponding projection file.

**Parameters:**

projFileName	String identifying the name and path of the projection file.
elevationFile	String identifying the name and path of elevation grid ASCII file.
slopeFile	String identifying the name and path of the slope grid ASCII file.
aspectFile	String identifying the name and path of the aspect grid ASCII file.

**Returns:**

S_OK	Successful
------	------------

ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_ACCESS_DENIED	Access denied.
ERROR_INVALID_HANDLE	The handle has been invalidated, or closed.
ERROR_FILE_EXISTS	The file exists.
ERROR_INVALID_PARAMETER	The parameter is incorrect.
ERROR_TOO_MANY_OPEN_FILES	System error; current application has too many files open.
ERROR_FILE_NOT_FOUND	The specified file could not be found.
ERROR_HANDLE_DISK_FULL	The disk is full.
E_POINTER	Address provided is invalid.
E_INVALIDARG	Invalid parameters.
ERROR_GRID_UNINITIALIZED	The grid is not initialized properly.

**(8.5) Ignition\_Export(BSTR IgnitionName, BSTR fileName)**

**Description:** Exports an ignition object identified by its name to the specified filename.

**Parameters:**

IgnitionName	Uniquely identifies the ignition.
fileName	The path and filename to export to.

**Returns:**

S_OK	Successful
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_SCENARIO_FIRE_UNKNOWN	There is no ignition with this name.
E_POINTER	Address provided is invalid.
E_INVALIDARG	Invalid parameters.

**(8.6) WxStream\_Export(BSTR filename, BSTR StationName, BSTR StreamName, int NeedHead)**

**Description:** Exports a weather stream identified by station and stream name with a possible header to the filename directed.

**Parameters:**

StationName	Uniquely identifies the weather station.
fileName	The path and filename to export to.
StreamName	Uniquely identifies the weather stream.
NeedHead	Whether or not to include header information.

**Returns:**

S_OK	Successful
------	------------

<a href="#">ERROR_SEVERITY_WARNING</a>	Failed initialization or a non-specific I/O error.
<a href="#">E_OUTOFMEMORY</a>	There is not enough memory to complete the operation.
<a href="#">ERROR_WEATHER_STATION_UNKNOW N</a>	There is no weather station with this name.
<a href="#">ERROR_WEATHER_STREAM_UNKNOW N</a>	There is no weather stream with this name.
<a href="#">ERROR_ACCESS_DENIED</a>	Access denied.
<a href="#">ERROR_INVALID_HANDLE</a>	The handle has been invalidated, or closed.
<a href="#">ERROR_FILE_EXISTS</a>	The file exists.
<a href="#">ERROR_INVALID_PARAMETER</a>	The parameter is incorrect.
<a href="#">ERROR_TOO_MANY_OPEN_FILES</a>	System error; current application has too many files open.
<a href="#">ERROR_FILE_NOT_FOUND</a>	The specified file could not be found.
<a href="#">ERROR_HANDLE_DISK_FULL</a>	The disk is full.

**(8.7) *Vector\_Export(BSTR filename, BSTR VectorName)***

**Description:** Exports a vector object identified by name to the filename directed.

**Parameters:**

<a href="#">fileName</a>	The path and filename to export to.
<a href="#">VectorName</a>	Uniquely identifies the vector.

**Returns:**

<a href="#">S_OK</a>	Successful
<a href="#">ERROR_SEVERITY_WARNING</a>	Failed initialization or a non-specific I/O error.
<a href="#">E_OUTOFMEMORY</a>	There is not enough memory to complete the operation.
<a href="#">ERROR_SCENARIO_VECTORENGINE _UNKNOWN</a>	
<a href="#">E_FAIL</a>	A general failure occurred.
<a href="#">E_POINTER</a>	Address provided is invalid.
<a href="#">S_FALSE</a>	False.
<a href="#">E_INVALIDARG</a>	Invalid parameters.
<a href="#">ERROR_NO_DATA</a>	There is no data.

