

**EIP**

**External Interfaces Specification**

**Version 1.24**

**July 05, 2022**

Document Revisions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Version | Description | Prod Release | Author(s) |
| 11/16/2021 | 1.21 | * NPRR1016: Added gray box information to section 6 to be used upon system implementation.   + Updates to Outage Scheduling Interface.   + DGR/DESR added to ErcotOutageTypes.xsd * NPRR863: Added gray box information to be used upon system implementation.   + Updated Submissions: ASTrade, Self-arranged AS, ASOffer and COP structures to handle the new PFR, UFR, and FFR RRS sub-types.   + Updated MarketInformation: AwardedAS to use asType of RRSPF, RRSFF, RRSUF and the OnLineReserves element would contain RRS.   + Updated MarketInformation: TotalASOffers and AggregatedASOfferCurves to use asType of RRSPF, RRSFF and RRSUF   + Removed the NCLR flag from the AS Offer submission structure. (Figure 11, sections 3.2.2, 3.3.4, 3.3.9 and 4.3.2)   + Updated ErcotCommonTypes.xsd and ErcotTransactionTypes.xsd | NPRR1016:  02/01/2022 – 02/03/2022  NPRR863:  Release targeted for:  10/13/2022 | SJ |
| 02/04/2022 | 1.22 | * NPRR1093: Added gray box information to Figure 11 and sections 3.3.2, 3.3.5 and 4.3.42.   + Updated Figure 11 which list ASTypes used in submissions.   + Updated Self-Arranged Ancillary Service to include nspnm\_value for Non-Spin provided from SCED dispatchable Resources   + Updated AS Trade to include a new ASType NSPNM to be used for Non-Spin provided from SCED dispatchable Resources   + Updated ErcotCommonTypes ASType to include NSPNM for ASTrade and TmPoint to include nspnm\_value for SelfArrangedAS. | NPRR1093:  05/24/2022 – 05/26/2022 | SJ |
| 04/28/2022 | 1.23 | * Added Market Information report content retrieval for the following reports:   + 4.3.47: NP4-197 - Price Correction to 15-min SOTG/SODG Price   + 4.3.48: NP4-197 - Price Correction to SCED SOTG/SODG LMPs * NPRR1093: Added gray box information to section 4.3.42.   + Updated ErcotInformationTypes.xsd to include additional column for SCEDPriceAdderORDC report NP6-323-CD (13221). | 05/02/2022 – 05/06/2022  05/24/2022 – 05/26/2022 | SJ |
| 07/05/2022 | 1.24 | * NPRR1120: Added gray box information to section 3.3.6   + Updated ErcotCommonTypes.xsd to include an additional Availability Type FFSS for Availability plan resource type. | NPRR1120:  Release targeted for:  10/13/2022 | SJ |

**Table of Contents**

[1. Introduction 1](#_Toc107912867)

[1.1. Purpose 1](#_Toc107912868)

[1.2. Scope 1](#_Toc107912869)

[1.3. Definitions, Acronyms, and Abbreviations 2](#_Toc107912870)

[1.4. References 4](#_Toc107912871)

[1.5. Overview 4](#_Toc107912872)

[1.6. Program-level Standards 4](#_Toc107912873)

[2. Services Organization 7](#_Toc107912874)

[2.1. Common Message Structure 7](#_Toc107912875)

[2.1.1. Message Header Structure 8](#_Toc107912876)

[2.1.2. Request Message Structures 11](#_Toc107912877)

[2.1.3. Payload Structures 12](#_Toc107912878)

[2.1.4. Response Message Structures 14](#_Toc107912879)

[2.2. Common Security Implementation 15](#_Toc107912880)

[2.2.1. Secure the Transport layer 15](#_Toc107912881)

[2.2.2. Secure SOAP messages 16](#_Toc107912882)

[2.3. Modeling and Conventions 17](#_Toc107912883)

[2.3.1. Use of the IEC CIM 17](#_Toc107912884)

[2.3.2. Representation of Time 22](#_Toc107912885)

[2.3.3. Market Products 23](#_Toc107912886)

[2.3.4. Management and Use of Transaction IDs (mRIDs) 24](#_Toc107912887)

[2.3.5. Other Conventions 29](#_Toc107912888)

[2.3.6. Precision 30](#_Toc107912889)

[2.3.7. XML Special Characters 31](#_Toc107912890)

[2.4. Delivery Approach 31](#_Toc107912891)

[2.5. Technical Interoperability 31](#_Toc107912892)

[2.6. Service Level Agreements 32](#_Toc107912893)

[2.7. Auditing, Monitoring and Management 32](#_Toc107912894)

[2.8. Versioning 32](#_Toc107912895)

[2.9. Governance 33](#_Toc107912896)

[2.10. Web Service Configuration Standards 33](#_Toc107912897)

[2.11. Web Service Design Assumptions and Limitations 34](#_Toc107912898)

[3. Market Transaction Service 36](#_Toc107912899)

[3.1. Interfaces Provided 36](#_Toc107912900)

[3.2. Interfaces Required 39](#_Toc107912901)

[3.3. Message Specifications 40](#_Toc107912902)

[3.3.1. Three-Part Supply Offer (TPO) 43](#_Toc107912903)

[3.3.2. Self-Arranged Ancillary Service Quantities (SAA) 50](#_Toc107912904)

[3.3.3. Incremental and Decremental Energy Offer Curves (IDO) 59](#_Toc107912905)

[3.3.4. Ancillary Service Offer (ASO) 63](#_Toc107912906)

[3.3.5. Ancillary Service Trade (AST) 74](#_Toc107912907)

[3.3.6. Availability Plan (AVP) 78](#_Toc107912908)

[3.3.7. Capacity Trade (CT) 82](#_Toc107912909)

[3.3.8. PTP Obligation w/ Links to Option (CRR) 85](#_Toc107912910)

[3.3.9. Current Operating Plan (COP) 89](#_Toc107912911)

[3.3.10. DAM Energy Bid (EB) 96](#_Toc107912912)

[3.3.11. DAM Energy-Only Offer (EOO) 99](#_Toc107912913)

[3.3.12. Energy Trade (ET) 102](#_Toc107912914)

[3.3.13. Output Schedule (OS) 106](#_Toc107912915)

[3.3.14. PTP Obligation Bid (PTP) 109](#_Toc107912916)

[3.3.15. Self-Schedule (SS) 112](#_Toc107912917)

[3.3.16. Real-Time Market Energy Bid (REB) 114](#_Toc107912918)

[3.3.17. Exceptional Fuel Cost (EFC) 116](#_Toc107912919)

[3.4. Error Handling 119](#_Toc107912920)

[3.5. Usage and XML Examples 121](#_Toc107912921)

[3.5.1. Offer and Bid Set Submission 121](#_Toc107912922)

[3.5.2. Querying Bids, Offers, Trades and Schedules 124](#_Toc107912923)

[3.5.3. Updating Bids, Offers, Trades and Schedules 127](#_Toc107912924)

[3.5.4. Canceling Bids, Offers, Trades and Schedules 128](#_Toc107912925)

[4. Market Information 130](#_Toc107912926)

[4.1. Interfaces Provided 130](#_Toc107912927)

[4.2. Interfaces Required 131](#_Toc107912928)

[4.3. Message Specifications 131](#_Toc107912929)

[4.3.1. AwardSet 132](#_Toc107912930)

[4.3.2. AwardedAS 135](#_Toc107912931)

[4.3.3. AwardedCRR 144](#_Toc107912932)

[4.3.4. AwardedEnergyBid 147](#_Toc107912933)

[4.3.5. AwardedEnergyOffer 149](#_Toc107912934)

[4.3.6. AwardedEnergyOnlyOffer 151](#_Toc107912935)

[4.3.7. AwardedPTPObligation 153](#_Toc107912936)

[4.3.8. Forecasted Load 156](#_Toc107912937)

[4.3.9. System Load \*\*\* Same as 4.3.33 - Total ERCOT Load \*\*\* 162](#_Toc107912938)

[4.3.10. Market Totals \*\*\* Same as 4.3.24 - Total DAM Energy \*\*\* 164](#_Toc107912939)

[4.3.11. Market LMPs, SPPs, Price Corrected LMPs and Price Corrected SPPs 167](#_Toc107912940)

[4.3.12. Market MCPCs 171](#_Toc107912941)

[4.3.13. Binding Constraints 174](#_Toc107912942)

[4.3.14. SCED Violated Constraints 177](#_Toc107912943)

[4.3.15. Price Corrected SCED Violated Constraints 180](#_Toc107912944)

[4.3.16. Ancillary Service Obligation 183](#_Toc107912945)

[4.3.17. Dynamic Ratings 186](#_Toc107912946)

[4.3.18. Load Ratio Share 189](#_Toc107912947)

[4.3.19. Aggregated Ancillary Service Offer Curves 192](#_Toc107912948)

[4.3.20. Ancillary Service Plan 196](#_Toc107912949)

[4.3.21. Startup and Shutdown Instructions 198](#_Toc107912950)

[4.3.22. Total Ancillary Service Offers 200](#_Toc107912951)

[4.3.23. Total DAM Energy \*\*\* Same as 4.3.10 - Market Totals \*\*\* 202](#_Toc107912952)

[4.3.24. Weather Data 205](#_Toc107912953)

[4.3.25. Responsive Reserve Capacity – Please See Section 4.3.39, System Parameters 207](#_Toc107912954)

[4.3.26. Non-Spinning Reserve – Please See Section 4.3.39, System Parameters 207](#_Toc107912955)

[4.3.27. Undeployed Reg-Up and Reg-Down – Please See Section 4.3.39, System Parameters 207](#_Toc107912956)

[4.3.28. Available Capacity – Please See Section 4.3.39, System Parameters 207](#_Toc107912957)

[4.3.29. Wind-Powered Generation Resource Production Potential 207](#_Toc107912958)

[4.3.30. Short-Term Wind Power Forecast 210](#_Toc107912959)

[4.3.31. Total ERCOT Load \*\*\* Same as 4.3.9 - System Load \*\*\* 211](#_Toc107912960)

[4.3.32. Total ERCOT Generation Operating Reserve – Please See Section 4.3.40, System Parameters 214](#_Toc107912961)

[4.3.33. UnconfirmedTrades 214](#_Toc107912962)

[4.3.34. ConfirmedTrades 215](#_Toc107912963)

[4.3.35. DAM Ancillary Service Offer Insufficiency Report 217](#_Toc107912964)

[4.3.36. End of Adjustment Period Results 219](#_Toc107912965)

[4.3.37. Two Hour Warning Results 221](#_Toc107912966)

[4.3.38. DAM Phase II Validation Results 223](#_Toc107912967)

[4.3.39. System Parameters 226](#_Toc107912968)

[4.3.40. RTD Indicative Base Points 229](#_Toc107912969)

[4.3.41. RTD Indicative LMPs 231](#_Toc107912970)

[4.3.42. SCED ORDC Price Adders 233](#_Toc107912971)

[4.3.43. RT Price Adder ORDC 237](#_Toc107912972)

[4.3.44. RTD Indicative ORDC Price Adders 239](#_Toc107912973)

[4.3.45. Photovoltaic Generation Resource Production Potential 242](#_Toc107912974)

[4.3.46. Short-Term Photovoltaic Power Forecast 244](#_Toc107912975)

[4.3.47. Price Corrected SOTG/SODG 15-min Prices 245](#_Toc107912976)

[4.3.48. Price Corrected SOTG/SODG LMPs 248](#_Toc107912977)

[5. Notifications 252](#_Toc107912978)

[5.1. Interfaces Provided 252](#_Toc107912979)

[5.2. Interfaces Required 253](#_Toc107912980)

[5.3. Message Specifications 254](#_Toc107912981)

[5.3.1. Notices and Alerts 254](#_Toc107912982)

[5.3.2. Offer and Bid Set Acceptance 276](#_Toc107912983)

[5.3.3. Offer and Bid Set Errors 277](#_Toc107912984)

[5.3.4. Confirmed and Unconfirmed Trades 277](#_Toc107912985)

[5.3.5. Energy Offer Awards 282](#_Toc107912986)

[5.3.6. DAM Energy-Only Offer Awards 283](#_Toc107912987)

[5.3.7. DAM Energy Bid Award 285](#_Toc107912988)

[5.3.8. Ancillary Service Awards 286](#_Toc107912989)

[5.3.9. CRR Awards 291](#_Toc107912990)

[5.3.10. PTP Obligation Awards 293](#_Toc107912991)

[5.3.11. Ancillary Service Obligations 294](#_Toc107912992)

[5.3.12. Outage Notifications 297](#_Toc107912993)

[5.3.13. Startup/Shutdown Instructions 298](#_Toc107912994)

[5.3.14. Wind Generation Forecast 300](#_Toc107912995)

[5.3.15. DAM Ancillary Service Offer Insufficiency Report 306](#_Toc107912996)

[5.3.16. End of Adjustment Period Results 308](#_Toc107912997)

[5.3.17. Two Hour Warning Results 310](#_Toc107912998)

[5.3.18. DAM Phase II Validation Results 312](#_Toc107912999)

[5.3.19. Solar Generation Forecast 314](#_Toc107913000)

[6. Outage Scheduling Interfaces 319](#_Toc107913001)

[6.1. Interfaces Provided 319](#_Toc107913002)

[6.2. Message Specifications 319](#_Toc107913003)

[6.2.1. Outage Creation 319](#_Toc107913004)

[6.2.2. Outage Query 345](#_Toc107913005)

[6.2.3. Outage Update 351](#_Toc107913006)

[6.2.4. Outage Cancellation 360](#_Toc107913007)

[7. Utility Interfaces 364](#_Toc107913008)

[7.1. Interfaces Provided 364](#_Toc107913009)

[7.2. Message Specifications 364](#_Toc107913010)

[7.2.1. Get mRID 364](#_Toc107913011)

[7.2.2. System Status 365](#_Toc107913012)

[7.2.3. Request Test Notification 366](#_Toc107913013)

[7.2.4. Change Active Notification URL 367](#_Toc107913014)

[7.2.5. Get SASM ID List 368](#_Toc107913015)

[8. Reports 370](#_Toc107913016)

[8.1. Message Specifications 370](#_Toc107913017)

[8.1.1. Get Reports 370](#_Toc107913018)

[9. Resource Parameter Transaction Service 374](#_Toc107913019)

[9.1. Interfaces Provided 374](#_Toc107913020)

[9.2. Interfaces Required 376](#_Toc107913021)

[9.2.1. Querying for Resource Parameters Set 376](#_Toc107913022)

[9.2.2. Updating Resource Parameters Set 377](#_Toc107913023)

[9.2.3. Canceling Resource Parameters Set 378](#_Toc107913024)

[9.3. Message Specifications 379](#_Toc107913025)

[9.3.1. Generator Resource Parameters 381](#_Toc107913026)

[9.3.2. Controllable Load Resource 385](#_Toc107913027)

[9.3.3. Non Controllable Load Resource 388](#_Toc107913028)

[9.3.4. Resource Parameters 390](#_Toc107913029)

[10. Verbal Dispatch Instructions 393](#_Toc107913030)

[10.1. Interfaces Provided 393](#_Toc107913031)

[10.2. Interfaces Required 394](#_Toc107913032)

[10.3. Message Specifications 396](#_Toc107913033)

[10.3.1. Verbal Dispatch Instructions Structure 397](#_Toc107913034)

[11. Get Notifications 402](#_Toc107913035)

[11.1. Message Specifications 403](#_Toc107913036)

[11.1.1. Get Notifications 403](#_Toc107913037)

[Appendix A: WS-Notifications 411](#_Toc107913038)

[Appendix B: WSDL for Market Requests 414](#_Toc107913039)

[Appendix C: XML Schemas for Message and Payload Definitions 416](#_Toc107913040)

[Appendix D: Annotated SOAP Message 418](#_Toc107913041)

[Appendix E: SOAP Examples 422](#_Toc107913042)

[Appendix F: XSL Examples 493](#_Toc107913043)

[Appendix G: Payload Compression Example 496](#_Toc107913044)

[Appendix H: DST XML Examples 497](#_Toc107913045)

# List of Figures

[Figure 1 - Web Service Interfaces 9](#_Toc107913046)

[Figure 2 - Message Header Structure 10](#_Toc107913047)

[Figure 3 - Request Message Structure 11](#_Toc107913048)

[Figure 4 - Message Request Parameters 12](#_Toc107913049)

[Figure 5 - Message Payload Container Structure 13](#_Toc107913050)

[Figure 6 - Response Message Structure 14](#_Toc107913051)

[Figure 7 - CIM Curves and Schedules 18](#_Toc107913052)

[Figure 8 - Time Schedule Structure 20](#_Toc107913053)

[Figure 9 - Required Keys for Market Products 24](#_Toc107913054)

[Figure 10 - mRID Structures 28](#_Toc107913055)

[Figure 11 - ASType Usage 29](#_Toc107913056)

[Figure 12 - Example BidSet Sequence Diagram 37](#_Toc107913057)

[Figure 13 - State Diagram for Bids, Trades, Offers, and Schedules 39](#_Toc107913058)

[Figure 14 - Message: create BidSet 40](#_Toc107913059)

[Figure 15 - Message: reply BidSet 40](#_Toc107913060)

[Figure 16 - BidSet Header Structure 41](#_Toc107913061)

[Figure 17 – Bid/Offer/Trade/Schedule Common Elements 43](#_Toc107913062)

[Figure 18 – ThreePartOffer Structure 44](#_Toc107913063)

[Figure 19 - PriceCurve Structure 45](#_Toc107913064)

[Figure 20 - EocFipFop, SuMeFipFop, StartupCost, and MinimumEnergy Structures 47](#_Toc107913065)

[Figure 21 - ThreePartOffer Requirements 49](#_Toc107913066)

[Figure 22 - SelfArrangedAS Structure 51](#_Toc107913067)

[Figure 23 - SelfASCapacitySchedule Structure 54](#_Toc107913068)

[Figure 24 - SelfArrangedAS Requirements 56](#_Toc107913069)

[Figure 25 - IncDecOffer Structure 60](#_Toc107913070)

[Figure 26 - IncDecOffer Requirements 62](#_Toc107913071)

[Figure 27 - ASOffer Structure 64](#_Toc107913072)

[Figure 28 - Price Curves using ASPriceCurve 67](#_Toc107913073)

[Figure 29 - ASOffer Requirements 70](#_Toc107913074)

[Figure 30 - ASTrade Structure 75](#_Toc107913075)

[Figure 31 - ASTrade Requirements 76](#_Toc107913076)

[Figure 32 - AVP Structure 79](#_Toc107913077)

[Figure 33 - AVP Requirements 80](#_Toc107913078)

[Figure 34 - CapacityTrade Structure 83](#_Toc107913079)

[Figure 35 - CapacityTrade Requirements 84](#_Toc107913080)

[Figure 36 – PTP Obligation w/ Links to Option - CRR Structure 86](#_Toc107913081)

[Figure 37 - MinimumReservationPrice Structure 87](#_Toc107913082)

[Figure 38 - CRR Requirements 88](#_Toc107913083)

[Figure 39 - COP Structure 90](#_Toc107913084)

[Figure 40 - Limits Structure 91](#_Toc107913085)

[Figure 41 - ASCapacity Structure 91](#_Toc107913086)

[Figure 42 - COP Requirements 92](#_Toc107913087)

[Figure 43 - EnergyBid Structure 96](#_Toc107913088)

[Figure 44 - EnergyBid Requirements 98](#_Toc107913089)

[Figure 45 – EnergyOnlyOffer Structure 100](#_Toc107913090)

[Figure 46 - EnergyOnlyOffer Requirements 101](#_Toc107913091)

[Figure 47 - EnergyTrade Structure 103](#_Toc107913092)

[Figure 48 - EnergySchedule Structure 104](#_Toc107913093)

[Figure 49 - EnergyTrade Requirements 105](#_Toc107913094)

[Figure 50 - OutputSchedule Structure 107](#_Toc107913095)

[Figure 51 - OutputSchedule Requirements 108](#_Toc107913096)

[Figure 52 - PTPObligation Requirements 109](#_Toc107913097)

[Figure 53 - MaximumPrice Sub-Structure 110](#_Toc107913098)

[Figure 54 - PTPObligation Requirements 111](#_Toc107913099)

[Figure 55 - SelfSchedule Structure 112](#_Toc107913100)

[Figure 56 – SelfSchedule Requirements 113](#_Toc107913101)

[Figure 57 - RTMEnergyBid Structure 114](#_Toc107913102)

[Figure 58 - RTMEnergyBid Requirements 115](#_Toc107913103)

[Figure 59 – EFC Bid Structure 117](#_Toc107913104)

[Figure 60 – EFC Submission Requirements 118](#_Toc107913105)

[Figure 61 - Error Handling 119](#_Toc107913106)

[Figure 62 - create BidSet Request Parameters 121](#_Toc107913107)

[Figure 63 - reply BidSet Parameters 121](#_Toc107913108)

[Figure 64 – Short mRID Key String 125](#_Toc107913109)

[Figure 65 - get BidSet Parameters 125](#_Toc107913110)

[Figure 66 - reply BidSet Parameters 126](#_Toc107913111)

[Figure 67 - change BidSet Parameters 127](#_Toc107913112)

[Figure 68 - reply BidSet Parameters 127](#_Toc107913113)

[Figure 69 - cancel BidSet Parameters 128](#_Toc107913114)

[Figure 70 - reply BidSet Parameters 129](#_Toc107913115)

[Figure 71 - Market Information Request Sequence Diagram 130](#_Toc107913116)

[Figure 72 - Parameters for 'get' Requests 131](#_Toc107913117)

[Figure 73 - Parameters for 'reply' Messages 131](#_Toc107913118)

[Figure 74 - Get AwardSet Parameters 132](#_Toc107913119)

[Figure 75 - Get AwardSet Response 132](#_Toc107913120)

[Figure 76 - AwardSet Container Structure 133](#_Toc107913121)

[Figure 77 - Generalized Award Structure 133](#_Toc107913122)

[Figure 78 - Get AwardedAS Parameters 135](#_Toc107913123)

[Figure 79 - Get AwardedAS Response 136](#_Toc107913124)

[Figure 80 - AwardedAS Container Structure 138](#_Toc107913125)

[Figure 81 - Get AwardedCRR Parameters 144](#_Toc107913126)

[Figure 82 - Get AwardedCRR Response 145](#_Toc107913127)

[Figure 83 - AwardedCRR Container Structure 145](#_Toc107913128)

[Figure 84 - Get AwardedEnergyBid Parameters 147](#_Toc107913129)

[Figure 85 - Get AwardedEnergyBid Response 147](#_Toc107913130)

[Figure 86 - AwardedEnergyBid Container Structure 148](#_Toc107913131)

[Figure 87 - Get AwardedEnergyOffer Parameters 149](#_Toc107913132)

[Figure 88 - Get AwardedEnergyOffer Response 150](#_Toc107913133)

[Figure 89 - AwardedEnergyOffer Container Structure 150](#_Toc107913134)

[Figure 90 - Get AwardedEnergyOnlyOffer Parameters 151](#_Toc107913135)

[Figure 91 - Get AwardedEnergyOnlyOffer Response 152](#_Toc107913136)

[Figure 92 - AwardedEnergyOnlyOffer Container Structure 152](#_Toc107913137)

[Figure 93 - Get AwardedPTPObligation Parameters 154](#_Toc107913138)

[Figure 94 - Get AwardedPTPObligation Response 154](#_Toc107913139)

[Figure 95 - AwardedPTPObligation Container Structure 156](#_Toc107913140)

[Figure 96 – Load Forecast Request Parameters 157](#_Toc107913141)

[Figure 97 – Load Forecast Response Parameters 157](#_Toc107913142)

[Figure 98 - Load Forecasts Structure 158](#_Toc107913143)

[Figure 99 - SystemLoad Request Parameters 162](#_Toc107913144)

[Figure 100 - SystemLoad Response Parameters 163](#_Toc107913145)

[Figure 101 - SystemLoad Structure 163](#_Toc107913146)

[Figure 102 - Market Totals Request Parameters 164](#_Toc107913147)

[Figure 103 - MarketTotals Response Parameters 165](#_Toc107913148)

[Figure 104 - MarketTotals Structure 166](#_Toc107913149)

[Figure 105 - LMP and SPP Request Parameters 168](#_Toc107913150)

[Figure 106 - LMP and SPP Response Parameters 168](#_Toc107913151)

[Figure 107 - LMP and LMPsPC Container Structure 169](#_Toc107913152)

[Figure 108 - SPP and SPPsPC Container Structure 170](#_Toc107913153)

[Figure 109 - MCPC Container Structure 173](#_Toc107913154)

[Figure 110 - BindingConstraint Container Structure 177](#_Toc107913155)

[Figure 111 - SCEDViolatedConstraints Container Structure 180](#_Toc107913156)

[Figure 112 - ShadowPriceCorrection Container Structure 182](#_Toc107913157)

[Figure 113 - ASObligations Container Structure 185](#_Toc107913158)

[Figure 114 - DynamicRatings Container Structure 188](#_Toc107913159)

[Figure 115 - LoadRatioShares Container Structure 191](#_Toc107913160)

[Figure 116 - AggregatedASOfferCurve Schedule 193](#_Toc107913161)

[Figure 117 - ASServicePlan Container Structure 197](#_Toc107913162)

[Figure 118 - ASSchedule Structure 197](#_Toc107913163)

[Figure 119 - StartupShutdownInstructions Container Structure 199](#_Toc107913164)

[Figure 120 - ASOffers Structure 201](#_Toc107913165)

[Figure 121 - TotalEnergy Structure 204](#_Toc107913166)

[Figure 122 – Weather Information Container Structure 207](#_Toc107913167)

[Figure 123 – WGRPP Structure 209](#_Toc107913168)

[Figure 124 - STWPF Structure 211](#_Toc107913169)

[Figure 125 – Total Load Container Structure 213](#_Toc107913170)

[Figure 126 – UnconfirmedTrades Container Structure 215](#_Toc107913171)

[Figure 127 – ConfirmedTrades Container Structure 216](#_Toc107913172)

[Figure 128 – DAM AS Offer Insufficiency Report Container Structure 218](#_Toc107913173)

[Figure 129 – End of Adjustment Period results: BidSet/OutputSchedule 220](#_Toc107913174)

[Figure 130 – Two Hour Warning results: BidSet/OutputSchedule 222](#_Toc107913175)

[Figure 131 – DAM Phase II Validation Results: BidSet/[Bid Type] 226](#_Toc107913176)

[Figure 132 - RTDIndicativeBasePoints Container Structure 230](#_Toc107913177)

[Figure 133 - RTDIndicativeLMPs Container Structure 232](#_Toc107913178)

[Figure 134 - SCEDPriceAdderORDC Container Structure 235](#_Toc107913179)

[Figure 135 - RTPriceAdderORDC Container Structure 238](#_Toc107913180)

[Figure 136 - RTDPriceAdderORDC Container Structure 240](#_Toc107913181)

[Figure 137 – PVGRPP Structure 243](#_Toc107913182)

[Figure 138 - STPPF Structure 245](#_Toc107913183)

[Figure 139 - RTMPriceCorrectionSOGPRICE Container Structure 247](#_Toc107913184)

[Figure 140 - RTMPriceCorrectionSOGLMP Container Structure 250](#_Toc107913185)

[Figure 141 - Notification Sequence Diagram 253](#_Toc107913186)

[Figure 142 - WS-Notifications Message Structure 254](#_Toc107913187)

[Figure 143 - Event Structure 255](#_Toc107913188)

[Figure 144 - MMS System generated Alerts and Notifications 266](#_Toc107913189)

[Figure 145 - EMS System generated Alerts and Notifications 269](#_Toc107913190)

[Figure 146 - NMMS System generated Alerts and Notifications 270](#_Toc107913191)

[Figure 147 - UnconfirmedTrades Container 278](#_Toc107913192)

[Figure 148 - Confirmed Trades 279](#_Toc107913193)

[Figure 149 - UnconfirmedTrades Container 280](#_Toc107913194)

[Figure 150 - Confirmed Trades (canceled) 281](#_Toc107913195)

[Figure 151 - AwardedEnergyOffer Structure 283](#_Toc107913196)

[Figure 152 - AwardedEnergyOnlyOffer Structure 284](#_Toc107913197)

[Figure 153 - AwardedEnergyBid Structure 286](#_Toc107913198)

[Figure 154 - AwardedAS Structure 290](#_Toc107913199)

[Figure 155 - AwardedCRR Structure 292](#_Toc107913200)

[Figure 156 - AwardedPTPObligation Structure 294](#_Toc107913201)

[Figure 157 – Ancillary Service Obligation Structure 296](#_Toc107913202)

[Figure 158 – Outage State Change Structure 298](#_Toc107913203)

[Figure 159 - StartupShutdownInstructions Structure 299](#_Toc107913204)

[Figure 160 - Wind Generation Resource Production Potential Structure 302](#_Toc107913205)

[Figure 161 – Insufficiency Report Structure 307](#_Toc107913206)

[Figure 162 – End of Adjustment Period results: BidSet/OutputSchedule 309](#_Toc107913207)

[Figure 163 – DAM Phase II Validation Results: BidSet/[Bid Type] 314](#_Toc107913208)

[Figure 164 – Photovoltaic Generation Resource Production Potential Structure 317](#_Toc107913209)

[Figure 165 - Outage Scheduling Interface Sequence Diagram 319](#_Toc107913210)

[Figure 166 - Outage Creation Request 320](#_Toc107913211)

[Figure 167 - Outage Creation Response 320](#_Toc107913212)

[Figure 168 - OutageSet Structure 320](#_Toc107913213)

[Figure 169 - High-level Outage Structure 321](#_Toc107913214)

[Figure 170 - Outage Information 322](#_Toc107913215)

[Figure 171 - Outage Grouping, Details, Schedule, Contacts, Notes, and Opportunity 329](#_Toc107913216)

[Figure 172 – Outage Creation Elements 341](#_Toc107913217)

[Figure 173 - OutageSet Query Request 346](#_Toc107913218)

[Figure 174 - OutageSet Query Response 346](#_Toc107913219)

[Figure 175 - OutageQuery Container Element structure 351](#_Toc107913220)

[Figure 176 - Outage Update Request 351](#_Toc107913221)

[Figure 177 - Outage Update Response 352](#_Toc107913222)

[Figure 178 - Outage Update XML payload Schema Diagram 353](#_Toc107913223)

[Figure 179 - Outage Cancel Request 360](#_Toc107913224)

[Figure 180 - Outage Cancel Response 363](#_Toc107913225)

[Figure 181 - Get mRID Request 364](#_Toc107913226)

[Figure 182 - Get mRID Response 365](#_Toc107913227)

[Figure 183 - SystemStatus Request 365](#_Toc107913228)

[Figure 184 - SystemStatus Response 366](#_Toc107913229)

[Figure 185 - TestNotification Request 366](#_Toc107913230)

[Figure 186 - TestNotification Response 367](#_Toc107913231)

[Figure 187 – Change ActiveNotificationURL Request 367](#_Toc107913232)

[Figure 188 – Accepted values for the Request/Option element 368](#_Toc107913233)

[Figure 189 – Change ActiveNotificationURL Response 368](#_Toc107913234)

[Figure 190 – Get SASM ID Request 368](#_Toc107913235)

[Figure 191 – Get SASM ID Response 369](#_Toc107913236)

[Figure 192 - Reports Request 370](#_Toc107913237)

[Figure 193 - Reports Response 371](#_Toc107913238)

[Figure 194 - Reports Container 372](#_Toc107913239)

[Figure 195 - Example ResParametersSet Sequence Diagram 375](#_Toc107913240)

[Figure 196 - Message: Request for a get ResParametersSet 376](#_Toc107913241)

[Figure 197 - Message: reply to get ResParametersSet request 376](#_Toc107913242)

[Figure 198 - Message: Request for a change ResParametersSet 377](#_Toc107913243)

[Figure 199 - Message: reply to change ResParametersSet request 378](#_Toc107913244)

[Figure 200 - Message: Request for cancel a ResParametersSet 378](#_Toc107913245)

[Figure 201 - Message: reply ResParametersSet 379](#_Toc107913246)

[Figure 202 – ResParametersRequest Common Elements 380](#_Toc107913247)

[Figure 203 – Generator Resource Parameters Structure 381](#_Toc107913248)

[Figure 204 – Normal and Emergency Curve Structures 382](#_Toc107913249)

[Figure 205 – Generator Resource Parameters Details Structures 382](#_Toc107913250)

[Figure 206 – GenResourceParameters Requirements 384](#_Toc107913251)

[Figure 207 – Controllable Load Resource Structure 385](#_Toc107913252)

[Figure 208 – Normal and Emergency Curve Structures 386](#_Toc107913253)

[Figure 209 – Controllable Load Resource Parameters Details Structures 386](#_Toc107913254)

[Figure 210 – Controllable Load Resource Requirements 387](#_Toc107913255)

[Figure 211 – Non Controllable Load Resource Structure 388](#_Toc107913256)

[Figure 212 – Non Controllable Load Resource Parameters Details Structures 388](#_Toc107913257)

[Figure 213 – NonControllableLoadResource Requirements 389](#_Toc107913258)

[Figure 214 – Resource Parameters Structure 390](#_Toc107913259)

[Figure 215 – ResourceParameters Requirements 391](#_Toc107913260)

[Figure 216 - Example Verbal Dispatch Instructions Sequence Diagram 393](#_Toc107913261)

[Figure 217 – Request message: get VDIs 394](#_Toc107913262)

[Figure 218 – Reply Message: reply to get VDIs 395](#_Toc107913263)

[Figure 219 – Request Message: change VDI 395](#_Toc107913264)

[Figure 220 – Reply Message: reply VDIs 396](#_Toc107913265)

[Figure 221 – VerbalDispatchInstruction Structure 398](#_Toc107913266)

[Figure 222 – VerbalDispatchInstruction Requirements 399](#_Toc107913267)

[Figure 223 – getNotifications Request 406](#_Toc107913268)

[Figure 224 – getNotifications Response 407](#_Toc107913269)

[Figure 225 – Notifications Container 408](#_Toc107913270)

# Introduction

This document describes the message structure machine to machine interfaces for Market Participant applications that need to interact with ERCOT Nodal Market systems. The intended audience of this document is developers that will be integrating Market Participant applications to the ERCOT Nodal systems through the use of the interfaces described within this specification.

Where sections 1 and 2 of this document apply to all interfaces, sections 3 and beyond describe specific groupings of interfaces. The appendices provide XML Schemas, WSDLs and additional examples.

The interfaces and related interactions described by this document define the externally-visible (black box view) perspective of the services provided by this project. It is the intent of this specification and interface architecture to shield Market Participants from the details of systems integration internal to ERCOT.

## Purpose

The interfaces described by this document are intended to be used by Market Participants for machine to machine integration. This document is intended to provide all the details of the message structures and technical interoperability standards required for the machine to machine interface.

## Scope

The scope of this document is to describe web services provided for integration by Market Participants from the perspective of external integration. This document has program level scope as related to web services that would be used by Market Participants for machine to machine interaction with nodal applications as detailed in an agreed list of interfaces to be managed by the Nodal project. The intent of this design is to leverage the integration layer (IL) to expose web services needed for external integration by Market Participants.

The following are specifically outside the scope of this document:

* The details of integration from the IL to specific applications (e.g. MMS, EMS) is outside the scope of this specification and will be described in subsequent design documents
* Inter Control Center Communications Protocol (ICCP) communications
* Interactions with application User Interfaces (UI)
* Access of information through MIS

This document is not intended as a replacement for the current services provided by the ERCOT used to access reports. The ‘get Reports’ interface described in section 9.2.1 provides a wrapper for this interface.

