REENGINEERING SWMM/EPANET   
USER INTERFACE APPLICATION SOFTWARE ARCHITECTURES:   
PRELIMINARY CONCEPTUAL ARCHITECTURAL DESIGN

**RSI-XXXX**

Revision 0

*prepared for*

U.S. Environmental Protection Agency

Office of Research and Development

26 West Martin Luther King Drive

Cincinnati, Ohio 45268

January 2016



REENGINEERING SWMM/EPANET   
USER INTERFACE APPLICATION SOFTWARE ARCHITECTURES:   
PRELIMINARY CONCEPTUAL ARCHITECTURAL DESIGN

EPA Contract #GS-10F-0041X

**RSI-XXXX**

Revision 0

*by*

AQUA TERRA Consultants

2685 Marine Way, 1314

Mountain View, California 94043

*prepared for*

U.S. Environmental Protection Agency

Office of Research and Development

26 West Martin Luther King Drive

Cincinnati, Ohio 45268

January 2016

TABLE OF CONTENTS

1. 0 INTRODUCTION 1

2. 0 DEVelopment stack and tool chain 2

3. 0 USER interface design 4

3.1 Sripting and plugin support 6

4. 0 SWMM and EPANET Class ARCHITECTURE 8

4.1 SWMM CORE CLASSES 9

4.2 EPANET CORE CLASSES 23

5. 0 SOFTWARE DEMONSTRATIONS 28

5.1 SCRIPTING 28

5.2 PLUG-INS 28

5.3 MAP 28

5.4 Main Window 28

# 0 INTRODUCTION

From the Statement of Work, Task 1.3 – *The contractor shall develop and document a preliminary conceptual software architectural design. The architectural design shall conform to the scope of general requirements described above and the AFRD. The design shall be organized around core functionality shared between the SWMM and EPANET and specialized functionality unique to each application. The architecture shall be designed using a logical combination of C++ and Python. The software architectural design may include technical demonstrations of key software architectural details on target platforms using development stack selected for the project as deemed necessary by the project team. The software architecture design shall also consider issues such as the complexity of development stack configuration and product builds, as well as the complexity of packaging, delivery, and installation on multiple target platforms.*

The purpose of this project is to reengineer the User Interface (UI) architecture for the EPA’s collection system and water distribution system simulation products, the Storm Water Management Model (SWMM) and software that models water distribution-piping systems (EPANET). The project focuses on the design of a modular and extensible user interface application software architecture for SWMM and EPANET. The modular and extensible architecture will enable deployment of new application features created by the EPA, third-party developers, and end users employing scripts and application “plug-ins.”

Factoring in the UI design of the current SWMM and EPANET software as well as the input file format for the two programs, we have outlined a preliminary conceptual design for both the UI components and the core classes for SWMM and EPANET input. The first section of this document presents the development platform, including the development stack and tool chain. The second section of this document describes UI components, much of which will be shared between the two programs. The document concludes with a detailed presentation of the core classes for both the SWMM and EPANET user interface architectures.

Other details of this design will be presented as technical demonstrations as described in the statement of work, particularly those pertaining to the development stack, plug-in and scripting architectures, and GIS functionality. This is a preliminary architectural document, and, as a living document, it is expected to be updated as needed during the development timeline.

# 0 DEVelopment stack and tool chain

The reengineering of SWMM and EPANET’s user interface software architecture will be designed using an open platform that employs open source software and tool chains. This open platform will foster the already sizeable user community, encouraging it to continue to grow and develop custom features for these two EPA products.

Python is the language of choice for the development work due to its widespread adoption by scientists, engineers, and software developers in many professions world-wide. It is a cross-platform programming language that has native support on all three major desktop operating systems, i.e. Linux, MacOS X, and Windows. In addition, a large pool of open source third party libraries for a great variety of purposes is already in existence and can be readily employed to accomplish many functionalities of the two software products ranging from user interface (UI) element to complex numerical analysis.

For this project, the GUI visual layout is created by the Qt GUI toolkit (from the Qt Company) via drag-n-drop of UI widgets (such as a button, text field, dropdown box, table view etc.) onto a design surface (i.e. window). PyQt, which consists of Python bindings for the Qt GUI visual elements, handles user interactions with the visual elements via a generic signal/slot mechanism to ensure implementation- neutral event handling. The choice of the Python-PyQt framework ensures this project benefits from the combined power of Qt and Python and to render the finished product to be truly cross-platform.

Required GIS functionality will be provided using the open source GIS map display from QGIS. Because QGIS is available as a set of libraries that can be incorporated into other standalone applications, QGIS can be brought into the reengineered EPANET and SWMM software as a component instead of using the entire QGIS main program. The utility of the QGIS library (namely qgis.core and qgis.gui) is incorporated by importing them into the new main application. These two libraries provide the main QGIS map control and drawing canvas where the actual GIS data layers will be displayed. A standalone python class (EmbedMap) is constructed as a wrapper of the QGIS libraries to arrive at an independent map control that can be added to a Qt UI container control.

Scripting will be implemented using the IPython command shell for interactive computing.

For this project, PyCharm Community Edition 5.0 from JetBrains was selected as the python source code integrated development environment (IDE). PyCharm has a large user community and it is available free of charge.

The overall project architecture is shown in Figure 1.