**(8.8) *Filter\_FuelGrid\_Export(BSTR FilterName, BSTR FileName)***

**Description:** Exports a fuel grid filter identified by name to the filename directed.

**Parameters:**

<a href="#">FileName</a>	The path and filename to export to.
<a href="#">FilterName</a>	Uniquely identifies the filter.

**Returns:**

<a href="#">S_OK</a>	Successful
<a href="#">ERROR_SEVERITY_WARNING</a>	Failed initialization or a non-specific I/O

	error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.

**(8.9) *Filter\_Greenup\_Export(BSTR FilterName, BSTR FileName)***

**Description:** Exports a green-up grid filter identified by name to the filename directed.

**Parameters:**

FileName	The path and filename to export to.
FilterName	Uniquely identifies the filter.

**Returns:**

S_OK	Successful
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.

**(8.10) *Filter\_FuelPoly\_Export(BSTR FilterName, BSTR FileName)***

**Description:** Exports a fuel patch identified by name to the filename directed.

**Parameters:**

FileName	The path and filename to export to.
FilterName	Uniquely identifies the filter.

**Returns:**

S_OK	Successful
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_GRIDFILTER_UNKNOWNN	There is no filter with that name.

**(8.11) *Filter\_WxPoly\_Export(BSTR FilterName, BSTR FileName)***

**Description:** Exports a weather patch identified by name to the filename directed.

**Parameters:**

FileName	The path and filename to export to.
FilterName	Uniquely identifies the filter.

**Returns:**

S_OK	Successful
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.
ERROR_GRIDFILTER_UNKNOWNN	There is no filter with that name.

**(8.12) *Filter\_WxLandscape\_Export(BSTR FilterName, BSTR FileName)***

**Description:** Exports a weather grid filter identified by name to the filename directed.

**Parameters:**

FileName	The path and filename to export to.
FilterName	Uniquely identifies the filter.

**Returns:**

S_OK	Successful
ERROR_SEVERITY_WARNING	Failed initialization or a non-specific I/O error.
E_OUTOFMEMORY	There is not enough memory to complete the operation.