## Definitions, Acronyms, and Abbreviations

| *Term/Acronym* | *Definition* |
| --- | --- |
| ADJ | Adjustment |
| AS | Ancillary Services, includes spinning reserve, non-spinning reserve, responsive reserve and regulation |
| award | An award identifies the acceptance of a bid or offer. |
| BidSet | A BidSet is an XML container type for a collection of bids, offers, trades and schedules as defined by the IEC Common Information Model |
| CIM | Common Information Model, an IEC standard |
| CRR | Congestion Revenue Rights, a system implemented at ERCOT by Nexant that is responsible for CRR auctions |
| CSV | A file format that uses values separated by commas |
| DAM | Day Ahead Market |
| DRUC | Day-Ahead Reliability Unit Commitment |
| EMS | Energy Management System, a system implemented at ERCOT by Areva |
| FIP | Fuel Index Price |
| FOP | Fuel Oil Price |
| FTP | File Transfer Protocol |
| GUID | Globally Unique Identifier. GUIDs are usually implemented as random 128 bit numbers that are usually represented as a string of hexadecimal digits. |
| HASL | High ancillary services limit |
| HDL | High dispatch limit |
| HEL | High emergency limit |
| HRUC | Hour-Ahead Reliability Unit Commitment |
| HSL | High sustained limit |
| IEC | International Electrotechnical Commission |
| LASL | Low ancillary services limit |
| LDL | Low dispatch limit |
| LEL | Low emergency limit |
| LMP | Location marginal price |
| LSL | Low sustained limit |
| MCPC | Market clearing price for capacity |
| MIS | Market Information System, an umbrella for the various interfaces provided to Market Participants by ERCOT |
| MMS | Market Management System, a system implemented at ERCOT by ABB |
| MP | Market Participant |
| MW | Megawatt, a measure of power |
| MWh | Megawatt hour, a measure of energy |
| NMMS | Network Model Management System, a system implemented at ERCOT by Siemens |
| NOMCR | Network operating model change request |
| Non-Spin | Non-spinning reserve |
| OASIS | Organization for the Advancement of Structured Information Systems |
| Operating Date | Synonymous with Trade Date |
| PMCR | Planning Model Change Request |
| POC | Proof Of Concept |
| QSE | Qualified Scheduling Entity. This is a type of Market Participant. Each QSE is identified by a certificate. A QSE is identified by a DUNS number, ‘long name’ and ‘short name’ |
| RDF | Resource Descriptor format. An XML format used by NMMS for model exchanges using IEC 61970-501. |
| Reg-Down | Regulation down |
| Reg-Up | Regulation up |
| RRS | Responsive reserve service |
| RTM | Real-Time Market |
| SAMR | Special Action Model Request |
| sink | Sink settlement point |
| SOAP | Simple Object Access Protocol |
| SoSA | System of Systems Architecture |
| source | Within a bid, offer, schedule, trade or award this refers to the source settlement point. Within a message header, this refers to the ID of the market participant. |
| SP | Settlement Point |
| SPP | Settlement Point Price |
| TP | Transmission Provider |
| Trading Date | Synonymous with Operating Date. Represented in XML using the ‘tradingDate’ tag. This is based on Texas prevailing time. |
| UTC | Universal Coordinated Time, an international standard |
| WS | Web Services. There are many web service standards that are commonly prefixed by ‘WS’. |
| WSDL | Web Services Definition Language |
| XML | eXtensible Markup Language |
| XSD | XML Schema, used to define the structure of XML documents |
| XSL | XML StyleSheet Language |
| Z | Zulu, an indicator for the use of GMT or UTC time |

## References

|  |  |
| --- | --- |
| Artifact | Definition |
| External Interfaces Conceptual Design | Conceptual design for external interfaces using web services |
| External Interfaces Security Design Specification | Detailed security design for external interfaces. This is a companion document to the External Interfaces Conceptual design. |
| OASIS WS-Notifications | OASIS Web Services Base Notification standard |
| OASIS WS-Security | OASIS Web Services Security |
| XSDs | Specific message structure and element details. |

Additional references to related standards are described in section 1.6.

## Overview

This document focuses on the external interface design and related interface definitions from all perspectives except for security, which is described in detail in a companion document. The interfaces are to be provided using web services, where a rationale is provided in subsequent sections. The web services defined by this document will support a wide variety of machine to machine information exchanges.

## Program-level Standards

In general, the design described by this document will leverage web services and related security standards as defined by the World-Wide Web Consortium (W3C) and OASIS. Program-level standards include those related to security, as well as basic web service standards including:

* XML
* XML Schema
* XPath
* XSL
* SOAP
* Web Services
* WSDL
* RDF

These are described in the companion security design document. W3C standards can be freely accessed from <http://www.w3.org>.

Another key program standard is the IEC Common Information Model (CIM), as defined by IEC 61970-301. This is used to define models used by ERCOT, which are exchanged using IEC 61970-501. It will also be leveraged by this design for the definition of messages used for interfaces. There is also a standard for message structures defined by IEC 61968-1. There standards can be purchased from the IEC web site at <http://www.iec.ch>. Materials related to IEC standards, including the CIM model it self can be freely obtained from the UCA International Users Group SharePoint at <http://sharepoint.ucausers.group.org/CIM>.

There are several key Internet Engineering Task Force references. These include:

* Internet Engineering Task Force RFC 2828: Internet Security Glossary
* Internet Engineering Task Force RFC 2119: Key words to indicate RFC requirement levels
* Internet Engineering Task Force RFC 2246: Transport Layer Security (TLS)
* Internet Engineering Task Force RFC 3275: XML Digital Signature and Processing

IETF documents can be freely obtained from <http://www.ietf.org>.

An ERCOT variation of the OASIS WS-Base Notification standard is used to define the mechanism for issuance of asynchronous notification messages to Market Participants. Given the application of WS-Notifications for Nodal, only a subset of the capabilities of WS-Notifications is required. OASIS standards being leveraged include:

* Web Services Notifications (WS-Notifications)
* Web Services Security (WS-Security)
* Web Services Security X.509 Token Profile
* Web Services Security Username Token Profile

OASIS standards can be freely obtained from <http://www.oasis-open.org>.

The definition of timestamps is specified by ISO-8601, with the exception that timestamps of 24:00:00 are not used for compatibility reasons. This is partly a consequence of the XML Schema definition for ‘dateTime’, where hour 24 is not explicitly allowed. There are some implementations of timestamps within software products that do not correctly handle timestamps of 24:00:00.

ISO standards can be purchased online from a variety of sources including ANSI, at <http://www.ansi.org>. Descriptions of ISO-8601 can be freely obtained from other sources including Wikipedia

# Services Organization

The services described by this document are defined using a combination of Web Services Definition Language (WSDL) and XML Schema. The WSDLs are organized as follows:

* One or more WSDLs defined by ERCOT, defining operations related to synchronous request/reply web service messages
* WSDL defined by OASIS for WS-Notifications to provide support for asynchronous messaging, using web services

In both of the above cases, one or more XML Schemas (XSD) is used to define the structure of message payloads.

XSD Diagrams:

Throughout this document there are diagrams representing XSD structures. Understanding the diagrams will assist in implementing any services based on these XSDs. Below is a link which elaborates the details and structures of the XSD diagrams.

<http://www.diversitycampus.net/Projects/TDWG-SDD/Minutes/SchemaDocu/SchemaDesignElements.html>

Example WSDL and XSD are provided in the appendices. It is anticipated that these would be key design artifacts for developers. The following diagram provides an overview of the interfaces described by this document.

## Common Message Structure

Unless otherwise specified, all messages use a common message envelope, where a predefined structure is used for requests and another structure is used for responses. This structure is based upon the IEC 61968-1 standard. Messages are constructed with several sections, including:

* Header: required for all messages, using a common structure for all service interfaces
* Request: optional, defining parameters needed to qualify request messages
* Reply: Used for response messages to indicate success, failure and error details
* Payload: optional, used to convey message information as a consequence of the ‘verb’ and ‘noun’ in the message Header. The payload structure provides options for payload compression.

### Message Header Structure

Common to both the request and response messages is a header structure. The header has several required fields that must be populated, these include:

* Verb, to identify a specific action to be taken. There are an enumerated set of valid verbs, where commonly used values include ‘get’, ‘create’, ‘change’, ‘cancel’, ‘close’ and ‘reply’. For notification messages ‘past tense’ verbs are used, which can include ‘created’, ‘changed’, ‘canceled’ and ‘closed’. Implementations should treat verbs ‘update’ and ‘updated’ as synonyms to ‘change’ and ‘changed’. There is also a special case when the noun is ‘BidSet’, where a ‘create’ is equivalent to a ‘change’ due to the nature of the market system.
* Noun: to identify the subject of the action and/or the type of the payload (e.g. BidSet, Notification) if a payload is provided.
* Source: identifying the source of the message, which should be the ID of the Market Participant or ERCOT (typically for reply messages)? This should be the ‘short name’ of the QSE. (This will be verified against the DUNS number in the certificate)
* Revision: To indicate the revision of the message definition. This should be ‘1’ by default.
* Nonce: A unique number that would not be repeated by the Market Participant within the period of at least a day. This could be a sequence number, large random number or a GUID. This is defined by WS-Security. A combination of this number and the timestamp make the message unique for a given time period.
* Created: A timestamp to indicate when the message was created. This value and the Nonce are used to protect against replay attacks. This is defined by WS-Security.
* UserID: Should be supplied as it may be required for some interfaces, depending upon underlying implementations.

The following diagram describes the header structure used for request and response messages.



Figure 1 - Web Service Interfaces

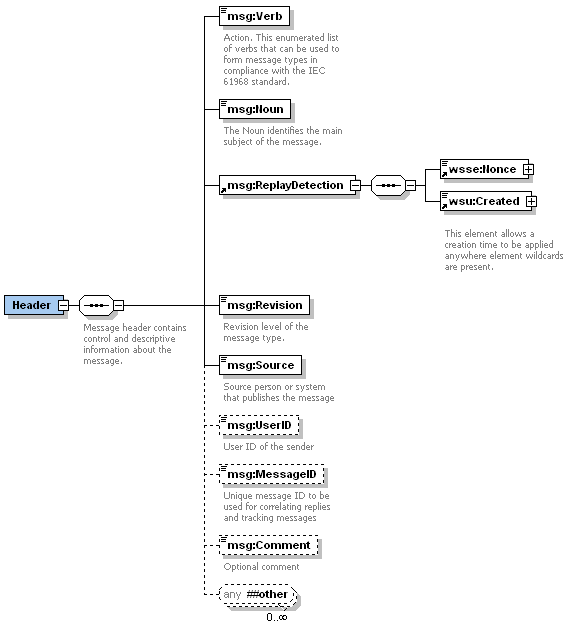


Figure 2 - Message Header Structure

There are several optional fields that may be populated. In the above diagram, the optional items are represented using dashed borders. If the MessageID is populated on a request, it will be returned on the reply. The Comment field is never used for any processing-related logic. The UserID may be used to indicate the person responsible for initiating a transaction, and will be logged as appropriate, but verification is the responsibility of the Source system.

In order to identify the Market Participant in a uniform manner, the registered ‘short name’ of the Market Participant should be supplied as the value of Header/Source. ERCOT will verify that this value is consistent with the Market Participant as identified by the certificate, noting the DUNS number is contained within the certificate.

### Request Message Structures

The following diagram describes the structure of a request message that would be used in conjunction with a WSDL operation.

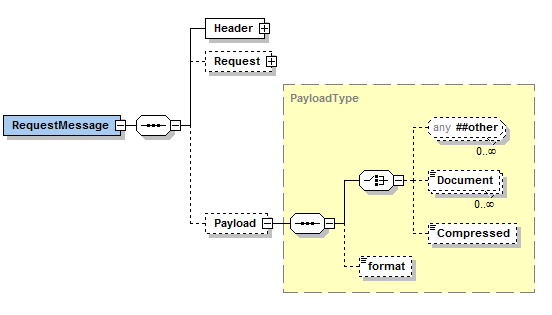


Figure 3 - Request Message Structure

The RequestMessage can also optionally contain a package with parameters relevant to the request, called Request. It is likely that different or variant Request packages may be defined to be used in conjunction with messages for a specific web service operation. In those cases, the corresponding WSDL and XSD would identify the optional parameters. The description of the interface (in subsequent sections of this document) would identify the usage of those parameters. The following is an example RequestType used in the definition of a Request package that defines some common parameters used for requests, however it is important to note that these are typically application specific. These parameters are most commonly used in conjunction with ‘get’ requests as qualifiers.

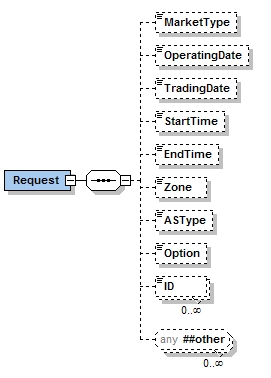


Figure 4 - Message Request Parameters

One key use of the RequestType is to avoid the placement of application specific request parameters in the header or within payload definitions. Also, where a set of requests that were supported by a specific web service operation had significantly different requirements for information in the RequestType, it could justify the use of RequestType variants, where each variant will be used for the definition of messages for the specific web service operation.

### Payload Structures

There are some requests where a Payload must be provided, as would be the case for a message with a verb of ‘create’ or ‘change’. Payloads are typically XML documents that conform to a defined XML schema. However, there are exceptions to this rule. Some XML payloads may not have useful XML schemas, as in the case of RDF files or dynamic query results, as well as non-XML formats such as CSV and PDF. There may also be cases where a large payload must be compressed, in the event that it would become very large and otherwise consume significant network bandwidth. In order to accommodate a variety of payload format options the following payload structure is used.

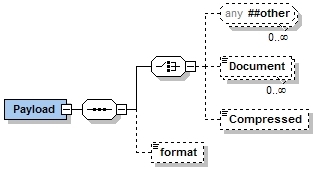


Figure 5 - Message Payload Container Structure

In the previous diagram, any type of XML document may be included, using the XML ‘any’ structure. While this provides options for loose-coupling, specific complex types defined by XML schemas (XSDs) can be used as well. The WSDL in the appendix provides an example of this case.

Payloads can also be supplied as XML encoded strings using the ‘Document’ tag, although this method is less preferred than used of the XML ‘any’.

There are also some cases where a zipped, base64 encoded string is necessary, and would be passed using the ‘Compressed’ tag. The Gnu Zip compression shall be used in order to provide compatibility within both Java and Microsoft .Net implementations. A Java example is provided in Appendix G. Specific examples of the usage of payload compression would be where:

1. An XML payload, conforming to a recognized XML schema exceeds a predefined size (e.g. 1MB). This would be very common for large Market Participant sets of bids, offers, trades or schedules
2. A payload has a non-XML format, such as PDF, Excel spreadsheet, CSV file or binary image
3. A payload is XML, but has no XML schema and exceeds a predefined size, as would be the case of a dynamic query that would return an XML result set

When a payload is compressed and base64 encoded, it is stored within the Payload/Compressed message element as a string.

The format tag can be used to identify specific data formats, such as XML, RDF, PDF, DOC, CSV, etc. This is especially useful if the payload is compressed. The use of this tag is optional, and would typically only be used when the payload is stored using the Payload/Compressed message element.

The above options provide an alternative to the use of SOAP attachments. SOAP attachments are more difficult to secure since the SOAP envelope signature signs the SOAP body but does not sign the attachment. This also requires that the payload is processed separately from the rest of the SOAP message (e.g. the message is parsed to extract the payload, and then the payload is parsed and processed). However, we believe this implementation approach is less complex than using SOAP attachments.

### Response Message Structures

The following diagram describes the structure of a response message that would be used in conjunction with a WSDL operation, as a response to the request message.

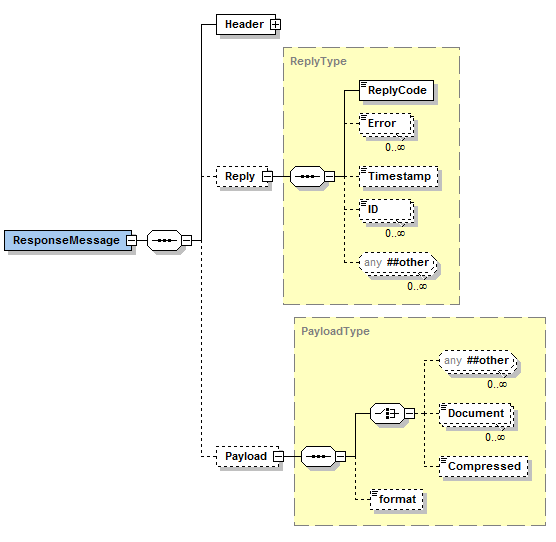


Figure 6 - Response Message Structure

The ReplyCode would be set to OK to indicate that the request was successful, otherwise it would be set to either ERROR or FATAL, and one or more Error elements would be provided to describe the error(s). FATAL ReplyCode is indicative of an internal error having occurred. There may also be more specific error information provided within the payload, as in the case of bids within a BidSet container.

If the MessageID was set in the Header for the RequestMessage, the value will be returned in the Header of the ResponseMessage.

## Common Security Implementation

This section will provide an overview of security from the perspective of implementation requirements for Market Participants. Market Participants MUST take two basic steps in securing their Web Services Interaction with ERCOT:

1. Secure the Transport layer
2. Secure SOAP messages

### Secure the Transport layer

The transport layer is secured by deploying Secure Socket Layer (SSL) and Transport Layer Security (TLS) following these steps:

1. Obtain client side certificate (these Certificates are issued by VeriSign under the ERCOT brand).
2. Implement mutual authentication (explained below).
3. Ensure minimum SSL/TLS security settings

Note that TLS is an enhanced specification based on SSL. References to SSL refer to both SSL and TLS.

SSL is a standard mechanism for Web services that is available on virtually all application servers. This widely used, mature technology, which secures the communication channel between client and server, will satisfy all of ERCOT’s use cases for secure Web Service communications. Since it works at the transport layer, SSL covers all information passed in the channel as part of a message exchange between a client and a server, including attachments. Authentication is an important aspect of establishing an HTTPS connection. Many platforms support the following authentication mechanisms for Web Services using HTTPS:

* The server authenticates itself to clients with SSL and makes its certificate available.
* The client uses basic authentication over an SSL channel.
* Mutual authentication with SSL, using the server certificate as well as the client certificate, so that both parties can authenticate to each other.

With Web Services, the interaction use case is usually machine to machine; that is, it is an interaction between two application components with no human involvement. Machine-to-machine interactions have a different trust model from typical website interactions. In a machine-to-machine interaction, trust must be established proactively, since there can be no real-time interaction with a user about whether to trust a certificate. Ordinarily, when a user interacts with a website via a browser and the browser does not have the certificate for the site, the user is prompted about whether to trust the certificate. The user can accept or reject the certificate at that moment. With Web Services, the individuals involved in the deployment of the Web Service interaction must distribute and exchange the server certificate, and the client certificate (for mutual authentication), prior to the interaction occurrence.

The combination of the two settings—CONFIDENTIAL for transport guarantee and CLIENT-CERT for auth-method—enables mutual authentication. When set to these values, the containers for the client and the target service both provide digital certificates sufficient to authenticate each other. (These digital certificates contain client-specific identifying information.)

### Secure SOAP messages

Besides creating a secure communication channel between a client and a Web Service, ERCOT Web Service message exchanges require that security information be embedded within the SOAP message itself. This is often the case when a message needs to be processed by several intermediary nodes before it reaches the target service or when a message must be passed among several services to be processed.

Message-level security is very useful in XML document-centric applications, since different sections of the XML document may have different security requirements or be intended for different users.

SOAP messages’ signing is done through the following:

* Obtain application/system signing certificate. (These certificates are issued by VeriSign under the ERCOT brand).
* Sign all SOAP messages, using Web Services Security Standards and its X.509 Certificate Token Profile
* Message headers MUST include a timestamp and a nonce
* Validate all SOAP messages have:
  + Signature
  + Certificates
  + Revocation status of certificates
  + Use of timestamp and nonce (to prevent replay attacks)

There are a variety of signature method/security tokens that are commonly used. ERCOT will sign outbound messages using SHA-1/RSA and strongly recommends its use by market participants for signing messages. However, the following SOAP signing methods will be supported for incoming messages:

1. MD5/RSA
2. DSA (which implies SHA-1)
3. SHA-256 and SHA-512 with RSA
4. ECDSA

Appendix D provides examples for the generation of signatures. Appendix E provides an annotated example of a SOAP message.

## Modeling and Conventions

There are several conventions that are used for definitions, data items and information models. *Note that additional values and conventions will be defined as market requirements are finalized.*

### Use of the IEC CIM

Where possible the IEC CIM should be leveraged. The ERCOT Nodal project and related systems use the CIM as a key standard. One example of this is for the submission and management of models through the NMMS system. Examples of leveraging the CIM for external web services include:

* The use of data structures defined by the IEC CIM where appropriate in payload definitions
* CIM naming conventions are used wherever possible, e.g. ClassName, propertyName
* The properties ‘startTime’ and ‘endTime’ are typically used to identify time intervals, as they are also used within many CIM classes. Instead of using combinations of start date, start hour and potentially an interval number (e.g. to represent 15 minute intervals), absolute times should be specified.
* The property ‘mRID’ is used to uniquely identify objects such as bids, offers, trades, awards and schedules
* A trade or operating date is always derived from the startTime of a bid, offer, trade, or schedule.

Where interfaces have references to resources, QSEs, settlement points and electrical buses, the ‘name’ property is used for the reference, as opposed to ‘mRID’. This is to maximize legibility and to provide for consistency with underlying market applications. Consequentially, simplified XML is used, where the use of the ‘name’ property is implied as opposed to explicit use of a <name> tag.

The following class diagram describes the structure of CIM Curves, IrregularIntervalSchedules, and RegularIntervalSchedules. These classes are CIM building blocks, where there are many other classes in the CIM that inherit from these three basic types of curves and schedules (e.g. PriceCurve).

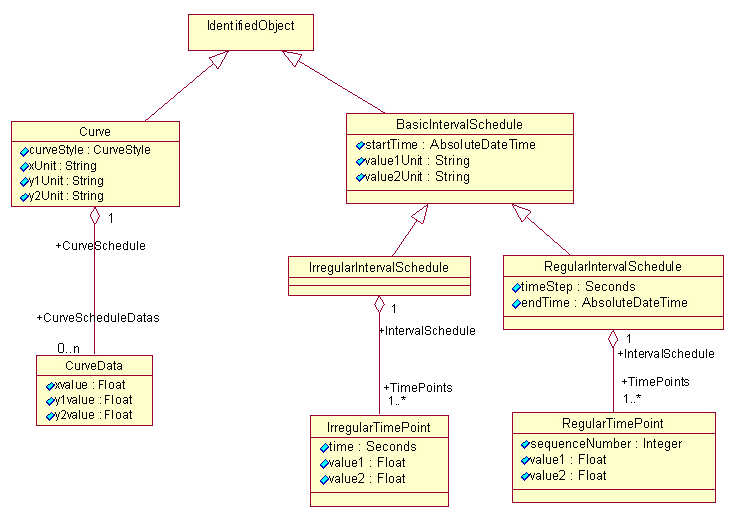


Figure 7 - CIM Curves and Schedules

For use within ERCOT Nodal, customizations and specializations of CIM curves and schedules are defined. One example is that units (e.g. xUnit, y1Unit, value1Unit) are not used.

For Curves:

* A curveStyle can be specified to identify the type of curve (e.g. CURVE, FIXED, VARIABLE) for interpretation of the curve by an application:
* A commonly used curve is a PriceCurve, where the curve is defined by

‘CurveData’ elements where the ‘xvalue’ is used for MW and the ‘y1value’ is used for $/MWh. Not all curve styles are allowed for each product type. The interpretation of curveStyle is as follows:

* + FIXED: only one point is provided, where the intent is that the bid or offer is for all or nothing (i.e. either ‘xvalue’ or 0)
  + VARIABLE: only one point is provided, but the intent is that any value in the range 0 to ‘Xvalue’ can be taken
  + CURVE: Up to 10 points (to define up to 9 segments) can be provided
* Curves are often used where values are NOT plotted against time (e.g. MW vs. $/MWh)
* For PriceCurve for ancillary services, ASCurveData elements are used (instead of CurveData elements) to define points for REGUP, REGDN, RRS and NSRS values.

The following is an example of a PriceCurve that is derived from a CIM Curve:

<PriceCurve>

<startTime>2008-01-01T00:00:00-06:00</startTime>

<endTime>2008-01-01T10:00:00-06:00</endTime>

<curveStyle>CURVE</curveStyle>

<CurveData>

<xvalue>60</xvalue>

<y1value>40.00</y1value>

</CurveData>

<CurveData>

<xvalue>80</xvalue>

<y1value>45.00</y1value>

</CurveData>

<CurveData>

<xvalue>90</xvalue>

<y1value>47.00</y1value>

</CurveData>

</PriceCurve>

In order to better meet the needs of ERCOT Market Participants for interfaces related to bidding, an alternative to an IrregularIntervalSchedule is defined. This is called a ‘TmSchedule’. A TmSchedule has one or more ‘TmPoints’, where one or more values can be specified at each point in time. This is illustrated by the following diagram:

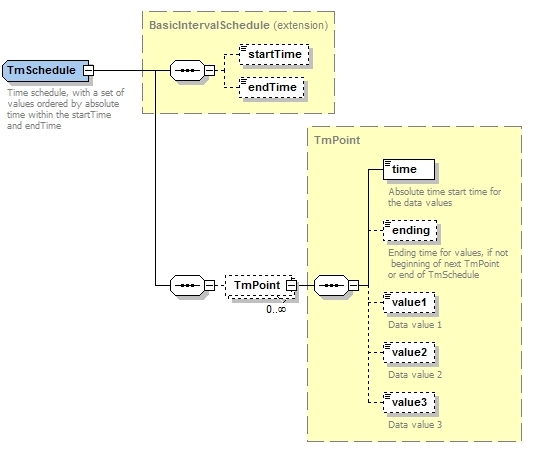


Figure 8 - Time Schedule Structure

The following is an XML example of an EnergySchedule, which is based upon a TmSchedule:

<EnergySchedule>

<startTime>2007-10-17T00:00:00-05:00</startTime>

<endTime>2007-10-18T00:00:00-06:00</endTime>

<TmPoint>

<time>2007-10-17T00:00:00-05:00</time>

<value1>120</value1>

</TmPoint>

<TmPoint>

<time>2007-10-17T10:00:00-06:00</time>

<value1>130</value1>

</TmPoint>

<TmPoint>

<time>2007-10-17T16:00:00-06:00</time>

<value1>115</value1>

</TmPoint>

</EnergySchedule>

It is important to note that a TmPoint identifies the *start* of an interval. The following diagram provides a graphic depiction of the XML example.



The optional ‘ending’ tag can be used to identify the end time for a specific TmPoint. This allows a schedule to have gaps. This is demonstrated by the following example.

<EnergySchedule>

<startTime>2007-10-17T00:00:00-05:00</startTime>

<endTime>2007-10-18T00:00:00-06:00</endTime>

<TmPoint>

<time>2007-10-17T00:00:00-05:00</time>

<ending>2007-10-17T05:00:00-05:00</ending>

<value1>120</value1>

</TmPoint>

<TmPoint>

<time>2007-10-17T10:00:00-06:00</time>

<value1>130</value1>

</TmPoint>

<TmPoint>

<time>2007-10-17T16:00:00-06:00</time>

<value1>115</value1>

</TmPoint>

</EnergySchedule>

The following diagram provides a graphic depiction of the previous XML example.



It is important to note a few conditions that may be treated as errors by applications using the data represented in a TmSchedule:

* TmPoints should be provided in chronological order
* TmPoints that result in a zero width time interval may be treated as errors
* The endTime should be greater than the startTime of the TmSchedule
* For any TmPoint, if specified, the ending must be greater than the time value
* Each TmPoint must exist within the startTime and endTime of the TmSchedule

Also, the use of ‘ending’ is optional. If no ‘ending’ is supplied, the last TmPoint is assumed to and at the ‘endTime’ for the TmSchedule.

### Representation of Time

The ISO 8601 standard is used to define the representations of time values that are conveyed through interfaces. This avoids issues related to time zones and daylight savings time changes.

* For timestamps in messages published by ERCOT would use Texas prevailing time, using the following format: 2007-03-27T14:00:00-05:00 (as time changes from CDT to CST, the -05:00 would change to -06:00)
* Timestamps in messages sent to ERCOT by market participants could use any ISO 8601 compliant timestamp (with the exception of 24:00, as described in section 1.6)

It is extremely important to note that the use of ISO 8601 timestamps within message definitions for the external interfaces defined by this document in no way constrains other representations of time that may include:

* User interfaces, where local time or market hours may be used as desired
* Reports, where reports would be generated using an appropriate local time
* Internal integration, where an application may internally require some other time structure

In some cases it may be desirable to convert between a timestamp and a local time or a market hour. This can be readily accomplished using software functions, XSL, and/or XPath expressions. Examples of time conversion using XSL are provided in Appendix F. Most development environments have a wide variety of APIs that can also be used for necessary time conversions.

In cases where there are a sequence of startTime and endTime intervals provided (as in the cases of ResourceStatus or MinimumEnergy structures as examples), overlapping time intervals are not allowed and will be flagged as an error. Typically in these cases the endTime for one interval should identically match the startTime for the next interval.

In cases where a time that is ‘aligned’ with an interval is expected, but not provided on a request message (e.g. 2008-01-01T01:03:23-06:00), the integration layer may round the time to the nearest minute. This may result in an error being generated by the target application if the minute is not a valid interval boundary. In cases where a time that is aligned with an hour boundary is expected, but not provided on a request message, the integration layer may round the time to the nearest hour. In this case, there may be no generation of an error message.

### Market Products

The market transaction interfaces described in this document use a number of different market products, as well as several types of schedules. The following table identifies each product or schedule type, along with the set of message fields (aside from the market participant ID of the submitting QSE, operating date, and operating hour) that are used to distinctly identify a bid, trade, offer, or schedule. In effect the primary key for each product type is formed by the QSE ID (short name), operating date, operating hour, and items described in the table of Figure 9. The table of Figure 10 provides a mapping of the XML tag names for each product to the approved protocol terms.

| *Product or Schedule Type (XML tag)* | *resource* | *As*  *Type* | *sp* | *bid*  *Id* | *CRR*  *Id* | *Offer*  *Id* | *CRR AH  Id* | *source* | *sink* | *buyer* | *seller* | *INC/*  *DEC* | *AVP Type* |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ASOffer | X | X |  |  |  |  |  |  |  |  |  |  |  |
| ASTrade |  | X |  |  |  |  |  |  |  | X | X |  |  |
| Availability Plan | X |  |  |  |  |  |  |  |  |  |  |  | X |
| Capacity Trade |  |  |  |  |  |  |  |  |  | X | X |  |  |
| COP | X |  |  |  |  |  |  |  |  |  |  |  |  |
| CRR (PTP Obl w/ Links to Option) |  |  |  |  | X | X | X | X | X |  |  |  |  |
| EFC | X |  |  |  |  |  |  |  |  |  |  |  |  |
| EnergyBid |  |  | X | X |  |  |  |  |  |  |  |  |  |
| Energy Only Offer |  |  | X | X |  |  |  |  |  |  |  |  |  |
| EnergyTrade |  |  | X |  |  |  |  |  |  | X | X |  |  |
| IncDecOffer | X |  |  |  |  |  |  |  |  |  |  | X |  |
| OutputSchedule | X |  |  |  |  |  |  |  |  |  |  |  |  |
| PTPObligation |  |  |  | X |  |  |  | X | X |  |  |  |  |
| RTMEnergyBid | X |  |  |  |  |  |  |  |  |  |  |  |  |
| SelfArrangedAS |  | X |  |  |  |  |  |  |  |  |  |  |  |
| SelfSchedule |  |  |  |  |  |  |  | X | X |  |  |  |  |
| ThreePartOffer | X |  |  |  |  |  |  |  |  |  |  |  |  |

Figure 9 - Required Keys for Market Products

The items marked with an ‘X’ are required key fields needed for submission. The operating date for each bid, trade, offer, or schedule is derived from the startTime provided by the bid, offer, trade, or schedule.

The XML schema provided to describe product types has all fields optional. This is because the schemas can be used for “create”, “get” and “cancel” operations. A bid, offer, trade or schedule can be referenced for a ‘get’ or ‘cancel’ request by specifying the transaction ID (mRID) of the desired bids, offers, trades or schedules (using the mRID that was returned by the reply to the submission request).

In each section related to market products, a table is provided that describes data requirements. Please note that fields that are used as ‘keys’ to identify a bid/offer/trade/schedule are identified by ‘K’. Required fields are identified by ‘Y’, option fields are identified by ‘N’. There are some exception cases where identified key fields are identified as not being required (e.g. expiration), and are notated as ‘K, N’.

### Management and Use of Transaction IDs (mRIDs)

In order to uniquely identify a bid, offer or trade, a transaction ID is allocated by ERCOT. The name of the tag used for the transaction ID is ‘mRID’, as this conforms to the IEC CIM. This transaction ID is returned immediately in response to the submission of a BidSet, where each bid, offer, or trade within the BidSet is given its own transaction ID. If a set of N valid unique bids are submitted using a BidSet, there will be N transaction IDs supplied, each corresponding to the order of the bid/offer/trade/schedule on the input BidSet. This transaction ID (mRID) may be used for the following purposes:

* To query the status of a specific transaction
* To cancel a specific transaction
* To relate an award to a bid

It is important to note that an update or change to a transaction does not use an mRID. To update a transaction it is simply resubmitted, where it will overwrite the previous submission of the bid. The resulting mRID for both the original submission and the revised submission will be the same.

A transaction ID (mRID) is a string that takes the following form:

<QSE ID>.<date>.<key string>

Within this, the following substrings are defined:

* QSE ID: ID of the market participant
* Date in format YYYYMMDD
* Key string, a derived string that includes required key fields for each market product

Examples of transaction ID (mRID) are as follows:

SOMEQSE.20081112.COP.UnitABC

QSE2.20081112.TPO.UnitXYZ

When referring to a specific hour of a bid, the hour may be appended to the transaction ID, resulting in a string of the following form:

<QSE ID>.<date>.<key string>.<hour>

Examples of transaction ID as would be used for hourly awards are as follows:

QSE2.20081112.TPO.UnitXYZ.1

QSE2.20081112.TPO.UnitXYZ.2

QSE2.20081112.TPO.UnitXYZ.2R (example for change to CST from CDT)

QSE2.20081112.TPO.UnitXYZ.3

QSE2.20081112.TPO.UnitXYZ.4

On a day with 25 hours, the mRID that represents the ‘repeated’ hour is suffixed with an ‘R’, e.g. ‘QSE2.20081112.TPO.UnitXYZ.2R’.

Additionally when referring to a specific continuous set of hours (i.e. 11, 12, 13), the following convention is used:

<QSE ID>.<date>.<key string>.<hour1>-<hour2>

Following table can be used as reference to convert CST, CDT time format values into hours for cancel and query(get) operations.

|  |  |  |  |
| --- | --- | --- | --- |
| StartTime | EndTime | Hours in mRID | Comments |
| 2010-01-01T00:00:00-06:00 | 2010-01-01T01:00:00-06:00 | Hour: 01  Example mRID:  <QSE ID>.<date>.<key string>.01 |  |
| 2010-01-01T02:00:00-06:00 | 2010-01-01T06:00:00-06:00 | Hours: 03-06  Example mRID:  <QSE ID>.<date>.<key string>.03-06 |  |
| 2010-01-01T04:00:00-06:00 | 2010-01-02T00:00:00-06:00 | Hours: 05-24  Example mRID:  <QSE ID>.<date>.<key string>.05-24 |  |
| 2010-03-14T01:00:00-06:00 | 2010-03-14T03:00:00-05:00 | Hours: 02  Example mRID:  <QSE ID>.<date>.<key string>.02 | DST short day. Hours 01:00:00-06:00 – 03:00:00-05:00 constitute one hour i.e hour 2. |
| 2010-03-14T00:00:00-06:00 | 2010-03-14T03:00:00-05:00 | Hours: 01-02  Example mRID:  <QSE ID>.<date>.<key string>.01-02 | DST short day. Hours 00:00:00-06:00 – 03:00:00-05:00 constitute hours 1 and 2. |
| 2010-03-14T03:00:00-05:00 | 2010-03-14T08:00:00-05:00 | Hours: 04-08  Example mRID:  <QSE ID>.<date>.<key string>.04-08 |  |
| 2010-11-07T00:00:00-05:00 | 2010-11-07T01:00:00-05:00 | Hours: 01  Example mRID:  <QSE ID>.<date>.<key string>.01 |  |
| 2010-11-07T00:00:00-05:00 | 2010-11-07T01:00:00-06:00 | Hours: 01-02  Example mRID:  <QSE ID>.<date>.<key string>.01-02 | Hours  (00:00:00-05:00 to 01:00:00-05:00 ) and  (01:00:00-05:00 to 01:00:00-06:00 ) |
| 2010-11-07T00:00:00-05:00 | 2010-11-07T02:00:00-06:00 | Hours: 01-2R  Example mRID:  <QSE ID>.<date>.<key string>.01-2R | Hours  (00:00:00-05:00 to 01:00:00-05:00 ),    (01:00:00-05:00 to 01:00:00-06:00 )  And  (01:00:00-06:00 to 02:00:00-06:00)\* DST extra hour |
| 2010-11-07T01:00:00-05:00 | 2010-11-07T01:00:00-06:00 | Hour: 02  Example mRID:  <QSE ID>.<date>.<key string>.02 | Normal Hour 2 |
| 2010-11-07T01:00:00-06:00 | 2010-11-07T02:00:00-06:00 | Hour: 2R  Example mRID:  <QSE ID>.<date>.<key string>.2R | DST extra hour |
| 2010-11-07T00:00:00-05:00 | 2010-11-07T03:00:00-06:00 | Hours: 01-03  Example mRID:  <QSE ID>.<date>.<key string>.01-03 | Including DST extra hour. |
| 2010-11-07T01:00:00-05:00 | 2010-11-07T02:00:00-06:00 | Hours: 02-2R  Example mRID:  <QSE ID>.<date>.<key string>.02-2R | Normal 2nd hour & DST extra hour |
| 2010-11-07T01:00:00-05:00 | 2010-11-07T04:00:00-06:00 | Hours: 02-04  Example mRID:  <QSE ID>.<date>.<key string>.02-04 |  |
| 2010-11-07T01:00:00-06:00 | 2010-11-07T04:00:00-06:00 | Hours: 2R-04  Example mRID:  <QSE ID>.<date>.<key string>.02-2R |  |
| 2010-04-01T00:00:00-05:00 | 2010-04-02T00:00:00-05:00 | Hour: 01-24  Example mRID:  <QSE ID>.<date>.<key string> | If all 24hous then not need to mention hours at the end of mRID. |

When referring to “trades” queries (AS Trades, Capacity Trades, and Energy Trades), querying for trades (confirmed or unconfirmed) specifying the trades which was entered by other parties other than the submitting entity, it can be stated as a last token in the pertinent trades mRID:

Token to Trades mRID – Please note that use of otherPartySubmitted (“true” or “false”) token is for querying purposes only (it is not populated on the create/change)

* + AS Trades: *<QSE>.<DATE>.AST.asType.buyer.seller.otherPartySubmitted*
  + Capacity Trades: *<QSE>.<DATE>.CT.buyer.seller.otherPartySubmitted*
  + Energy Trades: *<QSE>.<DATE>.ET.SP.buyer.seller.otherPartySubmitted*

The otherPartySubmitted element is returned in the payload of these three trade types returned result set. If no otherPartySubmitted is specified in these trade queries, it will be treated as a wild card and all trades will be returned for the stated mRID.

