AS	LLQMaxSizeBackboneCables	LLQ Max Size Backbone Cables		= NumberLLQBranchCables * LLQMaxSizeBranchCables + IF (NumberLLQBranchCables * LLQResidualBranchCableSize > MaxDistrSize, TRUNC (NumberLLQBranchCables * LLQResidualBranchCableSize / MaxDistrSize) , 0)
AT	LLQResidualBackboneCableSize	LLQ Residual Backbone Cable Size		= IF (LLQResidualBranchCableSize = 0, 0, INDEX (DistrCableSize, MATCH (IF (NumberLLQBranchCables * LLQResidualBranchCableSize > MaxDistrSize, NumberLLQBranchCables * LLQResidualBranchCableSize - MaxDistrSize * TRUNC (NumberLLQBranchCables * LLQResidualBranchCableSize / MaxDistrSize) , NumberLLQBranchCables * LLQResidualBranchCableSize) , DistrCableSize, - 1) , 1))
AU	LLQBranchBackboneAerialDistance	LLQ Branch & Backbone Aerial Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 4) * LLQBackboneBranchCableLength
AV	LLQBranchAerialCableMaterial	LLQ Branch Aerial Cable Material $		= IF (LLQResidualBranchCableSize = 0, 0, NumberLLQBranchCables * LLQBranchBackboneAerialDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LLQResidualBranchCableSize, IF (LLQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + LLQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (LLQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))
AW	LLQBackboneAerialCableMaterial	LLQ Backbone Aerial Cable Material $		= IF (LLQResidualBackboneCableSize = 0, 0, LLQBranchBackboneAerialDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LLQResidualBackboneCableSize, IF (LLQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + LLQMaxSizeBackboneCables * VLOOKUP (MaxFeederSize, IF (LLQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))
AX	LLQBranchBackboneBuriedDistance	LLQ Branch & Backbone Buried Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 3) * LLQBackboneBranchCableLength
AY	LLQBranchBuriedCableMaterial	LLQ Branch Buried Cable Material $		= IF (LLQResidualBranchCableSize = 0, 0, NumberLLQBranchCables * LLQBranchBackboneBuriedDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LLQResidualBranchCableSize, IF (LLQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + LLQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (LLQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))
AZ	LLQBackboneBuriedCableMaterial	LLQ Backbone Buried Cable Material $		= IF (LLQResidualBackboneCableSize = 0, 0, LLQBranchBackboneBuriedDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LLQResidualBackboneCableSize, IF (LLQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + LLQMaxSizeBackboneCables * VLOOKUP (MaxDistrSize, IF (LLQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))
BA	LLQBranchBackboneUGDistance	LLQ Branch & Backbone Underground Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 2) * LLQBackboneBranchCableLength
BB	LLQBranchUGCableMaterial	LLQ Branch Underground Cable Material $		= IF (LLQResidualBranchCableSize = 0, 0, NumberLLQBranchCables * LLQBranchBackboneUGDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LLQResidualBranchCableSize, IF (LLQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + LLQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (LLQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))
BC	LLQBackboneUGCableMaterial	LLQ Backbone Underground Cable Material $		= IF (LLQResidualBackboneCableSize = 0, 0, LLQBranchBackboneUGDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LLQResidualBackboneCableSize, IF (LLQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + LLQMaxSizeBackboneCables * VLOOKUP (MaxDistrSize, IF (LLQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

BD	LLQBranchDucts	Number LLQ Branch Ducts		= LLQMaxSizeBranchCables + 2
BE	LLQBranchManholes	Number LLQ Branch Manholes		= NumberLLQBranchCables * IF (LLQBranchBackboneUGDistance < 1, 0, LLQBranchBackboneUGDistance / VLOOKUP (Density, DistrSpacingTable, 2))
BF	LLQStandardBranchManholeCost	LLQ Standard Branch Manhole Cost		= IF (OR (LLQBranchManholes = 0, LLQBranchDucts > 9) , 0, VLOOKUP (LLQBranchDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
BG	LLQExpandedBranchManholeCost	LLQ Expanded Branch Manhole Cost		= IF (OR (LLQBranchManholes = 0, LLQBranchDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (LLQBranchDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
BH	LLQBackboneDucts	Number LLQ Backbone Ducts		= LLQMaxSizeBackboneCables + 2
BI	LLQBackboneManholes	Number LLQ Backbone Manholes		= IF (LLQBranchBackboneUGDistance < 1, 0, LLQBranchBackboneUGDistance / VLOOKUP (Density, DistrSpacingTable, 2))
BJ	LLQStandardBackboneManholeCost	LLQ Standard Backbone Manhole Cost		= IF (OR (LLQBackboneManholes = 0, LLQBackboneDucts > 9) , 0, VLOOKUP (LLQBackboneDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
BK	LLQExpandedBackboneManholeCost	LLQ Expanded Backbone Manhole Cost		= IF (OR (LLQBackboneManholes = 0, LLQBackboneDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (LLQBackboneDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
BL	LLQUGStructureCost	LLQ Underground Structure Cost		= LLQBranchManholes * IF (LLQBranchDucts < = 9, LLQStandardBranchManholeCost, LLQExpandedBranchManholeCost) + LLQBackboneManholes * IF (LLQBackboneDucts < = 9, LLQStandardBackboneManholeCost, LLQExpandedBackboneManholeCost) + (LLQBranchDucts + LLQBackboneDucts) * LLQBranchBackboneUGDistance * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer) + (NumberLLQBranchCables + 1) * LLQBranchBackboneUGDistance * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 3)
BM	URQHousingUnitlocations	URQ Housing Unit Locations		= IF (HousingUnits = 0, 0, TotalHousingUnitLocations * URQHousingUnits / HousingUnits)
BN	URQBusinesslocations	URQ Business Locations		= IF (TotalBusinessLines = 0, 0, CEILING (BusinessLocations * URQBusinessLines / TotalBusinessLines, 1))
BO	URQLots	URQ Lots		= URQHousingUnitlocations + URQBusinesslocations
BP	URQLotSideLength	URQ Lot Side Length		= IF (URQLots = 0, 0, SQRT (URQArea / URQLots) * 5280)
BQ	URQQuadrantSideLength	URQ Quadrant Side Length		= SQRT (URQArea) * 5280
BR	URQBackboneBranchCableLength	URQ Backbone & Branch Cable Length		= IF (NumberPairsInURQuadrant = 0, 0, IF (URQQuadrantSideLength - (2 * URQLotSideLength) < 0, 0, URQQuadrantSideLength - (2 * URQLotSideLength)))
BS	NumberURQbranchcables	Number Of URQ Branch Cables		= IF (OR (URQLotSideLength = 0, URQBackboneBranchCableLength = 0) , 0, CEILING (URQQuadrantSideLength / (2 * URQLotSideLength) , 1))

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
[illegible]ietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

BT	NumberpairsinUR quadrant	Number Of Pairs In UR Quadrant		= URQHousingUnits * Pairsperhousingunit + IF (URQBusinesslocations = 0, 0, IF ((URQBusinessLines / URQBusinesslocations) > Pairsperbusinesslocation, URQBusinessLines, Pairsperbusinesslocation * URQBusinesslocations))
BU	PairsperURQbranchcable	Pairs Per URQ Branch Cable		= IF (NumberURQBranchCables = 0, 0, CEILING (NumberPairsInURQuadrant / (2 * NumberURQBranchCables) / DistributionFillFactor, 1))
BV	URQMaxSizeBranchCables	URQ Max Size Branch Cables		= IF (PairsperURQbranchcable > MaxDistrSize, TRUNC (PairsperURQbranchcable / MaxDistrSize) , 0)
BW	URQResidualBranchCableSize	URQ Residual Branch Cable Size		= IF (PairsperURQbranchcable = 0, 0, INDEX (DistrCableSize, MATCH (PairsperURQbranchcable - URQMaxSizeBranchCables * MaxDistrSize, DistrCableSize, - 1) , 1))
BX	URQMaxSizeBackboneCables	URQ Max Size Backbone Cables		= NumberURQBranchCables * URQMaxSizeBranchCables + IF (NumberURQBranchCables * URQResidualBranchCableSize > MaxDistrSize, TRUNC (NumberURQBranchCables * URQResidualBranchCableSize / MaxDistrSize) , 0)
BY	URQResidualBackboneCableSize	URQ Residual Backbone Cable Size		= IF (URQResidualBranchCableSize = 0, 0, INDEX (DistrCableSize, MATCH (IF (NumberURQBranchCables * URQResidualBranchCableSize > MaxDistrSize, NumberURQBranchCables * URQResidualBranchCableSize - MaxDistrSize * TRUNC (NumberURQBranchCables * URQResidualBranchCableSize / MaxDistrSize) , NumberURQBranchCables * URQResidualBranchCableSize) , DistrCableSize, - 1) , 1))
BZ	URQBranchBackboneAerialDistance	URQ Branch & Backbone Aerial Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 4) * URQBackboneBranchCableLength
CA	URQBranchAerialCableMaterial	URQ Branch Aerial Cable Material $		= IF (URQResidualBranchCableSize = 0, 0, NumberURQBranchCables * URQBranchBackboneAerialDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (URQResidualBranchCableSize, IF (URQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + URQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (URQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))
CB	URQBackboneAerialCableMaterial	URQ Backbone Aerial Cable Material $		= IF (URQResidualBackboneCableSize = 0, 0, URQBranchBackboneAerialDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (URQResidualBackboneCableSize, IF (URQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + URQMaxSizeBackboneCables * VLOOKUP (MaxFeederSize, IF (URQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))
CC	URQBranchBackboneBuriedDistance	URQ Branch & Backbone Buried Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 3) * URQBackboneBranchCableLength
CD	URQBranchBuriedCableMaterial	URQ Branch Buried Cable Material $		= IF (URQResidualBranchCableSize = 0, 0, NumberURQBranchCables * URQBranchBackboneBuriedDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (URQResidualBranchCableSize, IF (URQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + URQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (URQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))
CE	URQBackboneBuriedCableMaterial	URQ Backbone Buried Cable Material $		= IF (URQResidualBackboneCableSize = 0, 0, URQBranchBackboneBuriedDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (URQResidualBackboneCableSize, IF (URQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + URQMaxSizeBackboneCables * VLOOKUP (MaxDistrSize, IF (URQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

