Requested Changes to 61970 and 61968 classes associated with ACLineSegment modeling

From the Device Datasheet Breakout Group

29 November, 2021

Changes to UML classes, attributes, and associations

61970 changes

Delete attribute PhaseImpedanceData.fromPhase

Delete attribute PhaseImpedanceData.toPhase

61968 changes

Change class name WireAssemblyInfo to WireAssembly

Change association role name PerLengthLineParameter.WireAssemblyInfo to PerLengthLineParameter.WireAssembly *[NOTE: also ensure that association has WireAssembly as source and PerLengthLineParameter as target since it crosses the 61968/61970 package boundary]*

Add association ACLineSegment.WireAssembly, *ensuring that association has WireAssembly as source and ACLineSegment as target since it crosses the 61968/61970 package boundary*

Add association WirePosition.WireAssembly, *ensuring that association has WirePosition as source and WireAssembly as target*

Change inheritance WireAssembly to inherit from IdentifiedObject instead of AssetInfo

Add association WirePosition.WireInfo *ensuring that association has WirePosition as source and WireInfo as target*

Add attribute WirePosition.phase

Delete class WirePhaseInfo

Delete association WirePhaseInfo.WireAssemblyInfo

Delete association WirePhaseInfo.WirePosition

Delete association WirePhaseInfo.WireInfo

Delete attribute WireSpacingInfo.isCable

Delete attribute WireSpacingInfo.usage

Delete class WireUsageKind

Change class name WireSpacingInfo to IntraphaseSpacing

Change attribute name WireSpacingInfo.phaseWireSpacing to IntraphaseSpacing.wireSpacing

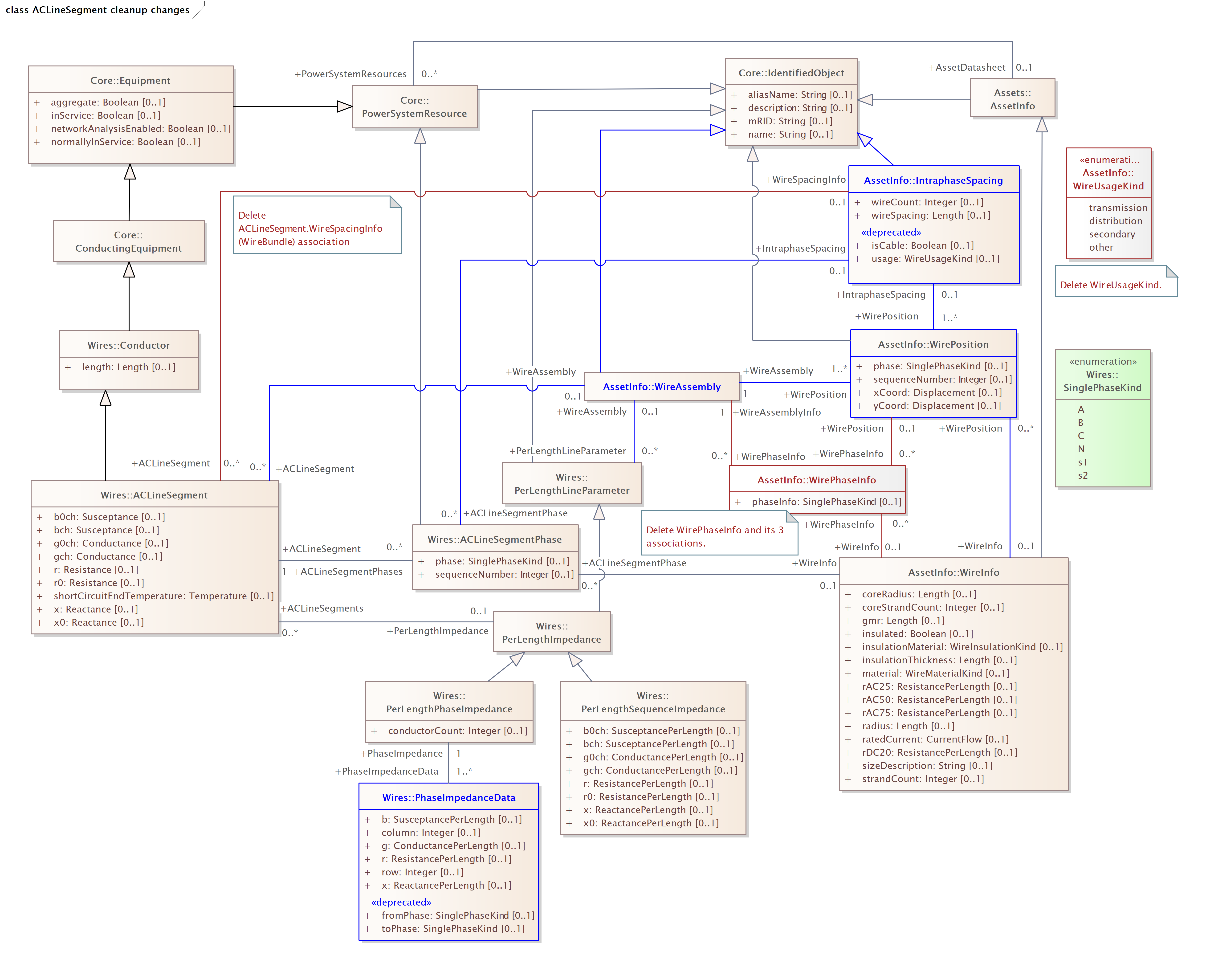
Change attribute name WireSpacingInfo.phaseWireCount to IntraphaseSpacing.wireCount