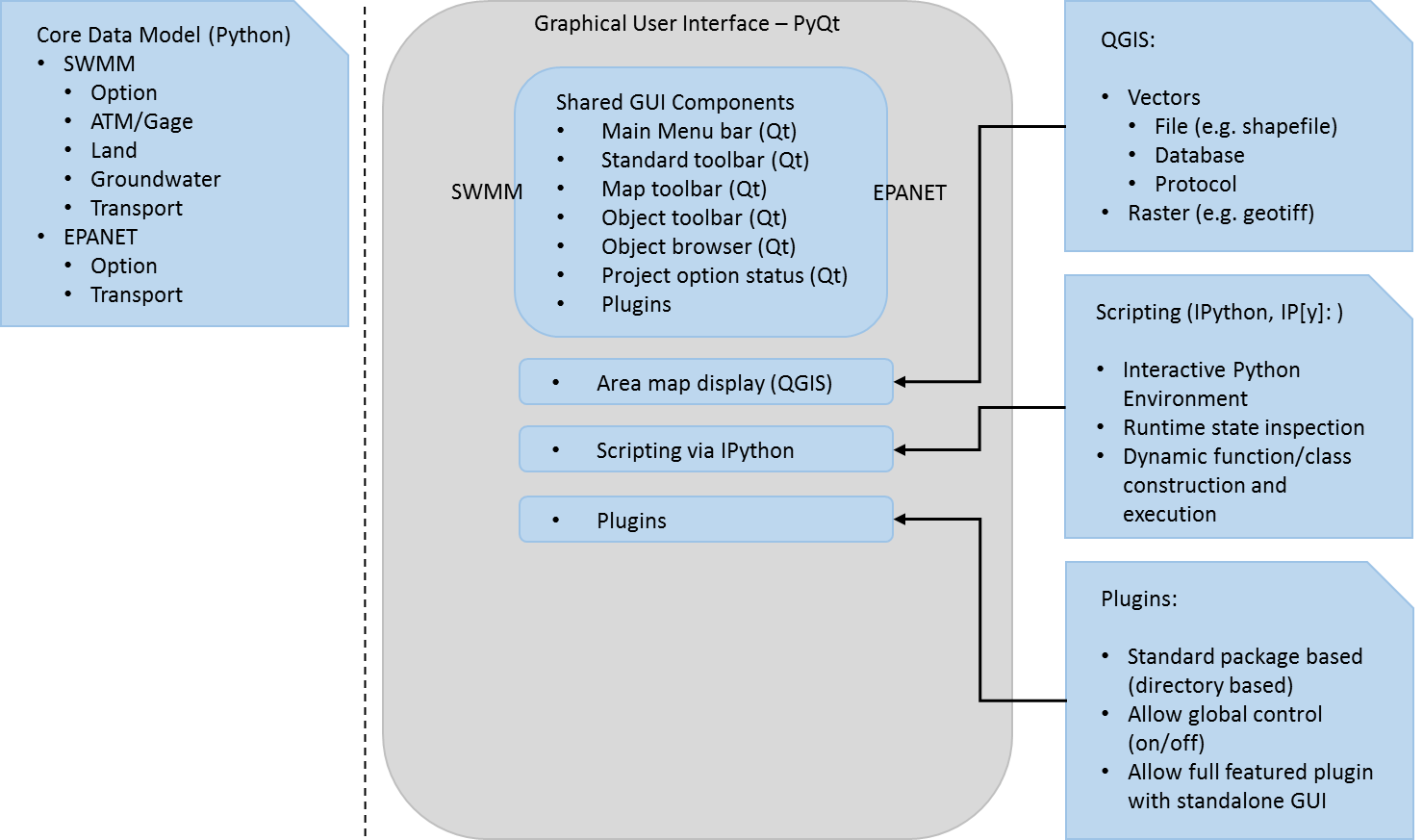


Figure 1. Overall project architectural design

# 0 USER interface design

The user interfaces for SWMM and EPANET are designed to closely align with the design of the previous official EPA versions, preserving familiarity for existing users. The current official SWMM and EPANET products have similar GUI layouts and share a common set of GUI control elements, as shown in Figures 2 and 3 below.