CF	URQBranchBackboneUGDistance	URQ Branch & Backbone Underground Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 2) * URQBackboneBranchCableLength
CG	URQBranchUGCableMaterial	URQ Branch Underground Cable Material $		= IF (URQResidualBranchCableSize = 0, 0, NumberURQBranchCables * URQBranchBackboneUGDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (URQResidualBranchCableSize, IF (URQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + URQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (URQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))
CH	URQBackboneUGCableMaterial	URQ Backbone Underground Cable Material $		= IF (URQResidualBackboneCableSize = 0, 0, URQBranchBackboneUGDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (URQResidualBackboneCableSize, IF (URQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + URQMaxSizeBackboneCables * VLOOKUP (MaxDistrSize, IF (URQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))
CI	URQBranchDucts	Number URQ Branch Ducts		= URQMaxSizeBranchCables + 2
CJ	URQBranchManholes	Number URQ Branch Manholes		= NumberURQBranchCables * IF (URQBranchBackboneUGDistance < 1, 0, URQBranchBackboneUGDistance / VLOOKUP (Density, DistrSpacingTable, 2))
CK	URQStandardBranchManholeCost	URQ Standard Branch Manhole Cost		= IF (OR (URQBranchManholes = 0, URQBranchDucts > 9) , 0, VLOOKUP (URQBranchDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
CL	URQExpandedBranchManholeCost	URQ Expanded Branch Manhole Cost		= IF (OR (URQBranchManholes = 0, URQBranchDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (URQBranchDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
CM	URQBackboneDucts	Number URQ Backbone Ducts		= URQMaxSizeBackboneCables + 2
CN	URQBackboneManholes	Number URQ Backbone Manholes		= IF (URQBranchBackboneUGDistance < 1, 0, URQBranchBackboneUGDistance / VLOOKUP (Density, DistrSpacingTable, 2))
CO	URQStandardBackboneManholeCost	URQ Standard Backbone Manhole Cost		= IF (OR (URQBackboneManholes = 0, URQBackboneDucts > 9) , 0, VLOOKUP (URQBackboneDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
CP	URQExpandedBackboneManholeCost	URQ Expanded Backbone Manhole Cost		= IF (OR (URQBackboneManholes = 0, URQBackboneDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (URQBackboneDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
CQ	URQUGStructureCost	URQ Underground Structure Cost		= URQBranchManholes * IF (URQBranchDucts < = 9, URQStandardBranchManholeCost, URQExpandedBranchManholeCost) + URQBackboneManholes * IF (URQBackboneDucts < = 9, URQStandardBackboneManholeCost, URQExpandedBackboneManholeCost) + (URQBranchDucts + URQBackboneDucts) * URQBranchBackboneUGDistance * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer) + (NumberURQBranchCables + 1) * URQBranchBackboneUGDistance * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 3)
CR	LRQHousingUnitlocations	LRQ Housing Unit Locations		= IF (HousingUnits = 0, 0, TotalHousingUnitLocations * LRQHousingUnits / HousingUnits)

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

CS	LRQBusinesslocations	LRQ Business Locations		= IF (TotalBusinessLines = 0, 0, CEILING (BusinessLocations * LRQBusinessLines / TotalBusinessLines, 1))
CT	LRQLots	LRQ Lots		= LRQHousingUnitlocations + LRQBusinesslocations
CU	LRQLotSideLength	LRQ Lot Side Length		= IF (LRQLots = 0, 0, SQRT (LRQArea / LRQLots) * 5280)
CV	LRQQuadrantSideLength	LRQ Quadrant Side Length		= SQRT (LRQArea) * 5280
CW	LRQBackboneBranchCableLength	LRQ Backbone & Branch Cable Length		= IF (NumberPairsInLRQuadrant = 0, 0, IF (LRQQuadrantSideLength - (2 * LRQLotSideLength) < 0, 0, LRQQuadrantSideLength - (2 * LRQLotSideLength)))
CX	NumberLRQbranchcables	Number Of LRQ Branch Cables		= IF (OR (LRQLotSideLength = 0, LRQBackboneBranchCableLength = 0) , 0, CEILING (LRQQuadrantSideLength / (2 * LRQLotSideLength) , 1))
CY	NumberpairsinLRquadrant	Number Of Pairs In LR Quadrant		= LRQHousingUnits * Pairsperhousingunit + IF (LRQBusinesslocations = 0, 0, IF ((LRQBusinessLines / LRQBusinesslocations) > Pairsperbusinesslocation, LRQBusinessLines, Pairsperbusinesslocation * LRQBusinesslocations))
CZ	PairsperLRQbranchcable	Pairs Per LRQ Branch Cable		= IF (NumberLRQBranchCables = 0, 0, CEILING (NumberPairsInLRQuadrant / (2 * NumberLRQBranchCables) / DistributionFillFactor, 1))
DA	LRQMaxSizeBranchCables	LRQ Max Size Branch Cables		= IF (PairsperLRQbranchcable > MaxDistrSize, TRUNC (PairsperLRQbranchcable / MaxDistrSize) , 0)
DB	LRQResidualBranchCableSize	LRQ Residual Branch Cable Size		= IF (PairsperLRQbranchcable = 0, 0, INDEX (DistrCableSize, MATCH (PairsperLRQbranchcable - LRQMaxSizeBranchCables * MaxDistrSize, DistrCableSize, - 1) , 1))
DC	LRQMaxSizeBackboneCables	LRQ Max Size Backbone Cables		= NumberLRQBranchCables * LRQMaxSizeBranchCables + IF (NumberLRQBranchCables * LRQResidualBranchCableSize > MaxDistrSize, TRUNC (NumberLRQBranchCables * LRQResidualBranchCableSize / MaxDistrSize) , 0)
DD	LRQResidualBackboneCableSize	LRQ Residual Backbone Cable Size		= IF (LRQResidualBranchCableSize = 0, 0, INDEX (DistrCableSize, MATCH (IF (NumberLRQBranchCables * LRQResidualBranchCableSize > MaxDistrSize, NumberLRQBranchCables * LRQResidualBranchCableSize - MaxDistrSize * TRUNC (NumberLRQBranchCables * LRQResidualBranchCableSize / MaxDistrSize) , NumberLRQBranchCables * LRQResidualBranchCableSize) , DistrCableSize, - 1) , 1))
DE	LRQBranchBackboneAerialDistance	LRQ Branch & Backbone Aerial Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 4) * LRQBackboneBranchCableLength
DF	LRQBranchAerialCableMaterial	LRQ Branch Aerial Cable Material $		= IF (LRQResidualBranchCableSize = 0, 0, NumberLRQBranchCables * LRQBranchBackboneAerialDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LRQResidualBranchCableSize, IF (LRQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + LRQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (LRQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))
DG	LRQBackboneAerialCableMaterial	LRQ Backbone Aerial Cable Material $		= IF (LRQResidualBackboneCableSize = 0, 0, LRQBranchBackboneAerialDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LRQResidualBackboneCableSize, IF (LRQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + LRQMaxSizeBackboneCables * VLOOKUP (MaxFeederSize, IF (LRQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))
DH	LRQBranchBackboneBuriedDistance	LRQ Branch & Backbone Buried Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 3) * LRQBackboneBranchCableLength

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

DI	LRQBranchBuried CableMaterial	LRQ Branch Buried Cable Material $		= IF (LRQResidualBranchCableSize = 0, 0, NumberLRQBranchCables * LRQBranchBackboneBuriedDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LRQResidualBranchCableSize, IF (LRQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + LRQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (LRQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))
DJ	LRQBackboneBur iedCableMaterial	LRQ Backbone Buried Cable Material $		= IF (LRQResidualBackboneCableSize = 0, 0, LRQBranchBackboneBuriedDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LRQResidualBackboneCableSize, IF (LRQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + LRQMaxSizeBackboneCables * VLOOKUP (MaxDistrSize, IF (LRQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))
DK	LRQBranchBackb oneUGDistance	LRQ Branch & Backbone Underground Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 2) * LRQBackboneBranchCableLength
DL	LRQBranchUGCa bleMaterial	LRQ Branch Underground Cable Material $		= IF (LRQResidualBranchCableSize = 0, 0, NumberLRQBranchCables * LRQBranchBackboneUGDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LRQResidualBranchCableSize, IF (LRQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + LRQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (LRQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))
DM	LRQBackboneUG CableMaterial	LRQ Backbone Underground Cable Material $		= IF (LRQResidualBackboneCableSize = 0, 0, LRQBranchBackboneUGDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LRQResidualBackboneCableSize, IF (LRQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + LRQMaxSizeBackboneCables * VLOOKUP (MaxDistrSize, IF (LRQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))
DN	LRQBranchDucts	Number LRQ Branch Ducts		= LRQMaxSizeBranchCables + 2
DO	LRQBranchManh oles	Number LRQ Branch Manholes		= NumberLRQBranchCables * IF (LRQBranchBackboneUGDistance < 1, 0, LRQBranchBackboneUGDistance / VLOOKUP (Density, DistrSpacingTable, 2))
DP	LRQStandardBran chManholeCost	LRQ Standard Branch Manhole Cost		= IF (OR (LRQBranchManholes = 0, LRQBranchDucts > 9) , 0, VLOOKUP (LRQBranchDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
DQ	LRQExpandedBra nchManholeCost	LRQ Expanded Branch Manhole Cost		= IF (OR (LRQBranchManholes = 0, LRQBranchDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (LRQBranchDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
DR	LRQBackboneDu cts	Number LRQ Backbone Ducts		= LRQMaxSizeBackboneCables + 2
DS	LRQBackboneMa nholes	Number LRQ Backbone Manholes		= IF (LRQBranchBackboneUGDistance < 1, 0, LRQBranchBackboneUGDistance / VLOOKUP (Density, DistrSpacingTable, 2))
DT	LRQStandardBac kboneManholeCos t	LRQ Standard Backbone Manhole Cost		= IF (OR (LRQBackboneManholes = 0, LRQBackboneDucts > 9) , 0, VLOOKUP (LRQBackboneDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