Delete association ACLineSegment.WireSpacingInfo

Change association role name WirePositions.WireSpacingInfo to WirePosition.IntraphaseSpacing *[NOTE: also ensure that association has WirePosition as source and IntraphaseSpacing as target]*

Add association IntraphaseSpacing.ACLineSegmentPhase *ensuring that association has IntraphaseSpacing as source and ACLineSegment as target since it crosses the 61968/61970 package boundary*

Change inheritance IntraphaseSpacing to inherit from IdentifiedObject instead of AssetInfo



Changes to UML descriptions

*Associations*

WireAssembly.PerLengthLineParameter

Wire Assembly - Wire assembly describing wire positions and optionally types, intraphase spacing and phases of wires with this per length line parameter.

PerLengthLineParameter - Per length line parameter associated with this wire assembly.

Delete existing description on association itself: A WireAssemblyInfo is used to compute the PerLengthParameter data in the Wires package

WireAssembly.ACLineSegment

Wire Assembly - Wire assembly describing wire positions and optionally types, intraphase spacing and phases of the wires comprising this line segement.

ACLineSegment – Line segment whose lines are described by this wire assembly.

WirePosition.WireAssembly

WirePosition – The position of one of the phase carrying or neutral wires making up this wire assembly.

Wire Assembly - Wire assembly to which this wire position belongs.

WirePosition.WireInfo

Wire Position - The position occupied by a wire with these characteristics.

WireInfo - The characteristics of the wire at this position.

WirePosition.IntraphaseSpacing

IntraphaseSpacing - Wire spacing of set of wires in this wire position.

WirePosition - Position at which this set of spaced wires is located.

ACLineSegmentPhase.IntraphaseSpacing

ACLineSegmentPhase - Line segment phase with which this set of spaced wires is associated.

IntraphaseSpacing - Wire spacing of set of wires associated with this line segment phase.

*Classes and Attributes*

Key:

## Green – 61970 class description changes

Blue – 61968 class description changes

## ACLineSegment

Class in package 'Wires'

Was: A wire or combination of wires, with consistent electrical characteristics, building a single electrical system, used to carry alternating current between points in the power system.

For symmetrical, transposed three phase lines, it is sufficient to use attributes of the line segment, which describe impedances and admittances for the entire length of the segment. Additionally impedances can be computed by using length and associated per length impedances.

The BaseVoltage at the two ends of ACLineSegments in a Line shall have the same BaseVoltage.nominalVoltage. However, boundary lines may have slightly different BaseVoltage.nominalVoltages and variation is allowed. Larger voltage difference in general requires use of an equivalent branch.

Requested: A line segment is a conductor or combination of conductors, with consistent electrical characteristics along its length, building a single electrical system that carries alternating current between two points in the power system.

The BaseVoltage at the two ends of a line segment shall have the same BaseVoltage.nominalVoltage. However, boundary lines may have slightly different BaseVoltage.nominalVoltages and variation is allowed. Larger voltage difference in general requires use of an equivalent branch.

Line segment impedances can be either directly described in electrical terms or physical line detail can be provided from which impedances can be calculated.