Within the definitions of bids, trades, offers and awards, the transaction ID is conveyed using the ‘mRID’ property. This is for consistency with the IEC CIM. The following table describes the initial scheme to be used for each product type.

| *Product or Schedule Type*  *(Protocol term)* | *Section Reference* | *Product or Schedule Type (XML tag)* | *Key String Composition*  *(all are prefixed with QSE ID and date to form mRID)* |
| --- | --- | --- | --- |
| Ancillary Service Offer | 3.3.4 | ASOffer | ASO.resource.asType |
| Ancillary Service Trade | 3.3.5 | ASTrade | AST.asType.buyer.seller |
| Availability Type | 3.3.6 | AVP | AVP.resource.avpType |
| Capacity Trade | 3.3.7 | CapacityTrade | CT.buyer.seller |
| Current Operating Plan | 3.3.9 | COP | COP.resource |
| PTP Obligation w/ Links to Option | 3.3.8 | CRR | CRR.crrId.offerid.crrAHId.source.sink |
| DAM Energy Bid | 3.3.10 | EnergyBid | EB.sp.bidID |
| DAM Energy-Only Offer | 3.3.11 | EnergyOnlyOffer | EOO.sp.bidID |
| Energy Trade | 3.3.12 | EnergyTrade | ET.sp.buyer.seller |
| Exceptional Fuel Cost | 3.3.12 | EFC | EFC.resource |
| Incremental and Decremental Energy Offer Curves | 3.3.3 | IncDecOffer | IDO.resource.type |
| Output Schedule | 3.3.13 | OutputSchedule | OS.resource |
| PTP Obligation Bid | 3.3.14 | PTPObligation | PTP.bidID.source.sink |
| Real-Time Market Energy Bid | 3.3.16 | RTMEnergyBid | REB.resource |
| Self-Arranged Ancillary Service Quantities | 3.3.2 | SelfArrangedAS | SAA.asType |
| Self-Schedule | 3.3.15 | SelfSchedule | SS.source.sink |
| Three Part Supply Offer | 3.3.1 | ThreePartOffer | TPO.resource |
| Outage Schedule | 6.2.1 | OutageSet | ID assigned by Outage Scheduler |

Figure 10 - mRID Structures

An important consideration has to do with the use of the interfaces described in this document in conjunction with the submission of bids by a user using a web-based user interface. The web-based user interface for bidding has no notion of the transaction ID itself, just the individual values used to construct the key string.

Another important note is that mRID values should not be used for purposes of non-repudiation. It is the intent of the signature of the SOAP message, as referenced in section 2.2.2 to provide mechanisms that can be leveraged for non-repudiation.

Please refer to following sections for the pertinent mRID compositions for non-bid centric transactions:

* Outage Scheduling Interface – Section 6.2.1
* Resource Parameter Transactions – Section 9.1
* Verbal Dispatch Instructions – Section 11.1

*Note: The structure of the transaction ID (mRID) is subject to change in the future.* The ‘get mRID’ interface described in section 8.2.1 provides a mechanism to generate an mRID given a set of ‘key’ values.

### Other Conventions

The following are other conventions that must be followed by this specification:

* Within XML definitions, tags should be namespace qualified. For example, a tag of <tag> should be prefixed by a specific namespace reference, e.g. <ns: tag>. This will help to eliminate ambiguity. *(Note that many examples in this document are not namespace qualified for brevity and to aid legibility)*
* Units for power quantities are in megawatts (MW).
* Units for capacity quantities are in megawatts (MW).
* Units for energy quantities are in megawatt-hours (MWh).
* Units for energy prices are in $/MWh.
* Valid market types include: DAM, RTM, DRUC, HRUC, SCED, and SASM . Note that these are not currently required by bidding interfaces
* Trading dates are specified using YYYY-MM-DD, which indicates the operating day
* Valid ancillary service types offered as products include:

|  |  |  |
| --- | --- | --- |
| EWS ASType | Description | Supported in |
| ERS | Emergency Response Service | MMS System Generated Notice: CM-ASM-NOTF |
| Non-Spin | SelfArrangedAS: Non-Spin from both SCED dispatchable and non-SCED dispatchable Resources  ASTrade: Non-Spin provided from SCED dispatchable Resources | SelfArrangedAS, ASTrade |
| NSPNM | ASTrade: Non-Spin provided from non-SCED dispatchable Resources | ASTrade |
| Off-Non-Spin | Off Line Non Spin | ASOffer, ASAwards |
| On-Non-Spin | On Line Non Spin | ASAwards |
| Reg-Down | Regulation Down | ASOffer, ASAwards, SelfArrangedAS, ASTrade |
| Reg-Up | Regulation Up | SelfArrangedAS, ASTrade, ASAwards |
| REGUP-RRS-ONNS | Regulation Up -RRS-Online non-spin | ASOffer |
| RRS | Responsive Reserve Service | SelfArrangedAS, ASAwards |
| RRSGN | RRS Generation Resource | ASTrade |
| RRSLD | RRS Controllable Load Resource | ASTrade |
| RRSNC | RRS Non-controllable Load Resource | ASTrade |

Figure 11 - ASType Usage

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***[NPRR863: Replace Figure 11 above with the following upon system implementation]***   |  |  |  | | --- | --- | --- | | EWS ASType | Description | Supported in | | ERS | Emergency Response Service | MMS System Generated Notice: CM-ASM-NOTF | | Non-Spin | *SelfArrangedAS*: Non-Spin from both SCED dispatchable and non-SCED dispatchable Resources  *ASTrade*: Non-Spin provided from SCED dispatchable Resources | SelfArrangedAS, ASTrade | | NSPNM | *ASTrade:* Non-Spin provided from non-SCED dispatchable Resources | ASTrade | | Off-Non-Spin | Off Line Non Spin | ASOffer, ASAwards | | On-Non-Spin | On Line Non Spin | ASAwards | | Reg-Down | Regulation Down | ASOffer, ASAwards, SelfArrangedAS, ASTrade | | Reg-Up | Regulation Up | SelfArrangedAS, ASTrade, ASAwards | | REGUP-RRS-ONNS | Regulation Up -RRS-Online non-spin | ASOffer | | RRS | Responsive Reserve Service | ASAwards (MCPCs) | | RRSUF | Load Resources controlled by high-set under-frequency relays | SelfArrangedAS, ASOffer, ASTrade ASAwards | | RRSPF | Resources providing Primary Frequency Response | SelfArrangedAS, ASOffer, ASTrade, ASAwards | | RRSFF | Fast Frequency Response (FFR) Resources | SelfArrangedAS, ASOffer, ASTrade, ASAwards | |

* Settlement point names are the same as the name of the associated electrical bus, if defined at a bus level. They may also be defined at other levels (e.g. hubs, load zones).
* QSE short names (as opposed to DUNS numbers) are used to identify the QSE as well as ‘buyer’ and ‘seller’ for trades.

### Precision

The following is a general guideline to be followed when dealing with input and output values while interacting with Market Management System. These restrictions are enforced at the XML schema level, upon submittals.

* Market Submission items
  + Enforcement of tenths of MW will be done at time of submission at the EWS XSD level to enforce submission of floating point numbers (excluding price related data) to 1 decimal point.
* All price related date elements (incoming to ERCOT via EWS or output from MMS) will have 2 decimal digits after the decimal point.
* All MMS generated Notifications sent to market participants will be compliant to:
  + One decimal digit after the decimal point for all non-price related floating point numbers that deal with MW awards, obligations, trades, etc.
  + Two digits after the decimal point for all price related floating point numbers.

### XML Special Characters

The following table contains reserved characters in XML.  In order to use these characters in an XML payload (for example, in Outage Creation “comments” field), they must contain entities to avoid unintended results such as XSD violations.  Refer to the following table for entities to be used with each reserved character:

|  |  |  |
| --- | --- | --- |
| XML Special Character | Description | To be replaced by |
| & | ampersand | &amp; |
| ‘ | apostrophe | &apos; |
| < | greater than | &gt; |
| > | less than | &lt; |
| “ | double quote | &quot; |

## Delivery Approach

ERCOT will provide the following:

* Interface specifications for web services
* Design artifacts, including XML schemas and WSDLs
* An environment for testing the interactions between the Market Participants and ERCOT. This environment will eventually be used for qualification of QSEs.

## Technical Interoperability

There are several strategies that are being employed in order to achieve technical interoperability of MPs with ERCOT. These include:

* + Use of open standards
  + API subgroup allows input from Market Participants on interoperability issues
  + The early deployment of a sandbox environment that enables ERCOT to work with the Market Participants to insure interoperability issues are addressed prior to market trials
  + Deployment of interfaces via the sandbox environment for early testing by Market Participants, so that feedback can be provided to ERCOT

Open standards are a key part of the strategy to achieve technical interoperability. Standards of particular interest include:

* W3C standards
* OASIS WS-\* standards
* IEC Common Information Model and related standards (e.g. IEC 61968-1)

It is very important that the implementation of Web Service interfaces not be dependent upon any proprietary, third party products. Another key requirement is that implementation of web service clients must be possible using both Java and .Net development tools.

More details on technical interoperability will be provided as a consequence of existing web services provided by ERCOT, detailed design and experience with the Sand Box environment.

## Service Level Agreements

Please see Service Level Agreement (SLA) for Nodal Early Delivery Systems (EDS) Environments document found on the ERCOT Website at <http://www.ercot.com/services/sla/> .

## Auditing, Monitoring and Management

ERCOT will perform auditing, monitoring and management for Web Services described by this specification using common services. Internal auditing by ERCOT will be used to track and insure that SLAs are met by ERCOT. All Web Service requests will be logged by ERCOT in order to permit calculations related to SLAs. The signatures supplied on SOAP messages will be recorded with transactions as a means of non-repudiation of each transaction.

It is important to recognize that ERCOT is not responsible for the monitoring and management of Market Participant software and network connectivity. Therefore ERCOT cannot guarantee that notification interfaces provided by Market Participants are accessible as needed for timely delivery of notifications.

## Versioning

It is important to recognize that new versions of interfaces may be provided over time, largely as a consequence of:

* Staging of initial implementation
* New requirements
* Upgrades to vendor products

Wherever possible, interfaces will be evolved through augmentation, where a newer version of an interface is compatible with a previous version of an interface. However, this will not always be possible. New versions of interfaces will be manifested by:

* Changes to WSDLs
* Changes to XML Schemas
* Changes to software implementations

New versions will be deployed within a Sand Box environment for a testing/trial period. WSDL and XML schemas namespaces will include a date reference. Messages will use the Header/Revision field to identify a specific revision - this will enable ERCOT to process multiple versions of an interface where appropriate.

A detailed versioning strategy will be developed and presented for approval to the API Subgroup and TPTF.

## Governance

The web service interfaces will be critical to the operations of both ERCOT and Market Participants. The Web Services will evolve for many reasons, especially as the needs of the market evolve. Governance policies and processes will need to be defined for the Web Service lifecycle that provide strict guidelines related to:

* Design
* Implementation
* Testing
* Deployment
* Management

A comprehensive governance strategy will need to be developed and implemented by ERCOT with input from the API Subgroup and TPTF.

## Web Service Configuration Standards

ERCOT will configure its web servers with specific parameters for use of Web Services by Market Participants (e.g. for security).  ERCOT will define specific configuration details and parameters to be used by Market Participants.

Detailed web service configuration standards will be provided to the API Subgroup and TPTF and will be refined through detailed design and experience with the Nodal Sand Box environment.

Market Participants will also need to set up Web Service listener(s) to handle outgoing notifications from ERCOT.  ERCOT Web services infrastructure supports and provisions for two notification listener URIs (Primary and Backup) for each market participant.  Additionally ERCOT provides an interface through which ERCOT is requested to change the Market Participant’s active listener (see section 7.2.4).  It is however recommended that Market Participant expose their Web Services Listeners’ URI (2 or more) through a load balance device which provides both real time redundancy as well as better performance.

## Web Service Design Assumptions and Limitations

Please note the following restrictions on the use of the ERCOT Nodal Web Services when dealing with submitting transactions to ERCOT Market.

Homogeneous files- BidSets submitted as part of the “create” and “change” BidSet Market Transactions must be limited to a single Product Type (Bid Type).  BidSets that consist of multiple Product Types, also referred to as “complex” or “heterogeneous” BidSets are not supported and will result in rejection and a fatal error upon submission.  ERCOT in response to the Bid Submittal requests will never send a heterogeneous BidSet response back to the market participants.  Please note that ERCOT external web services XML schema will enforce homogeneous BidSet payloads on submittals and conformance to these XML schemas is encouraged at the web services client side.

Size limits- The total size of each BidSet contained in the submission Web Services message payload must be less than 3 Mb in size(Pre-compression).  This limitation will be carried forward from the previous testing performed during the EDS testing in early 2008 and frequently re-evaluated during market trials to consider changes based on market trials results.  The goal of this threshold is to protect the Nodal architecture and mitigate the risk of excessively large submissions negatively impacting the infrastructure and market operation.

In the event that a Market Participant needs to submit transactions greater than this size, they should be prepared to divide the BidSet payload into smaller homogeneous BidSet transactions..  This can be done by limiting the number of resources being offered in the file (rather than submitting the entire portfolio) or by limiting to the number of hours of the day for the submissions (submit 12-hours at a time, rather than 24-hours).

It is also important to note that the 3MB threshold is pre-compression and the compressed payload size will be set to a lower limit.  ERCOT external web services infrastructure will enforce this payload size limit on submittals and exceeding such limit will be rejected by ERCOT.

Sequencing- It is important to note that the ERCOT Nodal Web Services will enforce sequential processing for each Market Participant.  The following verb and noun combinations will be processed in the order they are received by ERCOT (i.e. a first in, first out fashion).

Verbs: create, change, and cancel

Nouns: BidSet, OutageSet, ResParametersSet, Dispute, and VDIs

This will ensure any dependencies among transactions from a given Market Participant are maintained.  As such, the Market Participant is responsible for adhering to the payload size limits and homogeneous submission guidelines stated above while also ensuring their submissions are submitted to ERCOT in the proper sequence.  ERCOT infrastructure will then guarantee these submissions are processed in the order they are received.  ERCOT will base the sequence of submissions on receipt time where the “receipt time” is the time the submissions enters ERCOT’s infrastructure.  Therefore, Market Participants should avoid invoking the Nodal web service with two related and/or dependent submissions simultaneously as the risk exists that these requests could be processed out of order.  For related submissions, the Market Participant should instead submit one create/change/cancel BidSet request and wait for the synchronous response from ERCOT before submitting the second create/change/cancel BidSet request.

# Market Transaction Service

The purpose of the MarketTransaction Service is to support interfaces required for market transactions. This section describes the use of web services by Market Participants as required for bidding processes that involve the submission, change, and cancellation of bids, trades, offers, and schedules for specific markets.

On a given trading day, each Market Participant provides information (using a BidSet) to ERCOT that will be used at the close of the market to determine awards and obligations and to provide schedules needed for market operations. Up until the close of the market, a Market Participant may create, update, or cancel bids, trades, offers, and schedules. When submitted, ERCOT will validate the submission, reporting format or submission errors to Market Participants using notification messages.

## Interfaces Provided

The interfaces provide the means to create (i.e. submit), get (i.e. query) and cancel (i.e. withdraw) bids/trades/offers/schedules on a given trading date. A single container class ‘BidSet’ is used to hold a set of bids/trades/offers/schedules within the Payload section of the message, where each of the bids, trades, schedules or offers may be of a different type.

The following diagram shows an example message sequence, using the ‘verb’ and ‘noun’ convention. Where this section focuses on the requests made by Market Participant systems to the ERCOT Nodal Web Services, the sequence diagram also includes notification messages sent from ERCOT to Market Participant Notification services (as described in section 5).

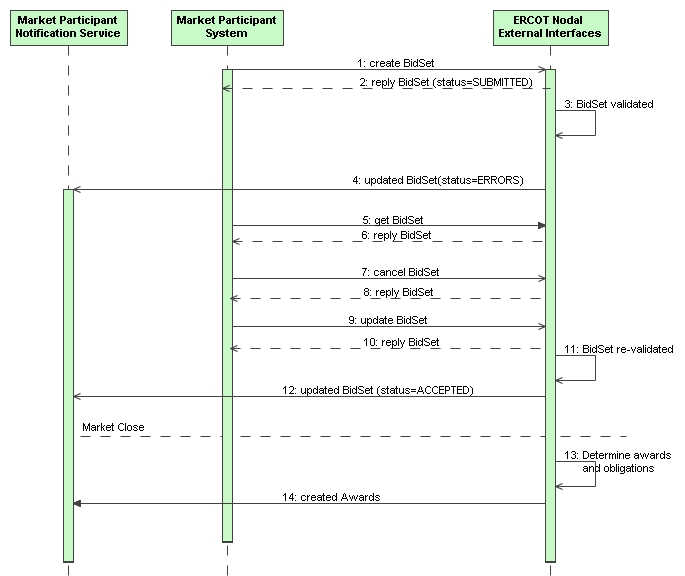


Figure 12 - Example BidSet Sequence Diagram

The message sequence example shown involves the following steps:

1. Market participant sends a RequestMessage for ‘create BidSet’ with an initial BidSet to ERCOT for a specific market
2. In response to step 1, ERCOT performs a simple syntax scan and typically sends a ResponseMessage with ReplyCode=OK. In the response payload, each bid/offer/trade/schedule will identify a ‘SUBMITTED’ status and an mRID value. (An alternative example could result in a reply of ‘ERRORS’ if the syntax check failed, in which case steps 3 and 4 would not occur). This reply is synchronous.
3. ERCOT validates the bids within the BidSet. This could take several minutes. This processing is done asynchronously.
4. A notification message (using verb=changed) is sent to the notification interface provided by the Market Participant. The status of the bids within the BidSet will indicate whether the bid/offer/trade/schedule was PENDING/ACCEPTED or had ERRORS. This message will not include the complete BidSet.
5. When multiple bids are submitted in a large payload, the Market Participant will asynchronously receive the notification holding the bids status for the "accepted" and "rejected" (in the event of ERRORS) bids along with the rejection reason.  As always, MPs may make a request to get the current bids within the submitted BidSet using a RequestMessage with verb=get. The request may be for individual bids using the mRID values, or for the entire set of bids within the BidSet using the TradeDate (using short mRIDs).  It will return all successful (accepted) bids and not the rejected bids.
6. If appropriate, reply is sent for get BidSet in step 5.
7. A Market Participant may choose to cancel one or more bids using a RequestMessage with verb=cancel. The mRID is used for cancellation of a bid. (See section 2.3.4)
8. In response to step 7, a ResponseMessage is sent with verb=’reply’. This response is synchronous.
9. The Market Participant may resubmit some bids (e.g. to correct errors), to change bids (e.g. change prices) and/or submit new bids, using a RequestMessage with verb=change (note that BidSets are a special case where the verbs ‘create’ and ‘change’ can be used interchangeably).
10. In response to step 9, a ResponseMessage is sent with verb=’reply’. This response is synchronous.
11. The newly submitted bids within the BidSet (i.e. the set of bids consequential to steps 7 and 9) are validated. This may take minutes, and processing is performed asynchronously.
12. As in step 4, a notification message is sent to the Market Participant to indicate whether or not the BidSet was validated and accepted (with either an ACCEPTED or PENDING state). The notification message uses verb=changed.
13. After the close of the market, awards and obligations are determined.
14. Market participants are notified of specific awards and/or obligations. The notification message uses verb=’created’, noun=’AwardSet’.

The following diagram describes the potential state transitions of each bid/offer/trade/schedule within a BidSet:



Figure 13 - State Diagram for Bids, Trades, Offers, and Schedules

## Interfaces Required

The following table describes the parameters used in the request message (RequestMessage) for market transactions, noting that each transaction has a request and a response message. The verbs create, update, get and cancel are respectively used to submit (create or change), query and cancel BidSets.

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | *create/get/change/cancel* |
| Header/Noun | BidSet |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Payload | BidSet (a single BidSet may be used for create and update request messages, and may optionally be used for get requests to identify specific bids of interest) |

Figure 14 - Message: create BidSet

The corresponding response messages (ResponseMessage) would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | BidSet |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *May be any number of error message if the ReplyCode is ERROR* |
| Reply/Timestamp | *The time the submission was received by ERCOT* |
| Payload | BidSet, where the status of each bid/offer/trade/schedule within the BidSet may be SUBMITTED, PENDING, ACCEPTED, UNCONFIRMED, REJECTED or ERRORS. Specific errors may be identified for each bid. |

Figure 15 - Message: reply BidSet

In the cases of payloads that would otherwise exceed 1 megabyte, the payloads should be zipped, base64 encoded and stored within the ‘Payload/Compressed’ tag.

For the purposes of BidSets, the verbs create and update can be used interchangeably. Note that only one BidSet is permitted for a given message, and all transactions within the BidSet must be for the same trading date.

## Message Specifications

A BidSet is the payload type used for the submission, query, and cancellation of bids and offers, and serves as a container for the different types of bids, offers, trades and scheduled that are submitted by a QSE for a given trading day. A BidSet identifies the trading date and Market Participant with a sequence of any number of bids, trades, offers, and schedules.

The header of a BidSet is shown in the following diagram:

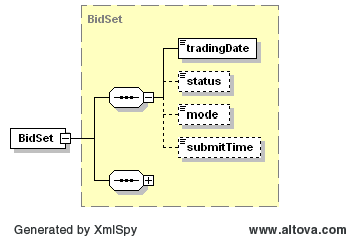


Figure 16 - BidSet Header Structure

An important note is that the BidSet uses an XSD sequence, requiring each type of bid, trade, offer, or schedule to be provided in a specific order. Please reference the XML Schema for the correct order.

The submission (using ’create’ verb and ‘BidSet’ noun in the message header) of a BidSet by a Market Participant will have the effect of either creating new bids (or offers, schedules, trades, etc.) for a given trading date, or overwriting existing bids that were previously submitted for a given BidSet for a given trading day.

Additionally, bids, offers, trades, and schedules may be submitted in partial batches, where more than one BidSet is submitted for a given trading day. When this is done, new bids are aggregated with previously submitted bids and updates to previously submitted bids overwrite the previous bid. The following sequence describes the aggregation of BidSets:

1. A first BidSet is submitted by a market participant for a trading day, with bids 1, 2, 3, 4.
2. A second BidSet is submitted by the market participant for the day with bids 5, 6, 7, 8.
3. The market participant ‘get’s the BidSet for the day, the reply returns the details of bids 1-8.
4. The market participant submits a cancel for bid 1.
5. A third BidSet is submitted by the market participant for the day, where there are updates to bids 5 and 7, as well as a new bid 9
6. The market participant ‘get’s the BidSet for the day, the reply returns the details of bids 2-9.

In order to cancel a bid, a BidSet must be sent using the ‘cancel’ verb, with the specific bids to be cancelled identified within the BidSet. Cancel can only be used to cancel specific bids, not a whole BidSet for a given market.

In order to query a BidSet, A BidSet is sent using the ‘get’ verb, where the desired bids for the specific BidSet are identified. There are then two options:

1. If no bids are identified within the BidSet, all bids for the particular trading date will be returned in the BidSet in the response message.
2. If specific bids are identified within the BidSet, only the details of the specified bids will be returned in the BidSet in the response message.

When a BidSet is returned by a ‘get’ request for a given trade date, the status value of each in the BidSet is populated. Values could include:

* + SUBMITTED (to indicate submission, but no further processing has been completed)
  + ACCEPTED (to indicate successful validation and acceptance of the transaction)
  + UNCONFIRMED (to indicate a trade that is not yet confirmed, once confirmed it would go to the ACCEPTED state)
  + PENDING (to indicate a transaction that is not yet accepted, as it may be submitted for a day after the next trading date or the second phase of validation within MMS has not yet completed)
  + REJECTED (to indicate a transaction that has been rejected)
  + CANCELED (to indicate a transaction was canceled, this can not be retrieved by a get)
  + ERRORS (to indicate that there are one or more errors for the transaction)

Additionally the “Submit Time” is returned with the response to a bid retrieval request, indicating the bid receipt timestamp by ERCOT.  Submit timestamp is an output only field and if populated by the submitter on the bid creation request, it will be overwritten by the ERCOT Integration layer at the time it is received (and returned via bid retrieval).

The following sub sections describe the structure of specific bid, offer, trade, and schedule types. Typically each bid/offer/trade/schedule will have a set of properties that along with the type of bid/offer/trade/schedule makes it unique (see section 2.3.3). Typically these would include:

* Bid, offer, schedule or trade type
* Resource
* Ancillary Service type

When submitting a bid/offer/schedule/trade using create or update, all properties for the bid must be specified. When performing a get or cancel request, only those parameters that uniquely identify the bid/offer/trade/schedule must be specified through the use of an mRID. The following diagram shows information commonly maintained for each type of bid/offer/trade/schedule. The status for a given bid/offer/trade/schedule may be SUBMITTED, PENDING, ACCEPTED, ERRORS or CANCELED. If the status is ERRORS, there may be one or more error stings identified.

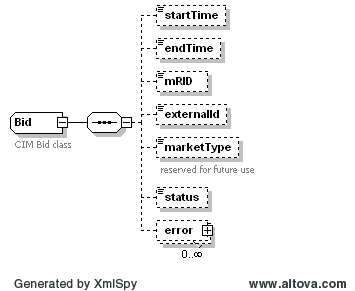


Figure 17 – Bid/Offer/Trade/Schedule Common Elements

The mRID is not supplied for the initial submission of a bid, but must be supplied for cancellations to previously submitted bids. The marketType is currently optional and is reserved for future use. When an mRID is specified for a ‘get’ or ‘cancel’ request, it is supplied using a message request/ID tag.

The ‘externalId’ may be populated by the QSE with an identifier of their choice. If supplied upon submission, the identifier will then be used in conjunction with notifications of acceptance or rejection due to errors.

### Three-Part Supply Offer (TPO)

The following diagram defines the structure of a Three-Part Supply Offer that could be included within a BidSet, using the ThreePartOffer tag. This is one of the more complex structures as it involves startup costs, minimum generation costs, and price curves. The PriceCurve is defined by CIM directly as a subclass of CurveSchedule.

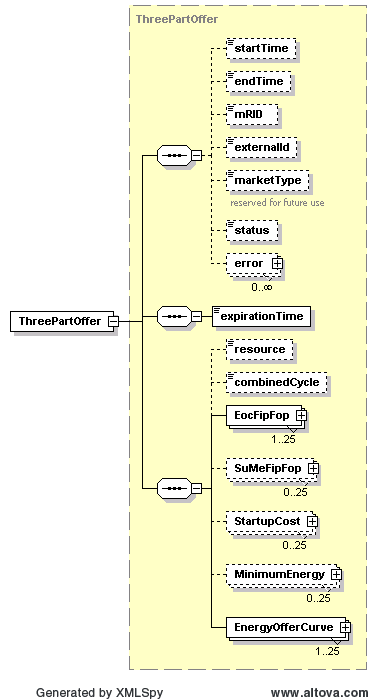


Figure 18 – ThreePartOffer Structure

The error tag is used to return one or more errors that may be the consequence of the failure of business or syntax validation rules for each type of bid/offer/trade/schedule. The EnergyOfferCurve element is based upon a PriceCurve type. Within PriceCurve, the values of ‘xvalue’ are used to identify the quantity in MW, and the values of ‘y1value’ are used to define the price in $/MWh. The number of points in each PriceCurve is limited to 10. Where more than one PriceCurve is supplied for a bid/offer/trade, the start and end times must not overlap. The incExcFlag is to indicate whether the capacity of the ThreePartOffer is inclusive of the capacity bid in the AS market. The reason code is typically used for adjustments. The structure of a PriceCurve type is detailed in the following diagram:

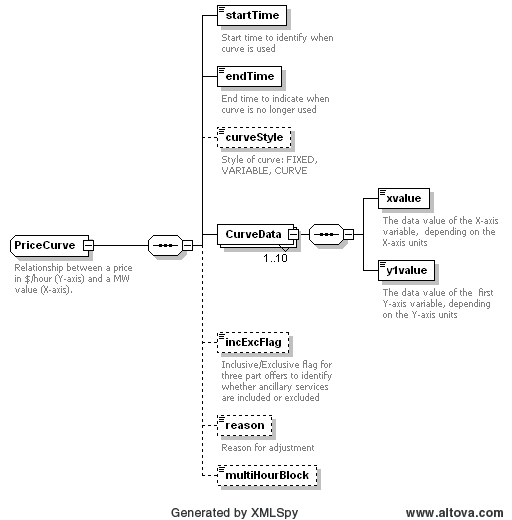


Figure 19 - PriceCurve Structure

The following diagrams describe the structures used for FIP/FOP, startup costs and minimum generation costs within the ThreePartOffer message. Note that if more than one StartupCost or MinimumEnergy is provided, there must not be overlapping start and end times. That is to say that more than one StartupCost or more than one MinimumEnergy can be specified within the day, provided that the time intervals for each instance are distinct.

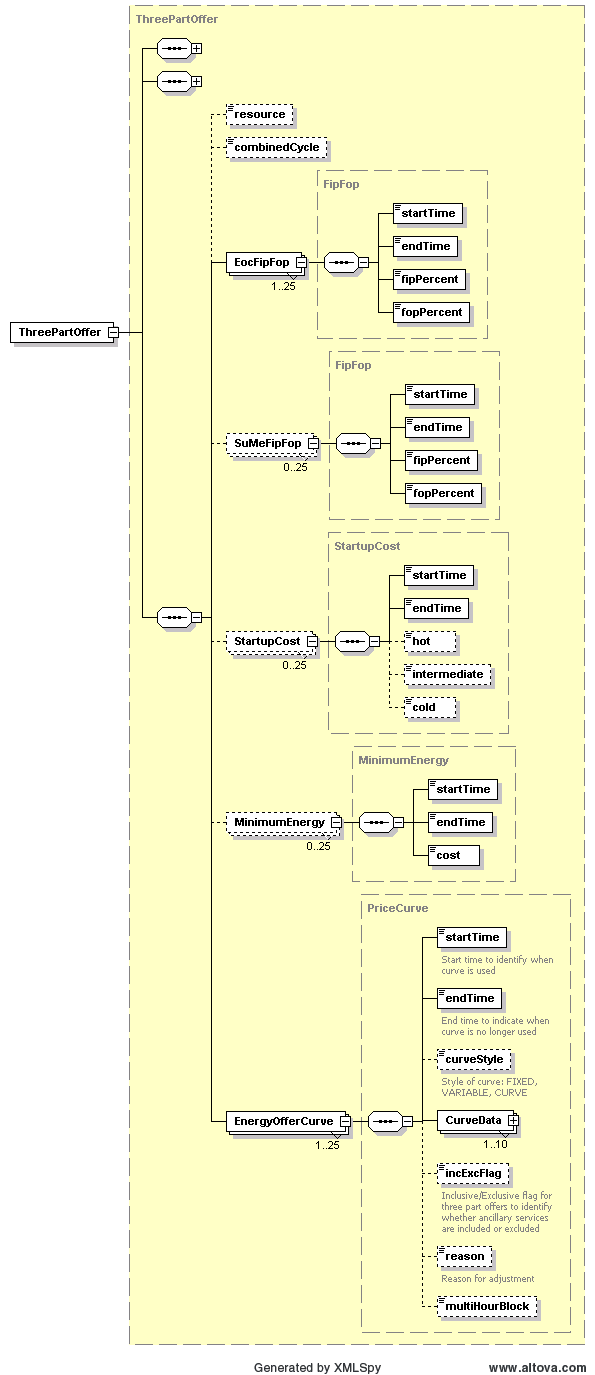


Figure 20 - EocFipFop, SuMeFipFop, StartupCost, and MinimumEnergy Structures

On submission, the following table describes the items used for a ThreePartOffer. Please note that fields that are used as ‘keys’ to identify a bid/offer/trade/schedule (see section 2.3.3) are identified by ‘K’. Required fields are identified by ‘Y’, option fields are identified by ‘N’. There are some exception cases where identified key fields are identified as not being required (e.g. expirationTime), and are notated as ‘K, N’.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| startTime | N | dateTime | Start time for bid | Valid start hour boundary for trade date |
| endTime | N | dateTime | End time for bid | Valid end hour boundary for trade date |
| externalId | N | string | External ID | QSE supplied |
| resource | K | string | Resource | Valid resource name |
| combinedCycle | N | string | Combined cycle to which resource is associated | NULL or Valid plant name if unit is part of a combined cycle |
| expirationTime | Y | dateTime | Time of offer expiration | Valid time within trade date adjustment period |
| EocFipFop/startTime | Y | dateTime | Start time for Energy Offer Curve | Valid hour boundary within the trade date |
| EocFipFop/endTime | Y | dateTime | End time for Energy Offer Curve | Valid hour boundary within the trade date |
| EocFipFop/fipPercent | Y | decimal | Fuel index price percent for Energy Offer Curve | >= 0, <= 100 |
| EocFipFop/fopPercent | Y | decimal | Fuel oil price percent for Energy Offer Curve | >= 0, <= 100 |
| SuMeFipFop/startTime | N | dateTime | Start time for Startup Minimum Energy | Valid hour boundary within the trade date |
| SuMeFipFop/endTime | N | dateTime | End time for Startup Minimum Energy | Valid hour boundary within the trade date |
| SuMeFipFop/fipPercent | N | decimal | Fuel index price percent for Startup Minimum Energy | >= 0, <= 100 |
| SuMeFipFop/fopPercent | N | decimal | Fuel oil price percent for Startup Minimum Energy | >= 0, <= 100 |
| Note: SuMeFipFop element is required when StartUpCost and MinimumEnergy are present. | | | | |
| StartupCost/startTime | N | dateTime | Start time for startup costs | Valid hour boundary within the trade date |
| StartupCost/endTime | N | dateTime | End time for startup costs | Valid hour boundary within the trade date |
| StartupCost/hot | N | decimal | Hot startup cost | >= 0 |
| StartupCost/intermediate | N | decimal | Intermediate startup cost | >= 0 |
| StartupCost/cold | N | decimal | Cold startup cost | >= 0 |
| MinimumEnergy/startTime | N | dateTime | Start time for minimum energy costs | Valid hour boundary within the trade date |
| MinimumEnergy/endTime | N | dateTime | End time for energy costs | Valid hour boundary within the trade date |
| MinimumEnergy/cost | N | decimal | Minimum energy cost | >= 0 |
| EnergyOfferCurve/startTime | Y | dateTime | Start time for curve | Valid hour boundary within the trade date |
| EnergyOfferCurve/endTime | Y | dateTime | End time for curve | Valid hour boundary within the trade date |
| EnergyOfferCurve/curveStyle | N | string | Not used |  |
| EnergyOfferCurve/CurveData/xvalue | Y | float | Megawatts | Quantity in MW |
| EnergyOfferCurve/CurveData/y1value | Y | float | $/MWh | Price in $/MWh |
| EnergyOfferCurve/incExcFlag | Y | string | Identifies whether or not an AS offer is linked to this offer. | INC or EXC. Default should be INC, where it is linked to an ASOffer |
| EnergyOfferCurve/reason | Y | string | Reason for adjustment | OUT or FUEL or DSCM |

Figure 21 - ThreePartOffer Requirements

There should minimally be a StartupCost, MinimumEnergy or PriceCurve specified for a Three Part Offer.

