NEPLAN®
Smarter Tools

NEPLAN Web-Services

NEPLAN AG Oberwachtstrasse 2 CH 8700 Küsnacht ZH www.neplan.ch
Phone +41 44 914 36 66 Fax +41 44 991 19 71

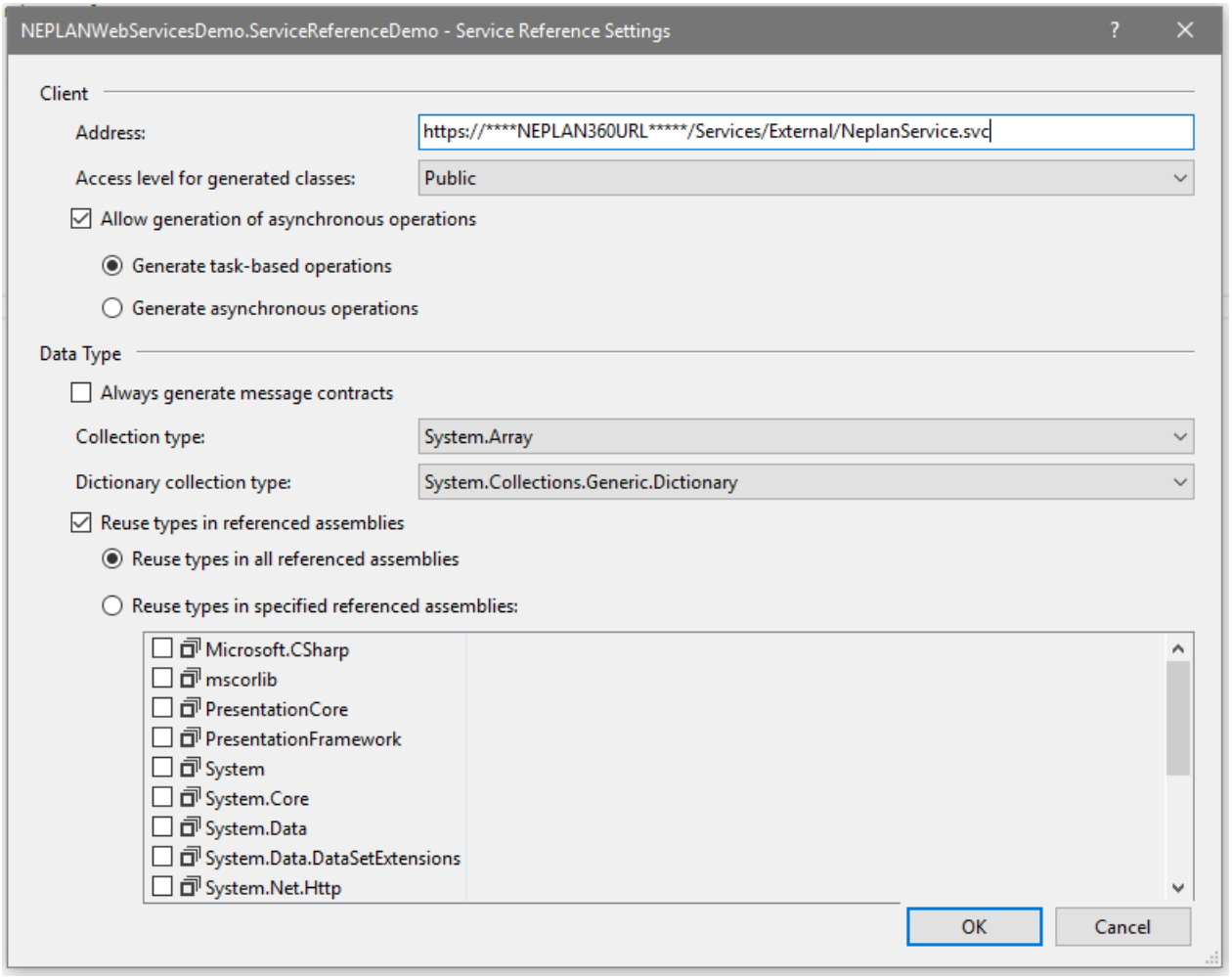
1) Requirements for Web-Services

The **https** protocol needs to be activated on the server

Download the file WPFWebServiceDemo and open it using Microsoft VisualStudio.

2) Configure the Web-Service in the WPFWebServiceDemo

Go to **Service References** in the Solution Explorer. Make a Right Click in NeplanService and click on **Configure Service References**.
Change the Address ********NEPLAN360URL******** with correct URL of NEPLAN360.



Addapt the App.config File to your needs. The following settings are available.

3) NEPLAN Web-Services

The NEPLAN database is accessible through web-services provided by the web version of NEPLAN. In this case external systems, like a GIS, needn't to know the NEPLAN database structure. All elements or assets in a network must have a unique ID, which are will be returned by the NEPLAN service.

Often the external system, like GIS, does not provide all electrical parameters, which are necessary to make the calculations. The external system must provide the library name and the library type, which is stored in the NEPLAN library, in order to select all the electrical, gas or water parameters for a network component.

A set of web-services are explained as follows:

Summary :
This function creates a new project

Parameter : "projectName" Name of project
Parameter : "variantName" Name of variant
Parameter : "diagram" Name of diagram
Parameter : "layerName" Name of graphic layer
Parameter : "mediumType" Type of Medium, for electrical networks: "Power", "Gas", "Water"
Parameter : "srid" Default value 0
Parameter : "copySettingsFromProjectName" The name of the project used as template for the settings
Returns : The project created as ExternalProject

ExternalProject CreateProject(string projectName, string variantName, string diagram, string layerName, string mediumType, int srid, string copySettingsFromProjectName);

Summary :
Gets a project based on the name

Parameter : "projectName" Name of project
Parameter : "variantName" Name of variant
Parameter : "diagramName" Name of diagram
Parameter : "layerName" Name of graphic layer
Returns : The project as ExternalProject

ExternalProject GetProject(string projectName, string variantName, string diagramName, string layerName);

Summary :
Get an url that the user can use to directly log in to neplan. Opens directly
the given project

Parameter : "project" The project to open can be null
Returns : Direct connection URL to NEPLAN 360 of the project to open

string GetLogOnUrlWithProject(ExternalProject project);

Summary :
Get an url that the user can use to directly log in to neplan.

Returns : Direct connection URL to NEPLAN 360

string GetLogOnUrl();

Summary :
Get the session id for login to neplan. Add the session id to the
base url of Neplan360

Parameter : "project" The project to open can be null
Returns : Session ID

string GetLogOnSessionID(ExternalProject project);

Summary :
Inserts a node in the specified project.