**Directly described impedances**

For symmetrical, transposed three phase line segments, it is sufficient to use attributes of the line segment, which describe impedances and admittances for the entire length of the line segment. Additionally, line segment impedances can be computed by using line segment length and associated per length impedances.

Unbalanced modeling of impedances is supported by the per length phase impedance matrix (PerLengthPhaseImpedance) in conjunction with phase-to-sequence number mapping supplied by either ACLineSegmentPhase or WirePosition. The sequence numbers are referenced by the row and column attributes of the per length phase impedance matrix. This method enables single-phase and two-phase line segments, and transpositions of phases, to be described using the same per length phase impedance matrix. The length of the line segment is used in the computation of total impedance values for the line segment.

**Line detail characteristics**

There are three approaches to providing line detail and all use WireAssembly to supply line positions:

* Option 1 - WireAssembly supplies only line positions. ACLineSegmentPhase points to wire type and intraphase spacing and supplies the phase-to-sequence number mapping.
* Option 2 - WireAssembly supplies line position and, for each position, also supplies wire type and intraphase spacing. ACLineSegmentPhase supplies the phase-to-sequence number mapping.
* Option 3 - WireAssembly supplies line position and, for each position, also supplies wire type and intraphase spacing and phase. WireAssembly therefore supplies the phase-to-sequence number mapping and ACLineSegmentPhase is not needed.

| ATTRIBUTES |
| --- |
| b0ch : Susceptance Public  Multiplicity: ( [0..1], Allow duplicates: , Is ordered: False )  Was: Zero sequence shunt (charging) susceptance, uniformly distributed, of the entire line section.  Requested: Zero sequence shunt (charging) susceptance, uniformly distributed, of the entire line segment.  [ Is static True. Containment is . ] |
| bch : Susceptance Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: Positive sequence shunt (charging) susceptance, uniformly distributed, of the entire line section. This value represents the full charging over the full length of the line.  Requested: Positive sequence shunt (charging) susceptance, uniformly distributed, of the entire line segment. This value represents the full charging over the full length of the line segment.  [ Is static True. Containment is Not Specified. ] |
| g0ch : Conductance Public  Multiplicity: ( [0..1], Allow duplicates: , Is ordered: False )  Was: Zero sequence shunt (charging) conductance, uniformly distributed, of the entire line section.  Requested: Zero sequence shunt (charging) conductance, uniformly distributed, of the entire line segment.  [ Is static True. Containment is . ] |
| gch : Conductance Public  Multiplicity: ( [0..1], Allow duplicates: , Is ordered: False )  Was: Positive sequence shunt (charging) conductance, uniformly distributed, of the entire line section.  Requested: Positive sequence shunt (charging) conductance, uniformly distributed, of the entire line segment.  [ Is static True. Containment is . ] |
| r : Resistance Public  Multiplicity: ( [0..1], Allow duplicates: , Is ordered: False )  Was: Positive sequence series resistance of the entire line section.  Requested: Positive sequence series resistance of the entire line segment.  [ Is static True. Containment is . ] |
| r0 : Resistance Public  Multiplicity: ( [0..1], Allow duplicates: , Is ordered: False )  Was: Zero sequence series resistance of the entire line section.  Requested: Zero sequence series resistance of the entire line segment.  [ Is static True. Containment is . ] |
| shortCircuitEndTemperature : Temperature Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Maximum permitted temperature at the end of SC for the calculation of minimum short-circuit currents. Used for short circuit data exchange according to IEC 60909.  [ Is static True. Containment is Not Specified. ] |
| x : Reactance Public  Multiplicity: ( [0..1], Allow duplicates: , Is ordered: False )  Was: Positive sequence series reactance of the entire line section.  Requested: Positive sequence series reactance of the entire line segment.  [ Is static True. Containment is . ] |
| x0 : Reactance Public  Multiplicity: ( [0..1], Allow duplicates: , Is ordered: False )  Was: Zero sequence series reactance of the entire line section.  Requested: Zero sequence series reactance of the entire line segment.  [ Is static True. Containment is . ] |

## ACLineSegmentPhase

Class in package 'Wires'

Was: Represents a single wire of an alternating current line segment.

Requested: A line segment phase represents one phase (or optionally the neutral) of an alternating current line segment.