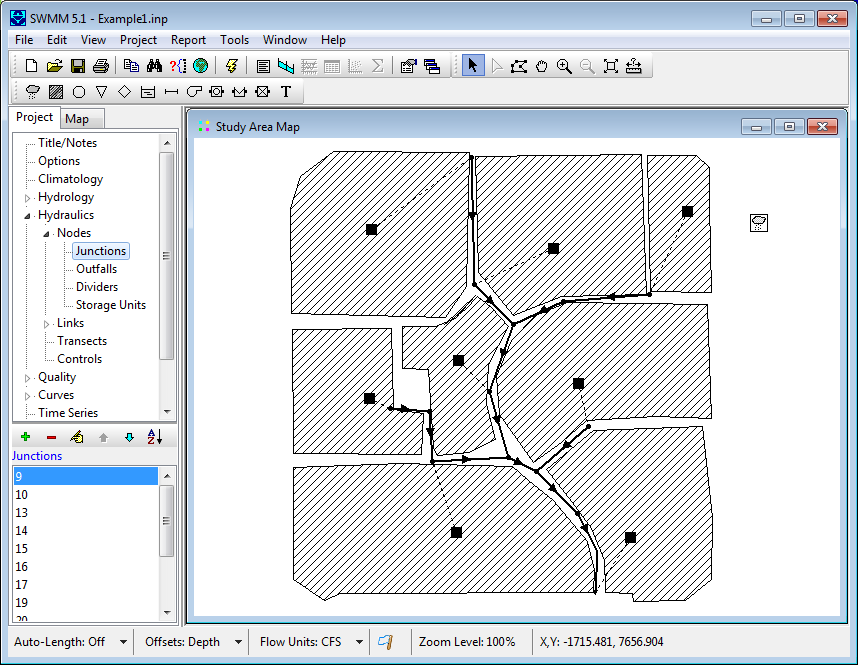


Figure 2. Existing SWMM User Interface

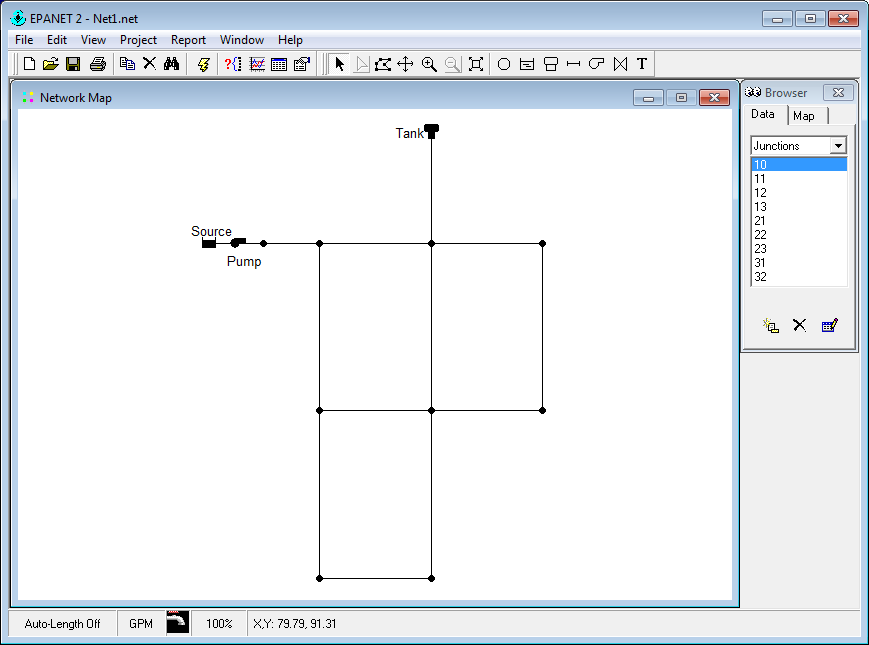


Figure 3. Existing EPANET User Interface

The two new software products will be designed using a shared architectural framework and the same set of event handling logic in all major UI components, including the map display, scripting, and plugins.

The UI components including main menu bar, toolbars, project browser, and project options, along with their subsequent chain of UI elements, will be designed and function in a similar fashion to the current software.

The current area/network map display control will be replaced with the open source GIS map display from QGIS. The QGIS libraries provide the main QGIS map control and drawing canvas where the actual GIS data layers will be displayed. In Figure 4 below, the map display is in a MDI control.

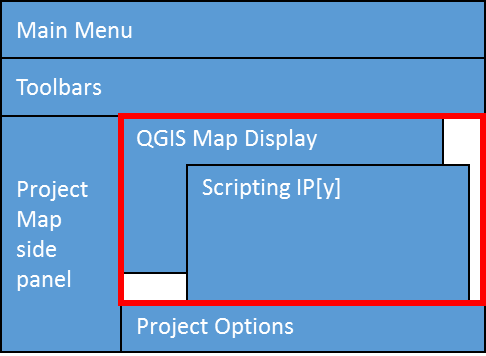


Figure 4. The QGIS map display and other controls (e.g. IPython scripting console) within the new EPANET/SWMM GUI

## Sripting and plugin support

The scripting control is implemented using IPython (a Qt-based python scripting console), which can be activated from the main menu of the application via a ‘Scripting🡪IPython’ menu option. The main program will open an IPython console in a stand-alone window that is within the same MDI control that also contains the QGIS map window (Figure 4). Within the console, users can type in Python commands, function and class definitions, and any other procedures for dynamic execution. Access to model inputs and outputs from within the scripting console is provided via class object sharing.

Plugins are placed in a special ‘plugins’ folder that is made known to the main program during the installation process. They are organized in packages that correspond to the file directory structures, namely, each plugin will reside in its own subdirectory under the ‘plugins’ directory. Each plugin can have unlimited level of packages (or subdirectories) of its own. At the minimum, each plugin must specify three pieces of information, i.e. its name (plugin\_name), whether to generate its own dropdown menu in the main menu bar of the main application (plugin\_create\_menu), and the operations that are to be exposed to (i.e. that can be executed by) end users (\_\_all\_\_ dictionary). Plugins can access model inputs and outputs via class object sharing at runtime. Upon program launch, the main program will scan the plugins directory for installed plugins and list them in a ‘Plugins’ menu item on the main form of the program, where they can be switched on/off (by mouse click) individually. We envision two major categories of plugins: core plugins and external plugins. Core plugins will be installed with the main program. These plugins will offer functionality desired by model users and will serve as examples for users who wish to develop a custom plugin. External plugins are those written by individual users that cater to their own specific needs.

 This design supports a future when there is a website where the community of model users and developers can share ideas, answer questions, and share peer-reviewed plugins and scripts. As such a website is developed, a future extension to plugin management can be constructed to retrieve and install shared plugins. This design pattern will provide maximum flexibility to users in constructing their own plugins at various levels of complexity ranging from a single function to an application with its own GUI.

# 0 SWMM and EPANET Class ARCHITECTURE

As described in the introduction, the core classes for SWMM and EPANET input follow a similar hierarchical design. This section presents the core classes for each program.

Code in this effort is divided into two major groups: **core** and **ui** (user interface) (Figure 5). Core classes contain the data structures needed to represent model parameters and data along with the code needed to read, write, and access these parameters and data. UI classes define the user interface: the graphical controls needed to display the data and allow a user to edit it. Within both **core** and **ui** groups, shared code appears at the top level and code specific to SWMM and EPANET is organized into sub-folders. Such a design lends to easier model-specific development and installation.

For example, at the top level of the core code group, the InputFile class contains general-purpose structure and methods for organizing, reading, and writing input files. One level deeper in the hierarchy, two classes inherit from InputFile: core.epanet.Project and core.swmm.Project. These classes use the shared code in InputFile plus application-specific logic to manage each kind of input file.