DU	LRQExpandedBackboneManholeCost	LRQ Expanded Backbone Manhole Cost		= IF (OR (LRQBackboneManholes = 0, LRQBackboneDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (LRQBackboneDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
DV	LRQUGStructure Cost	LRQ Underground Structure Cost		= LRQBranchManholes * IF (LRQBranchDucts < = 9, LRQStandardBranchManholeCost, LRQExpandedBranchManholeCost) + LRQBackboneManholes * IF (LRQBackboneDucts < = 9, LRQStandardBackboneManholeCost, LRQExpandedBackboneManholeCost) + (LRQBranchDucts + LRQBackboneDucts) * LRQBranchBackboneUGDistance * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer) + (NumberLRQBranchCables + 1) * LRQBranchBackboneUGDistance * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 3)
DW	SubtotalAerialCableDistributionCost	Subtotal Aerial Cable Distribution Cost		= ULQBranchAerialCableMaterial + ULQBackboneAerialCableMaterial + LLQBranchAerialCableMaterial + LLQBackboneAerialCableMaterial + URQBranchAerialCableMaterial + URQBackboneAerialCableMaterial + LRQBranchAerialCableMaterial + LRQBackboneAerialCableMaterial
DX	SubtotalBuriedCableDistributionCost	Subtotal Buried Cable Distribution Cost		= ULQBranchBuriedCableMaterial + ULQBackboneBuriedCableMaterial + LLQBranchBuriedCableMaterial + LLQBackboneBuriedCableMaterial + URQBranchBuriedCableMaterial + URQBackboneBuriedCableMaterial + LRQBranchBuriedCableMaterial + LRQBackboneBuriedCableMaterial
DY	TotalUGCableDistributionCost	Total Underground Cable Distribution Cost		= ULQBranchUGCableMaterial + ULQBackboneUGCableMaterial + LLQBranchUGCableMaterial + LLQBackboneUGCableMaterial + URQBranchUGCableMaterial + URQBackboneUGCableMaterial + LRQBranchUGCableMaterial + LRQBackboneUGCableMaterial
DZ	TotalAerialCableDistributionDistance	Total Aerial Cable Distribution Distance		= ULQBranchBackboneAerialDistance * (NumberULQBranchCables + 1) + LLQBranchBackboneAerialDistance * (NumberLLQBranchCables + 1) + URQBranchBackboneAerialDistance * (NumberURQBranchCables + 1) + LRQBranchBackboneAerialDistance * (NumberLRQBranchCables + 1)
EA	NumberPolesforDistribution	Number Of Poles for Distribution		= TotalAerialCableDistributionDistance / VLOOKUP (Density, DistrSpacingTable, 3) + NumberULQBranchCables + NumberLLQBranchCables + NumberURQBranchCables + NumberLRQBranchCables
EB	TotalPoleDistributionCost	Total Pole Distribution Cost		= NumberPolesForDistribution * WaterCostMultiplier * VLOOKUP (Density, CHOOSE (CopperDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 7)
EC	TotalBuriedCableDistributionDistance	Total Buried Cable Distribution Distance		= ULQBranchBackboneBuriedDistance * (NumberULQBranchCables + 1) + LLQBranchBackboneBuriedDistance * (NumberLLQBranchCables + 1) + URQBranchBackboneBuriedDistance * (NumberURQBranchCables + 1) + LRQBranchBackboneBuriedDistance * (NumberLRQBranchCables + 1)
ED	TotalBuriedDistributionStructureCost	Total Buried Distribution Structure Cost		= IF (TotalBuriedCableDistributionDistance < 1, 0, WaterCostMultiplier * TotalBuriedCableDistributionDistance * VLOOKUP (Density, CHOOSE (CopperDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 5))
EE	TotalDistributionConduitCost	Total Distribution Conduit Cost		= ULQUGStructureCost + LLQUGStructureCost + URQUGStructureCost + LRQUGStructureCost
EF	AverageDistributionDistance	Average Distribution Distance		= 0.75 * AVERAGE (ULQBackboneandBranchCableLength, LLQBackboneBranchCableLength, URQBackboneBranchCableLength, LRQBackboneBranchCableLength) + AVERAGE (ULQDropLength, LLQDropLength, URQDropLength, LRQDropLength)

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
This document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

Sheet: Investment

Col	Range Name	Column Name	Column Comment	Formula
A	AerialCopperCableInvestment	Aerial Copper Cable Investment $		= AllocationAerialCopperCable + SubfeederPt2AerialCopperMaterial + TotalDLCFDIAerialCableMaterial + AllocationFDIToAerialCopper + TotalAerialCableDistributionCost
B	BuriedCopperCableInvestment	Buried Copper Cable Investment $		= AllocationBuriedCopperCableinclStructure + SubfeederPt2BuriedCopperMaterial + SubfeederPt2BuriedCopperStructure + TotalDLCFDIBuriedCableMaterial + TotalBuriedDLCFDICableStructure + AllocationFDIToBuriedCopper + TotalBuriedCableDistributionCost
C	UndergroundCopperCableInvestment	Underground Copper Cable Investment $		= AllocationUGCopperCable + SubfeederPt2UGCopperMaterial + TotalDLCFDIUGCableMaterial + AllocationFDIToUGCopper + TotalUGCableDistributionCost
D	PoleInvestmentAssignedCopper	Pole Investment Assigned Copper		= AllocationCopperRelatedPoles + TotalAerialDLCFDICableStructure + IF (GridPlantType = "Cable", SubfeederPt2AerialStructure, 0) + TotalPoleDistributionCost
E	ConduitInvestmentAssignedCopper	Conduit Investment Assigned Copper		= AllocationCopperRelatedConduit + TotalUGDLCFDICableStructure + IF (GridPlantType = "Cable", SubfeederPt2UGStructure, 0) + TotalDistributionConduitCost
F	CopperRelatedPlantTotal	Copper Related Plant Total		= SUM (A3:E3)
G	AerialFiberCableInvestment	Aerial Fiber Cable Investment $		= AllocationAerialFiberCable + SubfeederPt2AerialFiberMaterial
H	BuriedFiberCableInvestment	Buried Fiber Cable Investment $		= AllocationBuriedFiberCableinclStructure + SubfeederPt2BuriedFiberMaterial + SubfeederPt2BuriedFiberStructure
I	UndergroundFiberCableInvestment	Underground Fiber Cable Investment $		= AllocationUGFiberCable + SubfeederPt2UGFiberMaterial
J	PoleInvestmentAssignedFiber	Pole Investment Assigned Fiber		= AllocationFiberRelatedPoles + IF (GridPlantType = "Cable", 0, SubfeederPt2AerialStructure)
K	ConduitInvestmentAssignedFiber	Conduit Investment Assigned Fiber		= AllocationFiberRelatedConduit + IF (GridPlantType = "Cable", 0, SubfeederPt2UGStructure)
L	TotalFiberRelatedPlant	Total Fiber Related Plant		= SUM (G3:K3)

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

Sheet: Term,Drop,&NID

Col	Range Name	Column Name	Column Comment	Formula
A	PercentDropsAerial	Percent Drops Aerial		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 4)
B	CostperRESNID	Cost Per Res NID		= VLOOKUP ("NID", RESNIDTABLE, DensityColumnPointer, FALSE) + (Pairsperhousingunit + 1) * (VLOOKUP ("PROTECTOR", RESNIDTABLE, DensityColumnPointer, FALSE) + VLOOKUP ("INTERFACE", RESNIDTABLE, DensityColumnPointer, FALSE))
C	CostperBUSNID	Cost Per Bus NID		= VLOOKUP ("NID", BUSNIDTABLE, DensityColumnPointer, FALSE) + Pairsperbusinesslocation * (VLOOKUP ("PROTECTOR", BUSNIDTABLE, DensityColumnPointer, FALSE) + VLOOKUP ("INTERFACE", BUSNIDTABLE, DensityColumnPointer, FALSE))
D	AerialDropCostperFoot	Aerial Drop Cost Per Foot		= VLOOKUP (1, AerDropCost, DensityColumnPointer)
E	BuriedDropCostperFoot	Buried Drop Cost Per Foot		= VLOOKUP (1, BurDropCost, DensityColumnPointer)
F	_1to2HUperDwellingGridLocations	1&2 HU Per Dwelling Grid Drops		= SingleUnitDetachedHUlocations + SingleUnitAttachedHUlocations + 2 * _1to2Buildings + MobileHUlocations + otherBuildings
G	PercentGT2MultiFamilyLots	Percent >2 Multi-Family Lots		= IF (TotalHousingUnitLocations = 0, 0, (TotalHousingUnitLocations - (SingleUnitDetachedHUlocations + SingleUnitAttachedHUlocations + _1to2Buildings + MobileHUlocations + otherBuildings)) / TotalHousingUnitLocations)
H	ULQTerminals	ULQ Terminals		= IF (OR (ULQLots = "", ULQLots = 0) , 0, IF (NumberULQBranchCables = 0, 1, NumberULQBranchCables^2))
I	ULQDropLength	ULQ Drop Length		= MIN (500, ULQLotSideLength * 0.5 * SQRT (2))
J	NumberULQResDrops	Number Of ULQ Res Drops		= IF (TotalHousingUnitLocations = 0, 0, (ULQHousingUnitslocations / TotalHousingUnitLocations) * _1to2HUperDwellingGridLocations)
K	AvgResDropsperULQterminal	Avg Res Drops Per ULQ Terminal		= IF (ULQTerminals = 0, 0, CEILING (NumberULQResDrops / ULQTerminals, 1))
L	SmallBusCheck	Small Business Line Per Location Check		= IF (BusinessLocations = 0, 0, IF (TotalBusinessLines / BusinessLocations > 6, 0, 1))
M	AvgBUSDropsperULQTerminal	Avg Bus Drops Per ULQ Terminal		= IF (ULQTerminals = 0, 0, CEILING (ULQBusinesslocations / ULQTerminals, 1)) * SmallBusCheck
N	ULQDropTerminalPairs	ULQ Drop Terminal Pairs		= AvgResDropsperULQterminal * Pairsperhousingunit + AvgBUSDropsperULQTerminal * Pairsperbusinesslocation
O	TotalULQAerialDropTerminalCost	Total ULQ Aerial Drop Terminal Cost		= IF (ULQDropTerminalPairs = 0, 0, ULQTerminals * PercentDropsAerial * VLOOKUP (ULQDropTerminalPairs, AerTermCostTable, DensityColumnPointer))
P	TotalULQBuriedDropTerminalCost	Total ULQ Buried Drop Terminal Cost		= IF (ULQDropTerminalPairs = 0, 0, ULQTerminals * (1 - PercentDropsAerial) * VLOOKUP (ULQDropTerminalPairs, BURTermCostTable, DensityColumnPointer))
Q	TotalULQNIDCost	Total ULQ NID Cost		= IF (NumberULQResDrops = 0, 0, NumberULQResDrops * CostperResNID) + IF (ULQBusinesslocations = 0, 0, ULQBusinesslocations * CostperBusNID)

