BCN Class

Creates and validates BCN objects. Note BCNs can belong to multiple alternatives

# Libraries/Classes calling on BCN Class

Validate/Read

Flows Library

AlternativeSummary Class

TotalRequiredFlows Class

TotalOptionalFLows Class

SensitivitySummary Class

UncertaintySummary Class

# Internal Library/Class dependencies

None

# External Library Dependencies

None

# Functions

init()

validateBCNObject()

updateObjec(“varName”,newValue)

# Pseudo Code

Optional inputs are in italics

Begin Pseudocode

init()

Standard class constructor method. Create object based off of list of inputs developed from json string. Class variables are provided in the following table. The STS document contains more information

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **VarName** | **VarType** | **Type** |
| AlternativeID | altID | Int | Whole digit array; object can apply to more than 1 alternative |
| BCN ID | bcnID | Int | Whole Digit |
| Type | bcnType | String | {Benefit, Cost, NonMonetary} or {0,1,2}; revenue treated as negative cost |
| SubType | bcnSubType | String | {Direct, Indirect, Externality} or {0,1,2} |
| Name | bcnName | String | String |
| Tag (Custom Category) | bcnTag | List of Strings | String array |
| DateofOccurrence | initialOcc | Int | Whole digit |
| NominalorReal | bcnRealBool | Boolean | {Nominal, Real} or {0,1} |
| Investment | bcnInvestBool | Boolean | {Yes,No} or {0,1} |
| InvestmentServiceLife | bcnLife | Int | Digit |
| CalculateResidualValue | rvBool | Bool | {Yes,No} or {0,1}; Yes uses Linear Depreciation while No means the user will input a separate object for Residual Value |
| Recurrence | recurBool | Bool | {Yes,No} or {0,1} |
| RecurrenceInterval | recurInterval | Int | Digit |
| RecurrenceVariabilityRateType | recurVarRate | String | {Percent Delta Timestep X-1} |
| RecurrenceVariabilityRateValues | recurVarValue | List of Floats | Array or Digit (if constant) |
| RecurrenceEndDate | recurEndDate | Date | Date |
| ValuePerQ | valuePerQ | Float | Float? |
| Quantity | quant | Float | Float? |
| QuantityVariabilityRateType | quantVarRate | String | {Fraction Delta Timestep X‑1} |
| QuantityVariabilityRateValues | quantVarValue | List of Floats | Array or Digit (if constant) |
| Quantity Unit | quantUnit | String | String; **If blank, report blank?** |

End init

Define validateBCNObject(objectList)

Verify that all inputs are of the required type and in valid ranges, if not

Return error: Invalid input for analysis object

Based on chosen bcnType, check that all required inputs are included (See table below, we are not going to do fuzzy matching on tag names, it will be the users responsibility to verify all tags are internally consistent)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Required Field** | | | |
| **Variable** | **Benefit** | **Cost** | **NonMonetary** |
| AlternativeID | X | X | X |
| bcnID | X | X | X |
| Type | X | X | X |
| SubType |  |  |  |
| Name |  |  |  |
| Tag (Custom Category) |  |  | Required for non-monetary to simplify measure calculations. |
| DateofOccurrence | X |  |  |
| NominalorReal | X | X |  |
| Investment |  | X |  |
| InvestmentServiceLife |  |  |  |
| CalculateResidualValue |  |  |  |
| Recurrence | X | X | X |
| RecurrenceInterval | Req if recurrence is True | Req if recurrence is True | Req if recurrence is True |
| RecurrenceVariabilityRateType | Req if recurrence is True | Req if recurrence is True | Req if recurrence is True |
| RecurrenceVariabilityRateValues | Req if recurrence is True | Req if recurrence is True | Req if recurrence is True |
| RecurrenceEndDate | Optional, if not provided BCN occurs for entire study period | Optional, if not provided BCN occurs for entire study period | Optional, if not provided BCN occurs for entire study period |
| ValuePerQ | X | X |  |
| Quantity | X | X | X |
| QuantityVariabilityRateType |  |  |  |
| QuantityVariabilityRateValues | Req. if QuantityVariabilityRateType exists | Req. if QuantityVariabilityRateType exists | Req. if QuantityVariabilityRateType exists |
| Quantity Unit | If not provided it is assumed the value is in dollars | If not provided it is assumed the value is in dollars | X |

if not

Return error: Invalid input for Analysis object using <analysisType> type

Else

Return: self **OR** call without assignment and return nothing

Valid Ranges for numeric input

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **VarName** | **VarType** | **Type** |
| AlternativeID | altID | Int | N/A |
| BCN ID | bcnID | Int | Whole Digit must be unique to bcn |
| Type | bcnType | String | N/A |
| SubType | bcnSubType | String | N/A |
| Name | bcnName | String | N/A |
| Tag (Custom Category) | bcnTag | List of Strings | String array |
| DateofOccurrence | initialOcc | Int | Whole digit, >= 0 must occur at a valid timestep and be less than the study period |
| NominalorReal | bcnRealBool | Boolean | N/A |
| Investment | bcnInvestBool | Boolean | N/A |
| InvestmentServiceLife | bcnLife | Int | >=1, counts the index of the timestep, not the timestep itself  Can be greater than study period |
| CalculateResidualValue | rvBool | Bool | N/A |
| Recurrence | recurBool | Bool | N/A |
| RecurrenceInterval | recurInterval | Int | >=1 if it exists, counts the index of the timestep, not the timestep itself |
| RecurrenceVariabilityRateType | recurVarRate | String | N/A |
| RecurrenceVariabilityRateValues | recurVarValue | List of Floats | Individual values should be in terms of decimal, |
| RecurrenceEndDate | recurEndDate | Date | Must be later than initial occur and less than or equal to study period |
| ValuePerQ | valuePerQ | Float | Any signed float |
| Quantity | quant | Float | Any signed float |
| QuantityVariabilityRateType | quantVarRate | String | Individual values should be in terms of decimal, |
| QuantityVariabilityRateValues | quantVarValue | List of Floats | Any signed float |
| Quantity Unit | quantUnit | String | String; **If blank, report blank?** |

End validateBCNObject

updateObject(“varName”,newValue)

use varName and newValue to change BCN variable value

end updateObject