## Appendix II: Prometheus Error Codes

---

```
#define ERROR_GRID_INITIALIZED (12001 | ERROR_SEVERITY_WARNING))
#define ERROR_GRID_LOCATION_OUT_OF_RANGE (12002 | ERROR_SEVERITY_WARNING))
#define ERROR_GRID_TIME_OUT_OF_RANGE (12003 | ERROR_SEVERITY_WARNING))
#define SUCCESS_GRID_DATA_UPDATED (12004)
#define ERROR_GRID_WEATHER_NOT_IMPLEMENTED (12005 | ERROR_SEVERITY_WARNING))
#define ERROR_GRID_WEATHER_INVALID_DATES (12006 | ERROR_SEVERITY_WARNING))
#define SUCCESS_GRID_TOTAL_AREA (12007)
#define SUCCESS_RETURNING_DEFAULTS (12008)
#define SUCCESS_RETURNING_CALCLED_SLOPE (12009)
#define ERROR_GRID_WEATHER_NO_DATA (12010 | ERROR_SEVERITY_WARNING))
#define SUCCESS_GRID_IMPORT_CONTAINED_ONLY_NODATA (12011)
#define SUCCESS_GRID_IMPORT_CONTAINED_NODATA (12012)
#define ERROR_GRID_NO_DATA (12013 | ERROR_SEVERITY_WARNING))
#define ERROR_FIREBREAK_NOT_FOUND (12050 | ERROR_SEVERITY_WARNING))
#define SUCCESS_FUELS_FUEL_ALREADY_ADDED (12500)
#define ERROR_FUELS_FILEINDEX_ALREADY_ASSIGNED (12501 | ERROR_SEVERITY_WARNING))
#define ERROR_FUELS_FUEL_UNKNOWN (12502 | ERROR_SEVERITY_WARNING))
#define ERROR_FUELS_FUEL_KNOWN (12503 | ERROR_SEVERITY_WARNING))
#define ERROR_FIRE_IGNITION_TYPE_UNKNOWN (12100 | ERROR_SEVERITY_WARNING))
#define SUCCESS_FIRE_NOT_STARTED (12102)
#define SUCCESS_FIRE_BURNED_OUT (12103)
#define ERROR_FIRE_INVALID_TIME (12104 | ERROR_SEVERITY_WARNING))
#define ERROR_FIRE_STAT_UNKNOWN (12105 | ERROR_SEVERITY_WARNING))
#define SUCCESS_FIRE_NO_HISTORY (12106)
#define ERROR_FIRE_OUT_OF_RANGE (12107 | ERROR_SEVERITY_WARNING))
#define ERROR_SCENARIO_FIRE_ALREADY_ADDED (12200 | ERROR_SEVERITY_WARNING))
#define ERROR_SCENARIO_FIRE_UNKNOWN (12201 | ERROR_SEVERITY_WARNING))
#define ERROR_SCENARIO_NO_FIRES (12202 | ERROR_SEVERITY_WARNING))
#define ERROR_SCENARIO_OPTION_INVALID (12203 | ERROR_SEVERITY_WARNING))
#define ERROR_SCENARIO_BAD_TIMES (12204 | ERROR_SEVERITY_WARNING))
#define ERROR_SCENARIO_BAD_TIMESTEPS (12212 | ERROR_SEVERITY_WARNING))
#define ERROR_SCENARIO_BAD_STATE (12205 | ERROR_SEVERITY_WARNING))
#define SUCCESS_SCENARIO_SIMULATION_RESET (12206)
#define SUCCESS_SCENARIO_SIMULATION_RUNNING (12207)
#define ERROR_SCENARIO_SIMULATION_RUNNING (SUCCESS_SCENARIO_SIMULATION_RUNNING
| ERROR_SEVERITY_WARNING)
#define SUCCESS_SCENARIO_SIMULATION_COMPLETE (12208)
#define ERROR_SCENARIO_VECTORENGINE_ALREADY_ADDED (12209 | ERROR_SEVERITY_WARNING))
#define ERROR_SCENARIO_VECTORENGINE_UNKNOWN (12210 | ERROR_SEVERITY_WARNING))
#define ERROR_SCENARIO_VECTORENGINE_KNOWN (12211 | ERROR_SEVERITY_WARNING))
#define ERROR_SCENARIO_WEATHERGRID_UNKNOWN (12212 | ERROR_SEVERITY_WARNING))
#define ERROR_SCENARIO_WEATHERGRID_ALREADY_ADDED (12213 | ERROR_SEVERITY_WARNING))
#define ERROR_SCENARIO_UNKNOWN (12214 | ERROR_SEVERITY_WARNING))
#define ERROR_WEATHER_STREAM_ALREADY_ADDED (12300 | ERROR_SEVERITY_WARNING))
#define ERROR_WEATHER_STREAM_UNKNOWN (12301 | ERROR_SEVERITY_WARNING))
#define ERROR_WEATHER_STREAM_NOT_ASSIGNED (12302 | ERROR_SEVERITY_WARNING))
#define ERROR_WEATHER_STREAM_ALREADY_ASSIGNED (12303 | ERROR_SEVERITY_WARNING))
#define ERROR_WEATHER_STATION_ALREADY_PRESENT (12304 | ERROR_SEVERITY_WARNING))
#define ERROR_WEATHER_OPTION_INVALID (12305 | ERROR_SEVERITY_WARNING))
#define ERROR_WEATHER_STATION_UNKNOWN (12306 | ERROR_SEVERITY_WARNING))
#define ERROR_POINT_NOT_IN_FIRE (12400 | ERROR_SEVERITY_WARNING))
#define ERROR_FILE_FORMAT_INVALID (12500 | ERROR_SEVERITY_WARNING))
#define SUCCESS_STATE_OBJECT_UNLOCKED (12600)
#define SUCCESS_STATE_OBJECT_LOCKED_WRITE (12601)
#define SUCCESS_STATE_OBJECT_LOCKED_READ (12602)
#define SUCCESS_STATE_OBJECT_LOCKED_SCENARIO (12603)
#define ERROR_GRIDFILTER_UNKNOWN (12700 | ERROR_SEVERITY_WARNING))
```