The following is an XML example of a Three Part Offer:

<BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotTransactions.xsd">

<tradingDate>2008-01-01</tradingDate>

<ThreePartOffer>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<marketType>DAM</marketType>

<expirationTime>2008-01-02T00:00:00-05:00</expirationTime>

<resource>Resource1</resource>

<EocFipFop>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<fipPercent>20</fipPercent>

<fopPercent>80</fopPercent>

</EocFipFop>

<SuMeFipFop>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<fipPercent>20</fipPercent>

<fopPercent>80</fopPercent>

</SuMeFipFop>

<StartupCost>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<hot>7</hot>

<intermediate>5</intermediate>

<cold>3</cold>

</StartupCost>

<MinimumEnergy>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<cost>5</cost>

</MinimumEnergy>

<EnergyOfferCurve>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<CurveData>

<xvalue>3.1</xvalue>

<y1value>3.1</y1value>

</CurveData>

<incExcFlag>INC</incExcFlag>

<reason>OUT</reason>

</EnergyOfferCurve>

</ThreePartOffer>

</BidSet>

And the corresponding response:

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2008-06-15</ns1:tradingDate>

<ns1:ThreePartOffer>

<ns1:mRID>QSE1.20080615.TPO.Resource1</ns1:mRID>

<ns1:externalId/>

<ns1:status>ACCEPTED</ns1:status>

<ns1:error>

<ns1:severity>INFORMATIVE</ns1:severity>

<ns1:text>Successfully Cancelled the Output Schedule</ns1:text>

</ns1:error>

<ns1:error>

<ns1:severity>INFORMATIVE</ns1:severity>

<ns1:text>Successfully processed the ERCOT Three Part Offer.</ns1:text>

</ns1:error>

</ns1:ThreePartOffer>

</ns1:BidSet>

### Self-Arranged Ancillary Service Quantities (SAA)

Self-Arranged Ancillary Service Quantities describe a schedule of ancillary services that a QSE has arranged. The following diagram describes the structure of a schedule for self arranged ancillary services:

Graphical user interface, diagram, application

Description automatically generated

Figure 22 - SelfArrangedAS Structure

Within the SelfArrangedAS is a CapacitySchedule of SelfASCapacitySchedule type. SelfASCapacitySchedule are defined using SelfASTmSchedules types, as shown in the following diagram. The values of ‘value1’ within the SelfASCapacitySchedule are used to identify megawatts. The corresponding rrs\_values are populated when RRS ASType is used in SelfArrangedAS submittals. The index of the TmPoint values should correspond to the index of the rrs\_values.

Please note that total values specified in the rrs\_values should be equal to or less than the AS obligation MW:

*rrsgn\_value + rrsld\_value + rrsnc\_value <= AS obligation MW*

MMS application rejects the SelfArrangedAS submittals if the above condition is not met.

The AS Obligation, which is a number produced by ERCOT, is the amount of AS per AS type that a QSE is obligated to provide.  QSEs may choose to self arrange all, a portion, or none of it.  The portion QSEs don't self arrange is procured by the DAM.

* To self arrange all of your RRS obligation, then enter:
  + *rrsgn\_value + rrsld\_value + and rrsnc\_value = RRS AS Obligation.*
* To self arrange a portion of your RRS obligation, then enter:
  + *rrsgn\_value + rrsld\_value + and rrsnc\_value < RRS AS Obligation.*
* To self arrange none of RRS obligation, then enter
  + *zeroes for rrsgn\_value, rrsld\_value, and rrsnc\_value.*

Keep in mind when ASType of RRS is used, TmPoint/value1 is ignored, and the corresponding RRS subtypes values are entered in the rrsgn\_value, rrsld\_value, and rrsnc\_value elements.

|  |
| --- |
| ***[NPRR863: Replace section above with the following upon system implementation]***  Please note that total values specified in the rrs\_values should be equal to or less than the AS obligation MW:  *rrsuf\_value + rrspf\_value + rrsff\_value <= AS obligation MW*  MMS application rejects the SelfArrangedAS submittals if the above condition is not met.    The AS Obligation, which is a number produced by ERCOT, is the amount of AS per AS type that a QSE is obligated to provide.  QSEs may choose to self arrange all, a portion, or none of it.  The portion QSEs don't self arrange is procured by the DAM.     * To self arrange all of your RRS obligation, then enter:   + *rrsuf\_value + rrspf\_value + and rrsff\_value = RRS AS Obligation.* * To self arrange a portion of your RRS obligation, then enter:   + *rrsuf\_value + rrspf\_value + and rrsff\_value < RRS AS Obligation.* * To self arrange none of RRS obligation, then enter   + *zeroes for rrsuf\_value, rrspf\_value, and rrsff\_value.*     Keep in mind when ASType of RRS is used, TmPoint/value1 is ignored, and the corresponding RRS subtypes values are entered in the rrsuf\_value, rrspf\_value, and rrsff\_value elements. |

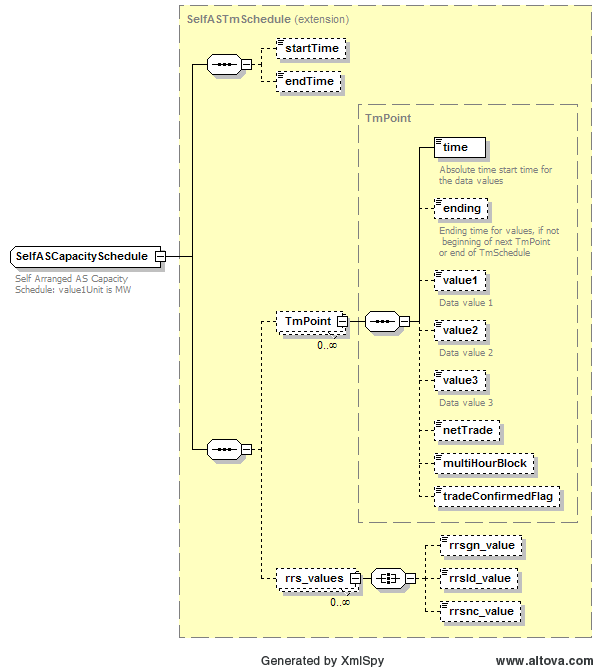


Figure 23 - SelfASCapacitySchedule Structure

|  |
| --- |
| Graphical user interface  Description automatically generated***[NPRR863: Replace Figure 23 above with the following upon system implementation]*** |

On submission, the following table describes the items used for a SelfArrangedAS:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| startTime | K | dateTime | Start time for bid | Valid start hour boundary for trade date |
| endTime | K | dateTime | End time for bid | Valid end hour boundary for trade date |
| externalId | N | string | External ID | QSE supplied |
| asType | K | string | Ancillary service type | See Figure 11 |
| CapacitySchedule/startTime | N | dateTime | not used | not used |
| CapacitySchedule/endTime | N | dateTime | not used | not used |
| CapacitySchedule/  TmPoint/time | Y | dateTime | Absolute time for start of interval | Valid time within trading date |
| CapacitySchedule/  TmPoint/ending | N | dateTime | Absolute time for end of interval | Valid time within trading date |
| CapacitySchedule/  TmPoint/value1 | Y | float | Megawatts | Less than or equal to AS Obligation.  For ASType of Non-Spin, value1 is Non-Spin provided from SCED dispatchable Resources. |
| CapacitySchedule/  TmPoint/nspnm\_value | N |  |  | For ASType of Non-Spin, nspnm\_value is Non-Spin provided from non-SCED dispatchable Resources. |
| CapacitySchedule/  rrs\_values/rrsgn\_value | Y if ASType is RRS | float | RRS Gen value (MW) | Total to be less than or equal to the AS Obligations |
| CapacitySchedule/  rrs\_values/rrsld\_value | float | RRS Load value (MW) |
| CapacitySchedule/  rrs\_values/rrsnc\_value | float | RRS Non-cont load value (MW) |

Figure 24 - SelfArrangedAS Requirements

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***[NPRR863: Replace Figure 24 above with the following upon system implementation]***   |  |  |  |  |  | | --- | --- | --- | --- | --- | | *Element* | *Req?* | *Datatype* | *Description* | *Values* | | startTime | K | dateTime | Start time for bid | Valid start hour boundary for trade date | | endTime | K | dateTime | End time for bid | Valid end hour boundary for trade date | | externalId | N | string | External ID | QSE supplied | | asType | K | string | Ancillary service type | See Figure 11 | | CapacitySchedule/startTime | N | dateTime | not used | not used | | CapacitySchedule/endTime | N | dateTime | not used | not used | | CapacitySchedule/  TmPoint/time | Y | dateTime | Absolute time for start of interval | Valid time within trading date | | CapacitySchedule/  TmPoint/ending | N | dateTime | Absolute time for end of interval | Valid time within trading date | | CapacitySchedule/  TmPoint/value1 | Y | float | Megawatts | Less than or equal to AS Obligation.  For ASType of Non-Spin, value1 is Non-Spin provided from SCED dispatchable Resources. | | CapacitySchedule/  TmPoint/nspnm\_value | N |  |  | For ASType of Non-Spin, nspnm\_value is Non-Spin provided from non-SCED dispatchable Resources. | | CapacitySchedule/  rrs\_values/rrspf\_value | Y if ASType is RRS | float | RRS PF value (MW) | Total to be less than or equal to the AS Obligations | | CapacitySchedule/  rrs\_values/rrsff\_value | float | RRS FF value (MW) | | CapacitySchedule/  rrs\_values/rrsuf\_value | float | RRS UF value  (MW) | |

The following XML is an example of a Self-Arranged Ancillary Service Quantities, showing the use of a CapacitySchedule:

<BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews">  
 <tradingDate>2022-01-12</tradingDate>  
 <SelfArrangedAS>  
 <startTime>2022-01-12T00:00:00-06:00</startTime>  
 <endTime>2022-01-12T01:00:00-06:00</endTime>  
 <externalId>246112203</externalId>  
 <asType>Non-Spin</asType>  
 <CapacitySchedule>  
 <startTime>2022-01-12T00:00:00-06:00</startTime>  
 <endTime>2022-01-12T01:00:00-06:00</endTime>  
 <TmPoint>  
 <time>2022-01-12T00:00:00-06:00</time>  
 <ending>2022-01-12T01:00:00-06:00</ending>  
 <value1>8</value1>  
 <nspnm\_value>0</nspnm\_value>  
 </TmPoint>  
 </CapacitySchedule>  
 </SelfArrangedAS>  
 <SelfArrangedAS>  
 <startTime>2022-01-12T00:00:00-06:00</startTime>  
 <endTime>2022-01-12T01:00:00-06:00</endTime>  
 <externalId>246112283</externalId>  
 <asType>RRS</asType>  
 <CapacitySchedule>  
 <startTime>2022-01-12T00:00:00-06:00</startTime>  
 <endTime>2022-01-12T01:00:00-06:00</endTime>  
 <TmPoint>  
 <time>2022-01-12T00:00:00-06:00</time>  
 <ending>2022-01-12T01:00:00-06:00</ending>  
 </TmPoint>  
 <rrs\_values>  
 <rrsgn\_value>8</rrsgn\_value>  
 <rrsld\_value>0</rrsld\_value>  
 <rrsnc\_value>11</rrsnc\_value>  
 </rrs\_values>  
 </CapacitySchedule>  
 </SelfArrangedAS>  
 </BidSet>

|  |
| --- |
| ***[NPRR863: Replace example above with the following upon system implementation]***  <BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews">  <tradingDate>2022-01-12</tradingDate>  <SelfArrangedAS>  <startTime>2022-01-12T00:00:00-06:00</startTime>  <endTime>2022-01-12T01:00:00-06:00</endTime>  <externalId>246112203</externalId>  <asType>Non-Spin</asType>  <CapacitySchedule>  <startTime>2022-01-12T00:00:00-06:00</startTime>  <endTime>2022-01-12T01:00:00-06:00</endTime>  <TmPoint>  <time>2022-01-12T00:00:00-06:00</time>  <ending>2022-01-12T01:00:00-06:00</ending>  <value1>8</value1>  <nspnm\_value>0</nspnm\_value>  </TmPoint>  </CapacitySchedule>  </SelfArrangedAS>  <SelfArrangedAS>  <startTime>2022-01-12T00:00:00-06:00</startTime>  <endTime>2022-01-12T01:00:00-06:00</endTime>  <externalId>246112283</externalId>  <asType>RRS</asType>  <CapacitySchedule>  <startTime>2022-01-12T00:00:00-06:00</startTime>  <endTime>2022-01-12T01:00:00-06:00</endTime>  <TmPoint>  <time>2022-01-12T00:00:00-06:00</time>  <ending>2022-01-12T01:00:00-06:00</ending>  </TmPoint>  <rrs\_values>  <rrspf\_value>100.1</rrspf\_value>  <rrsff\_value>50.1</rrsff\_value>  <rrsuf\_value>400.1</rrsuf\_value>  </rrs\_values>  </CapacitySchedule>  </SelfArrangedAS>  </BidSet> |

And the corresponding response:

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2022-01-12</ns1:tradingDate>

<ns1:submitTime>2022-01-09T11:53:36.736-06:00</ns1:submitTime>

<ns1:SelfArrangedAS>

<ns1:mRID>QSAMP.20220112.SAA.Non-Spin</ns1:mRID>

<ns1:externalId>246112203</ns1:externalId>

<ns1:status>SUBMITTED</ns1:status>

</ns1:SelfArrangedAS>

<ns1:SelfArrangedAS>

<ns1:mRID>QSAMP.20220112.SAA.RRS</ns1:mRID>

<ns1:externalId>246112283</ns1:externalId>

<ns1:status>SUBMITTED</ns1:status>

</ns1:SelfArrangedAS>

</ns1:BidSet>

### Incremental and Decremental Energy Offer Curves (IDO)

The following diagram describes the structure of Incremental and Decremental Energy Offer Curves. This shows the details of the FipFop and PriceCurve structures that are contained within:

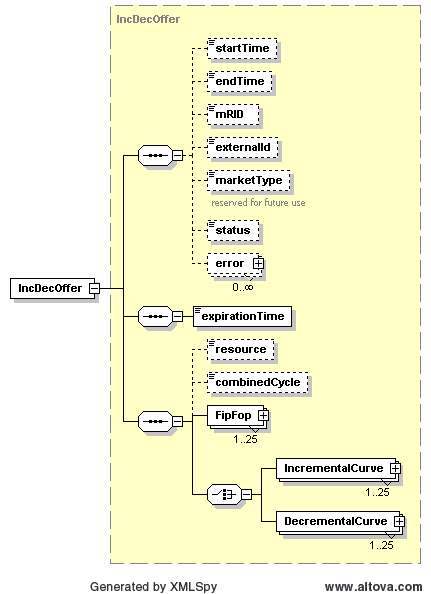


Figure 25 - IncDecOffer Structure

Both the IncrementalCurve and DecrementalCurve use the PriceCurve structure as described previously in this document, but only one (either an IncrementalCurve or a DecrementalCurve) can be specified per each IncDecOffer. On submission, the following table describes the items used for an IncDecOffer:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Element* | | *Req?* | | *Datatype* | | *Description* | *Values* | |
| startTime | | N | | dateTime | | Start time for bid | Valid start hour boundary for trade date | |
| endTime | | N | | dateTime | | End time for bid | Valid end hour boundary for trade date | |
| externalId | N | | string | | External ID | | | QSE supplied |
| resource | | K | | string | | Resource | Valid resource name | |
| combinedCycle | | N | | string | | Combined cycle to which resource is associated | Not required. Value ignored if provided. | |
| expirationTime | | Y | | dateTime | | Offer expiration time | Valid time before or during the trade date | |
| FipFop/fipPercent | | Y | | decimal | | Fuel index price percent | >= 0, <= 100 | |
| FipFop/fopPercent | | Y | | decimal | | Fuel oil price percent | >= 0, <= 100 | |
| IncrementalCurve/startTime | | If INC  offer | | dateTime | | Start time for curve | Valid hour boundary | |
| IncrementalCurve/endTime | | If INC  offer | | dateTime | | End time for curve | Valid hour boundary | |
| IncrementalCurve/curveStyle | | If INC  offer | | string | | Type of curve | CURVE | |
| IncrementalCurve/CurveData/xvalue | | If INC  offer | | float | | Megawatts | Quantity in MW | |
| IncrementalCurve/CurveData/y1value | | If INC  offer | | float | | $/MWh | Incremental price in $/MWh, must be greater than corresponding Decremental curve value | |
| DecrementalCurve/startTime | | If DEC offer | | dateTime | | Start time for curve | Valid hour boundary | |
| DecrementalCurve/endTime | | If DEC offer | | dateTime | | End time for curve | Valid hour boundary | |
| DecrementalCurve/curveStyle | | If DEC offer | | string | | Type of curve | CURVE | |
| DecrementalCurve/CurveData/xvalue | | If DEC offer | | float | | Megawatts | Quantity in MW | |
| DecrementalCurve/CurveData/y1value | | If DEC offer | | float | | $/MWh | Decremental price in $/MWh, must be less than corresponding incremental curve value | |

Figure 26 - IncDecOffer Requirements

The following is an XML example for an IncDecOffer (both Incremental and Decremental curves)

<BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotTransactions.xsd">

<tradingDate>2008-01-01</tradingDate>

<IncDecOffer>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<marketType>DAM</marketType>

<expirationTime>2008-01-02T00:00:00-05:00</expirationTime>

<resource>Resource123</resource>

<FipFop>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<fipPercent>25</fipPercent>

<fopPercent>75</fopPercent>

</FipFop>

<DecrementalCurve>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<curveStyle>FIXED</curveStyle>

<CurveData>

<!--The price value in the pq\_curve element of DEC Offer must be less than the existing INC offer price-->

<xvalue>2.14</xvalue>

<y1value>2.14</y1value>

</CurveData>

<reason>OUT</reason>

</DecrementalCurve>

</IncDecOffer>

</BidSet>

<BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotTransactions.xsd">

<tradingDate>2008-01-01</tradingDate>

<IncDecOffer>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<marketType>DAM</marketType>

<expirationTime>2008-01-02T00:00:00-05:00</expirationTime>

<resource>Resource123</resource>

<FipFop>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<fipPercent>15</fipPercent>

<fopPercent>85</fopPercent>

</FipFop>

<IncrementalCurve>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<curveStyle>FIXED</curveStyle>

<CurveData>

<xvalue>3.1</xvalue>

<y1value>3.1</y1value>

</CurveData>

<reason>OUT</reason>

</IncrementalCurve>

</IncDecOffer>

</BidSet>

And the corresponding response:

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2008-06-15</ns1:tradingDate>

<ns1:IncDecOffer>

<ns1:mRID>AEN.20080615.IDO.Resource1.DEC</ns1:mRID>

<ns1:externalId/>

<ns1:status>ACCEPTED</ns1:status>

<ns1:error>

<ns1:severity>INFORMATIVE</ns1:severity>

<ns1:text>Successfully processed the ERCOT Inc Dec Offer.</ns1:text>

</ns1:error>

</ns1:IncDecOffer>

</ns1:BidSet>

### Ancillary Service Offer (ASO)

An Ancillary Service Offer is used to offer ancillary services from a specified resource. The following diagram describes the structure of an Ancillary Service Offer (ASOffer):

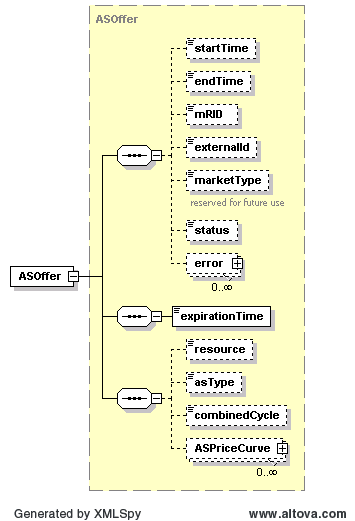
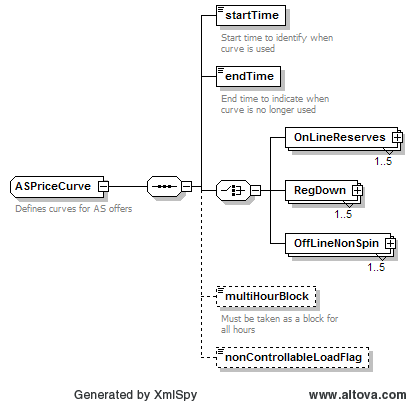


Figure 27 - ASOffer Structure

The ASPriceCurve structure when used for ASOffers uses OnLineReserves (to provide price points for Reg-Up, RRS, and/or Online Non-Spin), RegDown elements (to provide price points for Reg-Down), or OffLineNonSpin elements (for Offline Non-Spin). The choice depends upon the value of asType, where the values provided can be one of REGUP-RRS-ONNS, Reg-Down or Off-Non-Spin. Up to five points can be specified for OnLineReserves, RegDown, or OffLineNonSpin. This structure is described in the following figure:



|  |
| --- |
| ***[NPRR863: Replace image above with the following upon system implementation]***  Diagram  Description automatically generated |

The following diagram provides details of the OnLineReserves, RegDown, and OffLineNonSpin structures:

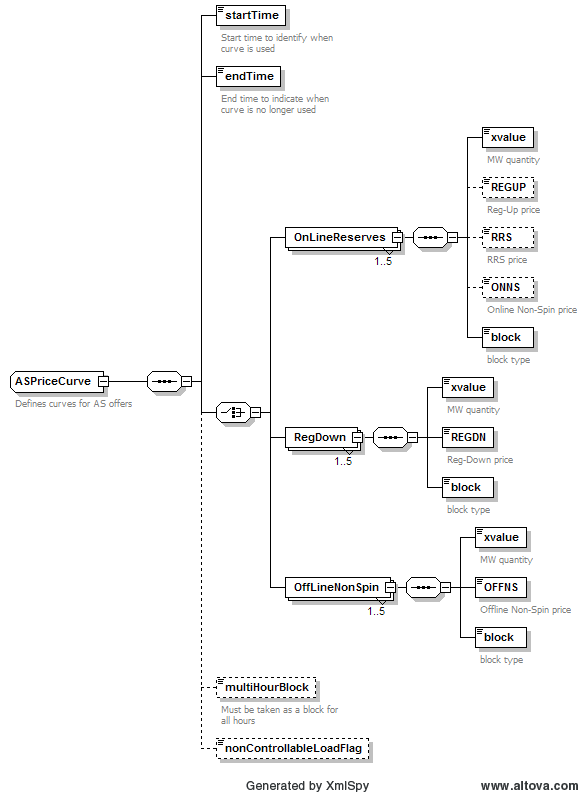


Figure 28 - Price Curves using ASPriceCurve

|  |
| --- |
| ***[NPRR863: Replace image above with the following upon system implementation]***  Diagram  Description automatically generated |

On submission, the following table describes the items used for an ASOffer:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| startTime | K | dateTime | Start time for bid | Valid start hour boundary for trade date |
| endTime | K | dateTime | End time for bid | Valid end hour boundary for trade date |
| externalId | N | string | External ID | QSE supplied |
| expirationTime | Y | dateTime | Time of offer expiration | Valid time before trade date |
| resource | K | string | Resource | Valid resource name |
| asType | K | string | AS key code | See Figure 11 |
| combinedCycle | N | string | Combined cycle | Not required. Value ignored if provided. |
| PriceCurve/startTime | Y | dateTime | Start time for curve | Valid hour boundary |
| PriceCurve/endTime | Y | dateTime | End time for curve | Valid hour boundary |
| ASPriceCurve/OnLineReserves/xvalue  Required if  asType = REGUP-RRS-ONNS | Y | float | Megawatts | Quantity in MW |
| ASPriceCurve/OnLineReserves/REGUP | N | float | $/MWh | Price in $/MWh |
| ASPriceCurve/ OnLineReserves /RRS | N | float | $/MWh | Price in $/MWh |
| ASPriceCurve/ OnLineReserves /ONNS | N | float | $/MWh | Price in $/MWh |
| ASPriceCurve/ OnLineReserves /block  Required if  asType = REGUP-RRS-ONNS | Y | string | Used as an indicator to describe the type of block | FIXED or VARIABLE |
| ASPriceCurve/RegDown/xvalue or ASPriceCurve/OffLineNonSpin/xvalue  Required if  asType = Reg-Down or Off-Non-Spin | Y | float | Megawatts | Quantity in MW |
| ASPriceCurve/RegDown/REGDN  Required if asType=Reg-Down | Y | float | $/MWh | Price in $/MWh |
| ASPriceCurve/OffLineNonSpin/OFFNS  Required if asType=Off-Non-Spin | Y | float | $/MWh | Price in $/MWh |
| ASPriceCurve/RegDown/block or ASPriceCurve/OffLineNonSpin/block  Required if  asType = Reg-Down or Off-Non-Spin | Y | String | Used as an indicator to describe the type of block | FIXED or VARIABLE |
| ASPriceCurve/multiHourBlock | N | Boolean | Indicates if offer must be taken as a block for all hours. | Default is false |
| ASPriceCurve/nonControllableLoadFlag | N | Boolean | Flag indicating Non-Controllable Load | default is false |

Figure 29 - ASOffer Requirements

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***[NPRR863: Replace Figure 29 above with the following upon system implementation]***   |  |  |  |  |  | | --- | --- | --- | --- | --- | | *Element* | *Req?* | *Datatype* | *Description* | *Values* | | startTime | K | dateTime | Start time for bid | Valid start hour boundary for trade date | | endTime | K | dateTime | End time for bid | Valid end hour boundary for trade date | | externalId | N | string | External ID | QSE supplied | | expirationTime | Y | dateTime | Time of offer expiration | Valid time before trade date | | resource | K | string | Resource | Valid resource name | | asType | K | string | AS key code | See Figure 11 | | combinedCycle | N | string | Combined cycle | Not required. Value ignored if provided. | | PriceCurve/startTime | Y | dateTime | Start time for curve | Valid hour boundary | | PriceCurve/endTime | Y | dateTime | End time for curve | Valid hour boundary | | ASPriceCurve/OnLineReserves/xvalue  Required if  asType = REGUP-RRS-ONNS | Y | float | Megawatts | Quantity in MW | | ASPriceCurve/OnLineReserves/REGUP | N | float | $/MWh | Price in $/MWh | | ASPriceCurve/OnLineReserves /RRSPF | N | float | $/MWh | Price in $/MWh | | ASPriceCurve/ OnLineReserves /RRSFF | N | float | $/MWh | Price in $/MWh | | ASPriceCurve/ OnLineReserves /RRSUF | N | float | $/MWh | Price in $/MWh | | ASPriceCurve/ OnLineReserves /ONNS | N | float | $/MWh | Price in $/MWh | | ASPriceCurve/ OnLineReserves /block  Required if  asType = REGUP-RRS-ONNS | Y | string | Used as an indicator to describe the type of block | FIXED or VARIABLE | | ASPriceCurve/RegDown/xvalue or ASPriceCurve/OffLineNonSpin/xvalue  Required if  asType = Reg-Down or Off-Non-Spin | Y | float | Megawatts | Quantity in MW | | ASPriceCurve/RegDown/REGDN  Required if asType=Reg-Down | Y | float | $/MWh | Price in $/MWh | | ASPriceCurve/OffLineNonSpin/OFFNS  Required if asType=Off-Non-Spin | Y | float | $/MWh | Price in $/MWh | | ASPriceCurve/RegDown/block or ASPriceCurve/OffLineNonSpin/block  Required if  asType = Reg-Down or Off-Non-Spin | Y | String | Used as an indicator to describe the type of block | FIXED or VARIABLE | | ASPriceCurve/multiHourBlock | N | Boolean | Indicates if offer must be taken as a block for all hours. | Default is false | |

The following is an XML example for an ASOffer, for the AS types of Reg-Up, RRS, and online Non-Spin:

<ASOffer>

<startTime>2008-01-01T00:00:00-06:00</startTime>

<endTime>2008-01-02T00:00:00-06:00</endTime>

<externalId>MyExternalID12343</externalId>

<expirationTime>2007-12-31T22:00:00-06:00</expirationTime>

<resource>Resource123</resource>

<asType>REGUP-RRS-ONNS</asType>

<ASPriceCurve>

<startTime>2008-01-01T00:00:00-06:00</startTime>

<endTime>2008-01-01T03:00:00-06:00</endTime>

<OnLineReserves>

<xvalue>60</xvalue>

<REGUP>20.00</REGUP>

<RRS>20.00</RRS>

<ONNS>20.00</ONNS>

<block>FIXED</block>

</OnLineReserves>

<multiHourBlock>false</multiHourBlock>

</ASPriceCurve>

<ASPriceCurve>

<startTime>2008-01-01T03:00:00-06:00</startTime>

<endTime>2008-01-02T00:00:00-06:00</endTime>

<OnLineReserves>

<xvalue>80</xvalue>

<REGUP>23.00</REGUP>

<RRS>20.00</RRS>

<ONNS>20.00</ONNS>

<block>FIXED</block>

</OnLineReserves>

<multiHourBlock>false</multiHourBlock>

</ASPriceCurve>

</ASOffer>

|  |
| --- |
| ***[NPRR863: Replace example above with the following upon system implementation]***  <BidSet>  <tradingDate>2021-11-16</tradingDate>  <ASOffer>  <startTime>2021-11-16T00:00:00-06:00</startTime>  <endTime>2021-11-17T00:00:00-06:00</endTime>  <externalId>ERCOTTEST</externalId>  <expirationTime>2021-11-16T23:59:59-06:00</expirationTime>  <resource>RES\_1</resource>  <asType>REGUP-RRS-ONNS</asType>  <ASPriceCurve>  <startTime>2021-11-16T00:00:00.000-06:00</startTime>  <endTime>2021-11-17T00:00:00.000-06:00</endTime>  <OnLineReserves>  <xvalue>11</xvalue>  <REGUP>23.00</REGUP>  <RRSPF>20.00</RRSPF>  <RRSFF>19.00</RRSFF>  <RRSUF>18.00</RRSUF>  <ONNS>1</ONNS>  <block>VARIABLE</block>  </OnLineReserves>  </ASPriceCurve>  </ASOffer> </BidSet> |

The following is another XML example for an ASOffer, where the AS type is only Reg-Down:

<ASOffer>

<startTime>2008-01-01T00:00:00-06:00</startTime>

<endTime>2008-01-02T00:00:00-06:00</endTime>

<externalId>MyExternalID12345</externalId>

<expirationTime>2007-12-31T22:00:00-06:00</expirationTime>

<resource>Resource1</resource>

<asType>Reg-Down</asType>

<ASPriceCurve>

<startTime>2008-01-01T00:00:00-06:00</startTime>

<endTime>2008-01-01T03:00:00-06:00</endTime>

<RegDown>

<xvalue>60</xvalue>

<REGDN>20.00</REGDN>

<block>FIXED</block>

</RegDown>

<multiHourBlock>false</multiHourBlock>

</ASPriceCurve>

<ASPriceCurve>

<startTime>2008-01-01T03:00:00-06:00</startTime>

<endTime>2008-01-02T00:00:00-06:00</endTime>

<RegDown>

<xvalue>80</xvalue>

<REGDN>23.00</REGDN>

<block>FIXED</block>

</RegDown>

<multiHourBlock>false</multiHourBlock>

</ASPriceCurve>

</ASOffer>

The following is another XML example for an ASOffer, where the AS type is Off-Non-Spin:

<ASOffer>

<startTime>2008-01-01T00:00:00-06:00</startTime>

<endTime>2008-01-02T00:00:00-06:00</endTime>

<externalId>MyExternalID12341</externalId>

<expirationTime>2007-12-31T22:00:00-06:00</expirationTime>

<resource>Resource1</resource>

<asType>Off-Non-Spin</asType>

<ASPriceCurve>

<startTime>2008-01-01T00:00:00-06:00</startTime>

<endTime>2008-01-01T03:00:00-06:00</endTime>

<OffLineNonSpin>

<xvalue>60</xvalue>

<OFFNS>20.00</OFFNS>

<block>VARIABLE</block>

</OffLineNonSpin >

<multiHourBlock>true</multiHourBlock>

</ASPriceCurve>

<ASPriceCurve>

<startTime>2008-01-01T03:00:00-06:00</startTime>

<endTime>2008-01-02T00:00:00-06:00</endTime>

<OffLineNonSpin >

<xvalue>80</xvalue>

<OFFNS>23.00</OFFNS>

<block>VARIABLE</block>

</OffLineNonSpin >

<multiHourBlock>true</multiHourBlock>

</ASPriceCurve>

</ASOffer>

And the corresponding response:

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2008-06-14</ns1:tradingDate>

<ns1:ASOffer>

<ns1:mRID>AEN.20080614.ASO. Resource1.Off-Non-Spin</ns1:mRID>

<ns1:externalId/>

<ns1:status>ACCEPTED</ns1:status>

<ns1:error>

<ns1:severity>INFORMATIVE</ns1:severity>

<ns1:text>Successfully processed the ERCOT As Offer.</ns1:text>

</ns1:error>

</ns1:ASOffer>

</ns1:BidSet>

### Ancillary Service Trade (AST)

An Ancillary Service Trade is used to describe a scheduled trade of ancillary services between a buyer and a seller. The following diagram describes the structure of an Ancillary Service Trade (ASTrade):

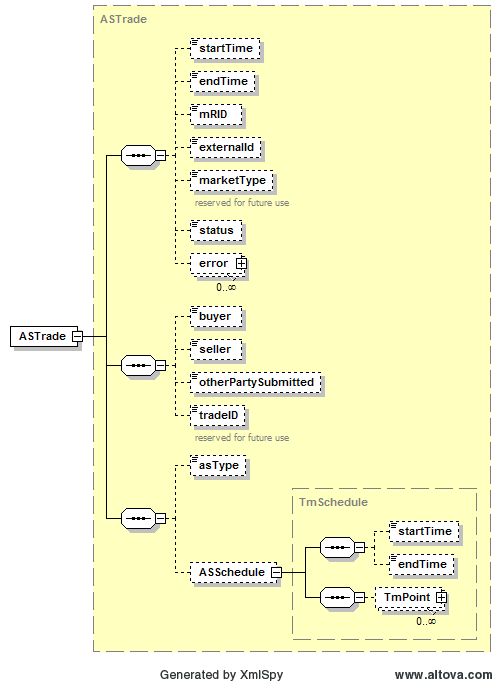


Figure 30 - ASTrade Structure

The ASSchedule structure is based upon a TmSchedule as described previously in this document. The submission of an ASTrade (or generally any trade) requires a matching trade to be submitted by the counter party before the close of the market. On submission, the following table describes the items used for an ASTrade:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| startTime | K | dateTime | Start time for bid | Valid start hour boundary for trade date |
| endTime | K | dateTime | End time for bid | Valid end hour boundary for trade date |
| externalId | N | string | External ID | QSE supplied |
| asType | K | string | Ancillary service type | See Figure 11 |
| buyer | K | string | Trade buyer | Valid QSE |
| seller | K | string | Trade seller | Valid QSE |
| otherPartySubmitted | N | Boolean | For querying purposes only:  True if trade was entered by other party | true (default)  or false |
| tradeID | N | string | reserved for future use |  |
| ASSchedule/startTime | N | dateTime | not used | not used |
| ASSchedule/endTime | N | dateTime | not used | not used |
| ASSchedule/  TmPoint/time | Y | dateTime | Absolute time for beginning of interval | Valid time within the trading date |
| ASSchedule/  TmPoint/ending | N | dateTime | Absolute time for end of interval | Valid time within the trading date |
| ASSchedule/  TmPoint/value1 | Y | float | Megawatts | >= 0 |

Figure 31 - ASTrade Requirements

The following is an XML example for an ASTrade:

<BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:ns2="http://www.ercot.com/schema/2007-06/nodal/ews">  
 <tradingDate>2022-01-12</tradingDate>  
 <ASTrade>  
 <startTime>2022-01-12T00:00:00-06:00</startTime>  
 <endTime>2022-01-12T08:00:00-06:00</endTime>  
 <externalId>123456</externalId>  
 <buyer>QSAMP2</buyer>  
 <seller>QSAMP1</seller>  
 <asType>Non-Spin</asType>  
 <ASSchedule>  
 <TmPoint>  
 <time>2022-01-12T00:00:00-06:00</time>  
 <ending>2022-01-12T06:00:00-06:00</ending>  
 <value1>38.0</value1>  
 </TmPoint>  
 <TmPoint>  
 <time>2022-01-12T06:00:00-06:00</time>  
 <ending>2022-01-12T08:00:00-06:00</ending>  
 <value1>35.0</value1>  
 </TmPoint>  
 </ASSchedule>  
 </ASTrade>  
 <ASTrade>  
 <startTime>2022-01-12T00:00:00-06:00</startTime>  
 <endTime>2022-01-12T03:00:00-06:00</endTime>  
 <externalId>123457</externalId>  
 <buyer>QSAMP3</buyer>  
 <seller>QSAMP1</seller>  
 <asType>NSPNM</asType>  
 <ASSchedule>  
 <TmPoint>  
 <time>2022-01-12T00:00:00-06:00</time>  
 <ending>2022-01-12T03:00:00-06:00</ending>  
 <value1>41.0</value1>  
 </TmPoint>  
 </ASSchedule>  
 </ASTrade>

