Business Energy Claim

Valuation Report

</< # First Page Setup  
acr1c1=Home.R94C3;acr1c1=num.acr1c1;acr1c2=Home.C100;acr1c2=num.acr1c2;acr2c1=Home.C95;acr2c1=num.acr2c1;acr2c2=Home.C101;acr2c2=num.acr2c2;acr3c1=Home.C96;acr3c1=num.acr3c1;acr3c2=Home.C102;acr3c2=num.acr3c2;rc1=Home.F91;rc1=num.rc1;rc2=Home.F92;rc2=num.rc2;rc3=Home.F93;rc3=num.rc3;  
"";>>

|  |  |
| --- | --- |
| **Claim Summary** | |
| Case ID | </<CaseID>> |
| Claimant Company | </<CNameCell>> |
| Energy Supplier | </<SNameCell>> |
| Date of Report | </<date.ClaimDate :sdate>> |

|  |  |  |  |
| --- | --- | --- | --- |
| **Total Refund of Agreement Cost** | | | |
| **Up to Date of Report** | | **After Date of Report** | |
| Quantum | </<acr1c1 :curr>> | Quantum | </<acr1c2 :curr>> |
| Compensatory Interest | </<acr2c1 :curr>> | Compensatory Interest | </<acr2c2 :curr>> |
| Quantum Plus Compensatory Interest | </<acr3c1 :curr>> | Quantum Plus Compensatory Interest | </<acr3c2 :curr>> |

|  |  |  |  |
| --- | --- | --- | --- |
| **Total Refund of Commission** | | | |
| Quantum | </<rc1 :curr>> | Compensatory Interest | </<rc2 :curr>> |
| Quantum Plus Compensatory Interest | | </<rc3 :curr>> | |

Table of Contents

[1. Claimant and Defendant Details 3](#_Toc155364431)

[2. Contract Data 4](#_Toc155364432)

[2.1 Contract #1 4](#_Toc155364433)

[3. Methodology 7](#_Toc155364434)

[3.1 Refund of Agreement Cost 7](#_Toc155364435)

[3.2 Refund of Commission 8](#_Toc155364436)

[4. Quantum 9](#_Toc155364437)

[4.1 Refund of Agreement Cost 9](#_Toc155364438)

[4.2 Refund of Commission 11](#_Toc155364439)

[5. Fair Rate Methodology 12](#_Toc155364440)

[6. Disclaimer and Confidentiality 13](#_Toc155364441)

[7. Notes and List of Assumptions 14](#_Toc155364442)

## Claimant and Defendant Details

|  |  |
| --- | --- |
| **Claimant Company Details** | |
| Company | </<CNameCell>> |
| Company Registered Address | </<CAddCell>> |
| Company Registration Number | </<CNumCell>> |

|  |  |
| --- | --- |
| **Energy Supplier Details:** | |
| Supplier | </<SNameCell>> |
| Supplier Address | </<SAddCell>> |
| Supplier Status | </<SStatusCell>> |
| Supplier Company Registration Number | </<SNumCell>> |

## Contract Data

### 2.1 Contract #</<CN\_Ord>>

C=2 S=2 E=5  
</<C\_RowStart>>  
</<C\_RowEnd>>  
</<C\_Count>>  
</<  
# Gather information from every row of a contract  
i = C\_RowStart  
while (i < C\_RowEnd + 1) {  
 let CurrentRow = row.CSiteCol + i; let prefix = "R" + int.CurrentRow + "C";  
   
 # Labelled Headers  
 let SiteCell = prefix + col.CSiteCol;  
 let TypeCell = prefix + col.CTypeCol;  
 let SignCell = prefix + col.CSignCol;  
 let StartCell = prefix + col.CStartCol;  
 let CEndCell = prefix + col.CCEndCol;  
 let TermCell = prefix + col.CTermCol;  
 let EAC1Cell = prefix + col.CEAC1Col;  
 let Bill1Cell = prefix + col.CBill1Col;  
 let EAC2Cell = prefix + col.CEAC2Col;  
 let Bill2Cell = prefix + col.CBill2Col;  
 let PeakCell = prefix + col.CPeakCol;  
 let BaseCell = prefix + col.CBaseCol;  
 let LengthCell = prefix + col.CLengthCol;  
 let EndCell = prefix + col.CEndCol;  
 let Usage1Cell = prefix + col.CUsage1Col;  
 let Stat1Cell = prefix + col.CStat1Col;  
 let Claim1Cell = prefix + col.CClaim1Col;  
 let EstUseCell = prefix + col.CEstUseCol;  
 let ActUseCell = prefix + col.CActUseCol;  
 let Rate2Cell = prefix + col.CRate2Col;  
 let Usage2Cell = prefix + col.CUsage2Col;  
 let Stat2Cell = prefix + col.CStat2Col;  
 let Claim2Cell = prefix + col.CClaim2Col;  
  
 # Unlabelled Headers  
 let UtilityCell = CX.C4; UtilityCell = prefix + col.UtilityCell;  
 let EAC3Cell = CX.N5; EAC3Cell = prefix + col.EAC3Cell;  
 let Bill3Cell = CX.O5; Bill3Cell = prefix + col.Bill3Cell;  
 let EAC4Cell = CX.P5; EAC4Cell = prefix + col.EAC4Cell;  
 let Bill4Cell = CX.Q5; Bill4Cell = prefix + col.Bill4Cell;  
 let EAC5Cell = CX.R5; EAC5Cell = prefix + col.EAC5Cell;  
 let Bill5Cell = CX.S5; Bill5Cell = prefix + col.Bill5Cell;  
 let EAC6Cell = CX.T5; EAC6Cell = prefix + col.EAC6Cell;  
 let Bill6Cell = CX.U5; Bill6Cell = prefix + col.Bill6Cell;  
  