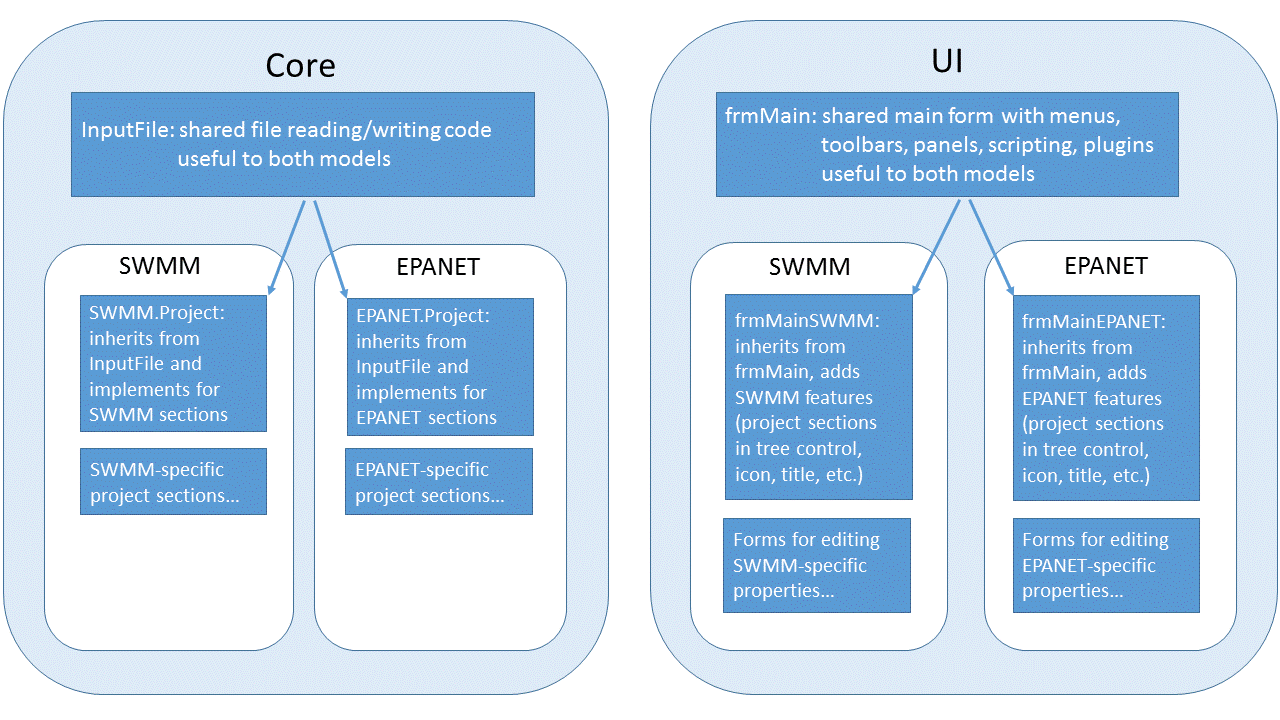


Figure 5. Class Architecture for SWMM/EPANET

A parallel example in the user interface group is ui.frmMain which is at the top level of ui and contains the shared definition of how a main form looks and acts. At the next level in the hierarchy, ui.epanet.frmMain and ui.swmm.frmMain each inherit the shared layout and functionality from frmMain and add the customizations needed to give the user an application-specific experience. For example, the window title and model data tree diagram are populated in the application-specific classes.

This design allows maximum sharing of common code between SWMM and EPANET applications without having any code from the “other” application get in the way. A programmer who is only interested in EPANET can safely ignore the SWMM-specific core classes and user interface and never have to step through code that references SWMM.

Note: Class documentation produced by Doxygen can be viewed in a browser using the following link:

<http://rawgit.com/USEPA/SWMM-EPANET_User_Interface/master/doc/Doxygen/html/annotated.html>

The class design for the SWMM/EPANET data model follows the conceptual model of SWMM and EPANET in terms of representing hydrologic/hydraulic compartments and physical processes. A schematic diagram for the data model to be used in this project to organize various objects and their properties is shown below in Figure 6:

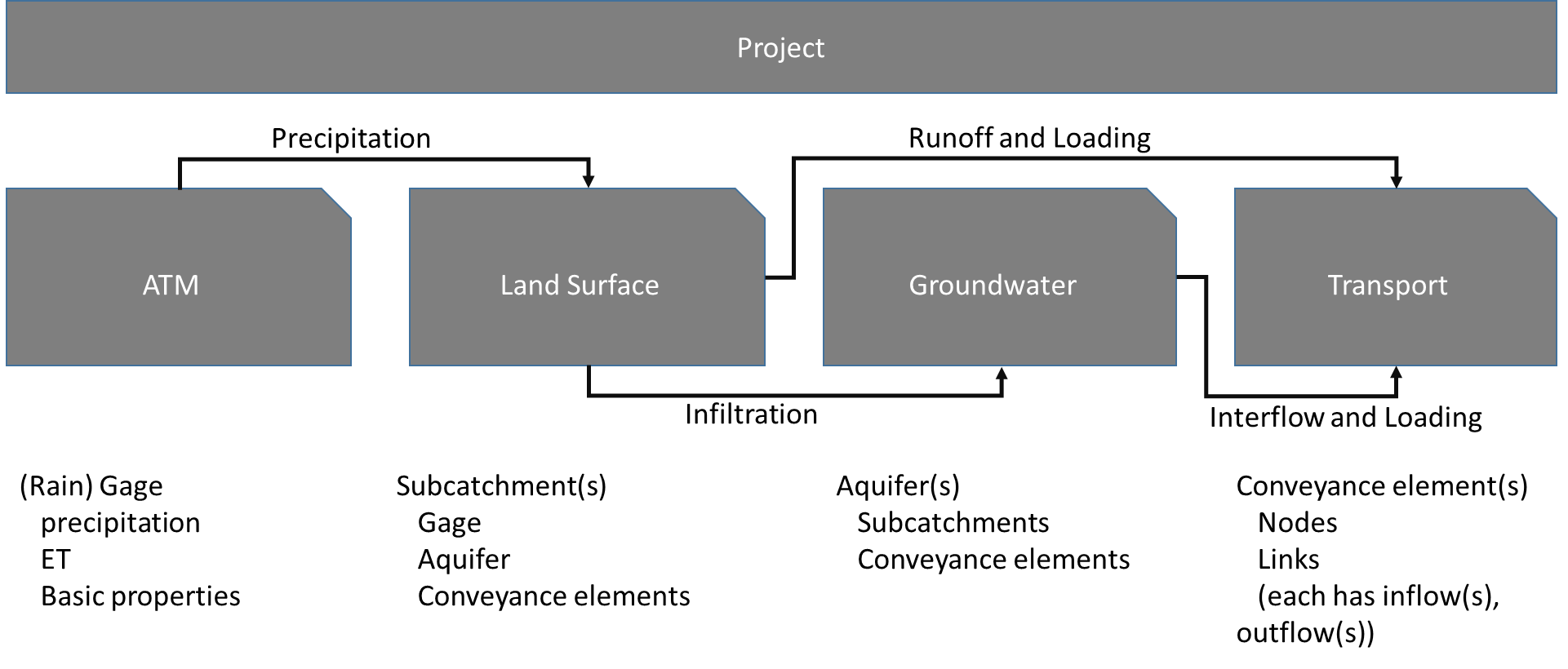


Figure 6. Conceptual data model for SWMM/EPANET

Note that EPANET encompasses only the Transport portion of the diagram above.

## SWMM CORE CLASSES

Core classes have been developed to provide a data model for interacting with SWMM input specifications and data. The data model is provided below. The second column of the model provides the type or valid values for each entity.

**SWMM Project**

Title

Text string

Options

General

FlowUnits CFS / GPM / MGD / CMS / LPS / MLD

Infiltration HORTON / MODIFIED\_HORTON / GREEN\_AMPT / MODIFIED\_GREEN\_AMPT / CURVE\_NUMBER

FlowRouting STEADY / KINWAVE / DYNWAVE

LinkOffsets DEPTH / ELEVATION

IgnoreRainfall YES / NO

IgnoreSnowmelt YES / NO

IgnoreGroundwater YES / NO

IgnoreRouting YES / NO

IgnoreQuality YES / NO

AllowPonding YES / NO

MinSlope real

TempDir string

Dates

StartDate month/day/year

StartTime hours:minutes

ReportStartDate month/day/year

ReportStartTime hours:minutes

EndDate month/day/year

EndTime hours:minutes

SweepStart month/day

SweepEnd month/day

DryDays integer

Time Steps

SkipSteadyState YES / NO

ReportStep hours:minutes:seconds

WetStep hours:minutes:seconds

DryStep hours:minutes:seconds

RoutingStep seconds

Dynamic Wave

InertialDamping NONE / PARTIAL / FULL

NormalFlowLimited SLOPE / FROUDE / BOTH

ForceMainEquation H-W / D-W

LengtheningStep real

VariableStep real

MinSurfaceArea real

Interface files

UseRainfall string

SaveRainfall string

UseRunoff string

SaveRunoff string

UseHotstart string

SaveHotstart string

UseRDII string

SaveRDII string

UseInflows string

SaveOutflows string

Reporting

Input YES / NO

Continuity YES / NO

FlowStats YES / NO

Controls YES / NO

Subcatchments ALL / NONE / <List>

Nodes ALL / NONE / <List>

Links ALL / NONE / <List>

Map

Dimensions real array

Units FEET / METERS / DEGREES / NONE

Backdrop

Dimensions real array

File string

Climatology

Temperature

Source TIMESERIES or FILE

Timeseries [Subclass Timeseries]

ClimateFile string

ClimateFileStartDate string

Evaporation

Format CONSTANT, MONTHLY, TIMESERIES, TEMPERATURE, or FILE

ConstantValue real

MonthlyValues real array

Timeseries [Subclass Timeseries]

MonthlyPanCoefficients real array

RecoveryPattern [Subclass Pattern]

DryOnly NO / YES

Wind Speed

Source MONTHLY or FILE

MonthlyValues real array

Snow Melt

SnowTemp real

ATIWeight real

NegativeMeltRatio real

Elevation real

Latitude real

TimeCorrection real

Areal Depletion

ImperviousFractions real array

PerviousFractions real array

Adjustments

Temperature real array

Evaporation real array

Rain real array

SoilConductivity real array

Hydrology

Rain Gages [Collection of Subclass RainGage]

Subcatchments [Collection of Subclass Subcatchment]

Aquifers [Collection of Subclass Aquifer]

Snow Packs [Collection of Subclass SnowPack]

Unit Hydrographs [Collection of Subclass UnitHydrograph]

LID Controls [Collection of Subclass LIDControl]

Hydraulics

Nodes

Junctions [Collection of Subclass JunctionNode]

Outfalls [Collection of Subclass OutfallNode]

Dividers [Collection of Subclass DividerNode]

Storage Units [Collection of Subclass StorageNode]

Links

Conduits [Collection of Subclass Conduit]

Pumps [Collection of Subclass Pump]

Orifices [Collection of Subclass Orifice]

Weirs [Collection of Subclass Weir]

Outlets [Collection of Subclass Outlet]

Transects [Collection of Subclass Transect]

Controls [Collection of Subclass ControlRule]

Quality

Pollutants [Collection of Subclass Pollutant]

Land Uses [Collection of Subclass Landuse]