</BidSet>

|  |
| --- |
| ***[NPRR863: Replace example above with the following upon system implementation]***  <BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:ns2="http://www.ercot.com/schema/2007-06/nodal/ews">  <tradingDate>2022-01-12</tradingDate>  <ASTrade>  <startTime>2022-01-12T00:00:00-06:00</startTime>  <endTime>2022-01-12T08:00:00-06:00</endTime>  <externalId>123456</externalId>  <buyer>QSAMP2</buyer>  <seller>QSAMP1</seller>  <asType>Non-Spin</asType>  <ASSchedule>  <TmPoint>  <time>2022-01-12T00:00:00-06:00</time>  <ending>2022-01-12T06:00:00-06:00</ending>  <value1>38.0</value1>  </TmPoint>  <TmPoint>  <time>2022-01-12T06:00:00-06:00</time>  <ending>2022-01-12T08:00:00-06:00</ending>  <value1>35.0</value1>  </TmPoint>  </ASSchedule>  </ASTrade>  <ASTrade>  <startTime>2022-01-12T00:00:00-06:00</startTime>  <endTime>2022-01-12T03:00:00-06:00</endTime>  <externalId>123457</externalId>  <buyer>QSAMP3</buyer>  <seller>QSAMP1</seller>  <asType>NSPNM</asType>  <ASSchedule>  <TmPoint>  <time>2022-01-12T00:00:00-06:00</time>  <ending>2022-01-12T03:00:00-06:00</ending>  <value1>41.0</value1>  </TmPoint>  </ASSchedule>  </ASTrade>  <ASTrade>  <startTime>2022-01-12T00:00:00-06:00</startTime>  <endTime>2022-01-12T01:00:00-06:00</endTime>  <buyer>QSAMP1</buyer>  <seller>QSAMP2</seller>  <otherPartySubmitted>false</otherPartySubmitted>  <asType>RRSPF</asType>  <ASSchedule>  <TmPoint>  <time>2021-11-02T00:00:00.000-05:00</time>  <ending>2021-11-02T01:00:00.000-05:00</ending>  <value1>22</value1>  </TmPoint>  </ASSchedule>  </ASTrade> </BidSet> |

And the corresponding response:

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2022-01-12</ns1:tradingDate>

<ns1:submitTime>2022-01-10T15:33:03.456-06:00</ns1:submitTime>

<ns1:ASTrade>

<ns1:mRID>QSAMP1.20220112.AST.Non-Spin.QSAMP2.QSAMP1</ns1:mRID>

<ns1:status>SUBMITTED</ns1:status>

</ns1:ASTrade>

<ns1:ASTrade>

<ns1:mRID>QSAMP1.20220112.AST.NSPNM.QSAMP3.QSAMP1</ns1:mRID>

<ns1:status>SUBMITTED</ns1:status>

</ns1:ASTrade>

</ns1:BidSet>

|  |
| --- |
| ***[NPRR863: Replace example above with the following upon system implementation]***  <ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">  <ns1:tradingDate>2022-01-12</ns1:tradingDate>  <ns1:submitTime>2022-01-10T15:33:03.456-06:00</ns1:submitTime>  <ns1:ASTrade>  <ns1:mRID>QSAMP1.20220112.AST.Non-Spin.QSAMP2.QSAMP1</ns1:mRID>  <ns1:status>SUBMITTED</ns1:status>  </ns1:ASTrade>  <ns1:ASTrade>  <ns1:mRID>QSAMP1.20220112.AST.NSPNM.QSAMP3.QSAMP1</ns1:mRID>  <ns1:status>SUBMITTED</ns1:status>  </ns1:ASTrade>  <ns1:mRID>QSAMP1.20220112.AST.RRSPF.QSAMP1.QSAMP2</ns1:mRID>  <ns1:status>SUBMITTED</ns1:status>  </ns1:ASTrade>  </ns1:BidSet> |

### Availability Plan (AVP)

The availability plan is an hourly representation of availability of RMR or Synchronous Condenser units or an hourly representation of the capability of Black Start Resources submitted to ERCOT by QSEs representing RMR Units, Synchronous Condenser Units or Black Start Resources. The following diagram describes the structure of an Availability Plan (AVP):

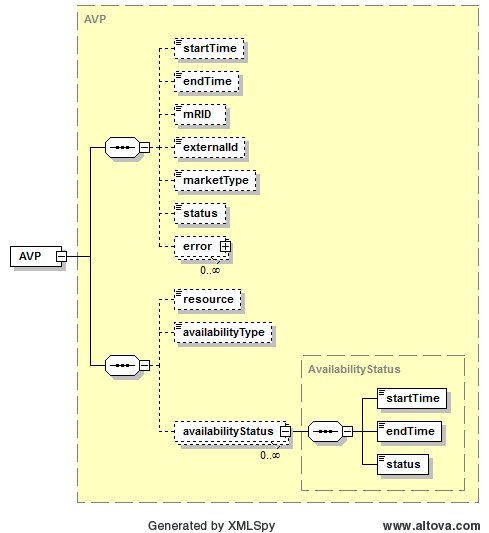


Figure 32 - AVP Structure

The AVP submission contains the resource name and the availabilityType (RMR, SYNCCOND and BLACKSTART) and the availabilityStatus of the resource (A for Available and U for Unavailable). If more than one availabilityStatus block is provided, start and end times must not overlap.

|  |
| --- |
| ***[NPRR1120: Replace section above with the following upon system implementation]***  The AVP submission contains the resource name and the availabilityType (RMR, SYNCCOND, BLACKSTART and FFSS) and the availabilityStatus of the resource (A for Available and U for Unavailable). If more than one availabilityStatus block is provided, start and end times must not overlap. |

AVP submissions can be cancelled. A resubmission of an AVP overwrites the previous submission. On submission, the following table describes the items used for a AVP:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| startTime | N | dateTime | Start time for AVP | Valid start hour boundary for trade date |
| endTime | N | dateTime | End time for AVP | Valid end hour boundary for trade date |
| externalId | N | string | External ID | QSE supplied |
| resource | K | string | Resource | Valid resource name |
| availabilityType | K | string | Resource Availability Type | RMR, SYNCCOND or BLACKSTART |
| availabilityStatus/startTime | Y | dateTime | Start time for status | Valid hour boundary |
| availabilityStatus/endTime | Y | dateTime | End time for status | Valid hour boundary |
| availabilityStatus/status | Y | dateTime | Resource Availability Status | A (Available) U (Unavailable) |

Figure 33 - AVP Requirements

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***[NPRR1120: Replace table above with the following upon system implementation]***   |  |  |  |  |  | | --- | --- | --- | --- | --- | | *Element* | *Req?* | *Datatype* | *Description* | *Values* | | startTime | N | dateTime | Start time for AVP | Valid start hour boundary for trade date | | endTime | N | dateTime | End time for AVP | Valid end hour boundary for trade date | | externalId | N | string | External ID | QSE supplied | | resource | K | string | Resource | Valid resource name | | availabilityType | K | string | Resource Availability Type | RMR, SYNCCOND BLACKSTART  FFSS | | availabilityStatus/startTime | Y | dateTime | Start time for status | Valid hour boundary | | availabilityStatus/endTime | Y | dateTime | End time for status | Valid hour boundary | | availabilityStatus/status | Y | dateTime | Resource Availability Status | A (Available) U (Unavailable) | |

The following is an XML example for an AVP:

<BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews">

<tradingDate>2012-11-08</tradingDate>

<AVP xmlns="http://www.ercot.com/schema/2007-06/nodal/ews"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews file:/C:/eint/tibco/EWS/Schemas/ErcotTransactions.xsd">

<startTime>2012-11-08T00:00:00-05:00</startTime>

<endTime>2012-11-09T00:00:00-05:00</endTime>

<resource>RESOURCE1</resource>

<availabilityType>BLACKSTART</availabilityType>

<availabilityStatus>

<startTime>2012-11-08T00:00:00-05:00</startTime>

<endTime>2012-11-08T01:00:00-05:00</endTime>

<status>A</status>

</availabilityStatus>

<availabilityStatus>

<startTime>2012-11-08T01:00:00-05:00</startTime>

<endTime>2012-11-09T00:00:00-05:00</endTime>

<status>U</status>

</availabilityStatus>

</AVP>

</BidSet>

And the corresponding response:

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2012-11-08</ns1:tradingDate>

<ns1:submitTime>2012-11-06T10:05:54.455-06:00</ns1:submitTime>

<ns1:AVP>

<ns1:mRID>QSE1.20121108.AVP.RESOURCE1.BLACKSTART</ns1:mRID>

<ns1:externalId/>

<ns1:status>SUBMITTED</ns1:status>

</ns1:AVP>

</ns1:BidSet>

|  |
| --- |
| ***[NPRR1120: Replace examples above with the following upon system implementation]***  The following is an XML example for an AVP:  <BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews">  <tradingDate>2012-11-08</tradingDate>  <AVP xmlns="http://www.ercot.com/schema/2007-06/nodal/ews"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews file:/C:/eint/tibco/EWS/Schemas/ErcotTransactions.xsd">  <startTime>2012-11-08T00:00:00-05:00</startTime>  <endTime>2012-11-09T00:00:00-05:00</endTime>  <resource>RESOURCE1</resource>  <availabilityType>FFSS</availabilityType>  <availabilityStatus>  <startTime>2012-11-08T00:00:00-05:00</startTime>  <endTime>2012-11-08T01:00:00-05:00</endTime>  <status>A</status>  </availabilityStatus>  <availabilityStatus>  <startTime>2012-11-08T01:00:00-05:00</startTime>  <endTime>2012-11-09T00:00:00-05:00</endTime>  <status>U</status>  </availabilityStatus>  </AVP>  </BidSet>  And the corresponding response:  <ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">  <ns1:tradingDate>2012-11-08</ns1:tradingDate>  <ns1:submitTime>2012-11-06T10:05:54.455-06:00</ns1:submitTime>  <ns1:AVP>  <ns1:mRID>QSE1.20121108.AVP.RESOURCE1.FFSS</ns1:mRID>  <ns1:externalId/>  <ns1:status>SUBMITTED</ns1:status>  </ns1:AVP>  </ns1:BidSet> |

### Capacity Trade (CT)

The following diagram describes the structure of a Capacity Trade:

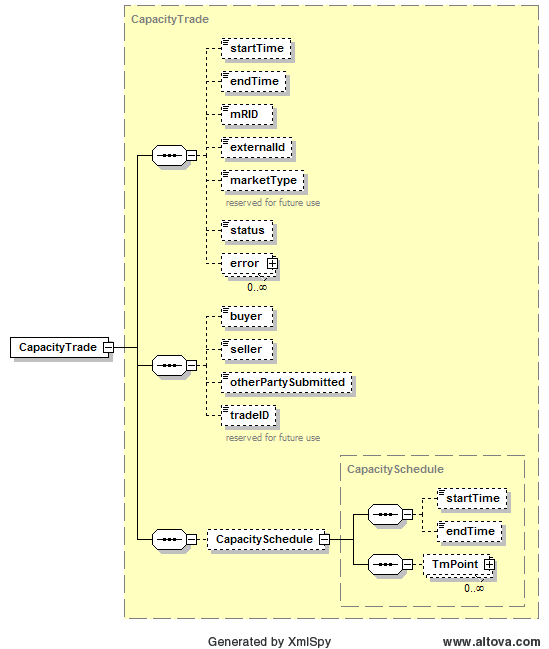


Figure 34 - CapacityTrade Structure

The CapacitySchedule structure is identical to the structure described previously in this document. The submission of a trade requires a matching trade to be submitted by the counter party. The following table describes the required fields:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| startTime | K | dateTime | Start time for bid | Valid start hour boundary for trade date |
| endTime | K | dateTime | End time for bid | Valid end hour boundary for trade date |
| externalId | N | string | External ID | QSE supplied |
| buyer | K | string | Trade buyer | Valid QSE |
| seller | K | string | Trade seller | Valid QSE |
| otherPartySubmitted | N | Boolean | For querying purposes only:  True if trade was entered by other party | true (default) or false |
| tradeID | N | string | reserved for future use |  |
| CapacitySchedule/startTime | N | dateTime | not used | not used |
| CapacitySchedule/endTime | N | dateTime | not used | not used |
| CapacitySchedule/  TmPoint/time | Y | dateTime | Absolute time for beginning of interval | Valid time within the trading date |
| CapacitySchedule/  TmPoint/ending | N | dateTime | Absolute time for end of interval | Valid time within the trading date |
| CapacitySchedule/  TmPoint/value1 | Y | float | Megawatts | >= 0 |

Figure 35 - CapacityTrade Requirements

The following is an XML example for a CapacityTrade:

<BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotTransactions.xsd">

<tradingDate>2008-01-01</tradingDate>

<status/>

<mode/>

<CapacityTrade>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<marketType>DAM</marketType>

<buyer>AEN</buyer>

<seller>LCRA</seller>

<tradeID>TradeID234</tradeID>

<CapacitySchedule>

<TmPoint>

<time>2008-01-01T00:00:00-05:00</time>

<ending>2008-01-02T00:00:00-05:00</ending>

<value1>88</value1>

</TmPoint>

</CapacitySchedule>

</CapacityTrade>

</BidSet>

And the corresponding response:

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2008-06-14</ns1:tradingDate>

<ns1:CapacityTrade>

<ns1:mRID>AEN.20080614.CT.AEN.LCRA</ns1:mRID>

<ns1:status>ACCEPTED</ns1:status>

<ns1:error>

<ns1:severity>INFORMATIVE</ns1:severity>

<ns1:text>Successfully processed the ERCOT Capacity Trade.</ns1:text>

</ns1:error>

</ns1:CapacityTrade>

</ns1:BidSet>

### PTP Obligation w/ Links to Option (CRR)

The following diagram describes the structure of a PTP Obligation w/ Links to Option:

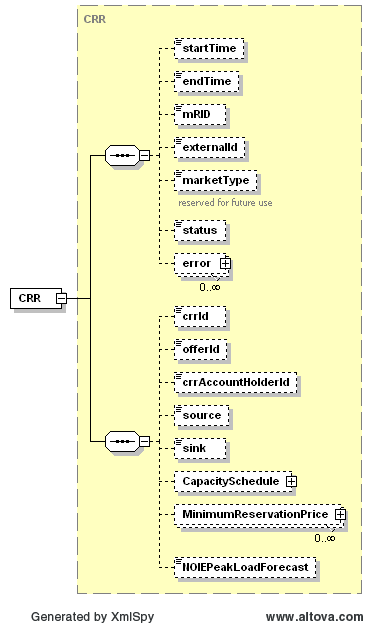


Figure 36 – PTP Obligation w/ Links to Option - CRR Structure

The following structure is used for minimum reservation prices. Note than if more than on MinimumReservationPrice is provided, the start and end times must not overlap.



Figure 37 - MinimumReservationPrice Structure

The CapacitySchedule structure is identical to the structure described previously in this document. On submission, the following table describes the items used for a CRR. In addition, the PeakLoadForecast must be specified for NOIEs.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| startTime | K | dateTime | Start time for offer | Valid start hour boundary for trade date |
| endTime | K | dateTime | End time for offer | Valid end hour boundary for trade date |
| externalId | N | string | External ID | QSE supplied |
| crrId | K | string | CRR ID | ID of CRR |
| offerId | K | string | Offer ID  ERCOT’s preference is for QSEs to submit the same Bid/Offer ID for each hour in the submission, versus submitting a different Bid/Offer ID for every hour.  The market system treats either submission format the exact same way.  The purpose of this is to optimize system performance. | QSE supplied  Value Restrictions:  Only alpha numeric, “\_”(underscore) and “-“(dash) are valid characters. First and last character should be alpha numeric.  Min Length: 2chars.  Max Length: 12 chars. |
| crrAccountHolderId | K | string | CRR account holder ID | ID of CRR account holder |
| Source | K | string | Source settlement point | Valid settlement point name |
| Sink | K | string | Sink settlement point | Valid settlement point name |
| CapacitySchedule/startTime | N | dateTime | not used | not used |
| CapacitySchedule/endTime | N | dateTime | not used | not used |
| CapacitySchedule/  TmPoint/time | Y | dateTime | Absolute start time for interval | Valid time in the trading date |
| CapacitySchedule/  TmPoint/ending | N | dateTime | Absolute end time for interval | Valid time in the trading date |
| CapacitySchedule/  TmPoint/value1 | Y | float | Megawatts | >= 0 |
| CapacitySchedule /  TmPoint/multiHourBlock | N | Boolean | Indicates if offer must be taken as a block for all hours | true or false (default=false) |
| MinimumReservationPrice/  startTime | Y | dateTime | Start time for minimum reservation price | Valid hour boundary |
| MinimumReservationPrice/  endTime | Y | dateTime | End time for minimum reservation price | Valid hour boundary |
| MinimumReservationPrice/  Price | N | decimal | Minimum price in $/MWh | Default (if not specified) is $2000 |
| NOIEPeakLoadForecast | Y | float | Peak load forecast | Peak load forecast in MW |

Figure 38 - CRR Requirements

The following is an XML example for a PTP Obligation w/ Links to Option:

<BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotTransactions.xsd">

<tradingDate>2008-01-01</tradingDate>

<CRR>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<marketType>DAM</marketType>

<crrId>779265</crrId>

<offerId>868558</offerId>

<crrAccountHolderId>XCRRAH</crrAccountHolderId>

<source>TNSCLP3\_\_\_8W</source>

<sink>TNSCLP3\_\_\_8X</sink>

<CapacitySchedule>

<TmPoint>

<time>2008-01-01T00:00:00-05:00</time>

<ending>2008-01-02T00:00:00-05:00</ending>

<value1>20</value1>

</TmPoint>

</CapacitySchedule>

<MinimumReservationPrice>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<price>20</price>

</MinimumReservationPrice>

<NOIEPeakLoadForecast>20</NOIEPeakLoadForecast>

</CRR>

</BidSet>

And the corresponding response:

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2008-06-14</ns1:tradingDate>

<ns1:CRR>

<ns1:mRID>AEN.20080614.CRR.779265.868558.8008717662000.TNSCLP3\_\_\_8W.TNSCLP3\_\_\_8X</ns1:mRID>

<ns1:externalId/>

<ns1:status>ACCEPTED</ns1:status>

<ns1:error>

<ns1:severity>INFORMATIVE</ns1:severity>

<ns1:text>Successfully processed the ERCOT CRR OFFER.</ns1:text>

</ns1:error>

</ns1:CRR>

</ns1:BidSet>

### Current Operating Plan (COP)

The following diagram describes the structure of a Current Operating Plan (COP), where the COP provides a schedule for a resource:

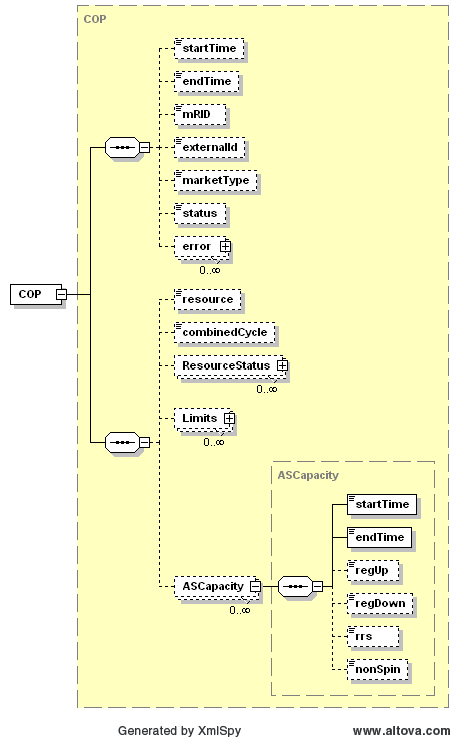


Figure 39 - COP Structure

The structures for Limits and ASCapacity are shown in the following diagrams. If more than one ResourceStatus, Limits, or ASCapacity block is provided, start and end times must not overlap:

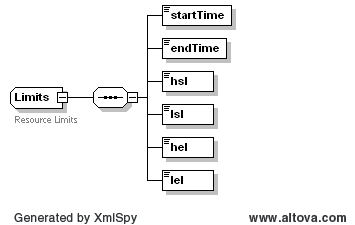


Figure 40 - Limits Structure

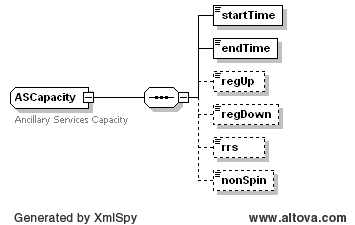


Figure 41 - ASCapacity Structure

COP submissions are never cancelled, as a COP must be submitted. A resubmission of a COP overwrites the previous submission. On submission, the following table describes the items used for a COP:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| startTime | N | dateTime | Start time for COP | Valid start hour boundary for trade date |
| endTime | N | dateTime | End time for COP | Valid end hour boundary for trade date |
| externalId | N | string | External ID | QSE supplied |
| resource | K | string | Resource | Valid resource name |
| combinedCycle | N | string | Combined cycle | Not required. Value ignored if provided. |
| ResourceStatus/startTime | Y | dateTime | Start time for status | Valid hour boundary |
| ResourceStatus /endTime | Y | dateTime | End time for status | Valid hour boundary |
| ResourceStatus /operatingMode | Y | string | status | Valid status code |
| Limits/startTime | Y | dateTime | Start time for limits | Valid hour boundary |
| Limits/endTime | Y | dateTime | End time for limits | Valid hour boundary |
| Limits/hsl | Y | float | High sustained limit in MW | >=0 |
| Limits/lsl | Y | float | Low sustained limit in MW | >=0 |
| Limits/hel | Y | float | High emergency limit in MW | >=0 |
| Limits/lel | Y | float | Low sustained limit in MW | >=0 |
| ASCapacity/startTime | Y | dateTime | Start time for AS capacity schedule | Valid hour boundary |
| ASCapacity/endTime | Y | dateTime | End time for AS capacity schedule | Valid hour boundary |
| ASCapacity/regUp | Y | float | Regulation up MW | >= 0 |
| ASCapacity/regDown | Y | float | Regulation down MW | >= 0 |
| ASCapacity/rrs | Y | float | Responsive reserve MW | >= 0 |
| ASCapacity/nonSpin | Y | float | Non-spinning MW | >= 0 |

Figure 42 - COP Requirements

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***[NPRR863: Replace Figure 42 above with the following upon system implementation]***   |  |  |  |  |  | | --- | --- | --- | --- | --- | | *Element* | *Req?* | *Datatype* | *Description* | *Values* | | startTime | N | dateTime | Start time for COP | Valid start hour boundary for trade date | | endTime | N | dateTime | End time for COP | Valid end hour boundary for trade date | | externalId | N | string | External ID | QSE supplied | | resource | K | string | Resource | Valid resource name | | combinedCycle | N | string | Combined cycle | Not required. Value ignored if provided. | | ResourceStatus/startTime | Y | dateTime | Start time for status | Valid hour boundary | | ResourceStatus /endTime | Y | dateTime | End time for status | Valid hour boundary | | ResourceStatus /operatingMode | Y | string | status | Valid status code | | Limits/startTime | Y | dateTime | Start time for limits | Valid hour boundary | | Limits/endTime | Y | dateTime | End time for limits | Valid hour boundary | | Limits/hsl | Y | float | High sustained limit in MW | >=0 | | Limits/lsl | Y | float | Low sustained limit in MW | >=0 | | Limits/hel | Y | float | High emergency limit in MW | >=0 | | Limits/lel | Y | float | Low sustained limit in MW | >=0 | | ASCapacity/startTime | Y | dateTime | Start time for AS capacity schedule | Valid hour boundary | | ASCapacity/endTime | Y | dateTime | End time for AS capacity schedule | Valid hour boundary | | ASCapacity/regUp | Y | float | Regulation up MW | >= 0 | | ASCapacity/regDown | Y | float | Regulation down MW | >= 0 | | ASCapacity/rrsPF | Y | float | Responsive reserve MW | >= 0 | | ASCapacity/rrsFF | Y | float | Responsive reserve MW | >= 0 | | ASCapacity/rrsUF | Y | float | Responsive reserve MW | >= 0 | | ASCapacity/nonSpin | Y | float | Non-spinning MW | >= 0 | |

The following is an XML example for a COP:

<BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews">

<tradingDate>2008-01-01</tradingDate>

<COP>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<marketType>DAM</marketType>

<resource>Resource123</resource>

<ResourceStatus>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<operatingMode>ON</operatingMode>

</ResourceStatus>

<Limits>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<hsl>45.50</hsl>

<!--HSL should not be less than (LSL+RRS+RegUP+NSpin) where resource status is ON

HSL minimum limit is 47 and LSL the minimum limit is 12-->

<lsl>15.14</lsl>

<hel>3.1</hel>

<lel>3.1</lel>

</Limits>

<ASCapacity>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<regUp>3.1</regUp>

<regDown>3.1</regDown>

<rrs>3.1</rrs>

<nonSpin>3.1</nonSpin>

</ASCapacity>

</COP>

</BidSet>

|  |
| --- |
| ***[NPRR863: Replace example above with the following upon system implementation]***  <BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:ns2="http://www.ercot.com/schema/2007-06/nodal/ews">  <tradingDate>2021-11-09</tradingDate>  <COP>  <startTime>2021-11-09T00:00:00-06:00</startTime>  <endTime>2021-11-10T00:00:00-06:00</endTime>  <resource>RES\_1</resource>  <ResourceStatus>  <startTime>2021-11-09T23:00:00.000-06:00</startTime>  <endTime>2021-11-10T00:00:00.000-06:00</endTime>  <operatingMode>ONRL</operatingMode>  </ResourceStatus>  <Limits>  <startTime>2021-11-09T23:00:00.000-06:00</startTime>  <endTime>2021-11-10T00:00:00.000-06:00</endTime>  <hsl>20</hsl>  <lsl>0</lsl>  <hel>20</hel>  <lel>0</lel>  </Limits>  <ASCapacity>  <startTime>2021-11-09T23:00:00.000-06:00</startTime>  <endTime>2021-11-10T00:00:00.000-06:00</endTime>  <regUp>0</regUp>  <regDown>0</regDown>  <rrsPF>0</rrsPF>  <rrsFF>0</rrsFF>  <rrsUF>20</rrsUF>  <nonSpin>0</nonSpin>  </ASCapacity>  </COP>  </BidSet> |

And the corresponding response:

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2021-11-09</ns1:tradingDate>

<ns1:COP>

<ns1:mRID>QSAMP1.20211109.COP.RES\_1</ns1:mRID>

<ns1:externalId/>

<ns1:status>ACCEPTED</ns1:status>

<ns1:error>

<ns1:severity>INFORMATIVE</ns1:severity>

<ns1:text>Successfully processed the ERCOT COP.</ns1:text>

</ns1:error>

</ns1:COP>

</ns1:BidSet>

The terms maxEmergency, maximumEconomic, minEmergency, and minimumEcononic are used to in order to be consistent with the IEC CIM.

Currently defined resource operatingModes include:

* ONRUC – On-Line and the hour is a RUC-Committed Interval
* ONREG – On-Line Resource with Energy Offer Curve providing Regulation Service
* ON – On-Line Resource with Energy Offer Curve
* ONDSR – On-Line Dynamically Scheduled Resource
* ONOS – On-Line Resource with Output Schedule
* ONOSREG – On-Line Resource with Output Schedule providing Regulation Service
* ONDSRREG – On-Line Dynamically Scheduled Resource providing Regulation Service
* ONTEST – On-Line Test with Output Schedule
* ONEMR – On-Line EMR (available for commitment or dispatch only for ERCOT-declared Emergency Conditions; the QSE may appropriately set LSL and HSL to reflect operating limits)
* ONRR – On-Line as a synchronous condenser (hydro) providing Responsive Reserve but unavailable for dispatch by SCED and available for commitment by RUC
* OUT – Off-Line and unavailable
* OFFNS – Off -Line but reserved for Non-Spin
* OFF – Off-Line but available for commitment by DAM and RUC
* EMR – Available for commitment only for ERCOT-declared Emergency Condition events; the QSE may appropriately set LSL and HSL to reflect operating limits
* ONRGL – Available for Dispatch of Regulation Service by Load Frequency Control (LFC) and, for any remaining Dispatchable capacity, by SCED with an Real-Time Market (RTM) Energy Bid
* ONCLR – Available for Dispatch as a Controllable Load Resource by SCED with an RTM Energy Bid
* ONRL – Available for Dispatch of RRS Service, excluding Controllable Load Resources
* OUTL – Not available
* ONOPTOUT – On-Line and the hour is a RUC Buy-Back Hour
* OFFQS – Off-Line but available for SCED deployment. Only qualified Quick Start Generation Resources (QSGRs) may utilize this status.
* EMRSWGR - status can be submitted individually for qualified SWGRs only and Operating Hour(s) for which they are applicable during the Adjustment Period

### DAM Energy Bid (EB)

The following diagram describes the structure of a DAM Energy Bid:

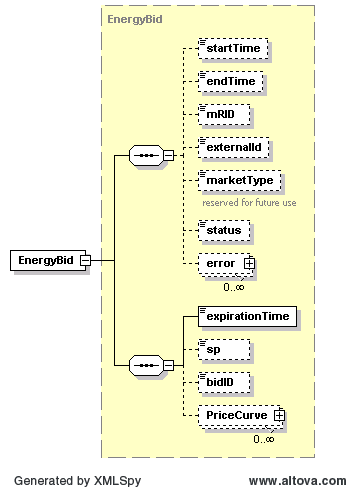


Figure 43 - EnergyBid Structure

The PriceCurve structure is identical to the structure described previously in this document. On submission, the following table describes the items used for an EnergyBid:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| startTime | K | dateTime | Start time for bid | Valid start hour boundary for trade date |
| endTime | K | dateTime | End time for bid | Valid end hour boundary for trade date |
| externalId | N | string | External ID | QSE supplied |
| expirationTime | Y | dateTime | Time of bid expiration | Valid time before trade date |
| sp | K | string | Settlement point | Valid settlement point name |
| bidID[[1]](#footnote-1) | K | string | Bid ID  ERCOT’s preference is for QSEs to submit the same Bid/Offer ID for each hour in the submission, versus submitting a different Bid/Offer ID for every hour.  The market system treats either submission format the exact same way.  The purpose of this is to optimize system performance. | MP supplied bid ID  Value Restrictions:  Only alpha numeric, “\_”(underscore) and “-“(dash) are valid characters. First and last character should be alpha numeric.  Min Length: 2chars.  Max Length: 12 chars. |
| PriceCurve/startTime | Y | dateTime | Start time for curve | Valid hour boundary |
| PriceCurve/endTime | Y | dateTime | End time for curve | Valid hour boundary |
| PriceCurve/curveStyle | Y | string | Used as an indicator to describe the type of ‘curve’ | FIXED, VARIABLE or CURVE |
| PriceCurve/CurveData/xvalue | Y | float | Megawatts | Quantity in MW |
| PriceCurve/CurveData/y1value | Y | float | Price in $/MWh | $/MWh |
| PriceCurve/multiHourBlock | N | Boolean | Indicates if offer must be taken as a block for all hours | true or false (default=false) |

Figure 44 - EnergyBid Requirements

The following is an XML example for an EnergyBid:

<BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotTransactions.xsd">

<tradingDate>2008-01-01</tradingDate>

<EnergyBid>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<marketType>DAM</marketType>

<expirationTime>2008-01-01T00:00:00-05:00</expirationTime>

<sp>TNSCLP3\_\_\_8X</sp>

<bidID>338601</bidID>

<PriceCurve>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<curveStyle>CURVE</curveStyle>

<CurveData>

<xvalue>41</xvalue>

<y1value>2790.00</y1value>

</CurveData>

<CurveData>

<xvalue>93</xvalue>

<y1value>2557.36</y1value>

</CurveData>

<CurveData>

<xvalue>150</xvalue>

<y1value>2312.44</y1value>

</CurveData>

<CurveData>

<xvalue>197</xvalue>

<y1value>2108.61</y1value>

</CurveData>

<CurveData>

<xvalue>256</xvalue>

<y1value>1896.05</y1value>

</CurveData>

<CurveData>

<xvalue>298</xvalue>

<y1value>1653.97</y1value>

</CurveData>

<CurveData>

<xvalue>353</xvalue>

<y1value>1383.49</y1value>

</CurveData>

<CurveData>

<xvalue>397</xvalue>

<y1value>1103.48</y1value>

</CurveData>

<CurveData>

<xvalue>454</xvalue>

<y1value>905.00</y1value>

</CurveData>

<CurveData>

<xvalue>558</xvalue>

<y1value>205.00</y1value>

</CurveData>

<multiHourBlock>false</multiHourBlock>

</PriceCurve>

</EnergyBid>

</BidSet>

And the corresponding response:

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2008-06-14</ns1:tradingDate>

<ns1:EnergyBid>

<ns1:mRID>AEN.20080614.EB.TNSCLP3\_\_\_8X.338601</ns1:mRID>

<ns1:externalId/>

<ns1:status>ACCEPTED</ns1:status>

<ns1:error>

<ns1:severity>INFORMATIVE</ns1:severity>

<ns1:text>Successfully processed the ERCOT Energy-Only Bid.</ns1:text>

</ns1:error>

</ns1:EnergyBid>

</ns1:BidSet>

### DAM Energy-Only Offer (EOO)

The following diagram describes the structure of a DAM Energy-Only Offer:

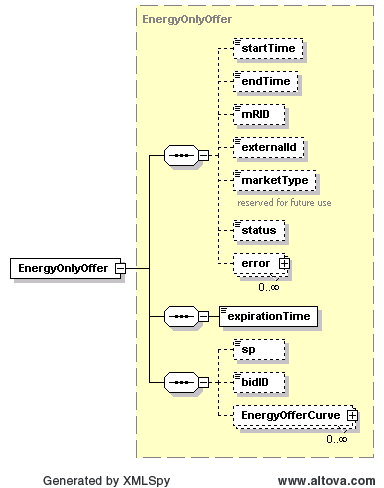


Figure 45 – EnergyOnlyOffer Structure

The EnergyOfferCurve structure uses the PriceCurve type structure described previously in this document. On submission, the following table describes the items used for an EnergyOnlyOffer:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* | |
| startTime | K | dateTime | Start time for bid | Valid start hour boundary for trade date | |
| endTime | K | dateTime | End time for bid | Valid end hour boundary for trade date | |
| externalId | N | string | External ID | | QSE supplied |
| sp | K | string | Settlement point | Valid settlement point name | |
| bidID[[2]](#footnote-2) | K | string | Bid ID  ERCOT’s preference is for QSEs to submit the same Bid/Offer ID for each hour in the submission, versus submitting a different Bid/Offer ID for every hour.  The market system treats either submission format the exact same way.  The purpose of this is to optimize system performance. | QSE supplied bid ID.  Value Restrictions:  Only alpha numeric, “\_”(underscore) and “-“(dash) are valid characters. First and last character should be alpha numeric.  Min Length: 2chars.  Max Length: 12 chars. | |
| expirationTime | Y | dateTime | Time of offer expiration | Valid time before trade date | |
| EnergyOfferCurve/startTime | Y | dateTime | Start time for curve | Valid hour boundary | |
| EnergyOfferCurve/endTime | Y | dateTime | End time for curve | Valid hour boundary | |
| EnergyOfferCurve/curveStyle | Y | string | Used as an indicator to describe the type of ‘curve’ | FIXED, VARIABLE or CURVE | |
| EnergyOfferCurve/CurveData/xvalue | Y | float | Quantity in megawatts | MW | |
| EnergyOfferCurve/CurveData/y1value | Y | float | Price in $/MWh | $/MWh | |
| EnergyOfferCurve/incExcFlag | N | string | Not used | Not used | |
| EnergyOfferCurve/reason | N | string | Not used | Not used | |
| EnergyOfferCurve/multiHourBlock | N | Boolean | Indicates if offer must be taken as a block for all hours | true or false (default=false) | |

Figure 46 - EnergyOnlyOffer Requirements

The following is an XML example for an EnergyOnlyOffer:

<BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotTransactions.xsd">

<tradingDate>2008-01-01</tradingDate>

<EnergyOnlyOffer>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<marketType>DAM</marketType>

<expirationTime>2008-01-01T00:00:00-05:00</expirationTime>

<sp>TNSCLP3\_\_\_8X</sp>

<bidID>338601</bidID>

<EnergyOfferCurve>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<curveStyle>FIXED</curveStyle>

<CurveData>

<xvalue>197</xvalue>

<y1value>999</y1value>

</CurveData>

<multiHourBlock>false</multiHourBlock>

</EnergyOfferCurve>

</EnergyOnlyOffer>

</BidSet>

And the corresponding response:

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2008-06-14</ns1:tradingDate>

<ns1:EnergyOnlyOffer>

<ns1:mRID>AEN.20080614.EOO.TNSCLP3\_\_\_8X.338601</ns1:mRID>

<ns1:externalId/>

<ns1:status>ACCEPTED</ns1:status>

<ns1:error>

<ns1:severity>INFORMATIVE</ns1:severity>

<ns1:text>Successfully processed the ERCOT Energy-Only Offer.</ns1:text>

</ns1:error>

</ns1:EnergyOnlyOffer>

</ns1:BidSet>

### Energy Trade (ET)

An EnergyTrade describes a scheduled trade of energy between a buyer and a seller at a specified settlement point. The following diagram describes the structure of an Energy Trade:

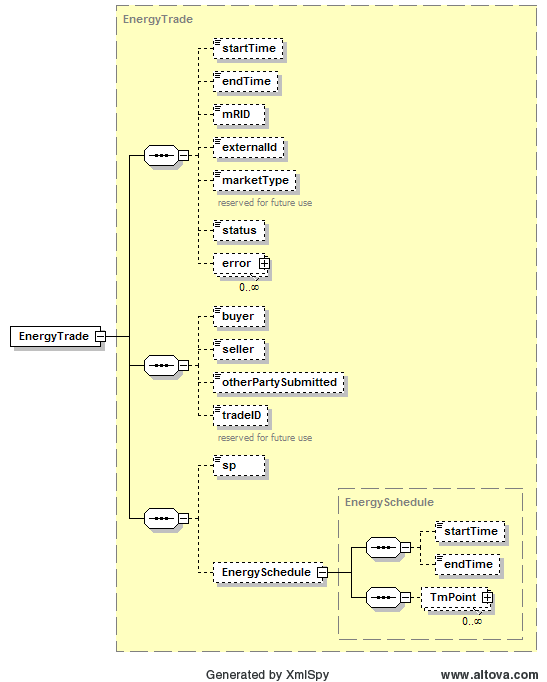


Figure 47 - EnergyTrade Structure

Within an EnergyTrade is an EnergySchedule. Energy Schedules are defined using the TmSchedule type, where intervals may be up to 15 minutes in granularity. This is shown in the following diagram:

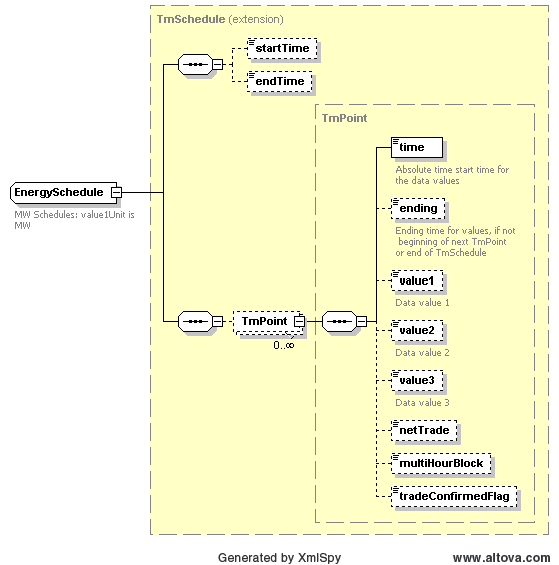


Figure 48 - EnergySchedule Structure

In order for a trade to be accepted, both the buyer and seller must submit matching trades, as identified using the buyer and seller tags. On submission, the following table describes the items used for an EnergyTrade:

* startTime
* endTime
* buyer (QSE ID)
* seller (QSE ID)
* sp (settlement point name)
* EnergySchedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| startTime | K | dateTime | Start time for bid | Valid start hour boundary for trade date |
| endTime | K | dateTime | End time for bid | Valid end hour boundary for trade date |
| externalId | N | String | External ID | QSE supplied |
| sp | K | String | Settlement point | Valid settlement point name |
| buyer | K | String | Trade buyer | Valid QSE |
| seller | K | String | Trade seller | Valid QSE |
| tradeID | N | String | reserved for future use |  |
| EnergySchedule/startTime | N | dateTime | not used | not used |
| EnergySchedule/endTime | N | dateTime | not used | not used |
| EnergySchedule/  TmPoint/time | Y | dateTime | Absolute time for start of 15 minute interval | Valid time in the trading date |
| EnergySchedule/  TmPoint/ending | N | dateTime | Absolute time for end of 15 minute interval | Valid time in the trading date |
| EnergySchedule/  TmPoint/value1 | Y | float | Megawatts | >= 0 |
| EnergySchedule/  TmPoint/netTrade | N | string | Trade description | Purchase (“P”) or  Sale (“S”)  for DSR resources. |

Figure 49 - EnergyTrade Requirements

The following is an XML example for an EnergyTrade:

<BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotTransactions.xsd">

<tradingDate>2008-01-01</tradingDate>

<status/>

<mode/>

<EnergyTrade>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<marketType>DAM</marketType>

<buyer>AEN</buyer>

<seller>LCRA</seller>

<tradeID>TradeID123</tradeID>

<sp>JUDKINS\_8</sp>

<EnergySchedule>

<TmPoint>

<time>2008-01-01T00:00:00-05:00</time>

<ending>2008-01-02T00:00:00-05:00</ending>

<value1>89</value1>

</TmPoint>

</EnergySchedule>

</EnergyTrade>

</BidSet>

And the corresponding response:

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotTransactions.xsd">

<ns1:tradingDate>2008-06-14</ns1:tradingDate>

<ns1:EnergyTrade>

<ns1:mRID>AEN.20080614.ET.JUDKINS\_8.AEN.LCRA</ns1:mRID>

<ns1:status>ACCEPTED</ns1:status>

<ns1:error>

<ns1:severity>INFORMATIVE</ns1:severity>

<ns1:text>Successfully processed the ERCOT Energy Trade.</ns1:text>

</ns1:error>

</ns1:EnergyTrade>

</ns1:BidSet>

### Output Schedule (OS)

An OutputSchedule describes the scheduled output for a resource at 5 minute intervals. The following diagram describes the structure of an Output Schedule:

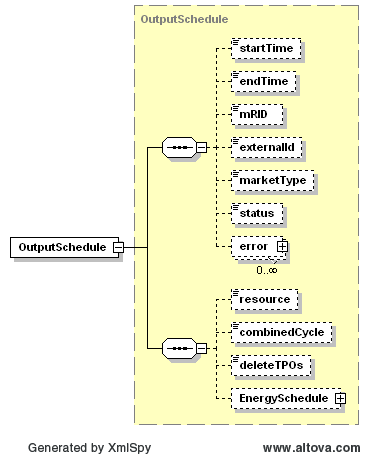


Figure 50 - OutputSchedule Structure

The EnergySchedule structure is identical to the structure described previously in this document. On submission, the following table describes the items used for an OutputSchedule:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| startTime | N | dateTime | Start time for schedule | Valid 5 minute interval boundary for trade date |
| endTime | N | dateTime | End time for schedule | Valid 5 minute interval boundary for trade date |
| externalId | N | string | External ID | QSE supplied |
| resource | K | string | Resource | Valid resource name |
| combinedCycle | N | string | Combined cycle | Not required. Value ignored if provided. |
| deleteTPOs | N | Boolean | Set to true if three part offers for resource are to be deleted | 1 or 0 (default=0) |
| EnergySchedule/startTime | N | dateTime | Not Used | Not Used |
| EnergySchedule/endTime | N | dateTime | Not Used | Not Used |
| EnergySchedule/  TmPoint/time | Y | dateTime | Absolute time for start of interval | Valid time between startTime and endTime for schedule |
| EnergySchedule/  TmPoint/ending | N | dateTime | Absolute time for end of interval | Valid time between startTime and endTime for schedule |
| EnergySchedule/  TmPoint/value1 | Y | float | Megawatts | >= 0 |

Figure 51 - OutputSchedule Requirements

The following is an XML example for an OutputSchedule:

<BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotTransactions.xsd">

<tradingDate>2008-01-01</tradingDate>

<OutputSchedule>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<marketType>DAM</marketType>

<resource> Resource1</resource>

<deleteTPOs>true</deleteTPOs>

<EnergySchedule>

<TmPoint>

<time>2008-01-01T00:00:00-05:00</time>

<ending>2008-01-02T00:00:00-05:00</ending>

<value1>20</value1>

</TmPoint>

</EnergySchedule>

</OutputSchedule>

</BidSet>

And the corresponding response:

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2008-06-15</ns1:tradingDate>

<ns1:OutputSchedule>

<ns1:mRID>AEN.20080615.OS. Resource1</ns1:mRID>

<ns1:externalId/>

<ns1:status>ACCEPTED</ns1:status>

<ns1:error>

<ns1:severity>WARNING</ns1:severity>

<ns1:text>Energy Offer Curve for Resource1 does not exist for cancellation.</ns1:text>

</ns1:error>

<ns1:error>

<ns1:severity>INFORMATIVE</ns1:severity>

<ns1:text>Successfully processed the ERCOT Output Schedule.</ns1:text>

</ns1:error>

</ns1:OutputSchedule>

</ns1:BidSet>

### PTP Obligation Bid (PTP)

The following diagram describes the structure of a PTP Obligation bid:

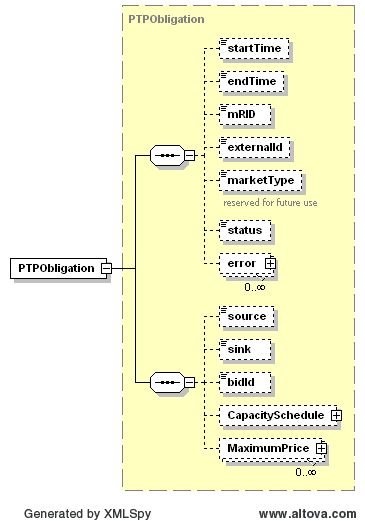


Figure 52 - PTPObligation Requirements

The CapacitySchedule structure is identical to the structure described previously in this document. The following diagram describes the structure for maximum prices. If more than one MaximumPrice block is provided, start and end times must not overlap.

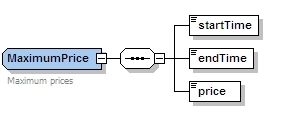


Figure 53 - MaximumPrice Sub-Structure

On submission, the following table describes the items used for a PTPObligation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| startTime | K | dateTime | Start time for bid | Valid start hour boundary for trade date |
| endTime | K | dateTime | End time for bid | Valid end hour boundary for trade date |
| externalId | N | string | External ID | QSE supplied |
| source | K | string | Source settlement point | Valid settlement point name |
| sink | K | string | Sink settlement point | Valid settlement point name |
| bidId[[3]](#footnote-3) | K | string | MP assigned bid ID  ERCOT’s preference is for QSEs to submit the same Bid/Offer ID for each hour in the submission, versus submitting a different Bid/Offer ID for every hour.  The market system treats either submission format the exact same way.  The purpose of this is to optimize system performance. | Value Restrictions:  Only alpha numeric, “\_”(underscore) and “-“(dash) are valid characters. First and last character should be alpha numeric.  Min Length: 2chars.  Max Length: 12 chars. |
| MaximumPrice/startTime | Y | dateTime | Start time for maximum price | Valid hour boundary |
| MaximumPrice/endTime | Y | dateTime | End time for maximum price | Valid hour boundary |
| MaximumPrice/price | Y | float | Maximum price | Any value |
| CapacitySchedule/startTime | N | dateTime | not used | not used |
| CapacitySchedule/endTime | N | dateTime | not used | not used |
| CapacitySchedule/  TmPoint/time | Y | dateTime | Absolute time for beginning of interval | Valid time between schedule start and end times |
| CapacitySchedule/  TmPoint/ending | N | dateTime | Absolute time for end of interval | Valid time between schedule start and end times |
| CapacitySchedule/  TmPoint/value1 | Y | float | Megawatts | >= 0 |
| CapacitySchedule /  TmPoint/multiHourBlock | N | Boolean | Indicates if offer must be taken as a block for all hours | true or false (default=false) |

Figure 54 - PTPObligation Requirements

The following is an XML example for a PTPObligation:

<BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotTransactions.xsd">

<tradingDate>2008-01-01</tradingDate>

<PTPObligation>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<marketType>DAM</marketType>

<source>JNSNCNTY8101</source>

<sink>JUDKINS\_8</sink>

<bidId>926606</bidId>

<CapacitySchedule>

<TmPoint>

<time>2008-01-01T00:00:00-05:00</time>

<ending>2008-01-02T00:00:00-05:00</ending>

<value1>327</value1>

</TmPoint>

</CapacitySchedule>

<MaximumPrice>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<price>15.00</price>

</MaximumPrice>

</PTPObligation>

</BidSet>

And the corresponding response:

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2008-06-14</ns1:tradingDate>

<ns1:PTPObligation>

<ns1:mRID>AEN.20080614.PTP.926606.AEN.JNSNCNTY8101.JUDKINS\_8</ns1:mRID>

<ns1:externalId/>

<ns1:status>ACCEPTED</ns1:status>

<ns1:error>

<ns1:severity>INFORMATIVE</ns1:severity>

<ns1:text>Successfully processed the ERCOT PTP BID.</ns1:text>

</ns1:error>

</ns1:PTPObligation>

</ns1:BidSet>

### Self-Schedule (SS)

The following diagram describes the structure of a Self-Schedule:

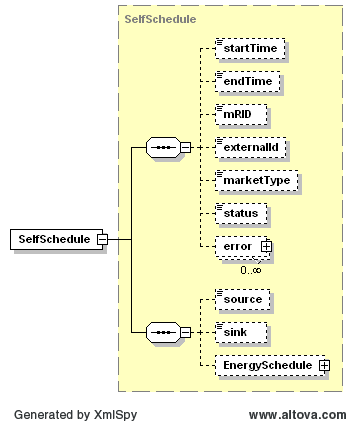


Figure 55 - SelfSchedule Structure

The EnergySchedule structure is defined using a TmSchedule type. On submission, the following table describes the items used for a SelfSchedule:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| startTime | K | dateTime | Start time for bid | Valid 15 minute boundary for trade date |
| endTime | K | dateTime | End time for bid | Valid 15 minute boundary for trade date |
| externalId | N | string | External ID | QSE supplied |
| source | K | string | Source settlement point | Valid settlement point name |
| sink | K | string | Sink settlement point | Valid settlement point name |
| EnergySchedule/startTime | N | dateTime | not used | not used |
| EnergySchedule/endTime | N | dateTime | not used | not used |
| EnergySchedule/  TmPoint/time | Y | dateTime | Absolute time for start of interval | Valid time within the trade date |
| EnergySchedule/  TmPoint/ending | N | dateTime | Absolute time for end of interval | Valid time within the trade date |
| EnergySchedule/  TmPoint/value1 | Y | float | Megawatts | >= 0 |

Figure 56 – SelfSchedule Requirements

The following is an XML example for a SelfSchedule:

<BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotTransactions.xsd">

<tradingDate>2008-01-01</tradingDate>

<SelfSchedule>

<startTime>2008-01-01T00:00:00-05:00</startTime>

<endTime>2008-01-02T00:00:00-05:00</endTime>

<marketType>DAM</marketType>

<source>TNSCLP3\_\_\_8Y</source>

<sink>TNSCLP3\_\_\_8X</sink>

<EnergySchedule>

<TmPoint>

<time>2008-01-01T00:00:00-05:00</time>

<ending>2008-01-02T00:00:00-05:00</ending>

<value1>6353</value1>

</TmPoint>

</EnergySchedule>

</SelfSchedule>

</BidSet>

And the corresponding response:

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2008-06-14</ns1:tradingDate>

<ns1:SelfSchedule>

<ns1:mRID>AEN.20080614.SS.TNSCLP3\_\_\_8Y.TNSCLP3\_\_\_8X</ns1:mRID>

<ns1:externalId/>

<ns1:status>ACCEPTED</ns1:status>

<ns1:error>

<ns1:severity>INFORMATIVE</ns1:severity>

<ns1:text>Successfully processed the ERCOT Self Schedule.</ns1:text>

</ns1:error>

</ns1:SelfSchedule>

</ns1:BidSet>

### Real-Time Market Energy Bid (REB)

The following diagram describes the structure of a Real-Time Market Energy Bid:

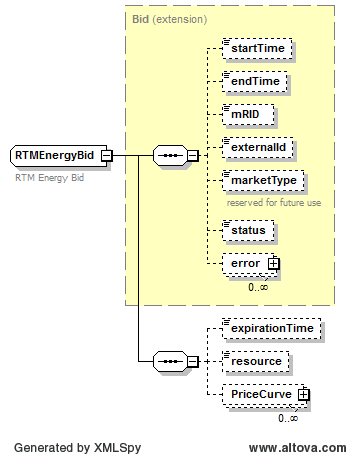


Figure 57 - RTMEnergyBid Structure

The PriceCurve structure is identical to the structure described previously in this document. On submission, the following table describes the items used for an RTMEnergyBid:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| startTime | K | dateTime | Start time for bid | Valid start hour boundary for trade date |
| endTime | K | dateTime | End time for bid | Valid end hour boundary for trade date |
| externalId | N | string | External ID | QSE supplied |
| expirationTime | Y | dateTime | Time of bid expiration | Valid time during the trade date |
| Resource | K | string | Resource | Valid resource name |
| PriceCurve/startTime | Y | dateTime | Start time for curve | Valid hour boundary |
| PriceCurve/endTime | Y | dateTime | End time for curve | Valid hour boundary |
| PriceCurve/CurveData/xvalue | Y | float | Megawatts | Quantity in MW |
| PriceCurve/CurveData/y1value | Y | float | Price in $/MWh | $/MWh |

Figure 58 - RTMEnergyBid Requirements

The following is an XML example for an EnergyBid:

<BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews">

<tradingDate>2014-04-01</tradingDate>

<RTMEnergyBid xmlns="http://www.ercot.com/schema/2007-06/nodal/ews"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<startTime>2014-04-01T00:00:00-05:00</startTime>

<endTime>2014-04-02T00:00:00-05:00</endTime>

<expirationTime>2014-04-02T00:00:00-05:00</expirationTime>

<resource>RESOURCE\_1</resource>

<PriceCurve>

<startTime>2014-04-01T00:00:00-05:00</startTime>

<endTime>2014-04-02T00:00:00-05:00</endTime>

<CurveData>

<xvalue>0</xvalue>

<y1value>250.00</y1value>

</CurveData>

<CurveData>

<xvalue>10</xvalue>

<y1value>225.00</y1value>

</CurveData>

<CurveData>

<xvalue>20</xvalue>

<y1value>220.00</y1value>

</CurveData>

<CurveData>

<xvalue>30</xvalue>

<y1value>215.00</y1value>

</CurveData>

<CurveData>

<xvalue>40</xvalue>

<y1value>200.00</y1value>

</CurveData>

<CurveData>

<xvalue>50</xvalue>

<y1value>185.00</y1value>

</CurveData>

<CurveData>

<xvalue>60</xvalue>

<y1value>160.00</y1value>

</CurveData>

<CurveData>

<xvalue>70</xvalue>

<y1value>145.00</y1value>

</CurveData>

<CurveData>

<xvalue>80</xvalue>

<y1value>130.00</y1value>

</CurveData>

<CurveData>

<xvalue>90</xvalue>

<y1value>115.00</y1value>

</CurveData>

</PriceCurve>

</RTMEnergyBid>

</BidSet>

And the corresponding response:

<ns2:BidSet xmlns:ns2="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns2:tradingDate>2014-04-01</ns2:tradingDate>

<ns2:submitTime>2014-03-27T16:24:23.797-05:00</ns2:submitTime>

<ns2:RTMEnergyBid>

<ns2:mRID>QSE1.20140401.REB.RESOURCE\_1</ns2:mRID>

<ns2:externalId/>

<ns2:status>ACCEPTED</ns2:status>

<ns2:error>

<ns2:severity>INFORMATIVE</ns2:severity>

<ns2:text>Successfully processed the ERCOT Real-Time Market Energy Bid.</ns2:text>

</ns2:error>

</ns2:RTMEnergyBid>

</ns2:BidSet>

### Exceptional Fuel Cost (EFC)

During the Adjustment Period, a QSE representing a Resource may submit Exceptional Fuel Cost as a volume-weighted average fuel price for use in the Mitigated Offer Cap (MOC) calculation for that Resource. The following diagram describes the structure of an Exceptional Fuel Cost Bid:

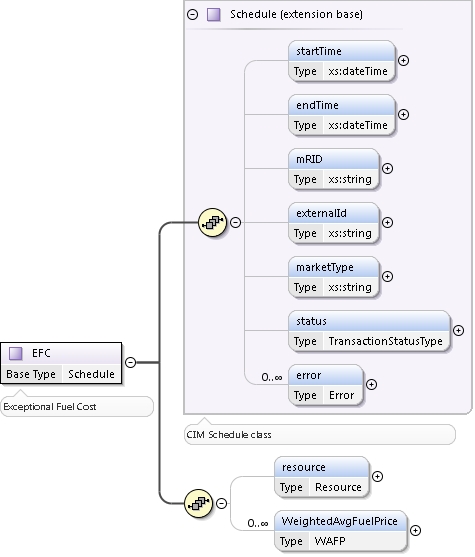


Figure 59 – EFC Bid Structure

The following table describes the items used for an Exception Fuel Cost submission:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| startTime | K | dateTime | Start time for bid | Valid start hour boundary for trade date |
| endTime | K | dateTime | End time for bid | Valid end hour boundary for trade date |
| externalId | N | string | External ID | QSE supplied |
| expirationTime | N | dateTime | Time of bid expiration | Valid time during the trade date |
| resource | K | string | Resource | Valid resource name |
| WeightedAvgFuelPrice/startTime | Y | dateTime | Start time for WAFP | Valid hour boundary |
| WeightedAvgFuelPrice/endTime | Y | dateTime | End time for WAFP | Valid hour boundary |
| WeightedAvgFuelPrice/price | Y | float | Weighted Average Fuel Price | The volume-weighted average intraday, same-day and spot price of fuel |

Figure 60 – EFC Submission Requirements

The following is an XML example for an EFC submission:

<BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews">

<tradingDate>2019-03-13</tradingDate>

<EFC>

<startTime>2019-03-13T19:00:00-05:00</startTime>

<endTime>2019-03-14T00:00:00-05:00</endTime>

<resource>RESOURCE\_1</resource>

<WeightedAvgFuelPrice>

<startTime>2019-03-13T19:00:00-05:00</startTime>

<endTime>2019-03-13T22:00:00-05:00</endTime>

<price>32</price>

</WeightedAvgFuelPrice>

<WeightedAvgFuelPrice>

<startTime>2019-03-13T22:00:00-05:00</startTime>

<endTime>2019-03-14T00:00:00-05:00</endTime>

<price>35</price>

</WeightedAvgFuelPrice>

</EFC>

</BidSet>

And the corresponding response:

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2019-03-13</ns1:tradingDate>

<ns1:submitTime>2019-03-13T18:13:00-05:00</ns1:submitTime>

<ns1:EFC>

<ns1:mRID>QSE1.20190313.EFC.RESOURCE\_1 </ns1:mRID>

<ns1:externalId/>

<ns1:status>ACCEPTED</ns1:status>

<ns1:error>

<ns1:severity>INFORMATIVE</ns1:severity>

<ns1:text>Successfully processed the ERCOT Exceptional Fuel Cost.</ns1:text>

</ns1:error>

</ns1:EFC>

</ns1:BidSet>

## Error Handling

The purpose of this section is to describe the handling of errors related to submissions, updates, queries, and cancellations. This is especially important, as BidSets can be partially accepted, where some bids, offers, trades, or schedules within a BidSet can be accepted, and others rejected.

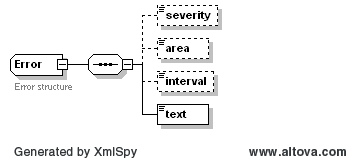
|  |  |  |
| --- | --- | --- |
| *Condition* | *Action* | *Error Reported* |
| Bad parameters in message header | Request is rejected | ERROR: INVALID REQUEST |
| Invalid Source or UserID | Request is rejected | ERROR: NOT AUTHORIZED |
| Payload can not be interpreted | Request is rejected | ERROR: BAD PAYLOAD |
| Parameters within BidSet header are invalid, e.g. bad trading date | No bids or updates are accepted | ERROR: BAD BIDSET |
| Request for a get, cancel or update identifies an unknown transaction (i.e. bid, offer, trade, schedule) | The request is processed only for the known transactions. | WARNING: UNKNOWN ID: …… |
| Some transactions within a BidSet are rejected | For submissions containing multiple offers/bids, valid offers/bids are accepted as submitted, invalid offers/bids are rejected and marked with errors. An offer/bid will be rejected entirely if any hour within it is invalid. | BidSet on reply identifies errors for those transactions that were rejected on a transaction by transaction basis. |

Figure 61 - Error Handling

It is also important to note that the submission of a BidSet involves multiple steps:

1. BidSet is syntax scanned, where only basic validity checks are performed. The reply identifies where basic validity checks failed. Transactions within the BidSet that pass basic validation checks are marked as ‘SUBMITTED’ and a transaction ID (mRID) value is returned.
2. A more thorough validity check is performed asynchronously by the Market Management System (MMS)
3. After acceptance or rejection by MMS, a notification is issued that identified whether the transaction was ‘ACCEPTED’, is ‘PENDING’, or had an error.

The following structure is used to report errors, where the severity, area of the error or the specific time can be identified.



The error structure is populated in the following manner:

* *severity* may be ‘error’, ‘warning’ or ‘informational’, as determined by the source system
* *area* may be used to indicate the tag or parent tag (e.g. EnergySchedule), where an error was identified
* *interval* may be used to identify the time of the data item of concern, where many data items may be related to a schedule submission. For market transaction interfaces, a market interval (using hour ending and minute ending) is used as opposed to a timestamp
* *text* as provided by the source system

When possible, the described error structure will be used. However, there are many cases where error5 structure can not be used, as in cases where schema validations fail and normal validation is not possible. These are typically the result of programming errors. Examples would include:

* XML is not well formed
* XML is not schema compliant, from the perspective of basic data structures
* XML is not schema compliant, from the perspective that values provided are invalid (e.g. invalid time strings, nulls where values are required, non-numeric values where numeric values are required, obviously out of range values, invalid enumerations)

## Usage and XML Examples

The purpose of this section is to provide usage details with XML examples for the submission (create), query (get), update, and cancellation (cancel) of bids, offers, trades, and schedules.

### Offer and Bid Set Submission

When submitting a set of bids, offers, trades, and/or schedules, the following structures are used within the message:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | create |
| Header/Noun | *BidSet* |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Payload | *BidSet* |

Figure 62 - create BidSet Request Parameters

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | *BidSet* |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *May be any number of error messages* |
| Payload | *BidSet, returning transaction IDs (mRIDs) for individual bids, offers, trades or schedules and/or error messages.* |

Figure 63 - reply BidSet Parameters

The following is an example message for a request message for the submission of a BidSet by a Market Participant. The following example uses a SelfArrangedAS.

<RequestMessage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.ercot.com/schema/2007-05/nodal/ews/msg">

<Header>

<Verb>create</Verb>

<Noun>BidSet</Noun>

<ReplayDetection>

<Nonce>3264874657467444949</Nonce>

<Created>2006-12-09 15:35:57Z</Created>

</ReplayDetection>

<Revision>001</Revision>

<Source>SOMEQSE</Source>

<UserID>Nemat</UserID>

<MessageID>231232466</MessageID>

<Comment>Example message</Comment>

</Header>

<Payload>

<BidSet xmlns="http://www.ercot.com/schema/2007-05/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<tradingDate>2007-01-04</tradingDate>

<SelfArrangedAS>

<startTime>2007-12-17T00:00:00-06:00</startTime>

<endTime>2007-12-18T00:00:00-06:00</endTime>

<externalId>myExternalID45473</externalId>

<asType>On-Non-Spin</asType>

<CapacitySchedule>

<TmPoint>

<time>2007-12-17T00:00:00-06:00</time>

<value1>120</value1>

</TmPoint>

<TmPoint>

<time>2007-12-17T02:00:00-06:00</time>

<value1>130</value1>

</TmPoint>

<TmPoint>

<time>2007-12-17T06:00:00-06:00</time>

<value1>115</value1>

</TmPoint>

</CapacitySchedule>

</SelfArrangedAS>

… additional bids inserted here …

</BidSet>

</Payload>

</RequestMessage>

The following is an example response message, where the submission request was successful:

<ResponseMessage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.ercot.com/schema/2007-05/nodal/ews/msg">

<Header>

<Verb>reply</Verb>

<Noun>BidSet</Noun>

<ReplayDetection>

<Nonce>74646464</Nonce>

<Created>2006-12-09T15:36:03-06:00</Created>

</ReplayDetection>

<Revision>001</Revision>

<Source>ERCOT</Source>

<UserID>MMS</UserID>

<MessageID>231232466</MessageID>

</Header>

<Reply>

<ReplyCode>OK</ReplyCode>

</Reply>

<Payload>

<BidSet xmlns="http://www.ercot.com/schema/2007-05/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<tradingDate>2007-01-04</tradingDate>

<SelfArrangedAS>

<mRID>SOMEQSE.20070104.SAA.Reg-Up</mRID>

<externalId>myExternalID45473</externalId>

<status>SUBMITTED</status>

</SelfArrangedAS>

</BidSet>

</Payload>

</ResponseMessage>

The following is an example response where a submission was unsuccessful, as a consequence of invalid parameters on the BidSet:

<ResponseMessage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.ercot.com/schema/2007-05/nodal/ews/msg">

<Header>

<Verb>reply</Verb>

<Noun>BidSet</Noun>

<ReplayDetection>

<Nonce>74646464</Nonce>

<Created>2006-12-09T15:36:03-06:00</Created>

</ReplayDetection>

<Revision>001</Revision>

<Source>ERCOT</Source>

<UserID>MMS</UserID>

<MessageID>231232466</MessageID>

</Header>

<Reply>

<ReplyCode>ERROR</ReplyCode>

<Error>Bad trading date</Error>

</Reply>

</ResponseMessage>

The following is an example response where a submission was unsuccessful, as a consequence of syntax failures within a specific bid/offer/trade/schedule within the BidSet. It is important to note that only the status and errors (if any exist) are returned for each bid.

<ResponseMessage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.ercot.com/schema/2007-05/nodal/ews/msg">

<Header>

<Verb>reply</Verb>

<Noun>BidSet</Noun>

<ReplayDetection>

<Nonce>74646464</Nonce>

<Created>2006-12-09T15:36:03-06:00</Created>

</ReplayDetection>

<Revision>001</Revision>

<Source>ERCOT</Source>

<UserID>MMS</UserID>

<MessageID>231232466</MessageID>

</Header>

<Reply>

<ReplyCode>ERROR</ReplyCode>

<Error>Bid syntax errors</Error>

</Reply>

<Payload>

<BidSet xmlns="http://www.ercot.com/schema/2007-05/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<tradingDate>2007-01-04</tradingDate>

<XYZ>

<status>ERROR</status>

<error><severity>error</severity><text>Unknown bid type XYZ</text></error>

</XYZ>

<ThreePartOffer>

<status>ERROR</status>

<error><severity>error</severity><text>Bad schema</text></error>

</ThreePartOffer>

<COP>

<mRID>SOMEQSE.20070104.COP.<Resource1></mRID>

<externalId>SOMEQSE external ID 123</externalId>

<status>SUBMITTED</status>

</COP>

*... sequence of bids with status and errors (if any) appears here ...*

</BidSet>

</Payload>

</ResponseMessage>

The more detailed validations errors would typically be available after a period of minutes, where a notification would be issued to identify that errors were found.

An important note is that the order of the bids, offers, trades, and schedules within the reply BidSet is identical to the order that was provided in the BidSet for the submission. This is important so that the mRID in the response can be associated with the correct bid, offer, trade, or schedule from the submission.

### Querying Bids, Offers, Trades and Schedules

A query request uses the ‘get’ verb on the message header to query specific bids, offers, trades and schedules. There are two options for querying a set of bids, offers, trades, or schedules:

* The transaction ID (mRID) for each bid, offer, trade, or schedule of interest is identified using the message Request/ID parameter. These must be for the same trade date.
* In all cases only the trading date and a product type is supplied, where only the bids, offers, schedules and trades for that trading date of the specified product type will be returned. This is done using a partial (shortened) mRID in the form ‘<QSE>.<date>.<product>’, such as ‘SOMEQSE.20081112.COP’.

Table below reflects the required short mRID tokens for every bid type:

| *BID Type* | *Type (XML tag)* | *Short mRID Key String* |
| --- | --- | --- |
| Ancillary Service Offer | ASOffer | <QSE>.<TradingDate>.ASO |
| Ancillary Service Trade | ASTrade | <QSE>.<TradingDate>.AST |
| Capacity Trade | CapacityTrade | <QSE>.<TradingDate>.CT |
| Current Operating Plan | COP | <QSE>.<TradingDate>.COP |
| PTP Obligation w/ Links to Option | CRR | <QSE>.<TradingDate>.CRR.Source.Sink |
| DAM Energy Bid | EnergyBid | <QSE>.<TradingDate>.EB.SP |
| DAM Energy-Only Offer | EnergyOnlyOffer | <QSE>.<TradingDate>.EOO.SP |
| Energy Trade | EnergyTrade | <QSE>.<TradingDate>.ET |
| Inc Dec Energy Offer | IncDecOffer | <QSE>.<TradingDate>.IDO |
| Output Schedule | OutputSchedule | <QSE>.<TradingDate>.OS |
| PTP Obligation Bid | PTPObligation | <QSE>.<TradingDate>.PTP.Source.Sink |
| Real-Time Market Energy Bid | RTMEnergyBid | <QSE>.<TradingDate>.REB |
| Self-Arranged AS | SelfArrangedAS | <QSE>.<TradingDate>.SAA |
| Self-Schedule | SelfSchedule | <QSE>.<TradingDate>.SS |
| Three Part Supply Offer | ThreePartOffer | <QSE>.<TradingDate>.TPO |

Figure 64 – Short mRID Key String

In all cases the verb is ‘get’, the noun is ‘BidSet’. Please note that this interface will not return any bids, offers, trades or schedules that were invalid upon submission. The externalId will not be returned, as it is not maintained by MMS.

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | *BidSet* |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/ID | *One or more mRID values, where the products are of the same type and for the same trading date.* |

Figure 65 - get BidSet Parameters

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | *BidSet* |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *May be any number of error messages* |
| Payload | *BidSet, returning details for desired bids, offers, trades and schedules* |

Figure 66 - reply BidSet Parameters

The following is an example response payload:

<BidSet xmlns="http://www.ercot.com/schema/2007-05/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<tradingDate>2008-01-01</tradingDate>

<ASTrade>

<startTime>2008-01-01T00:00:00-06:00</startTime>

<endTime>2008-01-02T00:00:00-06:00</endTime>

<mRID>ACME.20080101.AST.Reg-Up.<BuyerQSE>.<SellerQSE></mRID>

<status>ACCEPTED</status>

<buyer>Acme</buyer>

<seller>Cogswell</seller>

<asType>Reg-Up</asType>

<ASSchedule>

<startTime>2008-01-01T00:00:00-06:00</startTime>

<endTime>2008-01-02T00:00:00-06:00</endTime>

<TmPoint>

<time>2008-01-01T00:00:00-06:00</time>

<value1>40</value1>

</TmPoint>

<TmPoint>

<time>2008-01-01T20:00:00-06:00</time>

<value1>400</value1>

</TmPoint>

</ASSchedule>

</ASTrade>

</BidSet>

### Updating Bids, Offers, Trades and Schedules

A request to update specific bids, offers, trades and schedules is identically the same as a submission request. If the bid, offer, schedule or trade was previously submitted it will be updated. If it did not previously exist, it will be created.

For the purpose of bidding, the ‘create’ and ‘change’ verbs are used interchangeably.

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | change |
| Header/Noun | *BidSet* |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Payload | *BidSet* |

Figure 67 - change BidSet Parameters

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | *BidSet* |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *May be any number of error messages* |
| Payload | *BidSet, returning transaction IDs (mRIDs) for individual bids, offers, trades or schedules and or error messages.* |

Figure 68 - reply BidSet Parameters

The following is an example request payload, where a previously submitted ASTrade will be updated. There may be more than one bid, offer, trade or schedule updated at a time. New submissions may also be mixed with updates.