 # Extra Headers  
 #let OverPreCell = CX.AJ4; OverPreCell = prefix + col.OverPreCell;  
 #let SCPreCell = CX.AK4; SCPreCell = prefix + col.SCPreCell;  
 #let StatPreCell = CX.AL4; StatPreCell = prefix + col.StatPreCell;  
 #let OverPostCell = CX.AM4; OverPostCell = prefix + col.OverPostCell;  
 #let SCPostCell = CX.AN4; SCPostCell = prefix + col.SCPostCell;  
 #let StatPostCell = CX.AO4; StatPostCell = prefix + col.StatPostCell;  
  
 # Subsequent Rows  
 #if (i > C\_RowStart) {   
 # if (i == C\_RowStart + 1) {   
 # let fromMsg = " (from " + C\_StartDate + ")\n";  
 # C\_drates = C\_drates + fromMsg;   
 # C\_nrates = C\_nrates + fromMsg;   
 # C\_erates = C\_erates + fromMsg;   
 # C\_wrates = C\_wrates + fromMsg;   
 # C\_xrates = C\_xrates + fromMsg;   
 # C\_yrates = C\_yrates + fromMsg;   
 # }  
  
 # let TempStart = Contracts.StartCell  
 # let fromMsg = " (from " + date.TempStart + ")\n";  
 # let temp\_drates = CX.Bill1Cell; C\_drates = C\_drates + num.temp\_drates + fromMsg;  
 # let temp\_nrates = CX.Bill2Cell; C\_nrates = C\_nrates + num.temp\_nrates + fromMsg;  
 # let temp\_erates = CX.Bill3Cell; C\_erates = C\_erates + num.temp\_erates + fromMsg;  
 # let temp\_wrates = CX.Bill4Cell; C\_wrates = C\_wrates + num.temp\_wrates + fromMsg;  
 # let temp\_xrates = CX.Bill5Cell; C\_xrates = C\_xrates + num.temp\_xrates + fromMsg;  
 # let temp\_yrates = CX.Bill6Cell; C\_yrates = C\_yrates + num.temp\_yrates + fromMsg;  
 #   
 # # Accumulate all summable fields like quantum  
 # let temp\_ACOverpayPre = CX.OverPreCell; C\_ACOverpayPre = C\_ACOverpayPre + num.temp\_ACOverpayPre;  
 # let temp\_ACOverpayPost = CX.OverPostCell; C\_ACOverpayPost = C\_ACOverpayPost + num.temp\_ACOverpayPost;  
 # let temp\_ACSChargePre = CX.SCPreCell; C\_ACSChargePre = C\_ACSChargePre + num.temp\_ACSChargePre;  
 # let temp\_ACSChargePost = CX.SCPostCell; C\_ACSChargePost = C\_ACSChargePost + num.temp\_ACSChargePost;  
 # let temp\_ACStatPre = CX.StatPreCell; C\_ACStatPre = C\_ACStatPre + num.temp\_ACStatPre;  
 # let temp\_ACStatPost = CX.StatPostCell; C\_ACStatPost = C\_ACStatPost + num.temp\_ACStatPost;  
 # let temp\_RCQuantum = CX.Usage2Cell; C\_RCQuantum = C\_RCQuantum + num.temp\_RCQuantum;  
 # let temp\_RCStat = CX.Stat2Cell; C\_RCStat = C\_RCStat + num.temp\_RCStat;  
 # let temp\_RCQuantumStat = CX.Claim2Cell; C\_RCQuantum = C\_RCQuantum + num.temp\_RCQuantumStat;  
 #}  
  