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

R	ULQAerialDropCost	ULQ Aerial Drop Cost		= ULQDropLength * (NumberULQResDrops + ULQBusinesslocations) * PercentDropsAerial * AerialDropCostperFoot
S	ULQBuriedDropCost	ULQ Buried Drop Cost		= ULQDropLength * (NumberULQResDrops + ULQBusinesslocations) * (1 - PercentDropsAerial) * BuriedDropCostperFoot
T	LLQTerminals	LLQ Terminals		= IF (OR (LLQLots = "", LLQLots = 0) , 0, IF (NumberLLQBranchCables = 0, 1, NumberLLQBranchCables^2))
U	LLQDropLength	LLQ Drop Length		= IF (LLQLotSideLength * 0.5 * SQRT (2) > 500, 500, LLQLotSideLength * 0.5 * SQRT (2))
V	NumberLLQRESDrops	Number Of LLQ Res Drops		= IF (TotalHousingUnitLocations = 0, 0, (LLQHousingUnitlocations / TotalHousingUnitLocations) * _1to2HUperDwellingGridLocations)
W	AvgRESDropsperLLQterminal	Avg Res Drops Per LLQ Terminal		= IF (LLQTerminals = 0, 0, CEILING (NumberLLQResDrops / LLQTerminals, 1))
X	AvgBUSDropsperLLQTerminal	Avg Bus Drops Per LLQ Terminal		= IF (LLQTerminals = 0, 0, CEILING (LLQBusinesslocations / LLQTerminals, 1)) * SmallBusCheck
Y	LLQDropTerminalPairs	LLQ Drop Terminal Pairs		= AvgRESDropsperLLQterminal * Pairsperhousingunit + AvgBUSDropsPerLLQTerminal * Pairsperbusinesslocation
Z	TotalLLQAerialDropTerminalCost	Total LLQ Aerial Drop Terminal Cost		= IF (LLQDropTerminalPairs = 0, 0, LLQTerminals * PercentDropsAerial * VLOOKUP (LLQDropTerminalPairs, AerTermCostTable, DensityColumnPointer))
AA	TotalLLQBuriedDropTerminalCost	Total LLQ Buried Drop Terminal Cost		= IF (LLQDropTerminalPairs = 0, 0, LLQTerminals * (1 - PercentDropsAerial) * VLOOKUP (LLQDropTerminalPairs, BURTermCostTable, DensityColumnPointer))
AB	TotalLLQNIDCost	Total LLQ NID Cost		= IF (NumberLLQResDrops = 0, 0, NumberLLQResDrops * CostperResNID)
AC	LLQAerialDropCost	LLQ Aerial Drop Cost		= LLQDropLength * (NumberLLQResDrops + LLQBusinesslocations) * PercentDropsAerial * AerialDropCostperFoot
AD	LLQBuriedDropCost	LLQ Buried Drop Cost		= LLQDropLength * (NumberLLQResDrops + LLQBusinesslocations) * (1 - PercentDropsAerial) * BuriedDropCostperFoot
AE	URQTerminals	URQ Terminals		= IF (OR (URQLots = "", URQLots = 0) , 0, IF (NumberURQBranchCables = 0, 1, NumberURQBranchCables^2))
AF	URQDropLength	URQ Drop Length		= IF (URQLotSideLength * 0.5 * SQRT (2) > 500, 500, URQLotSideLength * 0.5 * SQRT (2))
AG	NumberURQRESDrops	Number Of URQ Res Drops		= IF (TotalHousingUnitLocations = 0, 0, (URQHousingUnitlocations / TotalHousingUnitLocations) * _1to2HUperDwellingGridLocations)
AH	AvgRESDropsperURQterminal	Avg Res Drops Per URQ Terminal		= IF (URQTerminals = 0, 0, CEILING (NumberURQResDrops / URQTerminals, 1))
AI	AvgBUSDropsperURQTerminal	Avg Bus Drops Per URQ Terminal		= IF (URQTerminals = 0, 0, CEILING (URQBusinesslocations / URQTerminals, 1)) * SmallBusCheck
AJ	URQDropTerminalPairs	URQ Drop Terminal Pairs		= AvgRESDropsperURQterminal * Pairsperhousingunit + AvgBUSDropsperURQTerminal * Pairsperbusinesslocation
AK	TotalURQAerialDropTerminalCost	Total URQ Aerial Drop Terminal Cost		= IF (URQDropTerminalPairs = 0, 0, URQTerminals * PercentDropsAerial * VLOOKUP (URQDropTerminalPairs, AerTermCostTable, DensityColumnPointer))
AL	TotalURQBuriedDropTerminalCost	Total URQ Buried Drop Terminal Cost		= IF (URQDropTerminalPairs = 0, 0, URQTerminals * (1 - PercentDropsAerial) * VLOOKUP (URQDropTerminalPairs, BURTermCostTable, DensityColumnPointer))

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

AM	TotalURQNIDCost	Total URQ NID Cost		= IF (NumberURQResDrops = 0, 0, NumberURQResDrops * CostperResNID) + IF (URQBusinesslocations = 0, 0, URQBusinesslocations * CostperBusNID)
AN	URQAerialDropCost	URQ Aerial Drop Cost		= URQDropLength * (NumberURQResDrops + URQBusinesslocations) * PercentDropsAerial * AerialDropCostperFoot
AO	URQBuriedDropCost	URQ Buried Drop Cost		= URQDropLength * (NumberURQResDrops + URQBusinesslocations) * (1 - PercentDropsAerial) * BuriedDropCostperFoot
AP	LRQTerminals	LRQ Terminals		= IF (OR (LRQLots = "", LRQLots = 0) , 0, IF (NumberLRQBranchCables = 0, 1, NumberLRQBranchCables^2))
AQ	LRQDropLength	LRQ Drop Length		= MIN (500, LRQLotSideLength * 0.5 * SQRT (2))
AR	NumberLRQRESDrops	Number Of LRQ Res Drops		= IF (TotalHousingUnitLocations = 0, 0, (LRQHousingUnitlocations / TotalHousingUnitLocations) * _1to2HUperDwellingGridLocations)
AS	AvgRESDropsperLRQterminal	Avg Res Drops Per LRQ Terminal		= IF (LRQTerminals = 0, 0, CEILING (NumberLRQResDrops / LRQTerminals, 1))
AT	AvgBUSDropsperLRQTerminal	Avg Bus Drops Per LRQ Terminal		= IF (LRQTerminals = 0, 0, CEILING (LRQBusinesslocations / LRQTerminals, 1)) * SmallBusCheck
AU	LRQDropTerminalPairs	LRQ Drop Terminal Pairs		= AvgRESDropsperLRQterminal * Pairsperhousingunit + AvgBUSDropsPerLRQTerminal * Pairsperbusinesslocation
AV	TotalLRQAerialDropTerminalCost	Total LRQ Aerial Drop Terminal Cost		= IF (LRQDropTerminalPairs = 0, 0, LRQTerminals * PercentDropsAerial * VLOOKUP (LRQDropTerminalPairs, AerTermCostTable, DensityColumnPointer))
AW	TotalLRQBuriedDropTerminalCost	Total LRQ Buried Drop Terminal Cost		= IF (LRQDropTerminalPairs = 0, 0, LRQTerminals * (1 - PercentDropsAerial) * VLOOKUP (LRQDropTerminalPairs, BURTermCostTable, DensityColumnPointer))
AX	TotalLRQNIDCost	Total LRQ NID Cost		= IF (NumberLRQResDrops = 0, 0, NumberLRQResDrops * CostperResNID) + IF (LRQBusinesslocations = 0, 0, LRQBusinesslocations * CostperBusNID)
AY	LRQAerialDropCost	LRQ Aerial Drop Cost		= LRQDropLength * (NumberLRQResDrops + LRQBusinesslocations) * PercentDropsAerial * VLOOKUP (1, AerDropCost, DensityColumnPointer)
AZ	LRQBuriedDropCost	LRQ Buried Drop Cost		= LRQDropLength * (NumberLRQResDrops + LRQBusinesslocations) * (1 - PercentDropsAerial) * VLOOKUP (1, BurDropCost, DensityColumnPointer)
BA	TotalGridAerialDropTerminalCost	Total Grid Aerial Drop Terminal Cost		= TotalULQAerialDropTerminalCost + TotalLLQAerialDropTerminalCost + TotalURQAerialDropTerminalCost + TotalLRQAerialDropTerminalCost
BB	TotalGridBuriedDropTerminalCost	Total Grid Buried Drop Terminal Cost		= TotalULQBuriedDropTerminalCost + TotalLLQBuriedDropTerminalCost + TotalURQBuriedDropTerminalCost + TotalLRQBuriedDropTerminalCost
BC	TotalGridNIDCost	Total Grid NID Cost		= TotalULQNIDCost + TotalLLQNIDCost + TotalURQNIDCost + TotalLRQNIDCost
BD	TotalGridSingleAerialDropCost	Total Grid Single Aerial Drop Cost		= ULQAerialDropCost + LLQAerialDropCost + URQAerialDropCost + LRQAerialDropCost
BE	TotalGridSingleBuriedDropCost	Total Grid Single Buried Drop Cost		= ULQBuriedDropCost + LLQBuriedDropCost + URQBuriedDropCost + LRQBuriedDropCost
BF	WeightedGridMultiHUDropLength	Weighted Grid Multi-HU Drop Length		= IF (OR (TotalHousingUnitLocations = 0, PercentGt2MultiFamilyLots = 0) , 0, (ULQHousingUnitslocations * ULQDropLength + LLQHousingUnitlocations * LLQDropLength + URQHousingUnitlocations * URQDropLength + LRQHousingUnitlocations * LRQDropLength) / TotalHousingUnitLocations)

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
[illegible] is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

BG	CableSizeLargeBusLocations	Cable Size Large Business Locations		= IF (OR (SmallBusCheck = 1, BusinessLocations = 0) , 0, INDEX (DistrCableSize, MATCH (TotalBusinessLines / BusinessLocations, DistrCableSize, - 1) , 1))
BH	AerialDropCostLargeBus	Aerial Drop Cable Cost Large Business Locations		= IF (CableSizeLargeBusLocations = 0, 0, BusinessLocations * PercentDropsAerial * WeightedGridMultiHUDropLength * VLOOKUP (CableSizeLargeBusLocations, AERCopper24Table, DensityColumnPointer, FALSE))
BI	BuriedDropCostLargeBus	Buried Drop Cable Cost Large Business Locations		= IF (CableSizeLargeBusLocations = 0, 0, BusinessLocations * (1 - PercentDropsAerial) * WeightedGridMultiHUDropLength * VLOOKUP (CableSizeLargeBusLocations, BURCopper24Table, DensityColumnPointer, FALSE))
BJ	CostPerTerminalLargeBus	Cost Per Terminal Large At Large Business Locations		= IF (CableSizeLargeBusLocations = 0, 0, VLOOKUP (CableSizeLargeBusLocations, IndoorSAICostTable, DensityColumnPointer))
BK	CableSize3to4HUdwellings	Cable Size 3-4 HU Dwellings		= MAX (25, INDEX (DistrCableSize, MATCH (VLOOKUP ("3 - 4", HousingUnitsperDwelling, DensityColumnPointer, FALSE) * Pairsperhousingunit, DistrCableSize, - 1) , 1))
BL	AerialDropCableCost3to4HUDwellings	Aerial Drop Cable Cost 3-4 HU Dwellings		= _3to4Buildings * PercentDropsAerial * WeightedGridMultiHUDropLength * VLOOKUP (CableSize3to4HUdwellings, AERCopper24Table, DensityColumnPointer, FALSE)
BM	BuriedDropCableCost3to4HUDwellings	Buried Drop Cable Cost 3-4 HU Dwellings		= _3to4Buildings * (1 - PercentDropsAerial) * WeightedGridMultiHUDropLength * VLOOKUP (CableSize3to4HUdwellings, BURCopper24Table, DensityColumnPointer, FALSE)
BN	CostperTerminal3to4HUdwellings	Cost Per Terminal 3-4 HU Dwellings		= IF (CableSize3to4HUdwellings = 0, 0, VLOOKUP (CableSize3to4HUdwellings, IndoorSAICostTable, DensityColumnPointer))
BO	CableSize5to9HUdwellings	Cable Size 5-9 HU Dwellings		= MAX (25, INDEX (DistrCableSize, MATCH (VLOOKUP ("5 - 9", HousingUnitsperDwelling, DensityColumnPointer, FALSE) * Pairsperhousingunit, DistrCableSize, - 1) , 1))
BP	AerialDropCableCost5to9HUDwellings	Aerial Drop Cable Cost 5-9 HU Dwellings		= _5to9Buildings * PercentDropsAerial * WeightedGridMultiHUDropLength * VLOOKUP (CableSize5to9HUdwellings, AERCopper24Table, DensityColumnPointer, FALSE)
BQ	BuriedDropCableCost5to9HUDwellings	Buried Drop Cable Cost 5-9 HU Dwellings		= _5to9Buildings * (1 - PercentDropsAerial) * WeightedGridMultiHUDropLength * VLOOKUP (CableSize5to9HUdwellings, BURCopper24Table, DensityColumnPointer, FALSE)
BR	CostperTerminal5to9HUdwellings	Cost Per Terminal 5-9 HU Dwellings		= IF (CableSize5to9HUdwellings = 0, 0, VLOOKUP (CableSize5to9HUdwellings, IndoorSAICostTable, DensityColumnPointer))
BS	CableSize10to19HUdwellings	Cable Size 10-19 HU Dwellings		= MAX (25, INDEX (DistrCableSize, MATCH (VLOOKUP ("10 - 19", HousingUnitsperDwelling, DensityColumnPointer, FALSE) * Pairsperhousingunit, DistrCableSize, - 1) , 1))
BT	AerialDropCableCost10to19HUDwellings	Aerial Drop Cable Cost 10-19 HU Dwellings		= _10to19Buildings * PercentDropsAerial * WeightedGridMultiHUDropLength * VLOOKUP (CableSize10to19HUdwellings, AERCopper24Table, DensityColumnPointer, FALSE)
BU	BuriedDropCableCost10to19HUDwellings	Buried Drop Cable Cost 10-19 HU Dwellings		= _10to19Buildings * (1 - PercentDropsAerial) * WeightedGridMultiHUDropLength * VLOOKUP (CableSize10to19HUdwellings, BURCopper24Table, DensityColumnPointer, FALSE)