Under most circumstances there is not a line segment phase for the neutral. However, if a wire assembly is being used and it does not specify phase, a line segment phase must exist for each position in the assembly (including the neutral).

| ATTRIBUTES |
| --- |
| phase : SinglePhaseKind Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: The phase connection of the wire at both ends.  Requested: Phase identifies the phase or neutral connection of the conductor at both ends.  [ Is static True. Containment is Not Specified. ] |
| sequenceNumber : Integer Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: Number designation for this line segment phase. Each line segment phase within a line segment should have a unique sequence number. This is useful for unbalanced modelling to bind the mathematical model (PhaseImpedanceData of PerLengthPhaseImpedance) with the connectivity model (this class) and the physical model (WirePosition) without tight coupling.  Requested: Sequence number provides the numbering for this line segment phase. When impedance is directly described, sequence number binds the phase to the column and row for entries in the per length phase impedance matrix (PhaseImpedanceData of PerLengthPhaseImpedance). When used in conjunction with a wire assembly description, sequence number associates the line segment phase to a position.  For a line segment not a part of a parallel line segment group, sequence number starts at one and increments by one through each line segment phase associated with the line segment. If there is a line segment phase for neutral, it is numbered last.  [ Is static True. Containment is Not Specified. ] |

## Conductor

Class in package 'Wires'

Combination of conducting material with consistent electrical characteristics, building a single electrical system, used to carry current between points in the power system.

| ATTRIBUTES |
| --- |
| length : Length Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: Segment length for calculating line section capabilities.  Requested: Segment length for calculating line segment capabilities.  [ Is static True. Containment is Not Specified. ] |

## PerLengthImpedance

Class in package 'Wires'

Was: Common type for per-length impedance electrical catalogues.

Requested: Common type for per-length electrical impedances.

## PerLengthLineParameter

Class in package 'Wires'

Was: Common type for per-length electrical catalogues describing line parameters.

Requested: Common type for per-length electrical line parameters.

## PerLengthPhaseImpedance

Class in package 'Wires'

Was: Impedance and admittance parameters per unit length for n-wire unbalanced lines, in matrix form.

Requested: The per length phase impedance matrix expresses impedance and admittance parameters per unit length for n-conductor unbalanced line segments. A phase impedance matrix contains both self impedances for each phase and mutual impedances between pairs of phases. The matrix is stored in symmetric lower triangular format where the diagonal entries represent self-impedances (and have the same value in row and column) and the off diagonal entries represent phase-to-phase impedances (and have different row and column values).

The matrix can be use to express impedances for both non-coupled and coupled line segments. Coupled line segments share a single per length phase impedance matrix whose entries reflect the self and mutual impedances of all the phases of all the wires.

| ATTRIBUTES |
| --- |
| conductorCount : Integer Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: Number of phase, neutral, and other wires retained. Constrains the number of matrix elements and the phase codes that can be used with this matrix.  Requested: Conductor count identifies the number of phase, neutral, and other wires represented in the phase impedance matrix and constrains the number of elements in the matrix. Typically, Kron reduction is used for incorporating effects of neutral impedance in the phase conductor values (so conductor count only considers phase wires), but neutrals can be included in the matrix (in which case conductor count would include the neutral wire(s)). For a phase impedance matrix associated with the lines of a coupled line segment group, the conductor count reflects the total of the conductors associated with all the coupled line segments in the group.  [ Is static True. Containment is Not Specified. ] |

## PerLengthSequenceImpedance

Class in package 'Wires'

Was: Sequence impedance and admittance parameters per unit length, for transposed lines of 1, 2, or 3 phases. For 1-phase lines, define x=x0=xself. For 2-phase lines, define x=xs-xm and x0=xs+xm.

Requested: Sequence impedance and admittance parameters per unit length, for transposed line segments of 1, 2, or 3 phases. For 1-phase line segments, define x = x0 = xself. For 2-phase line segments, define x = xself - xmutual and x0 = xself + xmutual.