 # First Row  
 #else {  
  
 # C\_AverageMonthly = "IDK Where?";  
 # C\_SiteName = CX.SiteCell  
 # C\_ContractType = CX.TypeCell;  
 # C\_UtilityType = CX.UtilityCell;  
 # C\_ContractLength = CX.LengthCell;  
 # C\_SignDate = CX.SignCell; C\_SignDate = date.C\_SignDate;  
 # C\_StartDate = CX.StartCell; C\_StartDate = date.C\_StartDate;  
 # C\_EndDate = CX.CEndCell; C\_EndDate = date.C\_EndDate;  
 # C\_drates = CX.Bill1Cell; C\_drates = num.C\_drates;  
 # C\_nrates = CX.Bill2Cell; C\_nrates = num.C\_nrates;  
 # C\_erates = CX.Bill3Cell; C\_erates = num.C\_erates;  
 # C\_wrates = CX.Bill4Cell; C\_wrates = num.C\_wrates;  
 # C\_xrates = CX.Bill5Cell; C\_xrates = num.C\_xrates;  
 # C\_yrates = CX.Bill6Cell; C\_yrates = num.C\_yrates;  
 # C\_peakRate = CX.PeakCell; C\_peakRate = num.C\_peakRate;  
 # C\_baseRate = CX.BaseCell; C\_baseRate = num.C\_baseRate;  
 # C\_deac = CX.EAC1Cell; C\_deac = num.C\_deac;  
 # C\_neac = CX.EAC2Cell; C\_neac = num.C\_neac;  
 # C\_eeac = CX.EAC3Cell; C\_eeac = num.C\_eeac;  
 # C\_weac = CX.EAC4Cell; C\_weac = num.C\_weac;  
 # C\_xeac = CX.EAC5Cell; C\_xeac = num.C\_xeac;  
 # C\_yeac = CX.EAC6Cell; C\_yeac = num.C\_yeac;  
 # C\_ACOverpayPre = CX.OverPreCell; C\_ACOverpayPre = num.C\_ACOverpayPre;  
 # C\_ACOverpayPost = CX.OverPostCell; C\_ACOverpayPost = num.C\_ACOverpayPost;  
 # C\_ACSChargePre = CX.SCPreCell; C\_ACSChargePre = num.C\_ACSChargePre;  
 # C\_ACSChargePost = CX.SCPostCell; C\_ACSChargePost = num.C\_ACSChargePost;  
 # C\_ACStatPre = CX.StatPreCell; C\_ACStatPre = num.C\_ACStatPre;  
 # C\_ACStatPost = CX.StatPostCell; C\_ACStatPost = num.C\_ACStatPost;  
 # C\_RCQuantum = CX.Usage2Cell; C\_RCQuantum = num.C\_RCQuantum;  
 # C\_RCStat = CX.Stat2Cell; C\_RCStat = num.C\_RCStat;  
 # C\_RCQuantumStat = CX.Claim2Cell; C\_RCQuantumStat = num.C\_RCQuantumStat;  
 #   
 # # Get Site Details  
 # let j = 1;  
 # let looking2 = true;  
 # while (looking2) {  
 # let SiteRow = row.SNameCol + j;  
 # let SitePrefix = "R" + int.SiteRow + "C";  
 # let SiteNameCell = SitePrefix + col.SNameCol; SiteNameCell = Home.SiteNameCell  
 # if (str.SiteNameCell == str.C\_SiteName) {  
 # let SiteSAddressCell = SitePrefix + col.SAddCol; C\_SiteAddress = Home.SiteSAddressCell;  
 # let SiteAccountCell = SitePrefix + col.SAccCol; C\_SiteAccount = Home.SiteAccountCell;  
 # let SiteMPANCell = SitePrefix + col.SMPANCol; C\_SiteMPAN = Home.SiteMPANCell;  
 # let SiteMeterCell = SitePrefix + col.SMeterCol; C\_SiteMeter = Home.SiteMeterCell;  
  
 # looking2 = false;  
 # }  
 # else if (Home.SiteNameCell == "") {  
 # looking2 = false;  
 # }  
  
 # j = j + 1;  
 # }  
 #}  
  
 i = i + 1;  
}  
"";>>  
</< # Post While Loop Perhaps  
  
C\_RCSCharge = (C\_ACSChargePre + C\_ACSChargePost) \* num.ComRate;  
C\_RCCommission = C\_RCQuantum - C\_RCSCharge;  
C\_ACQuantumPre = C\_ACOverpayPre + C\_ACSChargePre;  
C\_ACQuantumPost = C\_ACOverpayPost + C\_ACSChargePost;  
C\_ACQuantumStatPre = C\_ACQuantumPre + C\_ACStatPre;  
C\_ACQuantumStatPost = C\_ACQuantumPost + C\_ACStatPost;  
"";>></< # Update Totals (Possibly unnecessary)  
T\_ACOverpayPre=T\_ACOverpayPre+C\_ACOverpayPre;  
T\_ACOverpayPost=T\_ACOverpayPost+C\_ACOverpayPost;  
T\_ACSChargePre=T\_ACSChargePre+C\_ACSChargePre;  
T\_ACSChargePost=T\_ACSChargePost+C\_ACSChargePost;  
T\_ACQuantumPre=T\_ACQuantumPre+C\_ACQuantumPre;  
T\_ACQuantumPost=T\_ACQuantumPost+C\_ACQuantumPost;  
T\_ACStatPre=T\_ACStatPre+C\_ACStatPre;  
T\_ACStatPost=T\_ACStatPost+C\_ACStatPost;  
T\_ACQuantumStatPre=T\_ACQuantumStatPre+C\_ACQuantumStatPre;  
T\_ACQuantumStatPost=T\_ACQuantumStatPost+C\_ACQuantumStatPost;  
T\_RCCommission=T\_RCCommission+C\_RCCommission;  
T\_RCSCharge=T\_RCSCharge+C\_RCSCharge;  
T\_RCQuantum=T\_RCQuantum+C\_RCQuantum;  
T\_RCStat=T\_RCStat+C\_RCStat;  
T\_RCQuantumStat=T\_RCQuantumStat+C\_RCQuantumStat;  
"";>>

|  |  |
| --- | --- |
| **Energy Broker Details #1:** | |
| Brokerage | </<BNameCell>> |
| Brokerage Address | </<BAddCell>> |
| Brokerage Company Registration Number | </<BNumCell>> |

|  |  |
| --- | --- |
| **Contract Details #1:** | |
| Supply Address | </<C\_SiteAddress>> |
| Agreement Number | </<C\_SiteAccount>> |
| MPAN/MPRN | </<C\_SiteMPAN>> |
| Meter Number | </<C\_SiteMeter>> |
| Type of Contract | </<# Filled in by the Audit Team>> |
| Utility Type | </<C\_UtilityType>> |
| Date of Contract (Signed) | </<C\_SignDate :sdate>> |
| Contract Length (Months) | </<int.C\_ContractLength >> |
| Supply Start Date | </<C\_StartDate :sdate>> |
| Supply End Date | </<C\_EndDate :sdate>> |
| Status of Contract | </<if (C\_EndDate > date.ClaimDate) {"Active";} else {"Inactive";}>> |
| Contractual Energy Rate (per kWh) | </<"[TBD]">> |
| Estimated Annual Consumption (kWh) | </<"[TBD]">> |
| Actual Annual Consumption (kWh) | </<"[TBD]">> |
| Fair Energy Rate (per kWh) | </<"Peak: " + ("%.4fp" % C\_peakRate)>> </<"Base: " + ("%.4fp" % C\_baseRate)>> |
| Was the fact of Commission disclosed at the point of Contract? | No |
| Was the amount of Commission disclosed at the point of the contract? | No |