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

BV	CostperTerminal10to19HUdwellings	Cost Per Terminal 10-19 HU Dwellings		= IF (CableSize10to19HUdwellings = 0, 0, VLOOKUP (CableSize10to19HUdwellings, IndoorSAICostTable, DensityColumnPointer))
BW	CableSize20to49HUdwellings	Cable Size 20-49 HU Dwellings		= MAX (50, INDEX (DistrCableSize, MATCH (VLOOKUP ("20 - 49", HousingUnitsperDwelling, DensityColumnPointer, FALSE) * Pairsperhousingunit, DistrCableSize, - 1) , 1))
BX	AerialDropCableCost20to49HUDwellings	Aerial Drop Cable Cost 20-49 HU Dwellings		= _20to49Buildings * PercentDropsAerial * WeightedGridMultiHUDropLength * VLOOKUP (CableSize20to49HUdwellings, AERCopper24Table, DensityColumnPointer, FALSE)
BY	BuriedDropCableCost20to49HUDwellings	Buried Drop Cable Cost 20-49 HU Dwellings		= _20to49Buildings * (1 - PercentDropsAerial) * WeightedGridMultiHUDropLength * VLOOKUP (CableSize20to49HUdwellings, BURCopper24Table, DensityColumnPointer, FALSE)
BZ	CostperTerminal20to49HUdwellings	Cost Per Terminal 20-49 HU Dwellings		= IF (CableSize20to49HUdwellings = 0, 0, VLOOKUP (CableSize20to49HUdwellings, IndoorSAICostTable, DensityColumnPointer))
CA	CableSizeOver50HUdwellings	Cable Size >50 HU Dwellings		= MAX (100, INDEX (DistrCableSize, MATCH (VLOOKUP (" > 50", HousingUnitsperDwelling, DensityColumnPointer, FALSE) * Pairsperhousingunit, DistrCableSize, - 1) , 1))
CB	AerialDropCableCostOver50HUDwellings	Aerial Drop Cable Cost >50 HU Dwellings		= over50Buildings * PercentDropsAerial * WeightedGridMultiHUDropLength * VLOOKUP (CableSizeOver50HUdwellings, AERCopper24Table, DensityColumnPointer, FALSE)
CC	BuriedDropCableCostOver50HUDwellings	Buried Drop Cable Cost >50 HU Dwellings		= over50Buildings * (1 - PercentDropsAerial) * WeightedGridMultiHUDropLength * VLOOKUP (CableSizeOver50HUdwellings, BURCopper24Table, DensityColumnPointer, FALSE)
CD	CostperTerminalOver50HUdwellings	Cost Per Terminal >50 HU Dwellings		= IF (CableSizeOver50HUdwellings = 0, 0, VLOOKUP (CableSizeOver50HUdwellings, IndoorSAICostTable, DensityColumnPointer))
CE	MultiHUAerialDropCable	Multi-HU Aerial Drop Cable		= AerialDropCableCost3to4HUDwellings + AerialDropCableCost5to9HUDwellings + AerialDropCableCost10to19HUDwellings + AerialDropCableCost20to49HUDwellings + AerialDropCableCostOver50HUDwellings + AerialDropCostLargeBus
CF	MultiHUBuriedDropCable	Multi-HU Buried Drop Cable		= BuriedDropCableCost3to4HUDwellings + BuriedDropCableCost5to9HUDwellings + BuriedDropCableCost10to19HUDwellings + BuriedDropCableCost20to49HUDwellings + BuriedDropCableCostOver50HUDwellings + BuriedDropCostLargeBus
CG	TotalMultiHUBuriedDropCableDistance	Total Multi-HU Buried Drop Cable Distance		= (_3to4Buildings + _5to9Buildings + _10to19Buildings + _20to49Buildings + over50Buildings + BusinessLocations * (1 - SmallBusCheck)) * WeightedGridMultiHUDropLength
CH	PlacementCostMHUBuriedDropCable	Placement Cost MHU Buried Drop Cable		= IF (TotalMultiHUBuriedDropCableDistance < 1, 0, TotalMultiHUBuriedDropCableDistance * VLOOKUP (Density, CHOOSE (CopperDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 4))
CI	MultiHUAerialBuildingTerminalCost	Multi-HU Aerial Building Terminal Cost		= (_3to4Buildings * CostperTerminal3to4HUdwellings + _5to9Buildings * CostperTerminal5to9HUdwellings + _10to19Buildings * CostperTerminal10to19HUdwellings + _20to49Buildings * CostperTerminal20to49HUdwellings + over50Buildings * CostperTerminalOver50HUdwellings + BusinessLocations * CostPerTerminalLargeBus) * PercentDropsAerial

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

CJ	MultiHUBuriedBuildingTerminalCost	Multi-HU Buried Building Terminal Cost		= (_3to4Buildings * CostperTerminal3to4HUdwellings + _5to9Buildings * CostperTerminal5to9HUdwellings + _10to19Buildings * CostperTerminal10to19HUdwellings + _20to49Buildings * CostperTerminal20to49HUdwellings + over50Buildings * CostperTerminalOver50HUdwellings + BusinessLocations * CostPerTerminalLargeBus) * (1 - PercentDropsAerial)
CK	SubtotalAerialCableRelatedcosts	Subtotal Aerial Cable Related costs		= TotalGridAerialDropTerminalCost + TotalGridSingleAerialDropCost + MultiHUAerialDropCable + MultiHUAerialBuildingTerminalCost
CL	SubtotalBuriedCableRelatedcosts	Subtotal Buried Cable Related costs		= TotalGridBuriedDropTerminalCost + TotalGridSingleBuriedDropCost + MultiHUBuriedDropCable + MultiHUBuriedBuildingTerminalCost + PlacementCostMHUBuriedDropCable
CM	TotalAerialCableDistributionCost	Total Aerial Cable Distribution Cost		= SubtotalAerialCableDistributionCost + SubtotalAerialCablerelatedcosts
CN	TotalBuriedCable DistributionCost	Total Buried Cable Distribution Cost		= SubtotalBuriedCableDistributionCost + TotalBuriedDistributionStructureCost + SubtotalBuriedCablerelatedcosts
CO		Total UG Cable Distribution Cost		= TotalUGCableDistributionCost
CP		Total Pole Distribution Cost		= TotalPoleDistributionCost
CQ		Total Conduit Distribution Cost		= TotalDistributionConduitCost
CR	TotalNIDCostAllocatedAerial	Total NID Cost Allocated Aerial		= TotalGridNIDCost * PercentDropsAerial
CS	TotalNIDCostAllocatedBuried	Total NID Cost Allocated Buried		= TotalGridNIDCost * (1 - PercentDropsAerial)
CT	TotalDistribution Cost	Total Distribution Cost		= TotalAerialCableDistributionCost + TotalBuriedCableDistributionCost + TotalUGCableDistributionCost + TotalPoleDistributionCost + TotalDistributionConduitCost + TotalGridNIDCost

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
'atory to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