| ATTRIBUTES |
| --- |
| b0ch : SusceptancePerLength Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Zero sequence shunt (charging) susceptance, per unit of length.  [ Is static True. Containment is Not Specified. ] |
| bch : SusceptancePerLength Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Positive sequence shunt (charging) susceptance, per unit of length.  [ Is static True. Containment is Not Specified. ] |
| g0ch : ConductancePerLength Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Zero sequence shunt (charging) conductance, per unit of length.  [ Is static True. Containment is Not Specified. ] |
| gch : ConductancePerLength Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Positive sequence shunt (charging) conductance, per unit of length.  [ Is static True. Containment is Not Specified. ] |
| r : ResistancePerLength Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Positive sequence series resistance, per unit of length.  [ Is static True. Containment is Not Specified. ] |
| r0 : ResistancePerLength Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Zero sequence series resistance, per unit of length.  [ Is static True. Containment is Not Specified. ] |
| x : ReactancePerLength Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Positive sequence series reactance, per unit of length.  [ Is static True. Containment is Not Specified. ] |
| x0 : ReactancePerLength Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Zero sequence series reactance, per unit of length.  [ Is static True. Containment is Not Specified. ] |

## PhaseImpedanceData

Class in package 'Wires'

Was: Impedance and conductance matrix element values.

The diagonal elements are described by the elements having the same toPhase and fromPhase value and the off diagonal elements have different toPhase and fromPhase values. The matrix can also be stored in symmetric lower triangular format using the row and column attributes, which map to ACLineSegmentPhase.sequenceNumber.

Requested: Per length phase impedance matrix entry describes impedance and conductance matrix element values for a specific row and column of the matrix.

The phases to which each entry applies can be determined by means of the row and column attributes which bind to a sequence number provided in either ACLineSegmentPhase or WirePosition (which also specify phase). Due to physical symmetry that is reflected in the matrix, only the lower triangle of the matrix is populated with the row and column method. That is, the column attribute is always less than or equal to the row attribute.

| ATTRIBUTES |
| --- |
| b : SusceptancePerLength Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: Susceptance matrix element value, per length of unit.  Requested: Susceptance matrix entry value, per length of unit.  [ Is static True. Containment is Not Specified. ] |
| column : Integer Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: The matrix element's column number, in the range 1 to row. Only the lower triangle needs to be stored. This column number matches ACLineSegmentPhase.sequenceNumber.  Requested: The matrix entry's column number has a range of possible values from 1 to the conductor count of the matrix, but due to symmetry, only entries in the lower triangle (including diagonal) of the matrix need be defined. Column number binds to the sequence number in either ACLineSegmentPhase or WirePosition, which then identifies the phase for this entry.  [ Is static True. Containment is Not Specified. ] |
| fromPhase : SinglePhaseKind Public **Attribute to be removed**  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: Refer to the class description.  [ Is static True. Containment is Not Specified. ] |
| g : ConductancePerLength Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: Conductance matrix element value, per length of unit.  Requested: Conductance matrix entry value, per length of unit.  [ Is static True. Containment is Not Specified. ] |
| r : ResistancePerLength Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: Resistance matrix element value, per length of unit.  Requested: Resistance matrix entry value, per length of unit.  [ Is static True. Containment is Not Specified. ] |
| row : Integer Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: The matrix element’s row number, in the range 1 to PerLengthPhaseImpedance.conductorCount. Only the lower triangle needs to be stored. This row number matches ACLineSegmentPhase.sequenceNumber.  Requested: The matrix entry's row number has a range of possible values from 1 to the conductor count of the matrix, but due to symmetry, only entries in the lower triangle (including diagonal) of the matrix need be defined. Row number binds to the sequence number in either ACLineSegmentPhase or WirePosition, which then identifies the phase for this entry.  [ Is static True. Containment is Not Specified. ] |
| toPhase : SinglePhaseKind Public **Attribute to be removed**  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: Refer to the class description.  [ Is static True. Containment is Not Specified. ] |
| x : ReactancePerLength Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: Reactance matrix element value, per length of unit.  Requested: Reactance matrix entry value, per length of unit.  [ Is static True. Containment is Not Specified. ] |

## WireAssembly

Class in package 'AssetInfo'

Was: Describes the construction of a multi-conductor wire.<-NOTE: period missing.

Requested: A wire assembly is a collection of wire positions for a line segment. It can be used to represent one or more positions. For example, a line with three phase wires and a neutral would have 4 wire positions, a line with three phase wires and two overhead ground wires would have 5 positions, a line with one phase and a neutral would have 2 positions, a delta-configuration line would have 3 positions, and so on. It can be used for overhead, underground or sub-sea lines. Each position can optionally have wire type, bundling and phase defined for it.

Positions, along with wire type, phase, and optionally other information, are used to calculate line segment impedances.

## CableInfo

Class in package 'AssetInfo'

Was Cable data.