|  |  |
| --- | --- |
| **Refund of Agreement Cost Quantum Details #1:** | |
| **Up to Date of Report** | |
| Overpayment Refund up to Date of Report | </<C\_ACOverpayPre :curr>> |
| Standing Charge Refund up to Date of Report | </<C\_ACSChargePre: curr>> |
| Quantum up to Date of Report | </<C\_ACQuantumPre :curr>> |
| Compensatory Interest up to Date of Report | </<C\_ACStatPre :curr>> |
| Quantum Plus Compensatory Interest up to Date of Report | </<C\_ACQuantumStatPre :curr>> |
| **After Date of Report** | |
| Overpayment Refund after the Date of Report to the Supply End Date | </<C\_ACOverpayPost :curr>> |
| Standing Charge Refund after the Date of Report to the Supply End Date | </<C\_ACSChargePost: curr>> |
| Quantum after the Date of Report to the Supply End Date | </<C\_ACQuantumPost :curr>> |
| Compensatory Interest after the Date of Report to the Supply End Date | </<C\_ACStatPost :curr>> |
| Quantum Plus Compensatory Interest after the Date of Report to the Supply End Date | </<C\_ACQuantumStatPost :curr>> |

|  |  |
| --- | --- |
| **Refund of Commission Quantum Details #1:** | |
| Commission Refund | </<C\_RCCommission :curr>> |
| Standing Charge Refund | </<C\_RCSCharge : curr>> |
| Quantum | </<C\_RCQuantum :curr>> |
| Compensatory Interest | </<C\_RCStat :curr>> |
| Quantum Plus Compensatory Interest | </<C\_RCQuantumStat :curr>> |

### 2.1 Contract #</<CN\_Ord>>

<< # Look for the start and end rows of the current contract  
while (looking and i < MAX\_ITERATION\_LIMIT) {  
 let CurrentRow = row.CSiteCol + i; let prefix = "R" + int.CurrentRow + "C";  
 let siteCell = prefix + col.CSiteCol; siteCell = Contracts.siteCell;  
 if ((f.siteCell == empty or f.siteCell == "") and not (str.siteCell == empty or str.siteCell == "")) {  
 C\_Count = C\_Count + 1;  
 }  
  
 # Break once we've found the contract row  
 if (C\_Count == C2\_Ord and C\_RowStart == 0) { C\_RowStart = i; }  
 else if (C\_Count > C2\_Ordor((str.siteCell == empty or str.siteCell == "") and C\_RowStart != 0)) { C\_RowEnd = i; looking = false; }  
  
 i = i + 1;   
}  
"C="+C\_Count+" S="+C\_RowStart+" E="+C\_RowEnd;>>  
</<C\_RowStart>>  
</<C\_RowEnd>>  
</<C\_Count>>  
</<  
# Gather information from every row of a contract  
i = C\_RowStart  
while (i < C\_RowEnd + 1) {  
 let CurrentRow = row.CSiteCol + i; let prefix = "R" + int.CurrentRow + "C";  
  
 # Labelled Headers  
 let SiteCell = prefix + col.CSiteCol;  
 let TypeCell = prefix + col.CTypeCol;  
 let SignCell = prefix + col.CSignCol;  
 let StartCell = prefix + col.CStartCol;  
 let CEndCell = prefix + col.CCEndCol;  
 let TermCell = prefix + col.CTermCol;  
 let EAC1Cell = prefix + col.CEAC1Col;  
 let Bill1Cell = prefix + col.CBill1Col;  
 let EAC2Cell = prefix + col.CEAC2Col;  
 let Bill2Cell = prefix + col.CBill2Col;  
 let PeakCell = prefix + col.CPeakCol;  
 let BaseCell = prefix + col.CBaseCol;  
 let LengthCell = prefix + col.CLengthCol;  
 let EndCell = prefix + col.CEndCol;  
 let Usage1Cell = prefix + col.CUsage1Col;  
 let Stat1Cell = prefix + col.CStat1Col;  
 let Claim1Cell = prefix + col.CClaim1Col;  
 let EstUseCell = prefix + col.CEstUseCol;  
 let ActUseCell = prefix + col.CActUseCol;  
 let Rate2Cell = prefix + col.CRate2Col;  
 let Usage2Cell = prefix + col.CUsage2Col;  
 let Stat2Cell = prefix + col.CStat2Col;  
 let Claim2Cell = prefix + col.CClaim2Col;  
  
 # Unlabelled Headers  
 let UtilityCell = CX.C4; UtilityCell = prefix + col.UtilityCell;  
 let EAC3Cell = CX.N5; EAC3Cell = prefix + col.EAC3Cell;  
 let Bill3Cell = CX.O5; Bill3Cell = prefix + col.Bill3Cell;  
 let EAC4Cell = CX.P5; EAC4Cell = prefix + col.EAC4Cell;  
 let Bill4Cell = CX.Q5; Bill4Cell = prefix + col.Bill4Cell;  
 let EAC5Cell = CX.R5; EAC5Cell = prefix + col.EAC5Cell;  
 let Bill5Cell = CX.S5; Bill5Cell = prefix + col.Bill5Cell;  
 let EAC6Cell = CX.T5; EAC6Cell = prefix + col.EAC6Cell;  
 let Bill6Cell = CX.U5; Bill6Cell = prefix + col.Bill6Cell;  
  