<BidSet xmlns="http://www.ercot.com/schema/2007-05/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<tradingDate>2008-01-01</tradingDate>

<ASTrade>

<startTime>2008-01-01T00:00:00-06:00</startTime>

<endTime>2008-01-02T00:00:00-06:00</endTime>

<externalId>My External ID 123454</externalId>

<buyer>Acme</buyer>

<seller>Cogswell</seller>

<asType>Reg-Up</asType>

<ASSchedule>

<startTime>2008-01-01T00:00:00-06:00</startTime>

<endTime>2008-01-02T00:00:00-06:00</endTime>

<TmPoint>

<time>2008-01-01T00:00:00-06:00</time>

<value1>40</value1>

</TmPoint>

<TmPoint>

<time>2008-01-01T20:00:00-06:00</time>

<value1>70</value1>

</TmPoint>

</ASSchedule>

</ASTrade>

</BidSet>

### Canceling Bids, Offers, Trades and Schedules

A cancellation request uses the ‘cancel’ verb on the message header in order to cancel one or more query specific bids, offers, trades and schedules, as identified using an mRID. It is important to note that a COP submission can not be canceled, as it is required and can otherwise only be updated. It is also important to note that the mRID values must identify bids, offers, trades and schedules for the same trade date.

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | cancel |
| Header/Noun | *BidSet* |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/ID | *One or more mRID values* |

Figure 69 - cancel BidSet Parameters

When canceling a bid, it is also possible to cancel individual hours of a bid. This would be done by suffixing the mRID of the bid to be canceled with the specific hour. If the hour is not provided, all hours will be canceled.

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | *BidSet* |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *May be any number of error messages* |
| Payload/BidSet | *BidSet identifying bids canceled, identifying mRID and status=CANCELED* |

Figure 70 - reply BidSet Parameters

The following is an example BidSet that would be provided in the response message:

<BidSet xmlns="http://www.ercot.com/schema/2007-05/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<tradingDate>2008-01-01</tradingDate>

<ASOffer>

<mRID>ACME.20080101.ASO.<Resource1>.<Reg-Up></mRID>

<status>CANCELED</status>

</ASOffer>

<ASOffer>

<mRID>ACME.20080101.ASO.<Resource2>.<Reg-Up></mRID>

<status>CANCELED</status>

</ASOffer>

</BidSet>

# Market Information

This service is used to request specific types of market-related information. These are read-only interfaces, as opposed to transactional interfaces.

*Note: The interfaces described in this section are subject to change based upon final EMS and/or MMS designs.*

## Interfaces Provided

Specific interfaces using specific combinations of verbs and nouns (i.e. payload types) are defined to permit a market participant to programmatically access market information. The verb to be used for requests would in all cases be ‘get’. The noun would identify the type of information being requested. Each request could use a message ‘Request’ package to specify one or more parameters that would qualify the request.

The processing sequence is shown in the following sequence diagram:

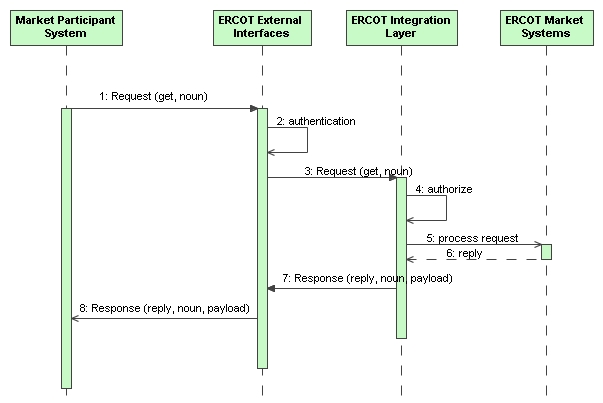


Figure 71 - Market Information Request Sequence Diagram

## Interfaces Required

The messages for market information requests would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | *Name of payload type* |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/? | *Optional: Other request parameters may be specified as needed* |
| Payload | *Message payload data with type defined by Noun* |

Figure 72 - Parameters for 'get' Requests

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | *Defined payload type name* |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *May be any number of error messages* |
| Payload | *Defined payload type* |

Figure 73 - Parameters for 'reply' Messages

In the cases of payloads that would otherwise exceed 1 megabyte, the payloads would be zipped, base64 encoded and stored within the ‘Payload/Compressed’ tag.

## Message Specifications

Specific payload definitions are not currently defined. These will be defined in future revisions.

### AwardSet

The Get AwardSet interface provides the means for a market participant to obtain awards. The following parameters are specified in the RequestMessage:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | AwardSet |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/MarketType | *DAM/SASM* |
| Request/TradingDate | *Trading date* |
| Request/Option | *Optional: SASM ID (required for SASM)*  *Format: YYYYMMDDHHMMSS* |

Figure 74 - Get AwardSet Parameters

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | AwardSet |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/AwardSet | *AwardSet* |

Figure 75 - Get AwardSet Response

An AwardSet is returned in the payload of the ResponseMessage using the following structure:

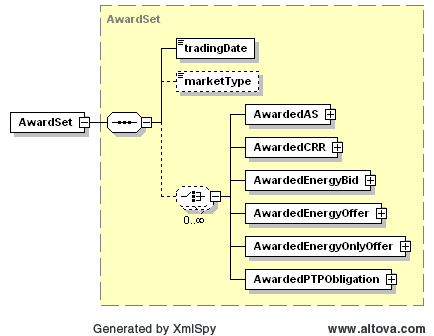


Figure 76 - AwardSet Container Structure

Within an AwardSet, there are a variety of different Award types. The following diagram describes the basic structure for each type of Award:

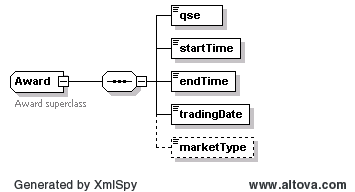


Figure 77 - Generalized Award Structure

The specific structures for each type of Award are described in section 5.3.

The table found in sections 4.3.2 through 4.3.7 describes each Award type elements.

The following is an XML example:

<?xml version="1.0" encoding="UTF-8"?>

<AwardSet>

<tradingDate>2008-04-30</tradingDate>

<marketType>DAM</marketType>

<AwardedEnergyOffer>

<qse>LUMN</qse>

<startTime>2008-04-30T23:00:00-05:00</startTime>

<endTime>2008-05-01T00:00:00-05:00</endTime>

<tradingDate>2008-04-30</tradingDate>

<marketType>DAM</marketType>

<resource> Resource1</resource>

<awardedMWh>0</awardedMWh>

</AwardedEnergyOffer>

<AwardedAS>

<qse>LUMN</qse>

<startTime>2008-04-30T00:00:00-05:00</startTime>

<endTime>2008-04-30T01:00:00-05:00</endTime>

<tradingDate>2008-04-30</tradingDate>

<marketType>DAM</marketType >

<resource>Resource2</resource>

<asType>On-Non-Spin</asType>

<awardedMW>

<startTime>2008-04-30T00:00:00-05:00</startTime>

<endTime>2008-04-30T01:00:00-05:00</endTime>

<OnLineReserves>

<xvalue>0</xvalue>

<block>1</block>

</OnLineReserves>

<OnLineReserves>

<xvalue>0</xvalue>

<block>2</block>

</OnLineReserves>

<OnLineReserves>

<xvalue>0</xvalue>

<block>3</block>

</OnLineReserves>

<OnLineReserves>

<xvalue>0</xvalue>

<block>4</block>

</OnLineReserves>

<OnLineReserves>

<xvalue>0</xvalue>

<block>5</block>

</OnLineReserves>

</awardedMW>

<mcpc>2000</mcpc>

<SASMid>20091108122027</SASMid>

</AwardedAS>

<AwardedPTPObligation>

<qse>LUMN</qse>

<startTime>2008-04-30T00:00:00-05:00</startTime>

<endTime>2008-04-30T01:00:00-05:00</endTime>

<tradingDate>2008-04-30</tradingDate>

<marketType>DAM</marketType >

<awardedMW>0</awardedMW>

<source>Source1</source>

<sink>SINK1</sink>

<price>0</price>

<bidId>01</bidId>

</AwardedPTPObligation>

<AwardedCRR>

<qse>LUMN</qse>

<startTime>2008-04-30T00:00:00-05:00</startTime>

<endTime>2008-04-30T01:00:00-05:00</endTime>

<tradingDate>2008-04-30</tradingDate>

<marketType>DAM</marketType>

<awardedMW>3</awardedMW>

<price>0</price>

<source>Source1</source>

<sink>Sink1</sink>

<crrId>12345</crrId>

<offerId>1234</offerId>

</AwardedCRR>

<AwardedEnergyBid>

<qse>LUMN</qse>

<startTime>2008-04-30T00:00:00-05:00</startTime>

<endTime>2008-04-30T01:00:00-05:00</endTime>

<tradingDate>2008-04-30</tradingDate>

<marketType>DAM</marketType>

<awardedMWh>3</awardedMWh>

<spp>34</spp>

<bidId>1234</bidId>

<sp>1234</sp>

</AwardedEnergyBid>

<AwardedEnergyOnlyOffer>

<qse>LUMN</qse>

<startTime>2008-04-30T00:00:00-05:00</startTime>

<endTime>2008-04-30T01:00:00-05:00</endTime>

<tradingDate>2008-04-30</tradingDate>

<marketType>DAM</marketType>

<awardedMWh>3</awardedMWh>

<spp>34</spp>

<bidId>456</bidId>

<sp>123</sp>

</AwardedEnergyOnlyOffer>

</AwardSet>

### AwardedAS

The Get AwardedAS interface provides the means for a market participant to obtain awarded AS. The following parameters are specified in the RequestMessage:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | AwardedAS |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/MarketType | *DAM/SASM* |
| Request/TradingDate | *Trading date* |
| Request/Option | *Optional: SASM ID (required for SASM)*  *Format: YYYYMMDDHHMMSS* |

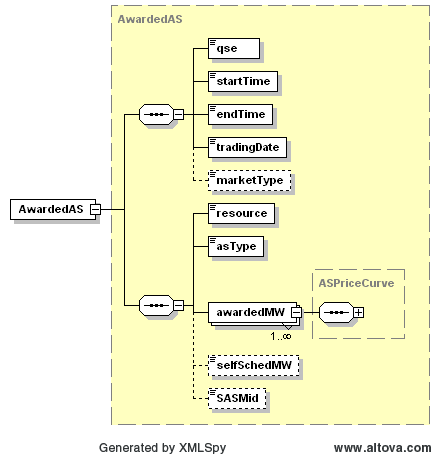
Figure 78 - Get AwardedAS Parameters

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | AwardedAS |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/AwardSet | *AwardedAS* |

Figure 79 - Get AwardedAS Response

On retrieval AwardedAS is returned in the payload of the ResponseMessage using the following structure:



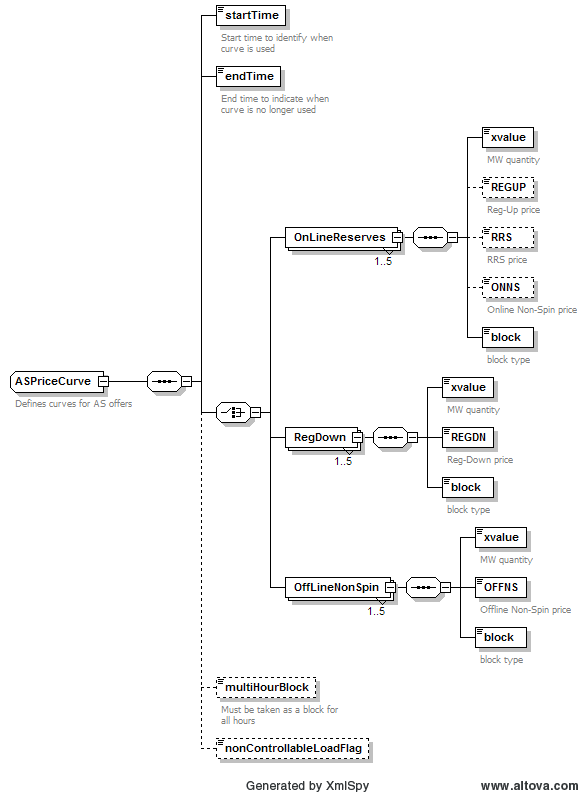


Figure 80 - AwardedAS Container Structure

|  |
| --- |
| ***[NPRR863: Replace Figure 80 above with the following upon system implementation]***  Diagram  Description automatically generated |

The following table describes the AwardedAS elements.

|  |  |  |  |
| --- | --- | --- | --- |
| *Element* | *Datatype* | *Description* | *Values* |
| qse | string | Participant ID |  |
| startTime | DateTime | Start time for the award | Valid award dateTime |
| endTime | DateTime | End time for the award | Valid award dateTime |
| tradingDate | Date | Award date | DAM/SASM execution date |
| marketType | string | Market type | DAM/SASM |
| resource | string | Name of resource |  |
| asType | string | Ancillary services type | See Figure 11 |
| awardedMW/startTime | DateTime | Start time for the award | Valid award dateTime |
| awardedMW/endTime | dateTime | End time for the award | Valid award dateTime |
| awardedMW/OnlineReserve/xvalue | float | Awarded quantity | Awarded MW |
| awardedMW/OnlineReserve/REGUP | float | Price | Note: Also provided by get MCPC for price |
| awardedMW/OnlineReserve/RRS | float | Price | Note: Also provided by get MCPC for price |
| awardedMW/OnlineReserve/ONNS | float | Price | Note: Also provided by get MCPC for price |
| awardedMW/OnlineReserve/block | string | Block number | 1 through 5 |
| awardedMW/RegDown/xvalue | float | Awarded quantity | Awarded MW |
| awardedMW/RegDown/REGDN | float | Price | Note: Also provided by get MCPC for price |
| awardedMW/RegDown/block | string | Block number | 1 through 5 |
| awardedMW/OffLineNonSpin/xvalue | float | Awarded quantity | Awarded MW |
| awardedMW/OffLineNonSpin/ OFFNS | float | Price | Note: Also provided by get MCPC for price |
| awardedMW/OffLineNonSpin/block | string | Block number | 1 through 5 |
| awardedMW/multiHourBlock | Boolean | Indicates if offer must be taken as a block for all hours. | Place Holder Not Used |
| awardedMW/nonControllableLoadFlag | Boolean | Flag indicating Non-Controllable Load | Place Holder Not Used |
| selfSchedMW | Integer |  | Place Holder Not Used |
| SASMid | String | SASM Identifier | provided by MMS  Format: YYYYMMDDHHMMSS |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***[NPRR863: Replace the table above with the following upon system implementation]***   |  |  |  |  | | --- | --- | --- | --- | | *Element* | *Datatype* | *Description* | *Values* | | qse | string | Participant ID |  | | startTime | DateTime | Start time for the award | Valid award dateTime | | endTime | DateTime | End time for the award | Valid award dateTime | | tradingDate | Date | Award date | DAM/SASM execution date | | marketType | string | Market type | DAM/SASM | | resource | string | Name of resource |  | | asType | string | Ancillary services type | See Figure 11 | | awardedMW/startTime | DateTime | Start time for the award | Valid award dateTime | | awardedMW/endTime | dateTime | End time for the award | Valid award dateTime | | awardedMW/OnlineReserve/xvalue | float | Awarded quantity | Awarded MW | | awardedMW/OnlineReserve/REGUP | float | Price | Note: Also provided by get MCPC for price | | awardedMW/OnlineReserve/RRS | float | Price | Note: Also provided by get MCPC for price | | awardedMW/OnlineReserve/ONNS | float | Price | Note: Also provided by get MCPC for price | | awardedMW/OnlineReserve/block | string | Block number | 1 through 5 | | awardedMW/RegDown/xvalue | float | Awarded quantity | Awarded MW | | awardedMW/RegDown/REGDN | float | Price | Note: Also provided by get MCPC for price | | awardedMW/RegDown/block | string | Block number | 1 through 5 | | awardedMW/OffLineNonSpin/xvalue | float | Awarded quantity | Awarded MW | | awardedMW/OffLineNonSpin/ OFFNS | float | Price | Note: Also provided by get MCPC for price | | awardedMW/OffLineNonSpin/block | string | Block number | 1 through 5 | | awardedMW/multiHourBlock | Boolean | Indicates if offer must be taken as a block for all hours. | Place Holder Not Used | | selfSchedMW | Integer |  | Place Holder Not Used | | SASMid | String | SASM Identifier | provided by MMS  Format: YYYYMMDDHHMMSS | |

The following is an XML example:

<ns0:AwardedAS>

<ns0:qse>QSE1</ns0:qse>

<ns0:startTime>2009-08-13T00:00:00-05:00</ns0:startTime>

<ns0:endTime>2009-08-13T01:00:00-05:00</ns0:endTime>

<ns0:tradingDate>2009-08-13</ns0:tradingDate>

<ns0:resource>Resource1</ns0:resource>

<ns0:asType>On-Non-Spin</ns0:asType>

<ns0:awardedMW>

<ns0:startTime>2009-08-13T00:00:00-05:00</ns0:startTime>

<ns0:endTime>2009-08-13T01:00:00-05:00</ns0:endTime>

<ns0:OnLineReserves>

<ns0:xvalue>0</ns0:xvalue>

<ns0:ONNS>0</ns0:ONNS>

<ns0:block>1</ns0:block>

</ns0:OnLineReserves>

<ns0:OnLineReserves>

<ns0:xvalue>0</ns0:xvalue>

<ns0:ONNS>0</ns0:ONNS>

<ns0:block>2</ns0:block>

</ns0:OnLineReserves>

<ns0:OnLineReserves>

<ns0:xvalue>0</ns0:xvalue>

<ns0:ONNS>0</ns0:ONNS>

<ns0:block>3</ns0:block>

</ns0:OnLineReserves>

<ns0:OnLineReserves>

<ns0:xvalue>0</ns0:xvalue>

<ns0:ONNS>0</ns0:ONNS>

<ns0:block>4</ns0:block>

</ns0:OnLineReserves>

……..

……..

<ns0:AwardedAS>

<ns0:qse>QSE1</ns0:qse>

<ns0:startTime>2009-08-13T23:00:00-05:00</ns0:startTime>

<ns0:endTime>2009-08-14T00:00:00-05:00</ns0:endTime>

<ns0:tradingDate>2009-08-13</ns0:tradingDate>

<ns0:resource>Resource1</ns0:resource>

<ns0:asType>RRS</ns0:asType>

<ns0:awardedMW>

<ns0:startTime>2009-08-13T23:00:00-05:00</ns0:startTime>

<ns0:endTime>2009-08-14T00:00:00-05:00</ns0:endTime>

<ns0:OnLineReserves>

<ns0:xvalue>5</ns0:xvalue>

<ns0:RRS>66.47</ns0:RRS>

<ns0:block>1</ns0:block>

</ns0:OnLineReserves>

<ns0:OnLineReserves>

<ns0:xvalue>3</ns0:xvalue>

<ns0:RRS>66.47</ns0:RRS>

<ns0:block>2</ns0:block>

</ns0:OnLineReserves>

<ns0:OnLineReserves>

<ns0:xvalue>0</ns0:xvalue>

<ns0:RRS>66.47</ns0:RRS>

<ns0:block>3</ns0:block>

</ns0:OnLineReserves>

<ns0:OnLineReserves>

<ns0:xvalue>0</ns0:xvalue>

<ns0:RRS>66.47</ns0:RRS>

<ns0:block>4</ns0:block>

</ns0:OnLineReserves>

<ns0:OnLineReserves>

<ns0:xvalue>0</ns0:xvalue>

<ns0:RRS>66.47</ns0:RRS>

<ns0:block>5</ns0:block>

</ns0:OnLineReserves>

<ns0:nonControllableLoadFlag>false</ns0:nonControllableLoadFlag>

</ns0:awardedMW>

</ns0:AwardedAS>

|  |
| --- |
| ***[NPRR863: Replace example above with the following upon system implementation]***  <ns0:AwardedAS>  <ns0:qse>QSE1</ns0:qse>  <ns0:startTime>2009-08-13T00:00:00-05:00</ns0:startTime>  <ns0:endTime>2009-08-13T01:00:00-05:00</ns0:endTime>  <ns0:tradingDate>2009-08-13</ns0:tradingDate>  <ns0:resource>Resource1</ns0:resource>  <ns0:asType>On-Non-Spin</ns0:asType>  <ns0:awardedMW>  <ns0:startTime>2009-08-13T00:00:00-05:00</ns0:startTime>  <ns0:endTime>2009-08-13T01:00:00-05:00</ns0:endTime>  <ns0:OnLineReserves>  <ns0:xvalue>0</ns0:xvalue>  <ns0:ONNS>0</ns0:ONNS>  <ns0:block>1</ns0:block>  </ns0:OnLineReserves>  <ns0:OnLineReserves>  <ns0:xvalue>0</ns0:xvalue>  <ns0:ONNS>0</ns0:ONNS>  <ns0:block>2</ns0:block>  </ns0:OnLineReserves>  <ns0:OnLineReserves>  <ns0:xvalue>0</ns0:xvalue>  <ns0:ONNS>0</ns0:ONNS>  <ns0:block>3</ns0:block>  </ns0:OnLineReserves>  <ns0:OnLineReserves>  <ns0:xvalue>0</ns0:xvalue>  <ns0:ONNS>0</ns0:ONNS>  <ns0:block>4</ns0:block>  </ns0:OnLineReserves>  ……..  ……..  <ns0:AwardedAS>  <ns0:qse>QSE1</ns0:qse>  <ns0:startTime>2009-08-13T23:00:00-05:00</ns0:startTime>  <ns0:endTime>2009-08-14T00:00:00-05:00</ns0:endTime>  <ns0:tradingDate>2009-08-13</ns0:tradingDate>  <ns0:resource>Resource1</ns0:resource>  <ns0:asType>RRSPF</ns0:asType>  <ns0:awardedMW>  <ns0:startTime>2009-08-13T23:00:00-05:00</ns0:startTime>  <ns0:endTime>2009-08-14T00:00:00-05:00</ns0:endTime>  <ns0:OnLineReserves>  <ns0:xvalue>5</ns0:xvalue>  <ns0:RRS>66.47</ns0:RRS>  <ns0:block>1</ns0:block>  </ns0:OnLineReserves>  <ns0:OnLineReserves>  <ns0:xvalue>3</ns0:xvalue>  <ns0:RRS>66.47</ns0:RRS>  <ns0:block>2</ns0:block>  </ns0:OnLineReserves>  <ns0:OnLineReserves>  <ns0:xvalue>0</ns0:xvalue>  <ns0:RRS>66.47</ns0:RRS>  <ns0:block>3</ns0:block>  </ns0:OnLineReserves>  <ns0:OnLineReserves>  <ns0:xvalue>0</ns0:xvalue>  <ns0:RRS>66.47</ns0:RRS>  <ns0:block>4</ns0:block>  </ns0:OnLineReserves>  <ns0:OnLineReserves>  <ns0:xvalue>0</ns0:xvalue>  <ns0:RRS>66.47</ns0:RRS>  <ns0:block>5</ns0:block>  </ns0:OnLineReserves>  </ns0:awardedMW>  </ns0:AwardedAS> |

### AwardedCRR

The Get AwardedCRR interface provides the means for a market participant to obtain awarded CRRs from the DAM market, not the CRR auction. The following parameters are specified in the RequestMessage:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | AwardedCRR |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/TradingDate | *Trading date* |

Figure 81 - Get AwardedCRR Parameters

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | AwardedCRR |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/AwardSet | *AwardedCRR* |

Figure 82 - Get AwardedCRR Response

An AwardedCRR is returned in the payload of the ResponseMessage using the following structure:

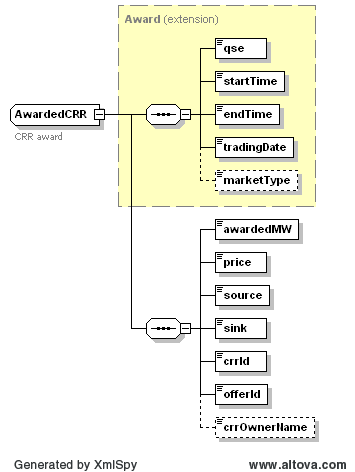


Figure 83 - AwardedCRR Container Structure

The following table describes the AwardedCRR elements.

|  |  |  |  |
| --- | --- | --- | --- |
| *Element* | *Datatype* | *Description* | *Values* |
| qse | string | Participant ID |  |
| startTime | DateTime | Start time for the award | Valid award dateTime |
| endTime | DateTime | End time for the award | Valid award dateTime |
| tradingDate | Date | Award date | Valid award date |
| marketType | string | Market type | DAM |
| awardedMW | Float | Awarded quantity | Awarded MW |
| price | Float | Awarded Price |  |
| source | string | Source settlement point | Valid settlement point name |
| sink | string | Sink settlement point | Valid settlement point name |
| crrId | string | CRR ID | ID of CRR |
| offerId | string | Offer ID | QSE supplied |
| crrOwnerName | string | CRR Owner |  |

The following is an XML example:

<?xml version="1.0" encoding="UTF-8"?>

<AwardSet>

<tradingDate>2008-04-30</tradingDate>

<marketType>DAM</marketType>

<AwardedCRR>

<qse>LUMN</qse>

<startTime>2008-04-30T00:00:00-05:00</startTime>

<endTime>2008-04-30T01:00:00-05:00</endTime>

<tradingDate>2008-04-30</tradingDate>

<marketType>DAM</marketType>

<awardedMW>3</awardedMW>

<price>0</price>

<source>BBSES\_UNIT1</source>

<sink>BBSES\_UNIT2</sink>

<crrId>12345</crrId>

<offerId>1234</offerId>

</AwardedCRR>

</AwardSet>

### AwardedEnergyBid

The Get AwardedEnergyBid interface provides the means for a market participant to obtain EnergyBid awards. The following parameters are specified in the RequestMessage:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | AwardedEnergyBid |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/TradingDate | *Trading date* |

Figure 84 - Get AwardedEnergyBid Parameters

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | AwardedEnergyBid |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/AwardSet | *AwardedEnergyBid* |

Figure 85 - Get AwardedEnergyBid Response

An AwardedEnergyBid is returned in the payload of the ResponseMessage using the following structure:

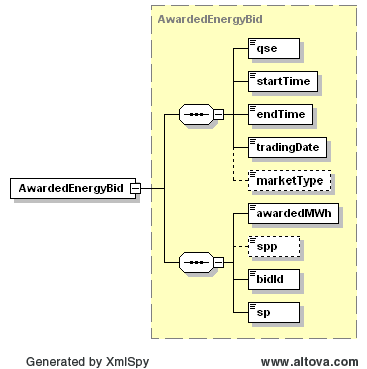


Figure 86 - AwardedEnergyBid Container Structure

The following table describes the AwardedEnergyBid elements.

|  |  |  |  |
| --- | --- | --- | --- |
| *Element* | *Datatype* | *Description* | *Values* |
| qse | string | Participant ID |  |
| startTime | DateTime | Start time for the award | Valid award dateTime |
| endTime | DateTime | End time for the award | Valid award dateTime |
| tradingDate | Date | Award date | Valid award date |
| marketType | string | Market type | DAM |
| awardedMW | Float | Awarded quantity | Awarded MWh |
| spp | Float | SPP value | Place Holder |
| bidId[[4]](#footnote-4) | string | Bid ID | MP supplied bid ID |
| sp | string | Settlement point | Valid settlement point name |

The following is an XML example:

<?xml version="1.0" encoding="UTF-8"?>

<AwardSet>

<tradingDate>2008-04-30</tradingDate>

<marketType>DAM</marketType>

<AwardedEnergyBid>

<qse>LUMN</qse>

<startTime>2008-04-30T00:00:00-05:00</startTime>

<endTime>2008-04-30T01:00:00-05:00</endTime>

<tradingDate>2008-04-30</tradingDate>

<marketType>DAM</marketType>

<awardedMWh>3</awardedMWh>

<spp>34</spp>

<bidId>1234</bidId>

<sp>1234</sp>

</AwardedEnergyBid>

</AwardSet>

### AwardedEnergyOffer

The Get AwardedEnergyOffer interface provides the means for a market participant to obtain EnergyOffer awards. The following parameters are specified in the RequestMessage:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | AwardedEnergyOffer |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/TradingDate | *Trading date* |

Figure 87 - Get AwardedEnergyOffer Parameters

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | AwardedEnergyOffer |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/AwardSet | *AwardedEnergyOffer* |

Figure 88 - Get AwardedEnergyOffer Response

An AwardedEnergyOffer is returned in the payload of the ResponseMessage using the following structure:

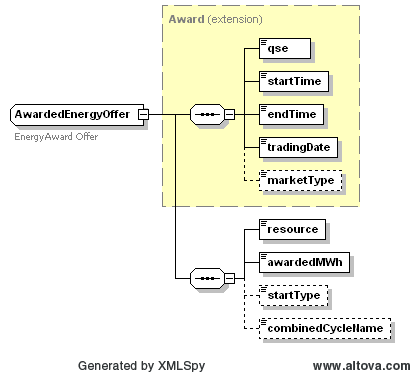


Figure 89 - AwardedEnergyOffer Container Structure

The following table describes the AwardedEnergyOffer elements.

|  |  |  |  |
| --- | --- | --- | --- |
| *Element* | *Datatype* | *Description* | *Values* |
| Qse | string | Participant ID |  |
| startTime | DateTime | Start time for the award | Valid award dateTime |
| endTime | DateTime | End time for the award | Valid award dateTime |
| tradingDate | Date | Award date | Valid award date |
| marketType | string | Market type | DAM |
| Resource | string | Resource | Valid resource name |
| awardedMWh | Float | Awarded quantity | Awarded MWh |
| startType | string |  | HOT or INTER or COLD |
| combinedCycleName | string |  |  |

The following is an XML example:

<?xml version="1.0" encoding="UTF-8"?>

<AwardSet>

<tradingDate>2008-04-30</tradingDate>

<marketType>DAM</marketType>

<AwardedEnergyOffer>

<qse>TestQSE</qse>

<startTime>2008-04-30T23:00:00-05:00</startTime>

<endTime>2008-05-01T00:00:00-05:00</endTime>

<tradingDate>2008-04-30</tradingDate>

<marketType>DAM</marketType>

<resource>Resource1</resource>

<awardedMWh>126.6</awardedMWh>

<startType>HOT</startType>

<combinedCycleName>CC1</combinedCycleName>

</AwardedEnergyOffer>

</AwardSet>

### AwardedEnergyOnlyOffer

The Get AwardedEnergyOnlyOffer interface provides the means for a market participant to obtain EnergyOnlyOffer awards. The following parameters are specified in the RequestMessage:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | AwardedEnergyOnlyOffer |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/TradingDate | *Trading date* |

Figure 90 - Get AwardedEnergyOnlyOffer Parameters

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | AwardedEnergyOnlyOffer |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/AwardSet | *AwardedEnergyOnlyOffer* |

Figure 91 - Get AwardedEnergyOnlyOffer Response

An AwardedEnergyOnlyOffer is returned in the payload of the ResponseMessage using the following structure:

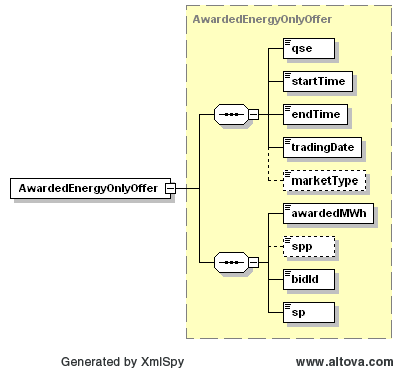


Figure 92 - AwardedEnergyOnlyOffer Container Structure

The following table describes the AwardedEnergyOnlyOffer elements.

|  |  |  |  |
| --- | --- | --- | --- |
| *Element* | *Datatype* | *Description* | *Values* |
| qse | string | Participant ID |  |
| startTime | DateTime | Start time for the award | Valid award dateTime |
| endTime | DateTime | End time for the award | Valid award dateTime |
| tradeingDate | Date | Award date | Valid award date |
| marketType | string | Market type | DAM |
| awardedMWh | Float | Awarded quantity | Awarded MWh |
| spp | Float | SPP value | Place Holder |
| bidId[[5]](#footnote-5) | string | Bid ID | MP supplied bid ID |
| sp | string | Settlement point | Valid settlement point name |

The following is an XML example:

<?xml version="1.0" encoding="UTF-8"?>

<AwardSet>

<tradingDate>2008-04-30</tradingDate>

<marketType>DAM</marketType>

<AwardedEnergyOnlyOffer>

<qse>LUMN</qse>

<startTime>2008-04-30T00:00:00-05:00</startTime>

<endTime>2008-04-30T01:00:00-05:00</endTime>

<tradingDate>2008-04-30</tradingDate>

<marketType>DAM</marketType>

<awardedMWh>3</awardedMWh>

<spp>34</spp>

<bidId>456</bidId>

<sp>123</sp>

</AwardedEnergyOnlyOffer>

</AwardSet>

### AwardedPTPObligation

The Get AwardedPTPObligation interface provides the means for a market participant to obtain PTPObligation awards. The following parameters are specified in the RequestMessage:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | AwardedPTPObligation |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/TradingDate | *Trading date* |

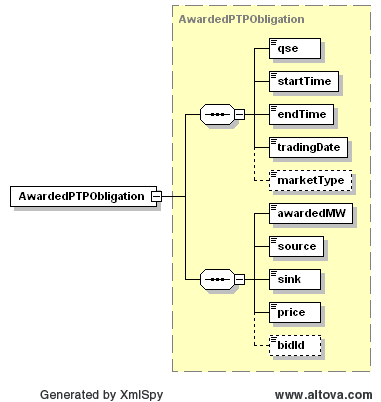
Figure 93 - Get AwardedPTPObligation Parameters

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | AwardedPTPObligation |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/AwardSet | *AwardedPTPObligation* |

Figure 94 - Get AwardedPTPObligation Response

An AwardedPTPObligation is returned in the payload of the ResponseMessage using the following structure:



The following table describes the AwardedPTPObligation Offer elements.

|  |  |  |  |
| --- | --- | --- | --- |
| *Element* | *Datatype* | *Description* | *Values* |
| qse | string | Participant ID |  |
| startTime | DateTime | Start time for the award | Valid award dateTime |
| endTime | DateTime | End time for the award | Valid award dateTime |
| tradingDate | Date | Award date | Valid award date |
| marketType | string | Market type | DAM |
| awardedMW | Float | Awarded quantity | Awarded MW |
| source | string | Source settlement point | Valid settlement point name |
| sink | string | Sink settlement point | Valid settlement point name |
| price | Float | Awarded Price |  |
| bidId[[6]](#footnote-6) | string | Bid ID | MP supplied bid ID |

Figure 95 - AwardedPTPObligation Container Structure

The following is an XML example:

<?xml version="1.0" encoding="UTF-8"?>

<AwardSet>

<tradingDate>2008-04-30</tradingDate>

<marketType>DAM</marketType>

<AwardedPTPObligation>

<qse>LUMN</qse>

<startTime>2008-04-30T00:00:00-05:00</startTime>

<endTime>2008-04-30T01:00:00-05:00</endTime>

<tradingDate>2008-04-30</tradingDate>

<marketType>DAM</marketType >

<awardedMW>0</awardedMW>

<source>BBSES\_UNIT1</source>

<sink>LZ\_NORTH</sink>

<price>0</price>

<bidId>01</bidId>

</AwardedPTPObligation>

</AwardSet>

### Forecasted Load

The purpose of this interface is to provide the means to obtain last available forecasted load values for next 7 days (168 hours) from ERCOT. The forecasted loads will be medium-term (MT). The input parameters to this request include:

* Possible values for “Zone” are:

1- "WEATHER" - Load forecast for each weather zone of:

* + Coast,
  + East,
  + FarWest,
  + North,
  + NorthCentral,
  + SouthCentral,
  + Southern,
  + West

2- "LOAD" - Load forecast for each load zone

* + North,
  + South,
  + West,
  + Houston

3 - "SYSTEM" – Total load forecast for all zones combined (one value)

The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | LoadForecasts |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/Zone | *Zone Type: "WEATHER" or "LOAD" or "SYSTEM"* |

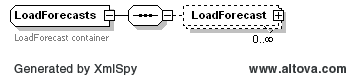
Figure 96 – Load Forecast Request Parameters

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | LoadForecasts |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload | LoadForecasts |

Figure 97 – Load Forecast Response Parameters

The following diagram describes the structure of a load forecast, which is based upon a TmSchedule. For medium term load forecasts, the values of ‘value1’ indicate the hourly forecasted megawatts:



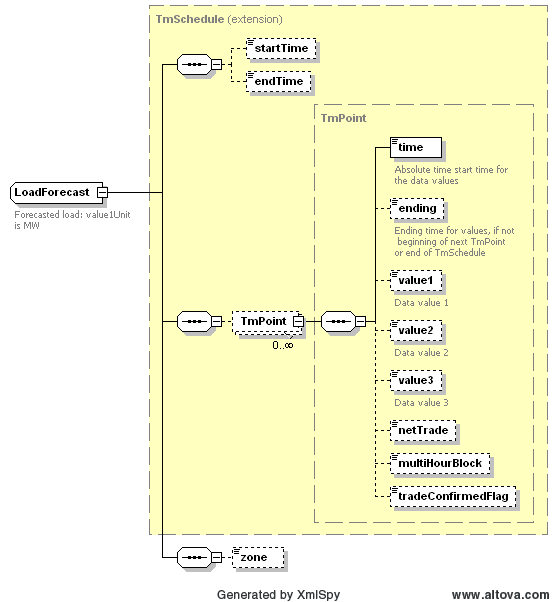


Figure 98 - Load Forecasts Structure

The following is an XML example for load forecast for *WEATHER* zone

<?xml version="1.0" encoding="UTF-8"?>

<ns0:LoadForecasts xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns0:LoadForecast>

<ns0:TmPoint>

<ns0:time>2009-06-10T00:00:00-05:00</ns0:time>

<ns0:ending>2009-06-10T01:00:00-05:00</ns0:ending>

<ns0:value1>31259.3</ns0:value1>

</ns0:TmPoint>

<ns0:TmPoint>

<ns0:time>2009-06-10T01:00:00-05:00</ns0:time>

<ns0:ending>2009-06-10T02:00:00-05:00</ns0:ending>

<ns0:value1>33056.3</ns0:value1>

</ns0:TmPoint>

<ns0:TmPoint>

<ns0:time>2009-06-10T02:00:00-05:00</ns0:time>

<ns0:ending>2009-06-10T03:00:00-05:00</ns0:ending>

<ns0:value1>31692.5</ns0:value1>

</ns0:TmPoint>

<ns0:TmPoint>

<ns0:time>2009-06-10T03:00:00-05:00</ns0:time>

<ns0:ending>2009-06-10T04:00:00-05:00</ns0:ending>

<ns0:value1>30107.5</ns0:value1>

</ns0:TmPoint>

<ns0:TmPoint>

<ns0:time>2009-06-10T04:00:00-05:00</ns0:time>

<ns0:ending>2009-06-10T05:00:00-05:00</ns0:ending>

<ns0:value1>30132.2</ns0:value1>

</ns0:TmPoint>

……………………….