Requested: Cable datasheet.

| ATTRIBUTES |
| --- |
| constructionKind : CableConstructionKind Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Kind of construction of this cable.  [ Is static True. Containment is Not Specified. ] |
| diameterOverCore : Length Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Diameter over the core, including any semi-con screen; should be the insulating layer's inside diameter.  [ Is static True. Containment is Not Specified. ] |
| diameterOverInsulation : Length Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Diameter over the insulating layer, excluding outer screen.  [ Is static True. Containment is Not Specified. ] |
| diameterOverJacket : Length Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Diameter over the outermost jacketing layer.  [ Is static True. Containment is Not Specified. ] |
| diameterOverScreen : Length Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Diameter over the outer screen; should be the shield's inside diameter.  [ Is static True. Containment is Not Specified. ] |
| isStrandFill : Boolean Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  True if wire strands are extruded in a way to fill the voids in the cable.  [ Is static True. Containment is Not Specified. ] |
| nominalTemperature : Temperature Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Maximum nominal design operating temperature.  [ Is static True. Containment is Not Specified. ] |
| outerJacketKind : CableOuterJacketKind Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Kind of outer jacket of this cable.  [ Is static True. Containment is Not Specified. ] |
| sheathAsNeutral : Boolean Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  True if sheath / shield is used as a neutral (i.e., bonded).  [ Is static True. Containment is Not Specified. ] |
| shieldMaterial : CableShieldMaterialKind Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Material of the shield.  [ Is static True. Containment is Not Specified. ] |

## ConcentricNeutralCableInfo

Class in package 'AssetInfo'

Was: Concentric neutral cable data.

Requested: Concentric neutral cable datasheet.

| ATTRIBUTES |
| --- |
| diameterOverNeutral : Length Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Diameter over the concentric neutral strands.  [ Is static True. Containment is Not Specified. ] |
| neutralStrandCount : Integer Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Number of concentric neutral strands.  [ Is static True. Containment is Not Specified. ] |
| neutralStrandGmr : Length Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Geometric mean radius of the neutral strand.  [ Is static True. Containment is Not Specified. ] |
| neutralStrandRadius : Length Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Outside radius of the neutral strand.  [ Is static True. Containment is Not Specified. ] |
| neutralStrandRDC20 : ResistancePerLength Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  DC resistance per unit length of the neutral strand at 20 °C.  [ Is static True. Containment is Not Specified. ] |

## OverheadWireInfo

Class in package 'AssetInfo'

Was: Overhead wire data.

Requested: Overhead wire datasheet.

## TapeShieldCableInfo

Class in package 'AssetInfo'

Was: Tape shield cable data.

Requested: Tape shield cable datasheet.

| ATTRIBUTES |
| --- |
| tapeLap : PerCent Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Percentage of the tape shield width that overlaps in each wrap, typically 10% to 25%.  [ Is static True. Containment is Not Specified. ] |
| tapeThickness : Length Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Thickness of the tape shield, before wrapping.  [ Is static True. Containment is Not Specified. ] |

## WireInfo

Class in package 'AssetInfo'

Was: Wire data that can be specified per line segment phase, or for the line segment as a whole in case its phases all have the same wire characteristics.

Requested: Wire datasheet.