 # Extra Headers  
#let OverPreCell = CX.AJ4; OverPreCell = prefix + col.OverPreCell;  
#let SCPreCell = CX.AK4; SCPreCell = prefix + col.SCPreCell;  
#let StatPreCell = CX.AL4; StatPreCell = prefix + col.StatPreCell;  
#let OverPostCell = CX.AM4; OverPostCell = prefix + col.OverPostCell;  
#let SCPostCell = CX.AN4; SCPostCell = prefix + col.SCPostCell;  
#let StatPostCell = CX.AO4; StatPostCell = prefix + col.StatPostCell;  
  
 # Subsequent Rows  
#if (i > C\_RowStart) {   
# if (i == C\_RowStart + 1) {   
# let fromMsg = " (from " + C\_StartDate + ")\n";  
# C\_drates = C\_drates + fromMsg;   
# C\_nrates = C\_nrates + fromMsg;   
# C\_erates = C\_erates + fromMsg;   
# C\_wrates = C\_wrates + fromMsg;   
# C\_xrates = C\_xrates + fromMsg;   
# C\_yrates = C\_yrates + fromMsg;   
# }  
  
# let TempStart = Contracts.StartCell  
# let fromMsg = " (from " + date.TempStart + ")\n";  
# let temp\_drates = CX.Bill1Cell; C\_drates = C\_drates + num.temp\_drates + fromMsg;  
# let temp\_nrates = CX.Bill2Cell; C\_nrates = C\_nrates + num.temp\_nrates + fromMsg;  
# let temp\_erates = CX.Bill3Cell; C\_erates = C\_erates + num.temp\_erates + fromMsg;  
# let temp\_wrates = CX.Bill4Cell; C\_wrates = C\_wrates + num.temp\_wrates + fromMsg;  
# let temp\_xrates = CX.Bill5Cell; C\_xrates = C\_xrates + num.temp\_xrates + fromMsg;  
# let temp\_yrates = CX.Bill6Cell; C\_yrates = C\_yrates + num.temp\_yrates + fromMsg;  
#  
# # Accumulate all summable fields like quantum  
# let temp\_ACOverpayPre = CX.OverPreCell; C\_ACOverpayPre = C\_ACOverpayPre + num.temp\_ACOverpayPre;  
# let temp\_ACOverpayPost = CX.OverPostCell; C\_ACOverpayPost = C\_ACOverpayPost + num.temp\_ACOverpayPost;  
# let temp\_ACSChargePre = CX.SCPreCell; C\_ACSChargePre = C\_ACSChargePre + num.temp\_ACSChargePre;  
# let temp\_ACSChargePost = CX.SCPostCell; C\_ACSChargePost = C\_ACSChargePost + num.temp\_ACSChargePost;  
# let temp\_ACStatPre = CX.StatPreCell; C\_ACStatPre = C\_ACStatPre + num.temp\_ACStatPre;  
# let temp\_ACStatPost = CX.StatPostCell; C\_ACStatPost = C\_ACStatPost + num.temp\_ACStatPost;  
# let temp\_RCQuantum = CX.Usage2Cell; C\_RCQuantum = C\_RCQuantum + num.temp\_RCQuantum;  
# let temp\_RCStat = CX.Stat2Cell; C\_RCStat = C\_RCStat + num.temp\_RCStat;  
# let temp\_RCQuantumStat = CX.Claim2Cell; C\_RCQuantum = C\_RCQuantum + num.temp\_RCQuantumStat;  
#}  
  
 # First Row  
#else {  
  
 # C\_AverageMonthly = "IDK Where?";  
# C\_SiteName = CX.SiteCell  
# C\_ContractType = CX.TypeCell;  
# C\_UtilityType = CX.UtilityCell;  
# C\_ContractLength = CX.LengthCell;  
# C\_SignDate = CX.SignCell; C\_SignDate = date.C\_SignDate;  
# C\_StartDate = CX.StartCell; C\_StartDate = date.C\_StartDate;  
# C\_EndDate = CX.CEndCell; C\_EndDate = date.C\_EndDate;  
# C\_drates = CX.Bill1Cell; C\_drates = num.C\_drates;  
# C\_nrates = CX.Bill2Cell; C\_nrates = num.C\_nrates;  
# C\_erates = CX.Bill3Cell; C\_erates = num.C\_erates;  
# C\_wrates = CX.Bill4Cell; C\_wrates = num.C\_wrates;  
# C\_xrates = CX.Bill5Cell; C\_xrates = num.C\_xrates;  
# C\_yrates = CX.Bill6Cell; C\_yrates = num.C\_yrates;  
# C\_peakRate = CX.PeakCell; C\_peakRate = num.C\_peakRate;  
# C\_baseRate = CX.BaseCell; C\_baseRate = num.C\_baseRate;  
# C\_deac = CX.EAC1Cell; C\_deac = num.C\_deac;  
# C\_neac = CX.EAC2Cell; C\_neac = num.C\_neac;  
# C\_eeac = CX.EAC3Cell; C\_eeac = num.C\_eeac;  
# C\_weac = CX.EAC4Cell; C\_weac = num.C\_weac;  
# C\_xeac = CX.EAC5Cell; C\_xeac = num.C\_xeac;  
# C\_yeac = CX.EAC6Cell; C\_yeac = num.C\_yeac;  
# C\_ACOverpayPre = CX.OverPreCell; C\_ACOverpayPre = num.C\_ACOverpayPre;  
# C\_ACOverpayPost = CX.OverPostCell; C\_ACOverpayPost = num.C\_ACOverpayPost;  
# C\_ACSChargePre = CX.SCPreCell; C\_ACSChargePre = num.C\_ACSChargePre;  
# C\_ACSChargePost = CX.SCPostCell; C\_ACSChargePost = num.C\_ACSChargePost;  
# C\_ACStatPre = CX.StatPreCell; C\_ACStatPre = num.C\_ACStatPre;  
# C\_ACStatPost = CX.StatPostCell; C\_ACStatPost = num.C\_ACStatPost;  
# C\_RCQuantum = CX.Usage2Cell; C\_RCQuantum = num.C\_RCQuantum;  
# C\_RCStat = CX.Stat2Cell; C\_RCStat = num.C\_RCStat;  
# C\_RCQuantumStat = CX.Claim2Cell; C\_RCQuantumStat = num.C\_RCQuantumStat;  
#  
# # Get Site Details  
# let j = 1;  
# let looking2 = true;  
# while (looking2) {  
# let SiteRow = row.SNameCol + j;  
# let SitePrefix = "R" + int.SiteRow + "C";  
# let SiteNameCell = SitePrefix + col.SNameCol; SiteNameCell = Home.SiteNameCell  
# if (str.SiteNameCell == str.C\_SiteName) {  
# let SiteSAddressCell = SitePrefix + col.SAddCol; C\_SiteAddress = Home.SiteSAddressCell;  
# let SiteAccountCell = SitePrefix + col.SAccCol; C\_SiteAccount = Home.SiteAccountCell;  
# let SiteMPANCell = SitePrefix + col.SMPANCol; C\_SiteMPAN = Home.SiteMPANCell;  
# let SiteMeterCell = SitePrefix + col.SMeterCol; C\_SiteMeter = Home.SiteMeterCell;  
  