Sheet: Electronics&FDI

Col	Range Name	Column Name	Column Comment	Formula
A	DLCLFiberElectronics	DLC-L On Fiber Electronics $		= IF (TotalLinesServedInGrid = 0, 0, CHOOSE (GridPlantIndex, 0, 0, DLC_LDiscount * (NumberDLCLsAtFullCapacity * VLOOKUP (673, DigitalCarrierCost, 2) + NumberDLCLsAtFullCapacity * 1344 * VLOOKUP (673, DigitalCarrierCost, 3) + VLOOKUP (SwitchedLinesInOverflowDLCLTerminal, DigitalCarrierCost, 2) + SwitchedLinesInOverflowDLCLTerminal * VLOOKUP (SwitchedLinesInOverflowDLCLTerminal, DigitalCarrierCost, 3))))
B	GridLinesonDLCL	Grid Lines Served by DLC-L	Grid Lines served by DLC-L	= IF (GridPlantIndex = 3, VoiceGradeLinesEquippedAtTerminal, 0)
C	TotalCLLILinesonDLCL	Total CLLI Lines Served by DLC-L	Total CLLI Lines served by DLC-L	= SUMIF (OFFSET (GridPlantIndex, 0, 0, LastDataRow, 1) , 3, OFFSET (VoiceGradeLinesEquippedAtTerminal, 0, 0, LastDataRow, 1))
D	COTLElectronics	COT-L Electronics $	Central Office DLC-L Terminal Investment	= IF (TotalCLLILinesOnDLCL = 0, 0, CHOOSE (GridPlantIndex, 0, 0, DLC_LDiscount * ((INT (TotalCLLILinesOnDLCL / MaxCOTDLCL) * VLOOKUP (MaxCOTDLCL, DLCCOTInvestment, 2) + VLOOKUP (MAX (MOD (TotalCLLILinesOnDLCL, MaxCOTDLCL) , 673) , DLCCOTInvestment, 2))) / TotalCLLILinesOnDLCL + COTDLCLPerLine) * GridLinesOnDLCL)
E	DLCSFiberElectronics	DLC-S On Fiber Electronics $		= IF (TotalLinesServedInGrid = 0, 0, CHOOSE (GridPlantIndex, 0, DLC_SDiscount * (VLOOKUP (VoiceGradeLinesEquippedAtTerminal, DigitalCarrierCost, 2) + VoiceGradeLinesEquippedAtTerminal * VLOOKUP (VoiceGradeLinesEquippedAtTerminal, DigitalCarrierCost, 3)) , 0))
F	GridLinesonDLCS	Grid Lines Served by DLC-S	Grid Lines served by DLC-L	= IF (GridPlantIndex = 2, VoiceGradeLinesEquippedAtTerminal, 0)
G	TotalCLLILinesonDLCS	Total CLLI Lines Served by DLC-S	Total CLLI Lines served by DLC-L	= SUMIF (OFFSET (GridPlantIndex, 0, 0, LastDataRow, 1) , 2, OFFSET (VoiceGradeLinesEquippedAtTerminal, 0, 0, LastDataRow, 1))
H	COTSElectronics	COT-S Electronics $	Central Office DLC-L Terminal Investment	= IF (TotalCLLILinesOnDLCS = 0, 0, CHOOSE (GridPlantIndex, 0, DLC_SDiscount * (CEILING ((TotalCLLILinesOnDLCS / MaxCOTDLCS) , 1) * VLOOKUP (MaxCOTDLCS, DLCCOTInvestment, 2) / TotalCLLILinesOnDLCS + COTDLCSPerLine) , 0) * GridLinesOnDLCS)
I	CopperDS1Electronics	Copper DS1 Electronics		= CHOOSE (GridPlantIndex, DLC_SDiscount * CEILING (GridLinesProvisionedAsDS1s / 24, 1) * CopperT1, 0, 0)
J	FiberDS1Electronics	Fiber DS1 Electronics		= DLC_LDiscount * DS3DS1UnitsPerTerminalLocation * OpticsCost
K	TotalElectronics	Total Electronics		= DLCLFiberElectronics + DLCSFiberElectronics + CopperDS1Electronics + FiberDS1Electronics + COTSElectronics + COTLElectronics
L	MaxSizeFDICost	Max Size FDI Cost		= VLOOKUP (MaxSizeFDI, OutdoorSAICostTable, DensityColumnPointer)
M	ULQFDITotalPairs	ULQ FDI Total Pairs		= ULQResidualCableSize + ULQMaxSizeConnectingCables * MaxFeederSize + 2 * (ULQMaxSizeBackboneCables * MaxDistrSize + ULQResidualBackboneCableSize)
N	ULQFDIsatMaxCapacity	ULQ FDIs At Max Capacity		= IF (ULQFDITotalPairs > MaxSizeFDI, TRUNC (ULQFDITotalPairs / MaxSizeFDI, 1) , 0)
O	ResidualULQFDIPairs	Residual ULQ FDI Pairs		= ULQFDITotalPairs - (MaxSizeFDI * ULQFDIsatMaxCapacity)
P	ULQFDICost	ULQ FDI Cost		= IF (ULQFDITotalPairs = 0, 0, VLOOKUP (ResidualULQFDIPairs, OutdoorSAICostTable, DensityColumnPointer) + ULQFDIsatMaxCapacity * MaxSizeFDICost)

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

Q	LLQFDITotalPairs	LLQ FDI Total Pairs		= LLQResidualCableSize + LLQMaxSizeConnectingCables * MaxFeederSize + 2 * (LLQMaxSizeBackboneCables * MaxDistrSize + LLQResidualBackboneCableSize)
R	LLQFDIsatMaxCapacity	LLQ FDIs At Max Capacity		= IF (LLQFDITotalPairs > MaxSizeFDI, TRUNC (LLQFDITotalPairs / MaxSizeFDI, 1) , 0)
S	ResidualLLQFDIPairs	Residual LLQ FDI Pairs		= LLQFDITotalPairs - (MaxSizeFDI * LLQFDIsatMaxCapacity)
T	LLQFDICost	LLQ FDI Cost		= IF (LLQFDITotalPairs = 0, 0, VLOOKUP (ResidualLLQFDIPairs, OutdoorSAICostTable, DensityColumnPointer) + LLQFDIsatMaxCapacity * MaxSizeFDICost)
U	URQFDITotalPairs	URQ FDI Total Pairs		= URQResidualCableSize + URQMaxSizeConnectingCables * MaxFeederSize + 2 * (URQMaxSizeBackboneCables * MaxDistrSize + URQResidualBackboneCableSize)
V	URQFDIsatMaxCapacity	URQ FDIs At Max Capacity		= IF (URQFDITotalPairs > MaxSizeFDI, TRUNC (URQFDITotalPairs / MaxSizeFDI, 1) , 0)
W	ResidualURQFDIPairs	Residual URQ FDI Pairs		= URQFDITotalPairs - (MaxSizeFDI * URQFDIsatMaxCapacity)
X	URQFDICost	URQ FDI Cost		= IF (URQFDITotalPairs = 0, 0, VLOOKUP (ResidualURQFDIPairs, OutdoorSAICostTable, DensityColumnPointer) + URQFDIsatMaxCapacity * MaxSizeFDICost)
Y	LRQFDITotalPairs	LRQ FDI Total Pairs		= LRQResidualCableSize + LRQMaxSizeConnectingCables * MaxFeederSize + 2 * (LRQMaxSizeBackboneCables * MaxDistrSize + LRQResidualBackboneCableSize)
Z	LRQFDIsatMaxCapacity	LRQ FDIs At Max Capacity		= IF (LRQFDITotalPairs > MaxSizeFDI, TRUNC (LRQFDITotalPairs / MaxSizeFDI, 1) , 0)
AA	ResidualLRQFDIPairs	Residual LRQ FDI Pairs		= LRQFDITotalPairs - (MaxSizeFDI * LRQFDIsatMaxCapacity)
AB	LRQFDICost	LRQ FDI Cost		= IF (LRQFDITotalPairs = 0, 0, VLOOKUP (ResidualLRQFDIPairs, OutdoorSAICostTable, DensityColumnPointer) + LRQFDIsatMaxCapacity * MaxSizeFDICost)
AC	FeederDistributionInterfaceCost	Feeder Distribution Interface Cost		= ULQFDICost + LLQFDICost + URQFDICost + LRQFDICost
AD	AllocationFDItoAerialCopper	Allocation Of FDI To Aerial Copper		= IF (TotalDLCFDICablematerial = 0, 0, FeederDistributionInterfaceCost * TotalDLCFDIAerialCableMaterial / TotalDLCFDICablematerial)
AE	AllocationFDItoBuriedCopper	Allocation Of FDI To Buried Copper		= IF (TotalDLCFDICablematerial = 0, 0, FeederDistributionInterfaceCost * TotalDLCFDIBuriedCableMaterial / TotalDLCFDICablematerial)
AF	AllocationFDItoUGCopper	Allocation Of FDI To UG Copper		= IF (TotalDLCFDICablematerial = 0, 0, FeederDistributionInterfaceCost * TotalDLCFDIUGCableMaterial / TotalDLCFDICablematerial)

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

Sheet: DLC to FDI

Col	Range Name	Column Name	Column Comment	Formula
A	LeftSideCommoncableDistance	Left Side Common Cable Distance		= IF (ULQHorizontalfttoRoadCentroid < = LLQHorizontalfttoRoadCentroid, ULQHorizontalfttoRoadCentroid, LLQHorizontalfttoRoadCentroid) * SoilSlopeFactor
B	ULQConnectingCableDistance	ULQ Connecting Cable Distance		= IF (ULQPairsrequired = 0, 0, (ULQHorizontalfttoRoadCentroid + ULQverticalfttoRoadCentroid) * SoilSlopeFactor - LeftSideCommoncableDistance)
C	ULQWorkingpairs	ULQ Working Pairs		= ResidenceLineBalancingFctr * (ULQHouseholds * (VLOOKUP (StateId, StateInformationTable, 2) - 1) + ULQHousingUnits) + BusinessLineBalancingFctr * ULQBusinessLines
D	ULQPairsrequired	ULQ Pairs Required		= ULQWorkingpairs / FeederFill
E	ULQMaxSizeConnectingCables	ULQ Max Size Connecting Cables		= IF (ULQPairsrequired > MaxFeederSize, TRUNC (ULQPairsrequired / MaxFeederSize) , 0)
F	ULQResidualCableSize	ULQ Residual Cable Size		= IF (ULQPairsrequired = 0, 0, INDEX (FeederCableSize, MATCH (ULQPairsrequired - MaxFeederSize * ULQMaxSizeConnectingCables, FeederCableSize, - 1) , 1))
G	ULQUGCableDistance	ULQ Underground Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 2) * ULQConnectingCableDistance
H	ULQDucts	ULQ Ducts		= IF (ULQUGCableDistance < 1, 0, ULQMaxSizeConnectingCables + 2)
I	ULQManholes	ULQ Manholes		= IF (ULQUGCableDistance < 1, 0, ULQUGCableDistance / VLOOKUP (Density, FeederSpacingTable, 2))
J	ULQStandardManholeCost	ULQ Standard Manhole Cost		= IF (OR (ULQManholes = 0, ULQDucts > 9) , 0, VLOOKUP (ULQDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
K	ULQExpandedManholeCost	ULQ Expanded Manhole Cost		= IF (OR (ULQManholes = 0, ULQDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (ULQDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
L	ULQConduitCost	ULQ Conduit Cost		= IF (ULQUGCableDistance < 1, 0, WaterCostMultiplier * ULQUGCableDistance * (IF (FiberDepthCondition = 1, VLOOKUP (Density, HardRockStructure, 2) , IF (OR (FiberDepthCondition = 2, SurfaceIndicator = 1) , VLOOKUP (Density, SoftRockStructure, 2) , VLOOKUP (Density, NormalStructure, 2))) + ULQDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer)) + ULQManholes * IF (ULQDucts > 9, ULQExpandedManholeCost, ULQStandardManholeCost))
M	ULQAerialcableDistance	ULQ Aerial Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 4) * ULQConnectingCableDistance
N	ULQCableGauge	ULQ Cable Gauge		= IF ((LeftSideCommoncableDistance + ULQConnectingCableDistance + SQRT (ULQArea) * 5280) > 10800, 24, 26)
O	ULQAerialcableMaterial	ULQ Aerial Cable Material $		= IF (ULQResidualCableSize = 0, 0, ULQAerialcableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (ULQResidualCableSize, IF (ULQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + ULQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (ULQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))
P	ULQBuriedCableDistance	ULQ Buried Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 3) * ULQConnectingCableDistance