| ATTRIBUTES |
| --- |
| coreRadius : Length Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  (if there is a different core material) Radius of the central core.  [ Is static True. Containment is Not Specified. ] |
| coreStrandCount : Integer Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  (if used) Number of strands in the steel core.  [ Is static True. Containment is Not Specified. ] |
| gmr : Length Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: Geometric mean radius. If we replace the conductor by a thin walled tube of radius GMR, then its reactance is identical to the reactance of the actual conductor.  Requested: Geometric mean radius. If we replace the wire by a thin walled tube of radius GMR, then its reactance is identical to the reactance of the actual wire.  [ Is static True. Containment is Not Specified. ] |
| insulated : Boolean Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: True if conductor is insulated.  Requested: True if wire is insulated.  [ Is static True. Containment is Not Specified. ] |
| insulationMaterial : WireInsulationKind Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: (if insulated conductor) Material used for insulation.  Requested: Material used for insulation, if insulated wire.  [ Is static True. Containment is Not Specified. ] |
| insulationThickness : Length Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: (if insulated conductor) Thickness of the insulation.  Requested: Thickness of the wire, if insulated wire.  [ Is static True. Containment is Not Specified. ] |
| material : WireMaterialKind Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: Conductor material.  Requested: Wire material.  [ Is static True. Containment is Not Specified. ] |
| rAC25 : ResistancePerLength Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: AC resistance per unit length of the conductor at 25 °C.  Requested: AC resistance per unit length of the wire at 25 °C.  [ Is static True. Containment is Not Specified. ] |
| rAC50 : ResistancePerLength Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: AC resistance per unit length of the conductor at 50 °C.  Requested: AC resistance per unit length of the wire at 50 °C.  [ Is static True. Containment is Not Specified. ] |
| rAC75 : ResistancePerLength Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: AC resistance per unit length of the conductor at 75 °C.  Requested: AC resistance per unit length of the wire at 75 °C.  [ Is static True. Containment is Not Specified. ] |
| radius : Length Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Outside radius of the wire.  [ Is static True. Containment is Not Specified. ] |
| ratedCurrent : CurrentFlow Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Current carrying capacity of the wire under stated thermal conditions.  [ Is static True. Containment is Not Specified. ] |
| rDC20 : ResistancePerLength Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: DC resistance per unit length of the conductor at 20 °C.  Requested: DC resistance per unit length of the wire at 20 °C.  [ Is static True. Containment is Not Specified. ] |
| sizeDescription : String Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Describes the wire gauge or cross section (e.g., 4/0, #2, 336.5).  [ Is static True. Containment is Not Specified. ] |
| strandCount : Integer Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: Number of strands in the conductor.  Requested: Number of strands in the wire.  [ Is static True. Containment is Not Specified. ] |

## WirePosition

Class in package 'AssetInfo'

Was: Identification, spacing and configuration of the wires of a conductor with respect to a structure.

Requested: Position describes the position of a wire in an assembly and optionally may include the definition of the associated wire type, intraphase spacing and phase.

| ATTRIBUTES |
| --- |
| phase : SinglePhaseKind Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Requested: The phase (or neutral) associated with this position.  [ Is static True. Containment is Not Specified. ] |
| sequenceNumber : Integer Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: Numbering for wires on a WireSpacingInfo. Neutrals should be numbered last.  Requested: Numbering of position within the wire assembly which supports the association of a phase with a position. (Phase is specified either by WirePosition or by ACLineSegmentPhase.) When impedance is to be calculated from line detail, sequence number binds the position to the column and row of an entry in the per length phase impedance matrix (PhaseImpedanceData of PerLengthPhaseImpedance).  When the wire assembly is associated with a line segment not a part of a coupled line segment group, numbering starts at one phase position in the assembly and increments by one through each phase position. Neutrals are numbered last.  When the wire assembly is associated with a line segment that is a part of a coupled line segment group, numbering starts with the phase wires in wire assembly associated with the first line segment (LineSegmentCoupling.coupledLineNumber=1), then increments through the phase wires of the wire assembly of each subsequent coupled line segment. Neutral positions are numbered at the end.  [ Is static True. Containment is Not Specified. ] |
| xCoord : Displacement Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: Signed horizontal distance from the wire at this position to a common reference point.  Requested: Signed horizontal distance from the wire assembly’s common reference point to this position.  [ Is static True. Containment is Not Specified. ] |
| yCoord : Displacement Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Was: Signed vertical distance from the wire at this position: above ground (positive value) or burial depth below ground (negative value).  Requested: Signed vertical distance from ground level or sea level to this position. The value shall be positive for above ground (or above sea) positions and negative for underground (or undersea) positions.  [ Is static True. Containment is Not Specified. ] |

## IntraphaseSpacing

Class in package 'AssetInfo'

Was: Wire spacing data that associates multiple wire positions with the line segment, and allows to calculate line segment impedances. Number of phases can be derived from the number of associated wire positions whose phase is not neutral.

Requested: The configuration of a set of 2 or more wires carrying the same phase at a position. This model assumes that the wires are deployed symmetrically 'around a circle'.

| ATTRIBUTES |
| --- |
| wireCount : Integer Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Number of wires in the symmetrical bundle (typically between 2 and 4).  Requested: Number of wires arranged 'around the circle' that are carrying the same phase at a position.  [ Is static True. Containment is Not Specified. ] |
| wireSpacing : Length Public  Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )  Distance between wires in a symmetrical bundle.  Requested: Distance between any two wires arranged 'around the circle'.  [ Is static True. Containment is Not Specified. ] |