# looking2 = false;  
# }  
# else if (Home.SiteNameCell == "") {  
# looking2 = false;  
# }  
  
# j = j + 1;  
# }  
#}  
  
 i = i + 1;  
}  
"";>>  
</< # Post While Loop Perhaps  
  
C\_RCSCharge = (C\_ACSChargePre + C\_ACSChargePost) \* num.ComRate;  
C\_RCCommission = C\_RCQuantum - C\_RCSCharge;  
C\_ACQuantumPre = C\_ACOverpayPre + C\_ACSChargePre;  
C\_ACQuantumPost = C\_ACOverpayPost + C\_ACSChargePost;  
C\_ACQuantumStatPre = C\_ACQuantumPre + C\_ACStatPre;  
C\_ACQuantumStatPost = C\_ACQuantumPost + C\_ACStatPost;  
"";>></< # Update Totals (Possibly unnecessary)  
T\_ACOverpayPre=T\_ACOverpayPre+C\_ACOverpayPre;  
T\_ACOverpayPost=T\_ACOverpayPost+C\_ACOverpayPost;  
T\_ACSChargePre=T\_ACSChargePre+C\_ACSChargePre;  
T\_ACSChargePost=T\_ACSChargePost+C\_ACSChargePost;  
T\_ACQuantumPre=T\_ACQuantumPre+C\_ACQuantumPre;  
T\_ACQuantumPost=T\_ACQuantumPost+C\_ACQuantumPost;  
T\_ACStatPre=T\_ACStatPre+C\_ACStatPre;  
T\_ACStatPost=T\_ACStatPost+C\_ACStatPost;  
T\_ACQuantumStatPre=T\_ACQuantumStatPre+C\_ACQuantumStatPre;  
T\_ACQuantumStatPost=T\_ACQuantumStatPost+C\_ACQuantumStatPost;  
T\_RCCommission=T\_RCCommission+C\_RCCommission;  
T\_RCSCharge=T\_RCSCharge+C\_RCSCharge;  
T\_RCQuantum=T\_RCQuantum+C\_RCQuantum;  
T\_RCStat=T\_RCStat+C\_RCStat;  
T\_RCQuantumStat=T\_RCQuantumStat+C\_RCQuantumStat;  
"";>>

|  |  |
| --- | --- |
| **Energy Broker Details #1:** | |
| Brokerage | </<BNameCell>> |
| Brokerage Address | </<BAddCell>> |
| Brokerage Company Registration Number | </<BNumCell>> |

|  |  |
| --- | --- |
| **Contract Details #1:** | |
| Supply Address | </<C\_SiteAddress>> |
| Agreement Number | </<C\_SiteAccount>> |
| MPAN/MPRN | </<C\_SiteMPAN>> |
| Meter Number | </<C\_SiteMeter>> |
| Type of Contract | </<# Filled in by the Audit Team>> |
| Utility Type | </<C\_UtilityType>> |
| Date of Contract (Signed) | </<C\_SignDate :sdate>> |
| Contract Length (Months) | </<int.C\_ContractLength>> |
| Supply Start Date | </<C\_StartDate :sdate>> |
| Supply End Date | </<C\_EndDate :sdate>> |
| Status of Contract | </<if (C\_EndDate > date.ClaimDate) {"Active";} else {"Inactive";}>> |
| Contractual Energy Rate (per kWh) | </<"[TBD]">> |
| Estimated Annual Consumption (kWh) | </<"[TBD]">> |
| Actual Annual Consumption (kWh) | </<"[TBD]">> |
| Fair Energy Rate (per kWh) | </<"Peak: " + ("%.4fp" % C\_peakRate)>> </<"Base: " + ("%.4fp" % C\_baseRate)>> |
| Was the fact of Commission disclosed at the point of Contract? | No |
| Was the amount of Commission disclosed at the point of the contract? | No |