Curves [Collection of Subclass Curve]

Time Series [Collection of Subclass TimeSeries]

Time Patterns [Collection of Subclass Pattern]

Map Labels [Collection of Subclass Label]

**SWMM Hydrology**

SubClass **RainGage**

Name string

Centroid [Subclass Vertex]

Description string

Tag string

Rain Format INTENSITY, VOLUME, or CUMULATIVE

Rain Interval hours:minutes

Snow Catch Factor real

Data Source TIMESERIES or FILE

Timeseries [Subclass Timeseries]

Data File Name string

Data File Station ID string

Data File Rain Units IN or MM

SubClass **Subcatchment**

Name string

Centroid [Subclass Vertex]

PolygonVertices [Collection of Subclass Vertex]

Description string

Tag string

Raingage [Subclass Raingage]

Outlet [Subclass Node]

Area real

PercentImperv real

Width real

Slope real

NImperv real

NPerv real

DstoreImperv real

DstorePerv real

PercentZero real

SubareaRouting IMPERVIOUS, PERVIOUS, or OUTLET

PercentRouted real

HortonMaxRate real

HortonMinRate real

HortonDecay real

HortonDryTime real

HortonMaxInf real

GreenAmptSuction real

GreenAmptConduct real

GreenAmptInitDef real

SCSCurveNo real

SCSConduct real

SCSDryTime real

Groundwater [Subclass Groundwater]

SnowPack [Subclass SnowPack]

LIDUsages [Collection of Subclass LIDUsage]

Coverages [Collection of Subclass Coverage]

InitialLoadings [Collection of Subclass InitialLoading]

CurbLength real

SubClass **Groundwater**

Aquifer [Subclass Aquifer]

Receiving Node string

Surface Elevation real

Groundwater Flow Coefficient real

Groundwater Flow Exponent real

Surface Water Flow Coefficient real

Surface Water Flow Exponent real

Surface-GW Interaction Coefficient real

Fixed Surface Water Depth real

Threshold Groundwater Elevation real

SubClass **LIDUsage**

ControlName string

NumberReplicateUnits integer

AreaEachUnit real

TopWidthOverlandFlowSurface real

PercentInitiallySaturated real

PercentImperviousAreaTreated real

SendOutflowPerviousArea integer

DetailedReportFile string

SubClass **Coverage**

LanduseName string

PercentSubcatchmentArea real

SubClass **InitialLoading**

PollutantName string

InitialBuildup real

SubClass **Aquifer**

Name string

Porosity real

WiltingPoint real

FieldCapacity real

Conductivity real

ConductSlope real

TensionSlope real

UpperEvapFraction real

LowerEvapDepth real

LowerGWLossRate real

BottomElevation real

WaterTableElevation real

UnsatZoneMoisture real

UpperEvapPattern [Subclass Pattern]

SubClass **SnowPack**

Name string

PlowableMinMeltCoefficient real

PlowableMaxMeltCoefficient real

PlowableBaseTemperature real

PlowableFractionFreeWaterCapacity real

PlowableInitialSnowDepth real

PlowableInitialFreeWater real

PlowableFractionImperviousArea real

ImperviousMinMeltCoefficient real

ImperviousMaxMeltCoefficient real

ImperviousBaseTemperature real

ImperviousFractionFreeWaterCapacity real

ImperviousInitialSnowDepth real

ImperviousInitialFreeWater real

ImperviousDepth100Cover real

PerviousMinMeltCoefficient real

PerviousMaxMeltCoefficient real

PerviousBaseTemperature real

PerviousFractionFreeWaterCapacity real

PerviousInitialSnowDepth real

PerviousInitialFreeWater real

PerviousDepth100Cover real

DepthSnowRemovalBegins real

FractionTransferredOutWatershed real

FractionTransferredImperviousArea real

FractionTransferredPerviousArea real

FractionConvertedImmediateMelt real

FractionMovedAnotherSubcatchment real

SubcatchmentTransfer string

SubClass **UnitHydrograph**

GroupName string

RainGageUsed [Subclass RainGage]

HydrographMonths string

UnitHydrographValues real array

InitialAbstractionDepth real

InitialAbstractionRate real

InitialAbstractionAmount real

SubClass **LIDControl**

ControlName string

LIDType BC for bio-retention cell; PP for porous pavement; IT for infiltration trench; RB for rain barrel; VS for vegetative swale

HasSurfaceLayer YES/NO

HasPavementLayer YES/NO

HasSoilLayer YES/NO

HasStorageLayer YES/NO

HasUnderdrainLayer YES/NO

SurfaceLayerStorageDepth real

SurfaceLayerVegetativeCoverFraction real

SurfaceLayerSurfaceRoughness real

SurfaceLayerSurfaceSlope real

SurfaceLayerSwaleSideSlope real

PavementLayerThickness real

PavementLayerVoidRatio real

PavementLayerImperviousSurfaceFraction real

PavementLayerPermeability real

PavementLayerClogging Factor real

SoilLayerThickness real

SoilLayerPorosity real

SoilLayerFieldCapacity real

SoilLayerWiltingPoint real

SoilLayerConductivity real

SoilLayerConductivitySlope real

SoilLayerSuctionHead real

StorageLayerHeight real

StorageLayerVoidRatio real

StorageLayerFiltrationRate real

StorageLayerCloggingFactor real

DrainCoefficient real

DrainExponent real

DrainOffsetHeight real

DrainDelay real

**SWMM Hydraulics**

Class **Node**

Name string

Centroid [Subclass Vertex]

Description string

Tag string

Direct Inflows [Collection of Subclass DirectInflow]

Dry Weather Inflows [Collection of Subclass DryWeatherInflow]

RDIInflows [Collection of Subclass RDIInflow]