Parameter : "project" Summary Object of the internal project
Parameter : "name" Name of node. Must be unique
Parameter : "un" Nominal system Voltage in kV of the node
Parameter : "fn" Nominal system Frequency in Hz of the node
Parameter : "libraryType" Name of the Element in the NEPLAN library to be used
Parameter : "xmlData" Specific XML technical data, empty by default
Parameter : "x" Coordinate X of the first point
Parameter : "y" Coordinate Y of the first point
Parameter : "symbolName" Name of the Symbol to be used
Returns : The Guid ID of the new Node as String

string InsertNode(ref ExternalProject project, string name, double un, double fn, string libraryType, string xmlData, double x, double y, string symbolName);

Summary :
Inserts a node without graphic in the specified project

Parameter : "project" Summary Object of the internal project
Parameter : "name" >Name of node. Must be unique
Parameter : "un" Nominal system Voltage in kV of the node
Parameter : "fn" Nominal system Frequency in Hz of the node
Parameter : "libraryType" Name of the Element in the NEPLAN library to be used
Parameter : "xmlData" Specific XML technical data, empty by default
Returns : The Guid ID of the new Node as String

string InsertNodeWithoutGraphic(ref ExternalProject project, string name, double un, double fn, string libraryType, string xmlData);

Summary :
Inserts a multipoint busbar with it's coordinates in the specified project

Parameter : "project" Summary Object of the internal project
Parameter : "name" >Name of node. Must be unique
Parameter : "un" Nominal system Voltage in kV of the node
Parameter : "fn" Nominal system Frequency in Hz of the node
Parameter : "libraryType" Name of the Element in the NEPLAN library to be used
Parameter : "xmlData" Specific XML technical data, empty by default
Parameter : "coordinates" List of the coordinates of the busbar
Returns : The Guid ID of the new Node as String

string InsertBusbar(ref ExternalProject project, string name, double un, double fn, string libraryType, string xmlData, List<double> coordinates);

Summary :
Inserts a multipoint line with it's coordinates in the specified project
the user has to specify the names of the nodes

Parameter : "project" Summary Object of the internal project
Parameter : "name" >Name of Line. Must be unique
Parameter : "fromNode" Unique Name of the "from" node
Parameter : "toNode" Unique Name of the "to" node
Parameter : "length" Length in km
Parameter : "libraryType" Name of the Element in the NEPLAN library to be used
Parameter : "xmlData" Specific XML technical data, empty by default
Parameter : "coordinates" List of the coordinates of the line
Returns : The Guid ID of the new Line as String

string InsertLine(ExternalProject project, string name, string fromNode, string toNode, double length, string libraryType, string xmlData, List<double> coordinates);

Summary :
Inserts a multipoint line with it's coordinates in the specified project
the user has to specify the nodes ID

Parameter : "project" Summary Object of the internal project
Parameter : "name" >Name of Line. Must be unique
Parameter : "fromNode" ID of the "from" node
Parameter : "toNode" ID of the "to" node
Parameter : "length" Length in km
Parameter : "libraryType" Name of the Element in the NEPLAN library to be used
Parameter : "xmlData" Specific XML technical data, empty by default
Parameter : "coordinates" List of the coordinates of the line
Returns : The Guid ID of the new Line as String

string InsertLineByNodeID(ExternalProject project, string name, string fromNodeID, string toNodeID, double length, string libraryType, string xmlData, List<double> coordinates);

Summary :
Inserts a multipoint asymetrical line with it's coordinates in the specified project

the user has to specify the names of the node

Parameter : "project" Summary Object of the internal project
Parameter : "name" >Name of Line. Must be unique
Parameter : "fromNode" Name of the "from" node
Parameter : "toNode" Name of the "to" node
Parameter : "length" Length in km
Parameter : "libraryType" Name of the Element in the NEPLAN library to be used
Parameter : "xmlData" Specific XML technical data, empty by default
Parameter : "coordinates" List of the coordinates of the line
Returns : The Guid ID of the new Line as String

string InsertLineAsym(ExternalProject project, string name, string fromNode, string toNode, double length, short phase, string libraryType, string xmlData, List<double> coordinates);

Summary :
Inserts a 2 port element (for ex: Transfo, Reactor), with it's coordinates in the specified project
the user has to specify the names of the node

Parameter : "project" Summary Object of the internal project
Parameter : "name" >Name of two port Element. Must be unique
Parameter : "fromNode" Name of the "from" node
Parameter : "toNode" Name of the "to" node
Parameter : "elementTypeName" Type of element, See the Appendix of the WebService Manual
Parameter : "libraryType" Name of the Element in the NEPLAN library to be used
Parameter : "xmlData" Specific XML technical data, empty by default
Parameter : "phase" Phase connection
Parameter : "coordinatesLink1" List of coordinates to describe the link1
Parameter : "coordinatesLink2" List of coordinates to describe the link2
Parameter : "symbolName" Name of the symbol to be used
Parameter : "symbolAngle" Angle of the symbol
Returns : The Guid ID of the new 2 port Element as String

string Insert2PortElement(ExternalProject project, string name, string fromNode, string toNode, string elementTypeName, string libraryType, string xmlData, int phase, List<double> coordinatesLink1, List<double> coordinatesLink2, string symbolName, double symbolAngle);

Summary :
Inserts a 1 port element (for ex: Load, Shund), with it's coordinates in the specified project
the user has to specify the name of the node where it is connected

Parameter : "project" Summary Object of the internal project
Parameter : "name" Name of One port Element. Must be unique
Parameter : "fromNode" Name of the "from" node
Parameter : "elementTypeName" Type of element, See the Appendix of the WebService Manual
Parameter : "libraryType" Name of the element in the NEPLAN library to be used
Parameter : "xmlData" Specific XML technical data, empty by default
Parameter : "phase" Phase connection
Parameter : "coordinatesLink" List of coordinates to describe the link

Parameter : "symbolName" Name of the symbol to be used
Parameter : "symbolAngle" Angle of the symbol
Returns : The Guid ID of the new 1 port Element as String

string Insert1PortElement(ref ExternalProject project, string name, string fromNode, string elementTypeName, string libraryType, string xmlData, int phase, List<double> coordinatesLink, string symbolName, double symbolAngle);

Summary :
Inserts a protection element in the specified project
the user has to specify the element ID where the protection is placed

Parameter : "project" Summary Object of the internal project
Parameter : "name" Name of Protection. Must be unique
Parameter : "onElementID" ID of the element where the protection is placed
Parameter : "atPort" Port of the element where the protection is placed
Parameter : "elementTypeName" Type of element, See the Appendix of the WebService Manual
Parameter : "libraryType" Name of the element in the NEPLAN library to be used
Parameter : "xmlData" Specific XML technical data, empty by default
Parameter : "phase" Phase connection
Parameter : "symbolName" Name of the symbol to be used
Returns : The Guid ID of the new Protection Element as String

string InsertProtectionElement(ExternalProject project, string name, string onElementID, short atPort, string elementTypeName, string libraryType, string xmlData, int phase, string symbolName);

Summary :
Inserts a feeder in the specified project

Parameter : "project" Summary Object of the internal project
Parameter : "feederName" Name of the feeder. Must be unique
Parameter : "nodeName" Name of the node where the feeder is connected
Parameter : "elementName" Name of the main element where the feeder is connected
Returns : The Guid ID of the feeder as String

string InsertFeeder(ExternalProject project, string feederName, string nodeName, string elementName);

Summary :
Inserts an area in the specified project

Parameter : "project" Summary Object of the internal project
Parameter : "areaName" Name of the Area. Must be unique
Parameter : "firstSubArea" Name of the subarea of the area
Parameter : "colorStringArea" Color of area. For e.g. "#FF207812"
Parameter : "colorStringSubArea" Color of subarea. For e.g. "#FF207812"
Returns : The Guid ID of the area as String

string InsertArea(ExternalProject project, string areaName, string firstSubArea, string colorStringArea, string colorStringSubArea);

Summary :
Inserts a zone in the specified project

Parameter : "project" Summary Object of the internal project
Parameter : "name" Name of the Zone. Must be unique
Parameter : "colorString" Color of zone. For e.g. "#FF207812"
Returns : The Guid ID of the zone as String

string InsertZone(ExternalProject project, string name, string colorString);

Summary :
Inserts a measurement on a element with its name in the specified project