|  |  |
| --- | --- |
| **Refund of Agreement Cost Quantum Details #1:** | |
| **Up to Date of Report** | |
| Overpayment Refund up toDate of Report | </<C\_ACOverpayPre :curr>> |
| Standing Charge Refund up toDate of Report | </<C\_ACSChargePre: curr>> |
| Quantum up toDate of Report | </<C\_ACQuantumPre :curr>> |
| Compensatory Interest up toDate of Report | </<C\_ACStatPre :curr>> |
| Quantum Plus Compensatory Interest up to Date of Report | </<C\_ACQuantumStatPre :curr>> |
| **After Date of Report** | |
| Overpayment Refund after the Date of Report to the Supply End Date | </<C\_ACOverpayPost :curr>> |
| Standing Charge Refund after the Date of Report to the Supply End Date | </<C\_ACSChargePost: curr>> |
| Quantum after the Date of Report to the Supply End Date | </<C\_ACQuantumPost :curr>> |
| Compensatory Interest after the Date of Report to the Supply End Date | </<C\_ACStatPost :curr>> |
| Quantum Plus Compensatory Interest after the Date of Report to the Supply End Date | </<C\_ACQuantumStatPost :curr>> |

|  |  |
| --- | --- |
| **Refund of Commission Quantum Details #1:** | |
| Commission Refund | </<C\_RCCommission :curr>> |
| Standing Charge Refund | </<C\_RCSCharge : curr>> |
| Quantum | </<C\_RCQuantum :curr>> |
| Compensatory Interest | </<C\_RCStat :curr>> |
| Quantum Plus Compensatory Interest | </<C\_RCQuantumStat :curr>> |

|  |  |
| --- | --- |
| **Claim Details:** | |
| Date of Report | </<date.ClaimDate :sdate>> |

## Methodology

We have been instructed to undertake a valuation of potential Business Energy Claim according to two separate methods. To quantify the effect of the non-disclosure of commission on each method, we have reconstructed the energy contract(s) from the supply start date to the supply end date in accordance with instructions provided by instructing party. The methodology for each method is outlined below.

### 3.1 Refund of Agreement Cost

A calculation is undertaken according to the principle that both parties ought to be put back into to the position they would be in were the agreement never entered into. The value of the overpayment is attributed to the difference between the contractual rate charged over the course of the agreement, and the cost of the agreement at the fair rate.

The breakdown of the Refund of Agreement Cost quantum is as follows:

1. Using the contract signed date, the initial fair rate is determined from the historical data of the future energy market. This initial fair rate is calculated over the portion of the agreement for which the contractual rate remained unchanged from its initial value. (Please see section 5 for an explanation of the ‘fair rate’ for the purposes of our calculation.)
2. It is assumed that the broker earned commission based on the prevailing cost of energy when the agreement was signed. If the energy cost over the duration of the agreement varied from what it initially cost, then that would result in a change in the contractual rate, and this would reflect a fair change in cost to the supplier purchasing the energy.
3. A new fair rate is calculated using the difference between the initial fair rate and initial contractual rate, assuming that the difference between the two rates remains constant across the agreement. This difference is deducted from the contractual rate every time it changes to determine the corresponding fair rate for the period for which the new contractual rate is charged for the energy supplied.
4. For each billing period, anything that was charged by the supplier above the fair rate is assumed to be an overpayment and is refunded to the claimant. The overpayment is defined as the energy bill total excluding VAT, standing charge, climate change levy (CCL) and fair rate.
5. If there are any separately billed pass-through costs, then these are included in the overpayment, as it is assumed that these costs have already been accommodated in the percentage uplift allowance applied for the supplier’s overheads. As part of our calculation process, we determine the portion of separately billed pass-through costs and include it in our overpayments if these costs exceed £10.00 on a given energy bill.
6. The energy supplier's daily standing charge to cover the cost of supplying energy to the property is refunded too.
7. The sum of the overpayments and standing charges on the agreement is calculated to be refunded to the claimant in order to put both parties back in the position that they would have been in if the agreement had never been signed.
8. Compensatory (Statutory) Interest at a rate of 8.00% is applied to the total of each overpayment and standing charges paid from the date of the bill invoice to the date of assessment (or the supply end date if applicable).

### 3.2 Refund of Commission

A calculation is undertaken according to the principle that a percentage of the total value of the agreement was undisclosed commission paid to the broker by the supplier. The undisclosed commission is ought to be refunded to the claimant company.

The breakdown of the Refund of Commission quantum calculation is as follows:

1. The initial contractual rate per kWh, the energy consumption, standing charge, and the length of the agreement are used to calculate the total value of the agreement. If an agreement has elapsed for more than 12 months, the energy consumption is calculated based on the average consumption across all bills; otherwise, the estimated annual consumption (EAC) from the agreement has been utilised.
2. It has been assumed that undisclosed commission at a rate of </<num.ComRate \* 100 :%.2f%%>> of the total value of agreement was paid to the broker. The commission rate specified above has been defined by the instructing party.
3. The undisclosed commission on the agreement is calculated to be refunded the claimant company.
4. Compensatory (Statutory) Interest at a rate of 8.00% is applied to the total amount of undisclosed commission calculated from the energy supply end date to the date of assessment. If an energy contract(s) remains active no statutory interest is applied.

## Quantum

The following is an illustration of the total quantum for each method due to the claimant company </<CNameCell>> for the energy contract(s) against </<SNameCell>> in accordance with the calculation principles outlined in Section 3. The total quantum for each method is the sum of the quantum of multiple contracts with the same energy supplier. Section 2 contains the separate quantum for each contract.