Treatments [Collection of Subclass Treatment]

InvertElev real

SubClass **DirectInflow**

Constituent string

Timeseries [Subclass Timeseries]

Format CONCEN or MASS

ConversionFactor real

ScaleFactor real

Baseline real

BaselinePattern [Subclass Pattern]

SubClass **DryWeatherInflow**

Constituent string

Average real

TimePattern [Subclass Pattern]

SubClass **RDIInflow**

HydrographGroup string

SewershedArea real

SubClass **Treatment**

Pollutant string

Result C or R(C – function computes effluent concentration, R – function computes fractional removal)

Function string

SubClass **Junction** (Inherits Node)

MaxDepth real

InitialDepth real

SurchargeDepth real

PondedArea real

SubClass **Outfall** (Inherits Node)

TideGate boolean

Type FREE, NORMAL, FIXED, TIDAL, or TIMESERIES

FixedStage real

TidalCurve [Subclass Curve]

Timeseries [Subclass Timeseries]

SubClass **Divider** (Inherits Node)

DivertedLink [Subclass Link]

Type OVERFLOW, CUTOFF, TABULAR, or WEIR

MaxDepth real

InitialDepth real

SurchargeDepth real

PondedArea real

CutoffFlow real

DividerCurve [Subclass Curve]

WeirMinFlow real

WeirHeight real

WeirCoefficient real

SubClass **StorageUnit** (Inherits Node)

MaxDepth real

InitialDepth real

Type TABULAR or FUNCTIONAL

StorageCurve [Subclass Curve]

Coefficient real

Exponent real

Constant real

PondedArea real

EvapFactor real

SeepageLoss YES, NO

SeepageSuctionHead real

SeepageHydraulicConductivity real

SeepageInitialMoistureDeficit real

Class **Link**

Name string

InletNode [Subclass Node]

OutletNode [Subclass Node]

Description string

Tag string

Vertices [Collection of Subclass Vertex]

SubClass **Conduit** (Inherits Link)

Length real

Roughness real

InletOffset real

OutletOffset real

InitialFlow real

MaximumFlow real

CrossSection [Subclass CrossSection]

EntryLossCoeff real

ExitLossCoeff real

LossCoeff real

FlapGate boolean

SubClass **Pump** (Inherits Link)

PumpCurve [Subclass Curve]

InitialStatus real

StartupDepth real

ShutoffDepth real

SubClass **Orifice** (Inherits Link)

Type SIDE or BOTTOM

CrossSection [Subclass CrossSection]

InletOffset real

DischargeCoeff real

FlapGate YES, NO

Orate real

SubClass **Weir** (Inherits Link)

Type TRANSVERSE, SIDEFLOW, V-NOTCH, TRAPEZOIDAL, or ROADWAY

CrossSection [Subclass CrossSection]

InletOffset real

DischargeCoeff real

FlapGate YES or NO

EndContractions real

EndCoeff real

CanSurcharge boolean

RoadWidth real

RoadSurface PAVED or GRAVEL

SubClass **Outlet** (Inherits Link)

InletOffset real

FlapGate YES or NO

Coefficient real

Exponent real

CurveType TABULAR/DEPTH, TABULAR/HEAD, FUNCTIONAL/DEPTH, FUNCTIONAL/HEAD

RatingCurve [Subclass Curve]

SubClass **CrossSection**

Shape CIRCULAR, FORCE\_MAIN, FILLED\_CIRCULAR2, RECT\_CLOSED, RECT\_OPEN, TRAPEZOIDAL, TRIANGULAR, HORIZ\_ELLIPSE, VERT\_ELLIPSE, ARCH (standard), ARCH (non-standard), PARABOLIC, POWER, RECT\_TRIANGULAR, RECT\_ROUND, MODBASKETHANDLE, EGG, HORSESHOE, GOTHIC, CATENARY, SEMIELLIPTICAL, BASKETHANDLE, SEMICIRCULAR, CUSTOM, or IRREGULAR

Geometry1 real

Geometry2 real

Geometry3 real

Geometry4 real

Barrels integer

Culvert integer

Curve [Subclass Curve]

Transect [Subclass Transect]

SubClass **Transect**

Name string

Description string

StationElevationDataGrid real array

RoughnessLeftBank real

RoughnessRightBank real

RoughnessChannel real

BankStationLeft real

BankStationRight real

StationsModifier real

ElevationsModifier real

MeanderModifier real

SubClass **ControlRule**

RuleID string

RuleText string

**SWMM Supporting Classes**

SubClass **Curve**

Name string

Type STORAGE / SHAPE / DIVERSION / TIDAL / PUMP1 / PUMP2 / PUMP3 / PUMP4 /RATING / CONTROL

XValues real array

YValues real array

SubClass **Label**

Centroid [Subclass Vertex]

Text string

AnchorID string

Font string

Size real

Bold YES/NO

Italic YES/NO

SubClass **Landuse**

Name string

Description string

StreetSweepingInterval real

StreetSweepingAvailability real

LastSwept real

Buildups [Collection of Subclass Buildup]

Washoffs [Collection of Subclass Washoff]

SubClass **Buildup**

Pollutant string

Function POW / EXP / SAT / EXT

MaxBuildup real

RateConstant real

PowerSatConstant real

ScalingFactor real

TimeSeries [Subclass Timeseries]

NormalizerName AREA or CURBLENGTH

SubClass **Washoff**

Pollutant string

Function EXP / RC / EMC

Coefficient real

Exponent real

Cleaning Efficiency real

BMP Efficiency real

SubClass **Pattern**

Name string

Type MONTHLY, DAILY, HOURLY, or WEEKEND

Description string

Multipliers real

SubClass **Pollutant**

Name string

Units MG/L, UG/L, #/L

RainConcentration real

GWConcentration real

I&IConcentration real

DWFConcentration real

DecayCoefficient real

SnowOnly YES or NO

Co-Pollutant string

Co-Fraction real

SubClass **Timeseries**

Name string

Description string

DateTime string array

Value real array

File string

SubClass **Vertex**

XCoordinate real

YCoordinate real

## EPANET CORE CLASSES

Similarly, core classes have been developed to provide a data model for interacting with EPANET input specifications and data. The data model is provided below. The second column of the model provides the type or valid values for each entity.