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

Q	ULQBuriedCable Material	ULQ Buried Cable Material $		= IF (ULQResidualCableSize = 0, 0, ULQBuriedCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (ULQResidualCableSize, IF (ULQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + ULQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (ULQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))
R	ULQUGCableMat erial	ULQ Underground Cable Material $		= IF (ULQResidualCableSize = 0, 0, ULQUGCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (ULQResidualCableSize, IF (ULQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + ULQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (ULQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))
S	LLQConnectingC ableDistance	LLQ Connecting Cable Distance		= IF (LLQPairsRequired = 0, 0, (LLQHorizontalftoRoadCentroid + LLQVerticalFtToRoadCentroid) * SoilSlopeFactor - LeftSideCommoncableDistance)
T	LLQWorkingPairs	LLQ Working Pairs		= ResidenceLineBalancingFctr * (LLQHouseholds * (VLOOKUP (StateId, StateInformationTable, 2) - 1) + LLQHousingUnits) + BusinessLineBalancingFctr * LLQBusinessLines
U	LLQPairsRequired	LLQ Pairs Required		= LLQWorkingPairs / FeederFill
V	LLQMaxSizeConn ectingCables	LLQ Max Size Connecting Cables		= IF (LLQPairsRequired > MaxFeederSize, TRUNC (LLQPairsRequired / MaxFeederSize) , 0)
W	LLQResidualCabl eSize	LLQ Residual Cable Size		= IF (LLQPairsRequired = 0, 0, INDEX (FeederCableSize, MATCH (LLQPairsRequired - LLQMaxSizeConnectingCables * MaxFeederSize, FeederCableSize, - 1) , 1))
X	LLQUGCableDist ance	LLQ Underground Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 2) * LLQConnectingCableDistance
Y	LLQDucts	LLQ Ducts		= IF (LLQUGCableDistance = 0, 0, LLQMaxSizeConnectingCables + 2)
Z	LLQManholes	LLQ Manholes		= IF (LLQUGCableDistance < 1, 0, LLQUGCableDistance / VLOOKUP (Density,FeederSpacingTable, 2))
AA	LLQStandardMan holeCost	LLQ Standard Manhole Cost		= IF (OR (LLQManholes = 0, LLQDucts > 9) , 0, VLOOKUP (LLQDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
AB	LLQExpandedMa nholeCost	LLQ Expanded Manhole Cost		= IF (OR (LLQManholes = 0, LLQDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (LLQDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
AC	LLQConduitCost	LLQ Conduit Cost		= IF (LLQUGCableDistance < 1, 0, WaterCostMultiplier * LLQUGCableDistance * (IF (FiberDepthCondition = 1, VLOOKUP (Density, HardRockStructure, 2) , IF (OR (FiberDepthCondition = 2, SurfaceIndicator = 1) , VLOOKUP (Density, SoftRockStructure, 2) , VLOOKUP (Density, NormalStructure, 2))) + LLQDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer)) + LLQManholes * IF (LLQDucts > 9, LLQExpandedManholeCost, LLQStandardManholeCost))
AD	LLQAerialcableDi stance	LLQ Aerial Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 4) * LLQConnectingCableDistance
AE	LLQCableGauge	LLQ Cable Gauge		= IF ((LeftSideCommoncableDistance + LLQConnectingCableDistance + SQRT (LLQArea) * 5280) > 10800, 24, 26)
AF	LLQAerialcableM aterial	LLQ Aerial Cable Material $		= IF (LLQResidualCableSize = 0, 0, LLQAerialCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LLQResidualCableSize, IF (LLQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + LLQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (LLQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))
AG	LLQBuriedCable Distance	LLQ Buried Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 3) * LLQConnectingCableDistance

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
...tion contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

AH	LLQBuriedCable Material	LLQ Buried Cable Material $		= IF (LLQResidualCableSize = 0, 0, LLQBuriedCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LLQResidualCableSize, IF (LLQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + LLQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (LLQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))
AI	LLQUGCableMat erial	LLQ Underground Cable Material $		= IF (LLQResidualCableSize = 0, 0, LLQUGCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LLQResidualCableSize, IF (LLQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + LLQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (LLQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))
AJ	LeftSideCommon CablePairsRequire d	Left Side Common Cable Pairs Required		= ULQPairsrequired + LLQPairsRequired
AK	NumberLeftComm onMaxSizeCables	Number Of Left Common Max Size Cables		= IF (LeftSideCommonCablePairsRequired > MaxFeederSize, CEILING (LeftSideCommonCablePairsRequired / MaxFeederSize, 1) , 0)
AL	LeftSideCommon CableResidualSize	Left Side Common Cable Residual Size		= INDEX (FeederCableSize, MATCH (LeftSideCommonCablePairsRequired - (MaxFeederSize * NumberLeftCommonMaxSizeCables) , FeederCableSize, - 1) , 1)
AM	LeftSideCommon UGCableDistance	Left Side Common Underground Cable Distance		= IF (LeftSideCommonCablePairsRequired = 0, 0, VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 2) * LeftSideCommoncableDistance)
AN	LeftSideCommon Ducts	Left Side Common Ducts		= IF (LeftSideCommonUGCableDistance = 0, 0, NumberLeftCommonMaxSizeCables + 2)
AO	LeftSideCommon Manholes	Left Side Common Manholes		= IF (LeftSideCommonUGCableDistance < 1, 0, LeftSideCommonUGCableDistance / VLOOKUP (Density, FeederSpacingTable, 2))
AP	LeftSideCommon StandardManhole Cost	Left Side Common Standard Manhole Cost		= IF (OR (LeftSideCommonManholes = 0, LeftSideCommonDucts > 9) , 0, VLOOKUP (LeftSideCommonDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
AQ	LeftSideCommon ExpandedManhole Cost	Left Side Common Expanded Manhole Cost		= IF (OR (LeftSideCommonManholes = 0, LeftSideCommonDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (LeftSideCommonDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
AR	LeftSideCommon ConduitCost	Left Side Common Conduit Cost		= IF (LeftSideCommonUGCableDistance < 1, 0, WaterCostMultiplier * LeftSideCommonUGCableDistance * (IF (FiberDepthCondition = 1, VLOOKUP (Density, HardRockStructure, 2) , IF (OR (FiberDepthCondition = 2, SurfaceIndicator = 1) , VLOOKUP (Density, SoftRockStructure, 2) , VLOOKUP (Density, NormalStructure, 2))) + LeftSideCommonDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer)) + LeftSideCommonManholes * IF (LeftSideCommonDucts > 9, LeftSideCommonExpandedManholeCost, LeftSideCommonStandardManholeCost))
AS	LeftSideCommon AerialcableDistan ce	Left Side Common Aerial Cable Distance		= IF (LeftSideCommonCablePairsRequired = 0, 0, VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 4) * LeftSideCommoncableDistance)

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
This document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

AT	LeftSideCommon CableGauge	Left Side Common Cable Gauge		= MIN (LLQCableGauge, ULQCableGauge)
AU	LeftSideCommon AerialcableMateri al	Left Side Common Aerial Cable Material $		= IF (LeftSideCommonCableResidualSize = 0, 0, LeftSideCommonAerialcableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LeftSideCommonCableResidualSize, IF (LeftSideCommonCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + NumberLeftCommonMaxSizeCables * VLOOKUP (MaxFeederSize, IF (LeftSideCommonCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))
AV	LeftSideCommon BuriedCableDista nce	Left Side Common Buried Cable Distance		= IF (LeftSideCommonCablePairsRequired = 0, 0, VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 3) * LeftSideCommoncableDistance)
AW	LeftSideCommon BuriedCableMater ial	Left Side Common Buried Cable Material $		= IF (LeftSideCommonCableResidualSize = 0, 0, LeftSideCommonBuriedCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LeftSideCommonCableResidualSize, IF (LeftSideCommonCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + NumberLeftCommonMaxSizeCables * VLOOKUP (MaxFeederSize, IF (LeftSideCommonCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))
AX	LeftSideCommon UGCableMaterial	Left Side Common Underground Cable Material $		= IF (LeftSideCommonCableResidualSize = 0, 0, LeftSideCommoncableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LeftSideCommonCableResidualSize, IF (LeftSideCommonCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + NumberLeftCommonMaxSizeCables * VLOOKUP (MaxFeederSize, IF (LeftSideCommonCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))
AY	RightSideCommo ncableDistance	Right Side Common Cable Distance		= IF (URQHorizontalfttoRoadCentroid < = LRQHorizontalfttoRoadCentroid, URQHorizontalfttoRoadCentroid, LRQHorizontalfttoRoadCentroid) * SoilSlopeFactor
AZ	URQConnectingC ableDistance	URQ Connecting Cable Distance		= IF (URQPairsrequired = 0, 0, (URQHorizontalfttoRoadCentroid + URQverticalfttoRoadCentroid) * SoilSlopeFactor - RightSideCommoncableDistance)
BA	URQWorkingpairs	URQ Working Pairs		= ResidenceLineBalancingFctr * (URQHouseholds * (VLOOKUP (StateId, StateInformationTable, 2) - 1) + URQHousingUnits) + BusinessLineBalancingFctr * URQBusinessLines
BB	URQPairsrequired	URQ Pairs Required		= URQWorkingpairs / FeederFill
BC	URQMaxSizeCon nectingCables	URQ Max Size Connecting Cables		= IF (URQPairsrequired > MaxFeederSize, TRUNC (URQPairsrequired / MaxFeederSize) , 0)
BD	URQResidualCabl eSize	URQ Residual Cable Size		= IF (URQPairsrequired = 0, 0, INDEX (FeederCableSize, MATCH (URQPairsrequired - URQMaxSizeConnectingCables * MaxFeederSize, FeederCableSize, - 1) , 1))
BE	URQUGCableDist ance	URQ Underground Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 2) * URQConnectingCableDistance
BF	URQDucts	URQ Ducts		= IF (URQUGCableDistance < 1, 0, URQMaxSizeConnectingCables + 2)
BG	URQManholes	URQ Manholes		= IF (URQUGCableDistance < 1, 0, URQUGCableDistance / VLOOKUP (Density, FeederSpacingTable, 2))
BH	URQStandardMan holeCost	URQ Standard Manhole Cost		= IF (OR (URQManholes = 0, URQDucts > 9) , 0, VLOOKUP (URQDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
e the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