### 4.1 Refund of Agreement Cost

|  |  |  |
| --- | --- | --- |
| **Refund of Agreement Cost Claim Summary** | | |
|  | Up to the Date of Report | </<T\_ACOverpayPre :curr>> |
|  | After the Date of Report to the Supply End Date | </<T\_ACOverpayPost :curr>> |
| 1. **Overpayment Refund** | | </<T\_ACOverpayPre + T\_ACOverpayPost :curr>> |
|  | Up to the Date of Report | </<T\_ACSChargePre :curr>> |
|  | After the Date of Report to the Supply End Date | </<T\_ACSChargePost :curr>> |
| 1. **Standing Charge Refund** | | **</<T\_ACSChargePre + T\_ACSChargePost :curr>>** |
|  | Up to the Date of Report | </<T\_ACQuantumPre :curr>> |
|  | After the Date of Report to the Supply End Date | </<T\_ACQuantumPost :curr>> |
| **All Refunds** | | **</<T\_ACQuantumPre + T\_ACQuantumPost :curr>>** |
|  | Until the Date of Report | </<T\_ACStatPre :curr>> |
|  | After the Date of Report to the Supply End Date | </<T\_ACStatPost :curr>> |
| 1. **PLUS Compensatory Interest (8.00%)** | | **</<T\_ACStatPre + T\_ACStatPost :curr>>** |
|  | After the Date of Report | </<T\_ACQuantumStatPre :curr>> |
|  | From the Date of Report to the Supply End Date | </<T\_ACQuantumStatPost :curr>> |
| **Total Claim for Recovery** | | **</<T\_ACQuantumStatPre + T\_ACQuantumStatPost :curr>>** |

The total claim for recovery consists of three components:

1. **Overpayment Refund**

The sum of all the overpayments made by the company is totalled for the entire agreement. These overpayments are the difference in total paid amount at the contractual rate and the fair rate and if applicable separately billed passed-through costs.

1. **Standing Charge Refund**

The sum of the fixed daily amount charged to energy bills for the duration of energy supply, regardless of quantity of energy used.

1. **Compensatory Interest**

Compensatory (Statutory) interest is added to the claim value at a rate of 8.00% per annum on each overpayment and standing charges paid from the date of the bill invoice to the supply end date.

### 4.2 Refund of Commission

|  |  |  |  |
| --- | --- | --- | --- |
| **Refund of Commission Claim Summary** | | | |
|  | Energy Charge refund | | </<T\_RCCommission :curr>> |
|  | Standing Charge | | </<T\_RCSCharge :curr>> |
| 1. **Commission Refund** | |  | **</<T\_RCQuantum :curr>>** |
| 1. **PLUS Compensatory Interest (8.00%)** | | | **</<T\_RCStat :curr>>** |
| **Total Claim for Recovery** | |  | **</<T\_RCQuantumStat :curr>>** |

The total claim for recovery consists of two components:

1. **Commission Refund**   
     
   The estimated undisclosed commission is calculated at a rate of </<(num.ComRate \* 100) :%.2f%%>> of the total estimated value of the agreement which is based on the total energy charge and the standing charge.
2. **Compensatory Interest**

Compensatory (Statutory) Interest at a rate of 8.00% is applied to the total amount of undisclosed commission that has been estimated to be paid to the broker from the supply end date to the date of report. This is only applicable if the supply end date is prior to the date of report.

## Fair Rate Methodology

The fair energy rate is simply the rate at which the power generator sells electricity or natural gas to the utility provider, plus operational costs.

The wholesale rate serves as the foundation for determining an appropriate fair rate. The term "wholesale energy rate" refers to the price at which a power generator sells electricity or natural gas to utility companies, who subsequently resell it to consumers. Supply and demand are prominent components of the wholesale price.

The wholesale rate is calculated using energy futures, which are exchange-traded derivative contracts with energy products as the underlying asset. Using the energy futures, we derive the fair value rate at which the market participants purchased and hedged energy commodities for the quantity and duration specified in the contract between the consumer and the energy supplier.

To account for the supplier's operating costs (which include transportation, network fees, levies, and other relevant charges), a 40.00% uplift is added to the determined wholesale rate to arrive at the fair rate.

## Disclaimer and Confidentiality

* 1. The methodology of our report, its terminology, bases for claim and the actions subsequent to its completion are entirely at the discretion of the instructing party.
  2. The instructing party requested that we carry out a report based solely on their instruction, and we do not bear responsibility for the use of this report after it has been produced.
  3. It has been assumed that all relevant documentation received by Central Chambers Law has been received by us regarding this matter.
     1. Central Chambers Law instructed us to make certain assumptions where the documentation we received from Central Chambers Law did not provide complete information to perform the calculations. If applicable, we have specified the assumption we have made for this case in Section 7. If the missing information is later revealed, changes to the report may be needed, which may change the conclusions reached.
  4. We guarantee the accuracy of our report only to the extent that the information provided at all points, by the instructing party, claimant and energy supplier is accurate and correct. We are not responsible for any errors or omissions in the received data which came into existence prior to our engagement with it.
  5. It is not the purpose of this report to amend, alter or correct any of the errors, inaccuracies, or omissions inherent in the data which has been provided by third parties.
  6. Where an individual has a complaint about the data used in this report, that complaint should be directed against the persons responsible for the safekeeping of that information.
  7. The information contained within this report is strictly confidential, for use only by the intended recipients. If you are not the intended recipient, be advised that you have received this report in error and that any use, dissemination, forwarding, printing, or copying of this is strictly prohibited, please return the document to the sender, and erase all copies on your system.

## Notes and List of Assumptions

* 1. ……………………………………………………………………………………………………………….
  2. ……………………………………………………………………………………………………………….
  3. ……………………………………………………………………………………………………………….