**EPANET Project**

Title

Text string

Options

Hydraulics

FlowUnits CFS/GPM/MGD/IMGD/AFD/LPS/LPM/MLD/CMH/CMD

HeadLossFormula H-W/D-W/C-M

SpecificGravity real

RelativeViscosity real

MaximumTrials integer

Accuracy real

Unbalanced STOP/CONTINUE

UnbalancedContinue integer

DefaultPattern [Subclass Pattern]

DemandMultiplier real

EmitterExponent real

CheckFrequency real

MaxCheck real

DampLimit real

Hydraulics USE/SAVE

HydraulicsFile string

Quality

Quality NONE/CHEMICAL/AGE/TRACE

ChemicalName string

MassUnits string

RelativeDiffusivity real

TraceNode string

Tolerance real

Reactions

OrderBulk real

OrderWall real

OrderTank real

GlobalBulk real

GlobalWall real

LimitingPotential real

RoughnessCorrelation real

Times

Duration hours:minutes

HydraulicTimestep hours:minutes

QualityTimestep hours:minutes

RuleTimestep hours:minutes

PatternTimestep hours:minutes

PatternStart hours:minutes

ReportTimestep hours:minutes

ReportStart hours:minutes

StartClocktime hours:minutes AM/PM

Statistic NONE/AVERAGED/MINIMUM/MAXIMUM/RANGE

Energy

GlobalPrice real

GlobalPattern real

GlobalEfficiency real

DemandCharge real

Report

Pagesize integer

File string

Status YES/NO/FULL

Summary YES/NO

Energy YES/NO

Parameters string

Map

MapFile string

Backdrop

Dimensions real

Units FEET/METERS/DEGREES/NONE

File string

OffsetX real

OffsetY real

Hydraulics

Nodes

Junctions [Collection of Subclass Junction]

Reservoirs [Collection of Subclass Reservoir]

Tanks [Collection of Subclass Tank]

Links

Pipes [Collection of Subclass Pipe]

Pumps [Collection of Subclass Pump]

Valves [Collection of Subclass Valve]

Controls [Collection of Subclass Control]

Curves [Collection of Subclass Curve]

Time Patterns [Collection of Subclass Pattern]

Map Labels [Collection of Subclass Label]

**EPANET Hydraulics**

Class **Node**

Name string

Centroid [Subclass Vertex]

Description string

Tag string

InitialQuality real

Source Quality [Subclass Source]

ReportFlag string

SubClass **Source**

Type CONCEN, MASS, FLOWPACED, or SETPOINT

Baseline real

SourcePattern [Subclass Pattern]

SubClass **Junction** (Inherits Node)

Elevation real

Demands [Collection of Subclass Demand]

EmitterCoefficient real

SubClass **Demand**

BaseDemand real

DemandPattern [Subclass Pattern]

Category string

SubClass **Reservoir** (Inherits Node)

TotalHead real

HeadPattern [Subclass Pattern]

SubClass **Tank** (Inherits Node)

Elevation real

InitialLevel real

MinimumLevel real

MaximumLevel real

Diameter real

MinimumVolume real

VolumeCurve [Subclass Curve]

MixingModel MIXED, 2COMP, FIFO, or LIFO

MixingFraction real

ReactionCoefficient real

Class **Link**

Name string

InletNode [Subclass Node]

OutletNode [Subclass Node]

Description string

Tag string

Vertices [Collection of Subclass Vertex]

ReportFlag string

SubClass **Pipe** (Inherits Link)

Length real

Diameter real

Roughness real

LossCoefficient real

InitialStatus OPEN, CLOSED, or CV

BulkReactionCoefficient real

WallReactionCoefficient real

SubClass **Pump** (Inherits Link)

Type POWER, HEAD, SPEED, or PATTERN

PumpCurve [Subclass Curve]

Power real

Speed real

Pattern [Subclass Pattern]

InitialStatus OPEN or CLOSED

Energy [Subclass PumpEnergy]

SubClass **PumpEnergy**

PricePatternEfficiency PRICE, PATTERN, or EFFIC

Value real

EnergyPattern [Subclass Pattern]

SubClass **Valve** (Inherits Link)

Diameter real

Type PRV, PSV, PBV, FCV, TCV, or GPV

Setting real

LossCoefficient real

ValveCurve [Subclass Curve]

FixedStatus OPEN or CLOSED

SubClass **Control**

SimpleControls [Collection of Subclass SimpleControl]

RuleBasedControls [Collection of Subclass RuleBasedControl]

SubClass **SimpleControl**

LinkID string

Status string

NodeID string

Value real

Time string

Clocktime string

ControlType ABOVE, BELOW, TIME, or CLOCKTIME

SubClass **RuleBasedControl**

RuleID string

RuleText string

**EPANET Supporting Classes**

SubClass **Curve**

Name string

Description string

Type PUMP, EFFICIENCY, VOLUME, or HEADLOSS

XValues real array

YValues real array

SubClass **Label**

Centroid [Subclass Vertex]

Text string

AnchorID string

SubClass **Pattern**

Name string

Description string

Multipliers real array

SubClass **Vertex**

XCoordinate real

YCoordinate real

# 0 SOFTWARE DEMONSTRATIONS

## SCRIPTING

## PLUG-INS

## MAP

## Main Window