Parameter : "project" Summary Object of the internal project
Parameter : "calcName" Name must be "Default"
Parameter : "elementName" Name of measurement Element. Must be unique
Parameter : "portNr" Port number (values: 0 for port 1, 1 for port 2, 2 for port 3, 3 for port 4)
Parameter : "dateTime" Time of the measurement
Parameter : "inputDataDefinition" 0: P and Q values are given [MW/Mvar], 1: P and Q values are given [kW/kvar], 2: I values are given in [A]
Parameter : "measuremens" List of measurement versus time
Parameter : "nameAliasName1AliasName2" Name of the Element where the measurement is defined by 0: Name, 1: Alias1, 2: Alias2
Returns : The Guid ID of the measurement device as String

string InsertMeasurement(ExternalProject project, string calcName, string elementName, short portNr, DateTime dateTime, int inputDataDefinition, List<double> measuremens, short nameAliasName1AliasName2);

Summary :
Inserts a measurement on a element with its ID in the specified project

Parameter : "elementID" ID of the Element
Parameter : "calcName" Name must be "Default"
Parameter : "portNr" Port number (values: 0 for port 1, 1 for port 2, 2 for port 3, 3 for port 4)
Parameter : "dateTime" Time of the measurement
Parameter : "inputDataDefinition" 0: P and Q values are given [MW/Mvar], 1: P and Q values are given [kW/kvar], 2: I values are given in [A]
Parameter : "measurements" List of measurement versus time
Returns : The Guid ID of the measurement device as String

string InsertMeasurementByElementID(string elementID, string calcName, short portNr, DateTime dateTime, int inputDataDefinition, List<double> measurements);

Summary :
Insert a new diagram in the specified project

Parameter : "project" Summary Object of the internal project

Parameter : "diagramName" Name of diagram
Parameter : "firstGraphicLayerName" Name of first Graphic layer
Returns : The Guid ID of the diagram as String

string InsertDiagram(ExternalProject project, string diagramName, string firstGraphicLayerName);

Summary :
Insert a new graphic layer in the diagram ID in the specified project

Parameter : "project" Summary Object of the internal project
Parameter : "diagramID" The ID of the diagram where to insert the graphic layer
Parameter : "graphicLayerName" The name of the graphic layer
Returns : The Guid ID of the graphic as String

string InsertGraphicLayer(ref ExternalProject project, string diagramID, string graphicLayerName);

Summary :
Inserts a subload inside a load

Parameter : "project" Summary Object of the internal project
Parameter : "loadID" The ID of the load to inseart a subload
Parameter : "loadData" The data of the load as LoadData
Returns : TRUE if successful

bool AddSubLoad(ExternalProject project, string loadID, LoadData loadData);

Summary :
Assigns the power to a 1 port element found by name.

Parameter : "project" Summary Object of the internal project
Parameter : "name" Name of 1 port element
Parameter : "elementTypeName" Type of element, See the Appendix of the WebService Manual
Parameter : "powerType" If the element is : a load 0: P Q, 1: P Cos(Phi), 2: I Cos(Phi), 3: P I, 4: S Cos(Phi), 5: E Cos(Phi)
Synchronous Machine or external grid: 0: Poper Qoper, 1: POper Uset, 2: Uset AngleSet, 3: POper CosOper
Asynchronous Machine: 0: POper, QOper, 1: POper, Cos(PhiOper)
Parameter : "value1" Value 1 to be inserted (See Power Type)
Parameter : "value2" Value 2 to be inserted (See Power Type)
Returns : TRUE if successful

bool AssignPowerTo1Port(ExternalProject project, string name, string elementTypeName, short powerType, double value1, double value2);

Summary :
Assigns the power to a 1 port element found by ID.

Parameter : "project" Summary Object of the internal project
Parameter : "elementID" ID of the 1 port element

Parameter : "powerType" if the element is : a load 0: P Q, 1: P Cos(Phi), 2: I Cos(Phi), 3: P I, 4: S Cos(Phi), 5: E Cos(Phi)
Synchronous Machine or external grid: 0: Poper Qoper, 1: POper Uset, 2: Uset AngleSet, 3: POper CosOper
Asynchronous Machine: 0: POper, QOper, 1: POper, Cos(PhiOper)
Parameter : "value1" Value 1 to be inserted (See Power Type)
Parameter : "value2" Value 2 to be inserted (See Power Type)
Returns : TRUE if successful

bool AssignPowerTo1PortByID(ExternalProject project, string elementID, short powerType, double value1, double value2);

Summary :
Switches off/on the port of the element found by its ID

Parameter : "elementID" ID of the element to be switched on/off
Parameter : "portNr" Port number (values: 0 for port 1, 1 for port 2, 2 for port 3, 3 for port 4)
Parameter : "isConnected" TRUE if it is connected, FALSE if it is disconnected

void SwitchOffElementAtPortByID(string elementID, short portNr, bool isConnected);

Summary :
Switches off/on the port of the element found by its name

Parameter : "project" Summary Object of the internal project
Parameter : "elementName" Name of the element to be switched on/off
Parameter : "elementTypeName" Type of element, See the Appendix of the WebService Manual
Parameter : "portNr" Port number (values: 0 for port 1, 1 for port 2, 2 for port 3, 3 for port 4)
Parameter : "isConnected" TRUE if it is connected, FALSE if it is disconnected

void SwitchOffElementAtPort(ExternalProject project, string elementName, string elementTypeName, short portNr, bool isConnected);

Summary :
Sets the general data of the element found by its name.
NOTE: The Zone ID and The Area ID must be created before.

Parameter : "project" Summary Object of the internal project
Parameter : "elementName" Name of the element
Parameter : "elementTypeName" Type of element, See the Appendix of the WebService Manual
Parameter : "aliasName1" Alias1 of the element
Parameter : "aliasName2" Alias2 of the element
Parameter : "description" Description of the element
Parameter : "isInMaintenance" TRUE if the element is in maintenance
Parameter : "IsProjected" TRUE if the element is projected
Parameter : "zoneGuid" Guid ID of the Zone
Parameter : "subAreaGuid" Guid ID of the subarea

void SetElementData(ExternalProject project, string elementName, string elementTypeName, string aliasName1, string aliasName2, string description, bool isInMaintenance, bool IsProjected, string zoneGuid, string subAreaGuid);

Summary :
Sets the general data of the element found by its ID.
NOTE: The Zone ID and The Area ID must be created before.

Parameter : "elementID" ID of the element
Parameter : "elementTypeName" Type of element, See the Appendix of the WebService Manual
Parameter : "aliasName1" Alias1 of the element
Parameter : "aliasName2" Alias2 of the element
Parameter : "description" Description of the element
Parameter : "isInMaintenance" TRUE if the element is in maintenance
Parameter : "IsProjected" TRUE if the element is projected
Parameter : "zoneGuid" Guid ID of the Zone
Parameter : "subAreaGuid" Guid ID of the subarea

void SetElementDataByID(string elementID, string elementTypeName, string aliasName1, string aliasName2, string description, bool isInMaintenance, bool IsProjected, string zoneGuid, string subAreaGuid);

Summary :
Gets the element data for the element specified with the id and the element-type.

Parameter : "project" Project where the element exists
Parameter : " elementID" Id of the element
Parameter : "elementTypeName" Type of element, See the Appendix of the WebService Manual
Parameter : "aliasName1" Alias1 of the element
Parameter : "aliasName2" Alias2 of the element
Parameter : "description" Description of the element
Parameter : "isInMaintenance" TRUE if the element is in maintenance
Parameter : "IsProjected" TRUE if the element is projected
Parameter : "zoneGuid" Guid ID of the Zone
Parameter : "subAreaGuid" Guid ID of the subarea

void void GetElementDataByIDandType(ExternalProject project, Guid elementID, string elementTypeName, ref string elementName, ref string aliasName1, ref string aliasName2, ref string description, ref bool isInMaintenance, ref bool IsProjected, ref string zoneGuid, ref string subAreaGuid);

Summary :
Gets the element data for the element specified with the name and the element-type.