BI	URQExpandedManholeCost	URQ Expanded Manhole Cost		= IF (OR (URQManholes = 0, URQDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (URQDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
BJ	URQConduitCost	URQ Conduit Cost		= IF (URQUGCableDistance < 1, 0, WaterCostMultiplier * URQUGCableDistance * (IF (FiberDepthCondition = 1, VLOOKUP (Density, HardRockStructure, 2) , IF (OR (FiberDepthCondition = 2, SurfaceIndicator = 1) , VLOOKUP (Density, SoftRockStructure, 2) , VLOOKUP (Density, NormalStructure, 2))) + URQDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer)) + URQManholes * IF (URQDucts > 9, URQExpandedManholeCost, URQStandardManholeCost))
BK	URQAerialcableDistance	URQ Aerial Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 4) * URQConnectingCableDistance
BL	URQCableGauge	URQ Cable Gauge		= IF ((RightSideCommoncableDistance + URQConnectingCableDistance + SQRT (URQArea) * 5280) > 10800, 24, 26)
BM	URQAerialcableMaterial	URQ Aerial Cable Material $		= IF (URQResidualCableSize = 0, 0, URQAerialcableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (URQResidualCableSize, IF (URQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + URQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (URQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))
BN	URQBuriedCable Distance	URQ Buried Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 3) * URQConnectingCableDistance
BO	URQBuriedCable Material	URQ Buried Cable Material $		= IF (URQResidualCableSize = 0, 0, URQBuriedCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (URQResidualCableSize, IF (URQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + URQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (URQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))
BP	URQUGCableMaterial	URQ Underground Cable Material $		= IF (URQResidualCableSize = 0, 0, URQUGCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (URQResidualCableSize, IF (URQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + URQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (URQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))
BQ	LRQConnectingCableDistance	LRQ Connecting Cable Distance		= IF (LRQPairsRequired = 0, 0, (LRQHorizontalfttoRoadCentroid + LRQVerticalFtToRoadCentroid) * SoilSlopeFactor - RightSideCommoncableDistance)
BR	LRQWorkingPairs	LRQ Working Pairs		= ResidenceLineBalancingFctr * (LRQHouseholds * (VLOOKUP (StateId, StateInformationTable, 2) - 1) + LRQHousingUnits) + BusinessLineBalancingFctr * LRQBusinessLines
BS	LRQPairsRequired	LRQ Pairs Required		= LRQWorkingPairs / FeederFill
BT	LRQMaxSizeConnectingCables	LRQ Max Size Connecting Cables		= IF (LRQPairsRequired > MaxFeederSize, TRUNC (LRQPairsRequired / MaxFeederSize) , 0)
BU	LRQResidualCableSize	LRQ Residual Cable Size		= IF (LRQPairsRequired = 0, 0, INDEX (FeederCableSize, MATCH (LRQPairsRequired - LRQMaxSizeConnectingCables * MaxFeederSize, FeederCableSize, - 1) , 1))
BV	LRQUGCableDistance	LRQ Underground Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 2) * LRQConnectingCableDistance
BW	LRQDucts	LRQ Ducts		= IF (LRQUGCableDistance = 0, 0, LRQMaxSizeConnectingCables + 2)
BX	LRQManholes	LRQ Manholes		= IF (LRQUGCableDistance < 1, 0, LRQUGCableDistance / VLOOKUP (Density, FeederSpacingTable, 2))
BY	LRQStandardManholeCost	LRQ Standard Manhole Cost		= IF (OR (LRQManholes = 0, LRQDucts > 9) , 0, VLOOKUP (LRQDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

BZ	LRQExpandedManholeCost	LRQ Expanded Manhole Cost		= IF (OR (LRQManholes = 0, LRQDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (LRQDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
CA	LRQConduitCost	LRQ Conduit Cost		= IF (LRQUGCableDistance < 1, 0, WaterCostMultiplier * LRQUGCableDistance * (IF (FiberDepthCondition = 1, VLOOKUP (Density, HardRockStructure, 2) , IF (OR (FiberDepthCondition = 2, SurfaceIndicator = 1) , VLOOKUP (Density, SoftRockStructure, 2) , VLOOKUP (Density, NormalStructure, 2))) + LRQDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer)) + LRQManholes * IF (LRQDucts > 9, LRQExpandedManholeCost, LRQStandardManholeCost))
CB	LRQAerialcableDistance	LRQ Aerial Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 4) * LRQConnectingCableDistance
CC	LRQCableGauge	LRQ Cable Gauge		= IF ((RightSideCommoncableDistance + LRQConnectingCableDistance + SQRT (LRQArea) * 5280) > 10800, 24, 26)
CD	LRQAerialcableMaterial	LRQ Aerial Cable Material $		= IF (LRQResidualCableSize = 0, 0, LRQAerialCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LRQResidualCableSize, IF (LRQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + LRQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (LRQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))
CE	LRQBuriedCable Distance	LRQ Buried Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 3) * LRQConnectingCableDistance
CF	LRQBuriedCable Material	LRQ Buried Cable Material $		= IF (LRQResidualCableSize = 0, 0, LRQBuriedCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LRQResidualCableSize, IF (LRQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + LRQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (LRQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))
CG	LRQUGCableMaterial	LRQ Underground Cable Material $		= IF (LRQResidualCableSize = 0, 0, LRQUGCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LRQResidualCableSize, IF (LRQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + LRQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (LRQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))
CH	RightSideCommonCablePairsRequired	Right Side Common Cable Pairs Required		= LRQPairsRequired + URQPairsrequired
CI	NumberMaxSizeRightCommonCables	Number Of Max Size Right Common Cables		= IF (RightSideCommonCablePairsRequired > MaxFeederSize, CEILING (RightSideCommonCablePairsRequired / MaxFeederSize, 1) , 0)
CJ	RightSideCommonCableResidualSize	Right Side Common Cable Residual Size		= INDEX (FeederCableSize, MATCH (RightSideCommonCablePairsRequired - NumberMaxSizeRightCommonCables * MaxFeederSize, FeederCableSize, - 1) , 1)
CK	RightSideCommonUGCableDistance	Right Side Common Underground Cable Distance		= IF (RightSideCommonCablePairsRequired = 0, 0, VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 2) * RightSideCommoncableDistance)
CL	RightSideCommonDucts	Right Side Common Ducts		= IF (RightSideCommonUGCableDistance = 0, 0, NumberMaxSizeRightCommonCables + 2)

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

CM	RightSideCommonManholes	Right Side Common Manholes		= IF (RightSideCommonUGCableDistance < 1, 0, RightSideCommonUGCableDistance / VLOOKUP (Density, FeederSpacingTable, 2))
CN	RightSideCommonStandardManholeCost	Right Side Common Standard Manhole Cost		= IF (OR (RightSideCommonManholes = 0, RightSideCommonDucts > 9) , 0, VLOOKUP (RightSideCommonDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
CO	RightSideCommonExpandedManholeCost	Right Side Common Expanded Manhole Cost		= IF (OR (RightSideCommonManholes = 0, RightSideCommonDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (RightSideCommonDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
CP	RightSideCommonConduitCost	Right Side Common Conduit Cost		= IF (RightSideCommonUGCableDistance < 1, 0, WaterCostMultiplier * RightSideCommonUGCableDistance * (IF (FiberDepthCondition = 1, VLOOKUP (Density, HardRockStructure, 2) , IF (OR (FiberDepthCondition = 2, SurfaceIndicator = 1) , VLOOKUP (Density, SoftRockStructure, 2) , VLOOKUP (Density, NormalStructure, 2))) + RightSideCommonDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer)) + RightSideCommonManholes * IF (RightSideCommonDucts > 9, RightSideCommonExpandedManholeCost, RightSideCommonStandardManholeCost))
CQ	RightSideCommonAerialcableDistance	Right Side Common Aerial Cable Distance		= IF (RightSideCommonCablePairsRequired = 0, 0, VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 4) * RightSideCommoncableDistance)
CR	RightSideCommonCableGauge	Right Side Common Cable Gauge		= MIN (LRQCableGauge, URQCableGauge)
CS	RightSideCommonAerialcableMaterial	Right Side Common Aerial Cable Material $		= IF (RightSideCommonCableResidualSize = 0, 0, RightSideCommonAerialcableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (RightSideCommonCableResidualSize, IF (RightSideCommonCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + NumberMaxSizeRightCommonCables * VLOOKUP (MaxFeederSize, IF (RightSideCommonCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))
CT	RightSideCommonBuriedCableDistance	Right Side Common Buried Cable Distance		= IF (RightSideCommonCablePairsRequired = 0, 0, VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 3) * RightSideCommoncableDistance)
CU	RightSideCommonBuriedCableMaterial	Right Side Common Buried Cable Material $		= IF (RightSideCommonCableResidualSize = 0, 0, RightSideCommonBuriedCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (RightSideCommonCableResidualSize, IF (RightSideCommonCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + NumberMaxSizeRightCommonCables * VLOOKUP (MaxFeederSize, IF (RightSideCommonCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))
CV	RightSideCommonUGCableMaterial	Right Side Common Underground Cable Material $		= IF (RightSideCommonCableResidualSize = 0, 0, RightSideCommonUGCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (RightSideCommonCableResidualSize, IF (RightSideCommonCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + NumberMaxSizeRightCommonCables * VLOOKUP (MaxFeederSize, IF (RightSideCommonCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))
CW	TotalDLCFDIAerialCableMaterial	Total DLC/FDI Aerial Cable Material $		= ULQAerialcableMaterial + LLQAerialCableMaterial + LeftSideCommonAerialcableMaterial + URQAerialcableMaterial + LRQAerialCableMaterial + RightSideCommonAerialcableMaterial

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
...ment is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

CX	TotalDLCFDIBuriedCableMaterial	Total DLC/FDI Buried Cable Material $		= ULQBuriedCableMaterial + LLQBuriedCableMaterial + LeftSideCommonBuriedCableMaterial + URQBuriedCableMaterial + LRQBuriedCableMaterial + RightSideCommonBuriedCableMaterial
CY	TotalDLCFDIUGCableMaterial	Total DLC/FDI Underground Cable Material $		= ULQUGCableMaterial + LLQUGCableMaterial + LeftSideCommonUGCableMaterial + URQUGCableMaterial + LRQUGCableMaterial + RightSideCommonUGCableMaterial
CZ	TotalDLCFDICablematerial	Total DLC/FDI Cable Material $		= SUM (CW3:CY3)
DA	TotalAerialDLCFDICableDistance	Total Aerial DLC/FDI Cable Distance		= ULQAerialcableDistance + LLQAerialCableDistance + LeftSideCommoncableDistance + URQAerialcableDistance + LRQAerialCableDistance + RightSideCommonAerialcableDistance
DB	NumberofDLCFDIPoles	Number Of DLC/FDI Poles		= IF (TotalAerialDLCFDICableDistance = 0, 0, TotalAerialDLCFDICableDistance / VLOOKUP (Density, FeederSpacingTable, 3))
DC	TotalAerialDLCFDICableStructure	Total Aerial DLC/FDI Cable Structure $		= NumberofDLCFDIPoles * WaterCostMultiplier * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 6)
DD	TotalburiedDLCFDICableDistance	Total buried DLC/FDI Cable Distance		= ULQBuriedCableDistance + LLQBuriedCableDistance + LeftSideCommonBuriedCableDistance + URQBuriedCableDistance + LRQBuriedCableDistance + RightSideCommonBuriedCableDistance
DE	TotalBuriedDLCFDICableStructure	Total Buried DLC/FDI Cable Structure $		= IF (TotalburiedDLCFDICableDistance < 1, 0, WaterCostMultiplier * TotalburiedDLCFDICableDistance * VLOOKUP (Density, CHOOSE (CopperDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 4))
DF	TotalUGDLCFDICableStructure	Total Underground DLC/FDI Cable Structure $		= ULQConduitCost + LLQConduitCost + LeftSideCommonConduitCost + URQConduitCost + LRQConduitCost + RightSideCommonConduitCost
DG	TotalDLCFDICableStructure	Total DLC/FDI Cable Structure $		= TotalAerialDLCFDICableStructure + TotalBuriedDLCFDICableStructure + TotalUGDLCFDICableStructure
DH	TotalDLCtoFDICableCost	Total DLC to FDI Cable Cost		= TotalDLCFDICablematerial + TotalDLCFDICableStructure

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
ent is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

ATTACHMENT 4

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.