Parameter : "project" Project where the element exists
Parameter : " elementID" Id of the element
Parameter : "elementTypeName" Type of element, See the Appendix of the WebService Manual
Parameter : "aliasName1" Alias1 of the element
Parameter : "aliasName2" Alias2 of the element

Parameter : "description" Description of the element
Parameter : "isInMaintenance" TRUE if the element is in maintenance
Parameter : "IsProjected" TRUE if the element is projected
Parameter : "zoneGuid" Guid ID of the Zone
Parameter : "subAreaGuid" Guid ID of the subarea

void GetElementDataByNameandType(ExternalProject project, string elementName, string elementTypeName, ref Guid elementID, ref string aliasName1, ref string aliasName2, ref string description, ref bool isInMaintenance, ref bool IsProjected, ref string zoneGuid, ref string subAreaGuid);

Summary :
Sets the particular attribute of particular element found by its name

Parameter : "project" Summary Object of the internal project
Parameter : "elementName" Name of the element
Parameter : "elementType" Type of element, See the Appendix of the WebService Manual
Parameter : "attributeName" Attribute to be modified, See the Appendix of the WebService Manual
Parameter : "attributeValue" Value of the attribute to be set
Returns : TRUE if successful

bool SetElementAttribute(ExternalProject project, string elementName, string elementType, string attributeName, string attributeValue);

Summary :
Sets the particular attribute of particular element found by its ID

Parameter : "project" Summary Object of the internal project
Parameter : "elementID" ID of the element
Parameter : "attributeName" Attribute to be modified, See the Appendix of the WebService Manual
Parameter : "attributeValue" Value of the attribute to be set
Returns : TRUE if successful

bool SetElementAttributeByID(ExternalProject project, string elementID, string attributeName, string attributeValue);

Summary :
Gets the particular attribute of particular element found by its ID

Parameter : "projectName" Name of the project
Parameter : "elementID" ID of the element
Parameter : "attributeName" Attribute to be modified, See the Appendix of the WebService Manual
Parameter : "attributeValue" Value of the attribute to be set
Returns : TRUE if successful

bool GetElementAttributeByID(string projectName, string elementID, string attributeName, ref string attributeValue);

Summary :
Gets the particular attribute of particular element found by its name

Parameter : "projectName" Name of the project
Parameter : "elementName" Name of the element
Parameter : "elementType" Type of element, See the Appendix of the WebService Manual
Parameter : "attributeName" Attribute to be modified, See the Appendix of the WebService Manual
Parameter : "attributeValue" Value of the attribute to be set
Returns : TRUE if successful

bool GetElementAttributeByName(string projectName, string elementName, string elementType, string attributeName, ref string attributeValue);

Summary :
Sets the the role to READ ALL to all users in this project.
The user must be Owner or Admin

Parameter : "project" Summary Object of the internal project
Returns : TRUE if successful

bool AddReadAllRoleToProject(ExternalProject project);

Summary :
Adds a voltage level to the network

Parameter : "project" Summary Object of the internal project
Parameter : "un" Nominal Voltage in [kV]
Parameter : "colorR" Color R in RGB format
Parameter : "colorG" Color G in RGB format
Parameter : "colorB" Color B in RGB format
Returns : The Guid ID of the voltage level as String

string AddVoltageLevel(ExternalProject project, double un, int colorR, int colorG, int colorB);

Summary :

Parameter : "project" Summary Object of the internal project
Parameter : "xMin"
Parameter : "yMin"
Parameter : "units"
Parameter : "factorXY"
Parameter : "angle"

void SetDiagramMinValues(ExternalProject project, double xMin, double yMin, short units, double factorXY, double angle);

Summary :
Sets the style of the line of a project

Parameter : "project" Summary Object of the internal project

Parameter : "lineType" "OHL" for OverHeadLine, "Cable", "Mixed"
Parameter : "lineStyle" Graphical Representation of the line, 0: Type 1, 1: Type 2, 2: Type 3, 3: Type 4
Returns : TRUE if successful

bool SetLineStyle(ExternalProject project, string lineType, short lineStyle);

Summary :
Sets the default values of the project

Parameter : "project" Summary Object of the internal project
Parameter : "defaultType" Set Default results labels for : 0: HV Transmission, 1: LV Distribution, 2: LV Unbalanced
Parameter : "phaseDomination" Set the Phase denomination : 0: IEC, 1: ANSI, 2: India
Parameter : "isAnsi" TRUE if the Settings And Variables are according to ANSI
Returns : TRUE if successful

bool SetDefaultValues(ref ExternalProject project, short defaultType, short phaseDomination, bool isAnsi);

Summary :
Sets the attributes of the diagram parameters

Parameter : "project" Summary Object of the internal project
Parameter : "parameterType" "color", "label", "colorrange", "width", "font"
Parameter : "attributeName" Attribute to be modified, See the Appendix of the WebService Manual
Parameter : "attributeValue" Value of the attribute to be set
Returns : TRUE if successful

bool SetDiagramParameterAttribute(ExternalProject project, string parameterType, string attributeName, string attributeValue);

Summary :
Sets the attributes of the calculation parameters

Parameter : "project" Summary Object of the internal project
Parameter : "parameterType" Type of the the calculation See the Appendix of the WebService Manual
Parameter : "attributeName" Attribute to be modified, See the Appendix of the WebService Manual
Parameter : "attributeValue" Value of the attribute to be set
Returns : TRUE if successful

bool SetCalcParameterAttribute(ExternalProject project, string parameterType, string attributeName, string attributeValue);

Summary :

Parameter : "project" Summary Object of the internal project
Parameter : "isDirectionUpY"
Parameter : "factor"
Parameter : "angle"

Parameter : "offsetZeroPoint"
Returns : TRUE if successful

bool SetGraphicLayerWorldCoordinates(ExternalProject project, bool isDirectionUpY, double factor, double angle, double offsetZeroPoint);

Summary :
Adds in the project a list of nodes

Parameter : "project" Summary Object of the internal project
Parameter : "listElements" List of the nodes to be added
Returns : TRUE if successful

bool AddNodesFromList(ref ExternalProject project, List<ImportElementItem> listElements);

Summary :
Adds in the project a list of nodes

Parameter : "project" Summary Object of the internal project
Parameter : "listElements" List of the elements to be added
Parameter : "elementTypeName" Type of element, See the Appendix of the WebService Manual
Returns : TRUE if successful

bool AddElementsFromList(ref ExternalProject project, List<ImportElementItem> listElements, string elementTypeName);

Summary :
Adds in the project a list of subloads in a load found by its ID

Parameter : "project" Summary Object of the internal project
Parameter : "listLoadData" List of the loads to be added
Parameter : "loadID" ID of the load
Returns : TRUE if successful

bool AddSubLoadsFromList(ref ExternalProject project, List<LoadData> listLoadData, string loadID);

Summary :
Sets the Date and Time of the last save

Parameter : "project" Summary Object of the internal project

void SetDateTimeLastSave(ExternalProject project);

Summary :
Gets as XML the summary of a calculation result

Parameter : "project" Summary Object of the internal project
Parameter : "analysisType" LoadFlow, ShortCircuit, CapacitorPlacement, InvestmentAnalysis, LoadFlowTimeSimulation,

SwitchingOptimization,PhaseSwapping, ThermalAnalysis, FeederReinforcement
Parameter : "simulationDateTime" can be null if not a time simulation, e.g load flow
Parameter : "networkTypeGroup" Group Type for summary: Network = 0, Feeder = 1, Zone = 2, Area = 3, SubArea =4, TieFlowZone = 5, TieFlowArea = 6, Un = 7
Parameter : "networkTypeGroupID" "" by default
Returns : Results as XML String

string GetResultSummary(ExternalProject project, string analysisType, DateTime simulationDateTime, int networkTypeGroup, string networkTypeGroupID);

Summary :
Gets as string the summary of a calculation result

Parameter : "project" Summary Object of the internal project
Parameter : "analysisType" LoadFlow, ShortCircuit, CapacitorPlacement, InvestmentAnalysis, LoadFlowTimeSimulation,
SwitchingOptimization,PhaseSwapping, ThermalAnalysis, FeederReinforcement
Parameter : "simulationDateTime" can be null if not a time simulation, e.g load flow
Parameter : "networkTypeGroup" Group Type for summary: Network = 0, Feeder = 1, Zone = 2, Area = 3, SubArea =4, TieFlowZone = 5, TieFlowArea = 6, Un = 7
Parameter : "networkTypeGroupID" "" by default
Returns : Results as a list of Strings

string[] GetListResultSummary(ExternalProject project, string analysisType, DateTime simulationDateTime, int networkTypeGroup, string networkTypeGroupID);

Summary :
Gets an XML string with the calculation settings

Parameter : "project" Summary Object of the internal project
Parameter : "analysisType" LoadFlow, ShortCircuit, CapacitorPlacement, InvestmentAnalysis, LoadFlowTimeSimulation,
SwitchingOptimization,PhaseSwapping, ThermalAnalysis, FeederReinforcement
Returns : Settings as XML String

string GetCalculationSettings(ExternalProject project, string analysisType);

Summary :
Gets the result of an element found by its ID at a specified port

Parameter : "project" Summary Object of the internal project
Parameter : "analysisType" LoadFlow, ShortCircuit, CapacitorPlacement, InvestmentAnalysis, LoadFlowTimeSimulation,
SwitchingOptimization,PhaseSwapping, ThermalAnalysis, FeederReinforcement
Parameter : "portNr" Port number (values: 0 for port 1, 1 for port 2, 2 for port 3, 3 for port 4)
Parameter : "simulationDateTime" cCan be null if it is not a time simultion
Returns : Settings as XML String

string GetResultElementByID(ExternalProject project, string elementID, int portNr, string analysisType, DateTime simulationDateTime);

Summary :
Get the result of an element found by its name at a specified port

Parameter : "project" Summary Object of the internal project
Parameter : "elementName" Name of the element
Parameter : "elementTypeName" Type of element, See the Appendix of the WebService Manual
Parameter : "portNr" Port number (values: 0 for port 1, 1 for port 2, 2 for port 3, 3 for port 4)
Parameter : "analysisType" LoadFlow, ShortCircuit, CapacitorPlacement, InvestmentAnalysis, LoadFlowTimeSimulation,
SwitchingOptimization,PhaseSwapping, ThermalAnalysis, FeederReinforcement
Parameter : "simulationDateTime" Can be null if it is not a time simultion
Returns : Settings as XML String

string GetResultElementByName(ExternalProject project, string elementName, string elementTypeName, int portNr, string analysisType, DateTime simulationDateTime);

Summary :
Runs an analysis on a variant

Parameter : "project" Summary Object of the internal project
Parameter : "analysisRefenceID" A reference ID to the analysis which will be started
Parameter : "analysisModule" Analysis Type, See the Appendix of the WebService Manual
Parameter : "calcNameID" Operational state name, "Default" by default
Parameter : "analysisMethode" At the moment not used
Parameter : "conditions" At the moment not used
Returns : Returns the Object AnalysisReturnInfo of the calculations where are stored the informations of the calculation

AnalysisReturnInfo AnalyseVariant(ExternalProject project, string analysisRefenceID, string analysisModule, string calcNameID, string analysisMethode, string conditions, string analysisLoadOptionXML);

Summary :
Runs an analysis on a variant with specified settings

Parameter : "project" Summary Object of the internal project
Parameter : "analysisRefenceID" A reference ID to the analysis which will be started
Parameter : "analysisModule" Analysis Type, See the Appendix of the WebService Manual
Parameter : "calcNameID" Operational state name, "Default" by default
Parameter : "analysisMethode" At the moment not used
Parameter : "conditions" At the moment not used
Parameter : "settings" Analysis settings for the calculation
Returns : Returns the Object AnalysisReturnInfo of the calculations where are stored the informations of the calculation

AnalysisReturnInfo AnalyseVariant(ExternalProject project, string analysisRefenceID, string analysisModule, string calcNameID, string analysisMethode, string conditions, string analysisLoadOptionXML);

Summary :
Checks if the analysis is finished

Parameter : "analysisRefenceID" A reference ID to the analysis
Returns : TRUE if the analysis is finished

bool IsAnalysisDone(string analysisRefenceID);

Summary :
Runs a calculation on a .NEPLST360 Project File

Parameter : "uploadName" Uploaded file to calculate
Parameter : "analysisModule" Analysis Type, See the Appendix of the WebService Manual
Parameter : "copySettingsFromProjectName" The name of the project used as template for the settings
Parameter : "fromTime" "2011-11-13T01:00:00" 13th November 2011
Parameter : "toTime" "2011-11-28T01:00:00" 28th November 2011
Parameter : "timeIncrement" For the time simulation, increment of Time in minutes
Returns : Returns the Object AnalysisReturnInfo of the calculations where are stored the informations of the calculation

AnalysisReturnInfo AnalyseWithListFileWithoutSaving(string uploadName, string analysisModule, string copySettingsFromProjectName, string fromTime, string toTime, int timeIncrement);

Summary :
Runs a calculation on a .NEP360 Project File

Parameter : "uploadName" Uploaded file to calculate
Parameter : "analysisModule" Analysis Type, See the Appendix of the WebService Manual
Parameter : "copySettingsFromProjectName" The name of the project used as template for the settings
Parameter : "fromTime" "2011-11-13T01:00:00" 13th November 2011
Parameter : "toTime" "2011-11-28T01:00:00" 28th November 2011
Parameter : "timeIncrement" For the time simulation, increment of Time in minutes
Returns : Returns the Object AnalysisReturnInfo of the calculations where are stored the informations of the calculation

AnalysisReturnInfo AnalyseWithProjectFileWithoutSaving(string uploadName, string analysisModule, string copySettingsFromProjectName, string fromTime, string toTime, int timeIncrement);

Summary :
Runs a calculation on a .CDE Project File

Parameter : "uploadName" Uploaded file to calculate
Parameter : "analysisModule" Analysis Type, See the Appendix of the WebService Manual
Parameter : "copySettingsFromProjectName" The name of the project used as template for the settings
Parameter : "fromTime" "2011-11-13T01:00:00" 13th November 2011

Parameter : "toTime" "2011-11-28T01:00:00" 28th November 2011
Parameter : "timeIncrement" For the time simulation, increment of Time in minutes
Returns : Returns the Object AnalysisReturnInfo of the calculations where are stored the informations of the calculation

AnalysisReturnInfo AnalyseWithCDEFileWithoutSaving(string uploadName, string analysisModule, string copySettingsFromProjectName, string fromTime, string toTime, int timeIncrement);

Summary :
Analysis with file (zip, cde, neplst360 nep30, nepmeas) and measurement file. The 2 files should be in a zip file.
Extension of the name of the list file should *.zip, *.cde, *.neplst360 *.nep30, The measuerment file extension should be *.Meas
The fromTime and toTime must be in the following format:
FromDateTime", "2011-11-13T01:00:00" 13th November 2011
ToDateTime", "2010-11-28T01:00:00" 28th November 2011

Parameter : "uploadName" Uploaded file to calculate
Parameter : "fromTime" "2011-11-13T01:00:00" 13th November 2011
Parameter : "toTime" "2011-11-28T01:00:00" 28th November 2011
Parameter : "copySettingsFromProjectName" The name of the project used as template for the settings
Returns : Returns the Object AnalysisReturnInfo of the calculations where are stored the informations of the calculation

AnalysisReturnInfo DoAnalysisFromFiles(string uploadName, string analysisModule, string copySettingsFromProjectName, string fromTime, string toTime, int timeIncrement);

Summary :
Energy loss calculation with file (zip, cde, neplst360 nep30, nepmeas) and measurement file. The 2 files should be in a zip file.
Extension of the name of the list file should *.zip, *.cde, *.neplst360 *.nep30, The measuerment file extension should be *.Meas
The fromTime and toTime must be in the following format:
FromDateTime", "2011-11-13T01:00:00" 13th November 2011
ToDateTime", "2010-11-28T01:00:00" 28th November 2011

Parameter : "uploadName" Uploaded file to calculate
Parameter : "fromTime" "2011-11-13T01:00:00" 13th November 2011
Parameter : "toTime" "2011-11-28T01:00:00" 28th November 2011
Parameter : "timeIncrement" For the time simulation, increment of Time in minutes
Parameter : "copySettingsFromProjectName" The name of the project used as template for the settings
Returns : Returns the Object AnalysisReturnInfo of the calculations where are stored the informations of the calculation

AnalysisReturnInfo EnergyCalculationFromFiles(string uploadName, string fromTime, string toTime, int timeIncrement, string copySettingsFromProjectName);

Summary :

Parameter : "projectName" Name of the project
Parameter : "calcNameID" Operational state name, "Default" by default
Parameter : "fromTime" 2011-11-13T01:00:00" 13th November 2011
Parameter : "toTime" "2011-11-28T01:00:00" 28th November 2011
Parameter : "timeIncrement" For the time simulation, increment of Time in minutes
Parameter : "settings" Special Settings of the calculation passed as AnalysisParameterSettings Object
Returns : Returns the Object AnalysisReturnInfo of the calculations where are stored the informations of the calculation

AnalysisReturnInfo LoadAllocationFromDB(string projectName, string calcNameID, string fromTime, string toTime, int timeIncrement, AnalysisParameterSettings settings);

Summary :

Parameter : "uploadedMeasurementFileName"
Parameter : "projectName" Name of the project
Parameter : "fromTime" 2011-11-13T01:00:00" 13th November 2011
Parameter : "toTime" "2011-11-28T01:00:00" 28th November 2011
Parameter : "timeIncrement" For the time simulation, increment of Time in minutes
Parameter : "settings" Special Settings of the calculation passed as AnalysisParameterSettings Object
Returns : Returns the Object AnalysisReturnInfo of the calculations where are stored the informations of the calculation

AnalysisReturnInfo LoadAllocationFromMeasurmentFileAndProjectName(string uploadedMeasurementFileName, string projectName, string fromTime, string toTime, int timeIncrement, AnalysisParameterSettings settings);

Summary :

Parameter : "project" Summary Object of the internal project
Parameter : "uploadedMeasurementFileName"
Parameter : "fromTime" 2011-11-13T01:00:00" 13th November 2011
Parameter : "toTime" "2011-11-28T01:00:00" 28th November 2011
Parameter : "timeIncrement" For the time simulation, increment of Time in minutes
Parameter : "settings" Special Settings of the calculation passed as AnalysisParameterSettings Object
Returns : Returns the Object AnalysisReturnInfo of the calculations where are stored the informations of the calculation

AnalysisReturnInfo LoadAllocationFromMeasurmentFile(ExternalProject project, string uploadedMeasurementFileName, string fromTime, string toTime, int timeIncrement, AnalysisParameterSettings settings);

Summary :

Parameter : "analysisReturnInfo" Returns the Object AnalysisReturnInfo of the calculations where are stored the informations of the calculation
Parameter : "uploadName" Uploaded file to calculate
Parameter : "projectName" Name of the project
Parameter : "settings" Special Settings of the calculation passed as AnalysisParameterSettings Object
Returns : Summary Object of the internal project

ExternalProject ImportFromListFileAndDoLoadAllocation(out AnalysisReturnInfo analysisReturnInfo, string uploadName, string projectName, AnalysisParameterSettings settings);

Summary :
Deletes the Analysis Log File

Parameter : "fileName" Name of the AnalysisLogFile
Returns : TRUE if successful

bool DeleteAnaylsisLogFile(string fileName);

Summary :
Deletes the Result Log File

Parameter : "fileName" Name of the AnalysisResultFile
Returns : TRUE if successful

bool DeleteAnalysisResultFile(string fileName);

Summary :
Deletes a project

Parameter : "project" Summary Object of the internal project
Returns : TRUE if successful

bool DeleteProject(ExternalProject project);

Summary :
Markes a project as deleted

Parameter : "project" Summary Object of the internal project
Returns : TRUE if successful

bool MarkedProjectAsDeleted(ExternalProject project);

Summary :
Imports or merge a NEP360 file.

Parameter : "uploadName" Uploaded file
Parameter : "projectName" Name of the project
Parameter : "withGraphic" TRUE if imported with Graphic
Parameter : "copySettingsFromProjectName" The name of the project used as template for the settings

Parameter : "merge" TRUE if the project should be merged
Parameter : "mergeOnNewDiagram" True if the projects should be merged in a new diagrams, FALSE if in the default diagram
Returns : 0=OK, -1=Logon error; -2=file format error; -3=number of nodes exceeded -4=create project error;-5=max project number exceeded -50=internal/other error

int ImportFromFile(string uploadName, string projectName, bool withGraphic, string copySettingsFromProjectName, bool merge, bool mergeOnNewDiagram);

Summary :
Adds a list of feeders in a project

Parameter : "project" Summary Object of the internal project
Parameter : "feederList" List of feeders
Returns : TRUE if successful

bool AddFeederFromList(ExternalProject project, List<ImportFeederItem> feederList);

Summary :
Imports a list of element in a project

Parameter : "listElements" The list of the elements to be imported
Parameter : "projectName" Name of the project
Parameter : "copySettingsFromProjectName" The name of the project used as template for the settings
Returns : Project object returned

ExternalProject ImportFromList(List<ImportElementItem> listElements, string projectName, string copySettingsFromProjectName);

Summary :
Imports a list of element from a file in a project

Parameter : "uploadName" Uploaded file
Parameter : "projectName" Name of the project
Parameter : "copySettingsFromProjectName" The name of the project used as template for the settings
Returns : Project object returned

ExternalProject ImportFromListFile(string uploadName, string projectName, string copySettingsFromProjectName);

Summary :
Imports or merge a .NEPLST360 file.

Parameter : "uploadName" Uploaded file
Parameter : "projectToMerge" Name of the project to merge
Parameter : "mergeOnNewDiagram" TRUE if the projects have to merge in a new diagram
Returns : Project object returned

ExternalProject MergeProjectFromListFile(string uploadName, string projectToMerge, bool mergeOnNewDiagram);

Summary :
Imports a .CDE file.

Parameter : "path1"
Parameter : "path2"
Parameter : "path3"
Parameter : "copySettingsFromProjectName" The name of the project used as template for the settings
Parameter : "withGraphic" TRUE if the Graphics have to be imported
Returns : Project object returned

ExternalProject ImportFromCDE(string path1, string path2, string path3, string copySettingsFromProjectName, bool withGraphic);

Summary :
Imports a project from a .CDE files

Parameter : "cdeFilename"
Parameter : "ndbFilename"
Parameter : "xmlFilename"
Parameter : "copySettingsFromProjectName" The name of the project used as template for the settings
Parameter : "withGraphic" TRUE if the Graphics have to be imported
Returns : Project object returned

ExternalProject ImportFromCDEFiles(string cdeFilename, string ndbFilename, string xmlFilename, string copySettingsFromProjectName, bool withGraphic);

Summary :
Imports a project from a .CDE in a ZIP file

Parameter : "zipFile" ZIP file containing the CDE
Parameter : "copySettingsFromProjectName" The name of the project used as template for the settings
Parameter : "withGraphic" TRUE if the Graphics have to be imported
Returns : Project object returned

ExternalProject ImportFromCDEZipFile(string zipFile, string copySettingsFromProjectName, bool withGraphic);

Summary :
Imports a project from a PSSE file

Parameter : "rawFilename" .RAW File
Parameter : "seqFilename" .SEQ File
Parameter : "dyrFilename" .DYR File
Parameter : "drwFilename" .DRW File
Parameter : "psseVersion" PSSE version number
Parameter : "copySettingsFromProjectName" The name of the project used as template for the settings

Returns : Project object returned

ExternalProject ImportFromPSSEFiles(string rawFilename, string seqFilename, string dyrFilename, string drwFilename, int psseVersion, string copySettingsFromProjectName);

Summary :
Imports a project from a PSSE in a ZIP file

Parameter : "zipFile" .ZIP File containing the PSSE
Parameter : "psseVersion" PSSE version number
Parameter : "copySettingsFromProjectName" The name of the project used as template for the settings
Returns : Project object returned

ExternalProject ImportFromPSSEZipFile(string zipFile, int psseVersion, string copySettingsFromProjectName);

Summary :
Import from a shape file

Parameter : "path" Path of the Shape file
Parameter : "fileName" Name of the Shape File
Returns : 0=OK, -1=Logon error; -2=file format error; -3=number of nodes exceeded -4=create project error;-10=max project number exceeded -50=internal/other error

int ImportFromShapeFile(string path, string fileName);

Summary :
Import from a .NDB file.

Parameter : "project" Summary Object of the internal project
Parameter : "uploadName" Uploaded file
Parameter : "origFileName" Not used at the moment
Returns : TRUE if successful

bool ImportFromNDB(ExternalProject project, string uploadName, string origFileName);

Summary :
Imports Measured data in a XML format

Parameter : "project" Summary Object of the internal project
Parameter : "uploadName" Uploaded file
Returns : TRUE if successful

bool ImportMeasuredDataFromXMlFile(ExternalProject project, string uploadName);

Summary :
Writes the message to the logger defined in the project

Parameter : "project" Summary Object of the internal project

Parameter : "text" Message to write in the logger
Parameter : "logLvl" Log Level. "Error", "Warning", "Info"

void WriteMessageToLogFile(ExternalProject project, string text, string logLvl);

Summary :
Delete the user log file

Returns : TRUE if successful

bool DeleteUserLogFile();

Summary :
Get the whole logfile as string

Returns : Returns the LOG information messages

string GetLogFileAsString();

Summary :
Get the whole logfile as a list

Returns : Returns the LOG information messages as List

List<string> GetLogFileAsList();

Summary :
Uploads a xml file to the server

Parameter : "stream" XML Stream to be uploaded
Returns : FileName or null if there was an error

string XMLUpload(System.IO.Stream stream);

Summary :
Uploads a cde file to the server

Parameter : "stream" CDE Stream to be uploaded<
Returns : FileName or null if there was an error

string CDEUpload(System.IO.Stream stream);

Summary :
Uploads a ndb file to the server

Parameter : "stream" NDB Stream to be uploaded<
Returns : FileName or null if there was an error

```
string NDBUpload(System.IO.Stream stream);
```

```
Summary :
Uploads a NEP360 file to the server

Parameter : "stream"  NEP360 Stream to be uploaded<
Returns :  FileName or null if there was an error

string Nep360Upload(System.IO.Stream stream);
```

```
Summary :
Uploads a ZIP file

Parameter : "stream"  ZIP Stream to be uploaded
Returns :  FileName or null if there was an error

string ZipUpload(System.IO.Stream stream);
```

```
Summary :
Uploads a Measurement file

Parameter : "stream"  NEPMEAS Stream to be uploaded
Returns :  FileName or null if there was an error

string NepMeasUpload(System.IO.Stream stream);
```

```
Summary :
Get the user Log file as stream

Returns :  The file stream or null

System.IO.Stream GetUserLogFile();
```

```
Summary :
Get the analysis log file as stream

Parameter : "fileName"  Name of the Analysis Log to download
Returns :  The file stream or null

System.IO.Stream GetAnaylsisLogFile(string fileName);
```

```
Summary :
Get the analysis result file as stream

Parameter : "fileName"  Name of the Results to download
Returns :  The file stream or null

System.IO.Stream GetAnalysisResultFile(string fileName);
```

Summary :
Gets the Feeder Ids by giving the feedernames

Parameter : "ProjectName" Name of the project
Parameter : " VariantName" Name of the variant
Parameter : " FeederNames" An array of the feeder names

Returns : An array of feeder Ids which corresponds to the feeder names

GetFeederIDByName(string ProjectName, string VariantName, string[] FeederNames);

4) NEPLAN classes associated to Web-Services

Most of the webservices are using classes defined by NEPLAN for input or output to Web-Service calls. Here, a comprehensive list of those classes is provided:

AnalysisParameterSettings

Summary :
Analysis settings for the calculation. Some settings refer to specific calculations, thus they do not need to be set for other calculations.

Member : "ListOfRelayIDs" List of relay IDs (for selectivity charts)
Member "ListOfShortCircuitLocationIDs" IDs of the elements with a short circuit (for selectivity charts)
Member "DoRecondResizing" Defines if reconductor sizing is to be done
Member "SelectedRecondResizElements" Defines a list of elements for reconducting
Member "LibNameForRecondResizing" Library name for reconductor sizing
Member "LibSubNameForRecondResizing" Library subname for reconductor sizing
Member "LibItemTypeNameForRecondResizing" Library subname for reconductor sizing
Member "ItemlibNameForRecondResizing" Library item name for reconductor sizing
"ItemlibSubNameForRecondResizing" Library item subname for reconductor sizing
ItemlibSubNameForRecondResizing" Library item subname for reconductor sizing
Member "ItemTypeNameForRecondResizingField" Library item type name for reconductor sizing
Member "ListOfSelectedFeederIDs" List of selected feeders for calculation (if calculation requires so)
Member "ListOfSelectedPartNetwIDs" List of selected partial networks for calculation (if calculation requires so)
Member "LogName" Name of log file
Member "NCalc" Integer describing the calculation type
Member "AnschlKnID" Node ID for connection point (only for DACH module)
Member "EleID" Element ID (only for DACH module)
Member "RelayID" ID of the relay element
Member "ConXML" XML file for connections e.g. PetersenCoil, Flicker, Dach
Member "ExcludeProjectedElements" Exclude projected elements from the calculation
Member "ExcludeOutOfOperElements" Exclude out of operation elements from the calculation
Member "ExclutionDateForOutOfOper" Exclude the datetime defining the "out of operation" check
Member "ExcludeElementsInMaint" Exclude elements in maintenance from the calculation
Member "ExclutionDateForInstalled" Exclude the datetime for the "installed elements" check
Member "ExcludeElementsInstalled" Exclude installed elements from the calculation
Member "MediumType" Medium type
Member "CalcOnlyLoadedData" Calculate only loaded data of the network
Member "IncludeBoundaryElementsInCalculation" Include boundary elements in the calculation

5) APPENDIX A

1. **InsertNode**, **InsertBusbar**

InsertNode creates a node which could be a bus bar or a pole. InsertBusbar creates a busbar. A busbar is represented by two coordinates pairs, a pole with one x,y pair.

Pole 

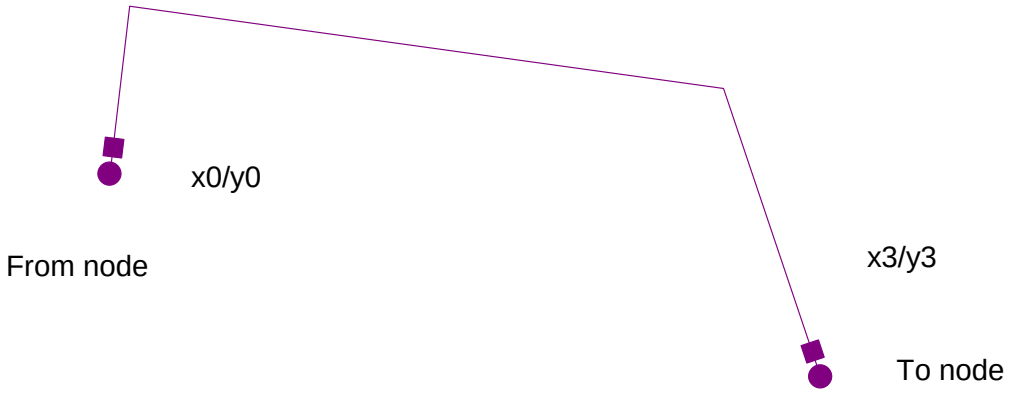
e.g. x1= 237788935 y1=190677860

Busbar

e.g. x1=100 y1=450, x2=120 y2=450

2. InsertLine

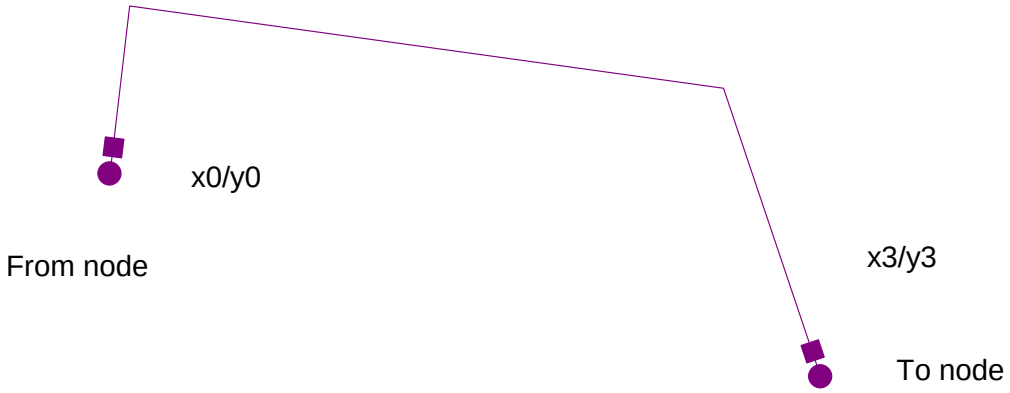
This function creates a line, which could be described by



e.g. x[0]=119 y[0]=816, x[1]=117 y[1]=831, x[2]=66 y[2]=824, x[3]=58 y[3]=800
Number of coordinates: 4

3. InsertLineAsym

This function creates an asymmetrical line, which could be described by

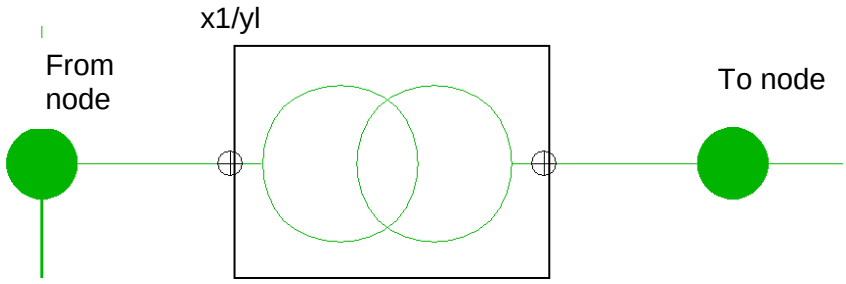


e.g. x[0]=119 y[0]=816, x[1]=117 y[1]=831, x[2]=66 y[2]=824, x[3]=58 y[3]=800
Number of coordinates: 4

Possible values for phases: L1L2L3N, L1N, L2N, L3N, L1L2N, L1L3N, L2L3N, L1L2L3N_AS

4. Insert2PortElement

This function creates an element, which is defined by 2 ports, like transformer, reactor, etc.



6) APPENDIX B

2-ports or more ports elements:

“Busbar“
“Trafo2Winding“
“Line“
“LineAsym“
“BusbarCoupler“
“Reactor“
“Trafo2WindingAsym“
“TrafoRegulator“
“SerieRLC“
“ParallelRLC“
“Trafo3Winding“
“Trafo4Winding“
“EquivalentSerieLF“
“EquivalentSerieSC“
“DisconnectSwitch“
“LoadSwitch“
“CircuitBreaker“
“TCSC“
“UPFC“
“PWM“
“SerieTransformator“

1-ports elements:

“ExternalGrid“
“SynchronousMachine“
“AsynchronousMachine“
“DFIG“
“Load“
“Shunt“
“SVS“
“Filter“
“SerieEarthRLC“
“EquivalentShuntLF“
“EquivalentShuntSC“
“HarmonicCurrentSource“
“HarmonicVoltageSource“
“STATCOM“

On elements (Protection, switches, etc)

“Fuse”
“DistanceRelais”
“OvercurrentRelais“
“CircuitBreakerOnElem“
“LineLoad“
“MeasurementDevice“

“CurrentTransformer“
“VoltageTransformer“
“DisconnectSwitchOnElem“
“LoadSwitchOnElem“
“EarthSwitch“
“SurgeArrester“
“FrequencyRelais“
“VoltageRelais“
“PowerRelais“
“MinMaxRelaisOnNode“
“MinMaxRelaisOnLink“

DC elements:

„DCNode“
„DCLine“
“DCVoltageSource“
“DCReactor“
“DCShunt“
“DCMotor“
“DCGround“
“DCConverter“
“DCConverter3Pole“
“DCLoad“

Others:

“Station“
“EarthConductor“
“Pylon“
“CustomerConnection“
“GroundElement“
“UserDefinedPort0“
“UserDefinedPort1“
“UserDefinedPort2“
“UserDefinedPort3“
“UserDefinedPort4“
“FaultIndicator“
“NestedBlockCCT“
“Regulator“
“Turbine“
“MechanicalLoad“
“Table“
“Inertia“
“PoleSlipRelais“
“BlockGeneral“