……………………….

<ns0:TmPoint>

<ns0:time>2009-06-16T22:00:00-05:00</ns0:time>

<ns0:ending>2009-06-16T23:00:00-05:00</ns0:ending>

<ns0:value1>38358</ns0:value1>

</ns0:TmPoint>

<ns0:TmPoint>

<ns0:time>2009-06-16T23:00:00-05:00</ns0:time>

<ns0:ending>2009-06-17T00:00:00-05:00</ns0:ending>

<ns0:value1>34050.8</ns0:value1>

</ns0:TmPoint>

<ns0:zone>Coast</ns0:zone>

………………………..

………………………..

<ns0:TmPoint>

<ns0:time>2009-06-16T22:00:00-05:00</ns0:time>

<ns0:ending>2009-06-16T23:00:00-05:00</ns0:ending>

<ns0:value1>11193.6</ns0:value1>

</ns0:TmPoint>

<ns0:TmPoint>

<ns0:time>2009-06-16T23:00:00-05:00</ns0:time>

<ns0:ending>2009-06-17T00:00:00-05:00</ns0:ending>

<ns0:value1>9952.9</ns0:value1>

</ns0:TmPoint>

<ns0:zone>East</ns0:zone>……………………..

……………………..

………………………..

………………………..

<ns0:TmPoint>

<ns0:time>2009-06-16T22:00:00-05:00</ns0:time>

<ns0:ending>2009-06-16T23:00:00-05:00</ns0:ending>

<ns0:value1>5550.2</ns0:value1>

</ns0:TmPoint>

<ns0:TmPoint>

<ns0:time>2009-06-16T23:00:00-05:00</ns0:time>

<ns0:ending>2009-06-17T00:00:00-05:00</ns0:ending>

<ns0:value1>4791.5</ns0:value1>

</ns0:TmPoint>

<ns0:zone>West</ns0:zone>

</ns0:LoadForecast>

</ns0:LoadForecasts>

The following is an XML example for load forecast for *LOAD* zone:

<ns0:LoadForecasts xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns0:LoadForecast>

<ns0:TmPoint>

<ns0:time>2009-06-10T00:00:00-05:00</ns0:time>

<ns0:ending>2009-06-10T01:00:00-05:00</ns0:ending>

<ns0:value1>14085</ns0:value1>

</ns0:TmPoint>

<ns0:TmPoint>

<ns0:time>2009-06-10T01:00:00-05:00</ns0:time>

<ns0:ending>2009-06-10T02:00:00-05:00</ns0:ending>

<ns0:value1>14338.6</ns0:value1>

</ns0:TmPoint>

<ns0:TmPoint>

<ns0:time>2009-06-10T02:00:00-05:00</ns0:time>

<ns0:ending>2009-06-10T03:00:00-05:00</ns0:ending>

<ns0:value1>13735.4</ns0:value1>

</ns0:TmPoint>

………………..

<ns0:TmPoint>

<ns0:time>2009-06-16T22:00:00-05:00</ns0:time>

<ns0:ending>2009-06-16T23:00:00-05:00</ns0:ending>

<ns0:value1>18145.2</ns0:value1>

</ns0:TmPoint>

<ns0:TmPoint>

<ns0:time>2009-06-16T23:00:00-05:00</ns0:time>

<ns0:ending>2009-06-17T00:00:00-05:00</ns0:ending>

<ns0:value1>15799.2</ns0:value1>

</ns0:TmPoint>

<ns0:zone>North</ns0:zone>

……………………..

<ns0:TmPoint>

<ns0:time>2009-06-16T22:00:00-05:00</ns0:time>

<ns0:ending>2009-06-16T23:00:00-05:00</ns0:ending>

<ns0:value1>12060</ns0:value1>

</ns0:TmPoint>

<ns0:TmPoint>

<ns0:time>2009-06-16T23:00:00-05:00</ns0:time>

<ns0:ending>2009-06-17T00:00:00-05:00</ns0:ending>

<ns0:value1>10513</ns0:value1>

</ns0:TmPoint>

<ns0:zone>South</ns0:zone>

…………………

<ns0:TmPoint>

<ns0:time>2009-06-16T22:00:00-05:00</ns0:time>

<ns0:ending>2009-06-16T23:00:00-05:00</ns0:ending>

<ns0:value1>9257</ns0:value1>

</ns0:TmPoint>

<ns0:TmPoint>

<ns0:time>2009-06-16T23:00:00-05:00</ns0:time>

<ns0:ending>2009-06-17T00:00:00-05:00</ns0:ending>

<ns0:value1>8217.2</ns0:value1>

</ns0:TmPoint>

<ns0:zone>Houston</ns0:zone>

</ns0:LoadForecast>

</ns0:LoadForecasts>

The following is an XML example for load forecast for *SYSTEM* zone:

<ns1:LoadForecasts xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotInformation.xsd">

<ns1:LoadForecast>

<ns1:TmPoint>

<ns1:time>2009-06-10T00:00:00-05:00</ns1:time>

<ns1:ending>2009-06-10T01:00:00-05:00</ns1:ending>

<ns1:value1>33407.8</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-10T01:00:00-05:00</ns1:time>

<ns1:ending>2009-06-10T02:00:00-05:00</ns1:ending>

<ns1:value1>34466.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-10T02:00:00-05:00</ns1:time>

<ns1:ending>2009-06-10T03:00:00-05:00</ns1:ending>

<ns1:value1>33133.3</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-10T03:00:00-05:00</ns1:time>

<ns1:ending>2009-06-10T04:00:00-05:00</ns1:ending>

<ns1:value1>31936.6</ns1:value1>

</ns1:TmPoint>

…………….

<ns1:TmPoint>

<ns1:time>2009-06-16T22:00:00-05:00</ns1:time>

<ns1:ending>2009-06-16T23:00:00-05:00</ns1:ending>

<ns1:value1>42035.6</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-16T23:00:00-05:00</ns1:time>

<ns1:ending>2009-06-17T00:00:00-05:00</ns1:ending>

<ns1:value1>36952.8</ns1:value1>

</ns1:TmPoint>

<ns1:zone>System</ns1:zone>

</ns1:LoadForecast>

</ns1:LoadForecasts>

### System Load \*\*\* Same as 4.3.33 - Total ERCOT Load \*\*\*

The purpose of this interface is to provide the means to obtain real-time system load from ERCOT. The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | SystemLoad |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/StartTime | *Start time of interest* |
| Request/EndTime | *End time of interest* |

Figure 99 - SystemLoad Request Parameters

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | SystemLoad |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload | SystemLoad |

Figure 100 - SystemLoad Response Parameters

The structure of SystemLoad is described by the following diagram. This used a TmSchedule:

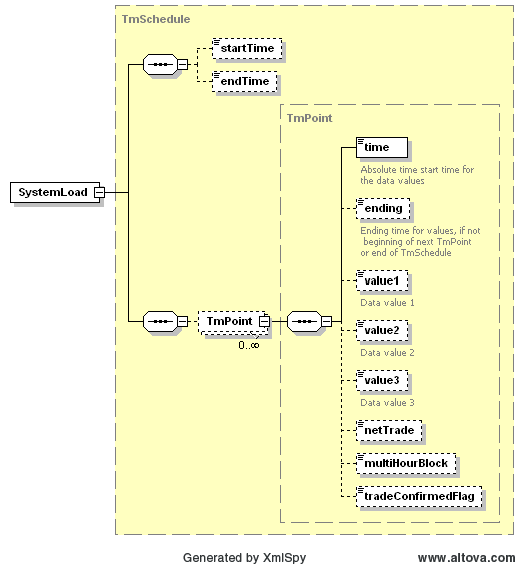


Figure 101 - SystemLoad Structure

The following is an XML example:

<?xml version="1.0" encoding="UTF-8"?>

<SystemLoad xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotInformation.xsd" xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<startTime>2007-07-25T00:00:00-06:00</startTime>

<endTime>2007-07-25T00:00:00-06:00</endTime>

<TmPoint>

<time>2007-07-25T00:00:00-06:00</time>

<ending>2007-07-25T00:00:00-06:00</ending>

<value1>12.14</value1>

</TmPoint>

</SystemLoad>

### Market Totals \*\*\* Same as 4.3.24 - Total DAM Energy \*\*\*

The purpose of this interface is to provide a query for market totals. Market totals supported by this interface include:

* Total energy bought in DAM
* Total energy sold in DAM

The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | MarketTotals |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/option | *One of:*   * *EnergyBoughtInDAM* * *EnergySoldInDAM* |
| Request/OperatingDate | *Day of the report* |

Figure 102 - Market Totals Request Parameters

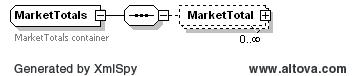
The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | MarketTotals |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload | MarketTotals |

Figure 103 - MarketTotals Response Parameters

The Totals structure is based upon a CIM RegularIntervalSchedules, where the values of ‘value1’ are total megawatt values.

The structure of a MarketTotals payload is shown by the following diagram:



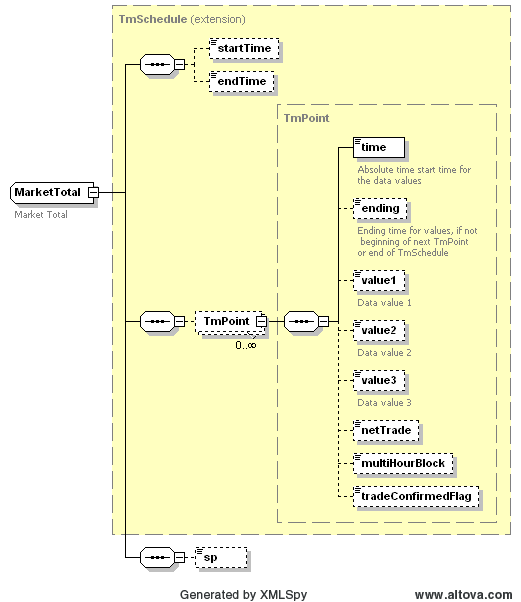


Figure 104 - MarketTotals Structure

The following is an XML example:

<ns0:MarketTotals xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns0:MarketTotal>

<ns0:TmPoint>

<ns0:time>2009-06-13T00:00:00-05:00</ns0:time>

<ns0:ending>2009-06-13T01:00:00-05:00</ns0:ending>

<ns0:value1>0</ns0:value1>

</ns0:TmPoint>

<ns0:sp>DIB\_PUN1</ns0:sp>

</ns0:MarketTotal>

<ns0:MarketTotal>

<ns0:TmPoint>

<ns0:time>2009-06-13T00:00:00-05:00</ns0:time>

<ns0:ending>2009-06-13T01:00:00-05:00</ns0:ending>

<ns0:value1>0</ns0:value1>

</ns0:TmPoint>

<ns0:sp>DUKE\_CC1</ns0:sp>

</ns0:MarketTotal>

<ns0:MarketTotal>

<ns0:TmPoint>

<ns0:time>2009-06-13T00:00:00-05:00</ns0:time>

<ns0:ending>2009-06-13T01:00:00-05:00</ns0:ending>

<ns0:value1>0</ns0:value1>

</ns0:TmPoint>

<ns0:sp>EXN\_PUN1</ns0:sp>

</ns0:MarketTotal>

<ns0:MarketTotal>

<ns0:TmPoint>

<ns0:time>2009-06-13T00:00:00-05:00</ns0:time>

<ns0:ending>2009-06-13T01:00:00-05:00</ns0:ending>

<ns0:value1>37</ns0:value1>

</ns0:TmPoint>

<ns0:sp>DIB\_DIB\_G1</ns0:sp>

</ns0:MarketTotal>

………………………..

<ns0:MarketTotal>

<ns0:TmPoint>

<ns0:time>2009-06-13T23:00:00-05:00</ns0:time>

<ns0:ending>2009-06-14T00:00:00-05:00</ns0:ending>

<ns0:value1>0</ns0:value1>

</ns0:TmPoint>

<ns0:sp>FTR\_CC1</ns0:sp>

</ns0:MarketTotal>

<ns0:MarketTotal>

<ns0:TmPoint>

<ns0:time>2009-06-13T23:00:00-05:00</ns0:time>

<ns0:ending>2009-06-14T00:00:00-05:00</ns0:ending>

<ns0:value1>0</ns0:value1>

</ns0:TmPoint>

<ns0:sp>\_SL\_PUN1</ns0:sp>

</ns0:MarketTotal>

</ns0:MarketTotals>

### Market LMPs, SPPs, Price Corrected LMPs and Price Corrected SPPs

The purpose of this interface is to provide a query for market LMPs, LMPsPC (price corrected LMPs), SPPs and SPPsPC (price corrected SPPs). The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | LMPs (or LMPsPC, SPPs, SPPsPC) |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/OperatingDate | *Day of the Report*  *For DAM MarketType* |
| Request/StartTime | *Start time of interest*  *For RTM MarketType* |
| Request/EndTime | *End time of interest*  *For RTM MarketType* |
| Request/MarketType | *DAM or RTM* |

Figure 105 - LMP and SPP Request Parameters

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | LMPs (or LMPsPC, SPPs, SPPsPC) |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | *LMPs (or SPPs)* |

Figure 106 - LMP and SPP Response Parameters

Please note that requesting LMPs and LMPsPC for DAM market is not supported at this time due to its large response payload size. LMPs and LMPsPC values for DAM market can be retrieved via CDR reports or “get Reports” interface specified in section 9.0.

For RTM Market type, the last SCED run time is returned in the TmPoint/time element. The structure of a LMPs/LMPsPC/SPPs/SPPsPC payload is shown by the following diagram, where each LMP/SPP uses a TmSchedule type for each bus or SettlementPoint:

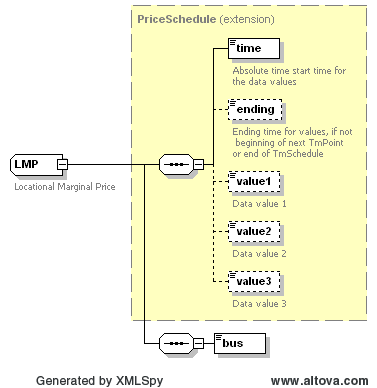
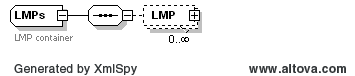
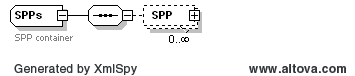


Figure 107 - LMP and LMPsPC Container Structure



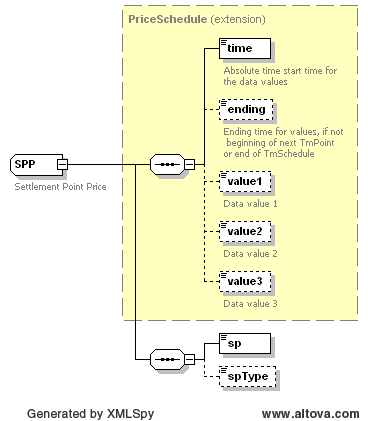


Figure 108 - SPP and SPPsPC Container Structure

In both cases the data structure is based upon a TmSchedule, where values of ‘value1’ and ‘value2’ are in $/MWh and values of ‘time’ is the absolute start time for the data values. The last point on the schedule is valid until the endTime. For LMPsPC and SPPsPC responses, ‘value1’ represents the original price and ‘value2’ represents the corrected price. The ‘value2’ element will only be populated for LMPsPC and SPPsPC responses and is not applicable to LMPs and SPPs responses.

For SPPs and SPPsPC the Settlement Point Type (spType) is an optional field and it is not applicable to DAM market but it is populated for the RTM market.  It’s possible values are: RES, LZ, HUB.

The following is an XML example for SPPs:

<ns0:SPPs xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns0:SPP>

<ns0:time>2009-06-20T00:00:00-05:00</ns0:time>

<ns0:ending>2009-06-20T01:00:00-05:00</ns0:ending>

<ns0:value1>43</ns0:value1>

<ns0:sp>PSA\_CC1</ns0:sp>

</ns0:SPP>

<ns0:SPP>

<ns0:time>2009-06-20T00:00:00-05:00</ns0:time>

<ns0:ending>2009-06-20T01:00:00-05:00</ns0:ending>

<ns0:value1>43</ns0:value1>

<ns0:sp>FORMOSA\_CC1</ns0:sp>

</ns0:SPP>

<ns0:SPP>

<ns0:time>2009-06-20T00:00:00-05:00</ns0:time>

<ns0:ending>2009-06-20T01:00:00-05:00</ns0:ending>

<ns0:value1>43</ns0:value1>

<ns0:sp>LZ\_SOUTH</ns0:sp>

</ns0:SPP>

<ns0:SPP>

<ns0:time>2009-06-20T00:00:00-05:00</ns0:time>

<ns0:ending>2009-06-20T01:00:00-05:00</ns0:ending>

<ns0:value1>43</ns0:value1>

<ns0:sp>LZ\_WEST</ns0:sp>

</ns0:SPP>

<ns0:SPP>

…

</ns0:SPPs>

And the following is an XML example for LMPs:

<ns0:LMPs xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns0:LMP>

<ns0:time>2008-06-19T17:15:22-05:00</ns0:time>

<ns0:value1>716.1</ns0:value1>

<ns0:bus>0ABEC\_\_0001</ns0:bus>

</ns0:LMP>

<ns0:LMP>

<ns0:time>2008-06-19T17:15:22-05:00</ns0:time>

<ns0:value1>716.1</ns0:value1>

<ns0:bus>0ABEC\_\_0002</ns0:bus>

</ns0:LMP>

<ns0:LMP>

<ns0:time>2008-06-19T17:15:22-05:00</ns0:time>

<ns0:value1>716.1</ns0:value1>

<ns0:bus>0AMIST\_0001</ns0:bus>

</ns0:LMP>

………….

<ns0:LMP>

<ns0:time>2008-06-19T17:15:22-05:00</ns0:time>

<ns0:value1>716.1</ns0:value1>

<ns0:bus>\_WZ\_\_\_\_0004</ns0:bus>

</ns0:LMP>

</ns0:LMPs>

### Market MCPCs

The purpose of this interface is to provide a query for MCPCs, for either the DAM or SASM market. Specific SASM ID is needed for SASM market type.

The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | MCPCs |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/MarketType | *DAM or SASM* |
| Request/TradingDate | *Trading date* |
| Request/Option | *Optional: SASM ID (required for SASM)*  *Format: YYYYMMDDHHMMSS* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | MCPCs |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | *MCPCs* |

The following diagram describes the structure of MCPCs:

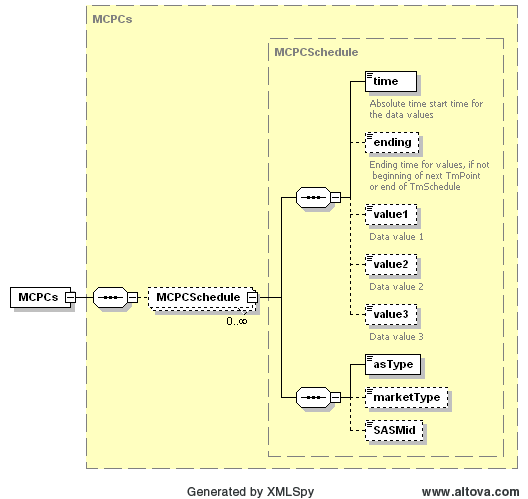


Figure 109 - MCPC Container Structure

Within each MCPCSchedule, an asType is specified and the values of ‘value1’ convey individual MCPC values.

The following is an XML example:

<ns1:MCPCs xmlns:ns0 = "http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1 = "http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:MCPCSchedule>

<ns1:time>2009-08-13T00:00:00-05:00</ns1:time>

<ns1:ending>2009-08-13T01:00:00-05:00</ns1:ending>

<ns1:value1>0</ns1:value1>

<ns1:asType>Non-Spin</ns1:asType>

<ns1:marketType>DAM</ns1:marketType>

</ns1:MCPCSchedule>

<ns1:MCPCSchedule>

<ns1:time>2009-08-13T00:00:00-05:00</ns1:time>

<ns1:ending>2009-08-13T01:00:00-05:00</ns1:ending>

<ns1:value1>20.59</ns1:value1>

<ns1:asType>Reg-Down</ns1:asType>

<ns1:marketType>DAM</ns1:marketType>

</ns1:MCPCSchedule>

<ns1:MCPCSchedule>

<ns1:time>2009-08-13T00:00:00-05:00</ns1:time>

<ns1:ending>2009-08-13T01:00:00-05:00</ns1:ending>

<ns1:value1>22.22</ns1:value1>

<ns1:asType>Reg-Up</ns1:asType>

<ns1:marketType>DAM</ns1:marketType>

</ns1:MCPCSchedule>

<ns1:MCPCSchedule>

<ns1:time>2009-08-13T00:00:00-05:00</ns1:time>

<ns1:ending>2009-08-13T01:00:00-05:00</ns1:ending>

<ns1:value1>52.39</ns1:value1>

<ns1:asType>RRS</ns1:asType>

<ns1:marketType>DAM</ns1:marketType>

……………….

……………….

<ns1:MCPCSchedule>

<ns1:time>2009-08-13T23:00:00-05:00</ns1:time>

<ns1:ending>2009-08-14T00:00:00-05:00</ns1:ending>

<ns1:value1>66.47</ns1:value1>

<ns1:asType>RRS</ns1:asType>

<ns1:marketType>DAM</ns1:marketType>

</ns1:MCPCSchedule>

</ns1:MCPCs> …………

### Binding Constraints

This section describes interfaces used to retrieve binding constraints. There are several options for retrieval of binding constraints:

* Binding constraints related to DRUC
  + Requested Operating Day
* Binding constraints related to HRUC
  + Requested End Hour
* DAM shadow prices for binding constraints

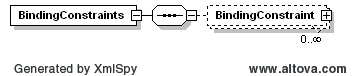
The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | BindingConstraints |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/OperatingDate | *Day of the Report*  *For DRUC & DAM MarketType* |
| Request/MarketType | *DAM/DRUC/HRUC* |
| Request/EndTime | *End time of interest*  *For HRUC MarketType* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | BindingConstraints |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | *BindingConstraints* |

The following diagram describes the structures that are used for the retrieval of binding constraints and shadow prices:



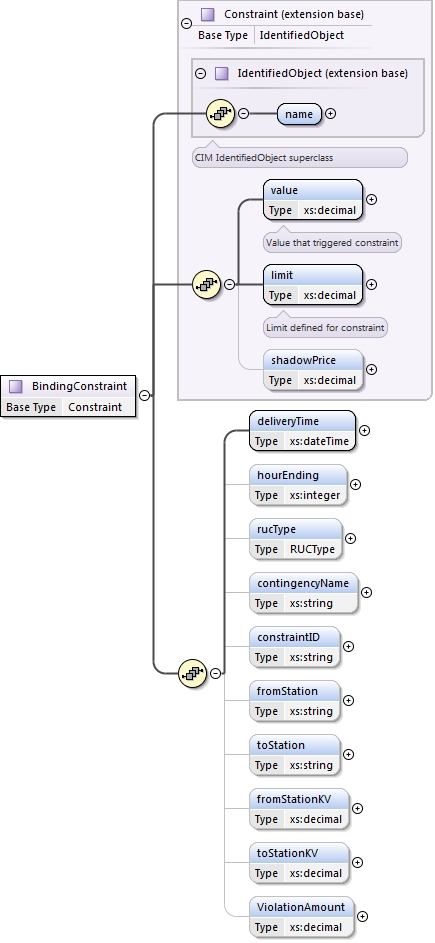


Figure 110 - BindingConstraint Container Structure

The following elements are used to report binding constraints:

* Name of the constraint
* Value that exceeded limit
* Limit defined for constraint
* Maximum shadow price
* Delivery time
* RUC type (DRUC, HRUC, and DAM)
* Contingency Name
* Contraint ID
* Station name which contingency is from
* Station name which contingency is to
* Station voltage which contingency is from
* Station voltage which contingency is to
* Violation amount

The following is an XML example:

<ns1:BindingConstraints xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:BindingConstraint>

<ns1:name>PNHNDL</ns1:name>

<ns1:value>3063.0</ns1:value>

<ns1:limit>3063.0</ns1:limit>

<ns1:shadowPrice>0.655</ns1:shadowPrice>

<ns1:deliveryTime>2017-09-21T01:00:00-05:00</ns1:deliveryTime>

<ns1:hourEnding>1</ns1:hourEnding>

<ns1:rucType>DAM</ns1:rucType>

<ns1:contingencyName>BASE CASE</ns1:contingencyName>

<ns1:constraintID>1</ns1:constraintID>

<ns1:fromStationKV>0.0</ns1:fromStationKV>

<ns1:toStationKV>0.0</ns1:toStationKV>

<ns1:ViolationAmount>0.0</ns1:ViolationAmount>

</ns1:BindingConstraint>

</ns1:BindingConstraints>

### SCED Violated Constraints

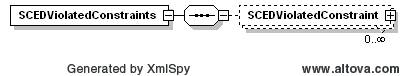
This section describes interfaces used to retrieve binding constraint in SCED. SCED Violated Constraints are only provided for RTM market. The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | SCEDViolatedConstraints |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/StartTime | *Start time of interest* |
| Request/EndTime | *End time of interest* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | SCEDViolatedConstraints |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | SCEDViolatedConstraints |

The structure of SCED violated constraints are described by the following diagram:



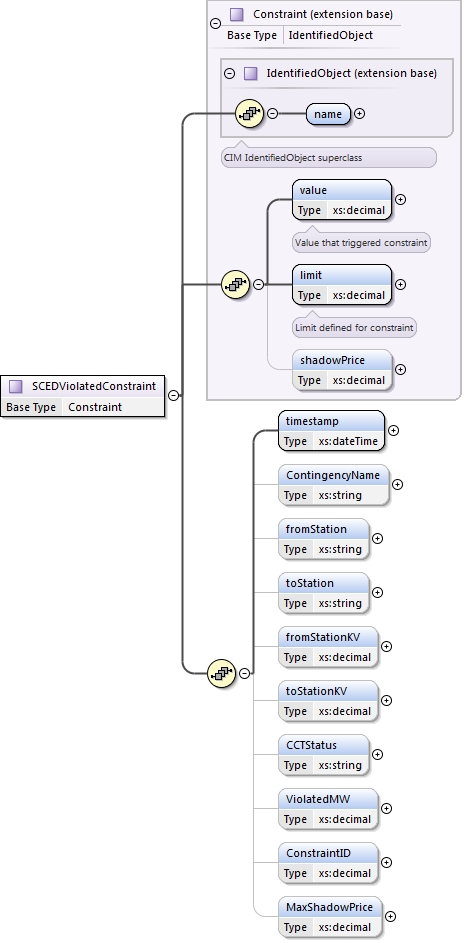


Figure 111 - SCEDViolatedConstraints Container Structure

The following elements are used to report violated constraints:

* Name of the constraint
* Value that exceeded limit
* Defined limit for constraint
* Timestamp of violation
* Contingency Name
* Station name which contingency is from
* Station name which contingency is to
* Station voltage which contingency is from
* Station voltage which contingency is to
* Constraint is competitive or not
* Violated MW
* Constraint ID
* Max shadow price

The following is an XML example:

<ns1:SCEDViolatedConstraints xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:SCEDViolatedConstraint>

<ns1:name>6485\_\_A</ns1:name>

<ns1:value>156.8</ns1:value>

<ns1:limit>156.8</ns1:limit>

<ns1:shadowPrice>431.39862</ns1:shadowPrice>

<ns1:timestamp>2017-09-20T10:55:12-05:00</ns1:timestamp>

<ns1:ContingencyName>SMNHODE8</ns1:ContingencyName>

<ns1:fromStation>MOSSW</ns1:fromStation>

<ns1:toStation>PBSES</ns1:toStation>

<ns1:fromStationKV>138.0</ns1:fromStationKV>

<ns1:toStationKV>138.0</ns1:toStationKV>

<ns1:CCTStatus>NONCOMP</ns1:CCTStatus>

<ns1:ViolatedMW>0.0</ns1:ViolatedMW>

<ns1:ConstraintID>3.0</ns1:ConstraintID>

<ns1:MaxShadowPrice>3500.0</ns1:MaxShadowPrice>

</ns1:SCEDViolatedConstraint> </ns1:SCEDViolatedConstraints>

### Price Corrected SCED Violated Constraints

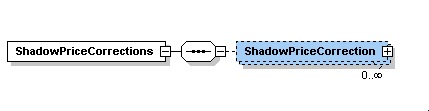
This section describes interfaces used to retrieve price corrected binding constraint in SCED. Price Corrected SCED Violated Constraints are only provided for RTM market. The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | SCEDViolatedConstraintsPC |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/StartTime | *Start time of interest* |
| Request/EndTime | *End time of interest* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | SCEDViolatedConstraintsPC |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | ShadowPriceCorrections |

The structure of ShadowPriceCorrections are described by the following diagram:



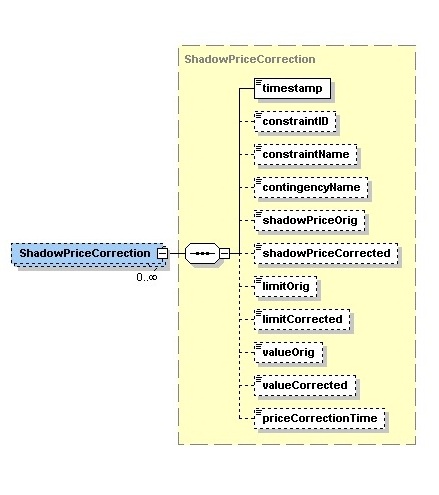


Figure 112 - ShadowPriceCorrection Container Structure

The following elements are used to report price corrections for real-time shadow prices:

* SCED Timestamp
* Constraint ID
* Constraint Name
* Contingency Name
* Original Shadow Price
* Corrected Shadow Price
* Original Limit
* Corrected Limit
* Original Value
* Corrected Value
* Price Correction Timestamp

The following is an XML example:

<ns1:ShadowPriceCorrections xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:ShadowPriceCorrection>

<ns1:timestamp>2012-01-05T11:00:00-06:00</ns1:timestamp>

<ns1:constraintID>1</ns1:constraintID>

<ns1:constraintName>ABC</ns1:constraintName>

<ns1:contingencyName>DEFGHIJ1</ns1:contingencyName>

<ns1:shadowPriceOrig>995.45</ns1:shadowPriceOrig>

<ns1:shadowPriceCorrected>786.59</ns1:shadowPriceCorrected>

<ns1:limitOrig>32.3</ns1:limitOrig>

<ns1:limitCorrected>32.3</ns1:limitCorrected>

<ns1:valueOrig>32.3</ns1:valueOrig>

<ns1:valueCorrected>32.3</ns1:valueCorrected>

<ns1:priceCorrectionTime>2012-01-10T14:06:42-06:00</ns1:priceCorrectionTime>

</ns1:ShadowPriceCorrection>

</ns1:ShadowPriceCorrections>

### Ancillary Service Obligation

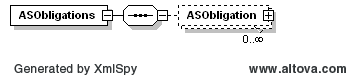
The purpose of this interface is to provide ancillary service obligation by service type for the requesting QSE. The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | ASObligations |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/MarketType | *DAM/SASM* |
| Request/TradingDate | *Trading date* |
| Request/ASType | *Optional: AS type* |
| Request/Option | *Optional: SASM ID (required for SASM)*  *Format: YYYYMMDDHHMMSS* |

If the AS type is not specified, all AS obligations will be returned. The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | ASObligations |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | ASObligations |

The structure of the ASObligations payload is described by the following diagram, using a CIM RegularIntervalSchedule. The values of ‘value1’ would represent the MW obligation and the values of ‘value2’ would represent a positive or negative difference from COP:



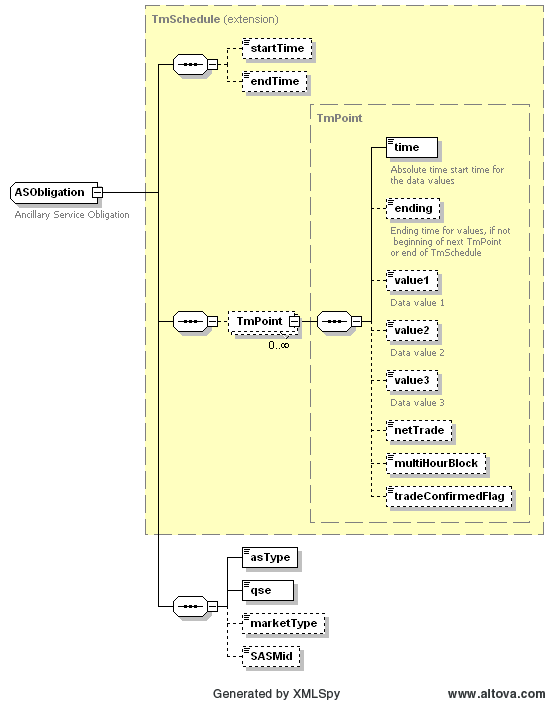


Figure 113 - ASObligations Container Structure

The following is an XML example:

<ns1:ASObligations xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotInformation.xsd">

<ns1:ASObligation>

<ns1:startTime>2009-06-10T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-10T01:00:00-05:00</ns1:endTime>

<ns1:TmPoint>

<ns1:time>2009-06-10T00:00:00-05:00</ns1:time>

<ns1:ending>2009-06-10T01:00:00-05:00</ns1:ending><ns1:value1>133.8</ns1:value1>

</ns1:TmPoint>

<ns1:asType>Reg-Down</ns1:asType>

<ns1:qse>QSE</ns1:qse>

</ns1:ASObligation>

<ns1:ASObligation>

<ns1:startTime>2009-06-10T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-10T01:00:00-05:00</ns1:endTime>

<ns1:TmPoint>

<ns1:time>2009-06-10T00:00:00-05:00</ns1:time>

<ns1:ending>2009-06-10T01:00:00-05:00</ns1:ending>

<ns1:value1>164.7</ns1:value1>

</ns1:TmPoint>

<ns1:asType>Reg-Up</ns1:asType>

<ns1:qse>QSE</ns1:qse>

</ns1:ASObligation>

<ns1:ASObligation>

<ns1:startTime>2009-06-10T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-10T01:00:00-05:00</ns1:endTime>

<ns1:TmPoint>

<ns1:time>2009-06-10T00:00:00-05:00</ns1:time>

<ns1:ending>2009-06-10T01:00:00-05:00</ns1:ending>

<ns1:value1>473.6</ns1:value1>

</ns1:TmPoint>

<ns1:asType>RRS</ns1:asType>

<ns1:qse>QSE</ns1:qse>

</ns1:ASObligation>

</ns1:ASObligations>

### Dynamic Ratings

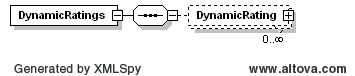
This section describes interfaces used to retrieve thermal ratings for conducting equipment. This interface provides current(realtime) Dynamic Ratings information for all equipments. The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | DynamicRatings |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/StartTime | *Start time of interest* |
| Request/EndTime | *End time of interest* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | DynamicRatings |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | DynamicRatings |

The payload structure is described by the following diagram:



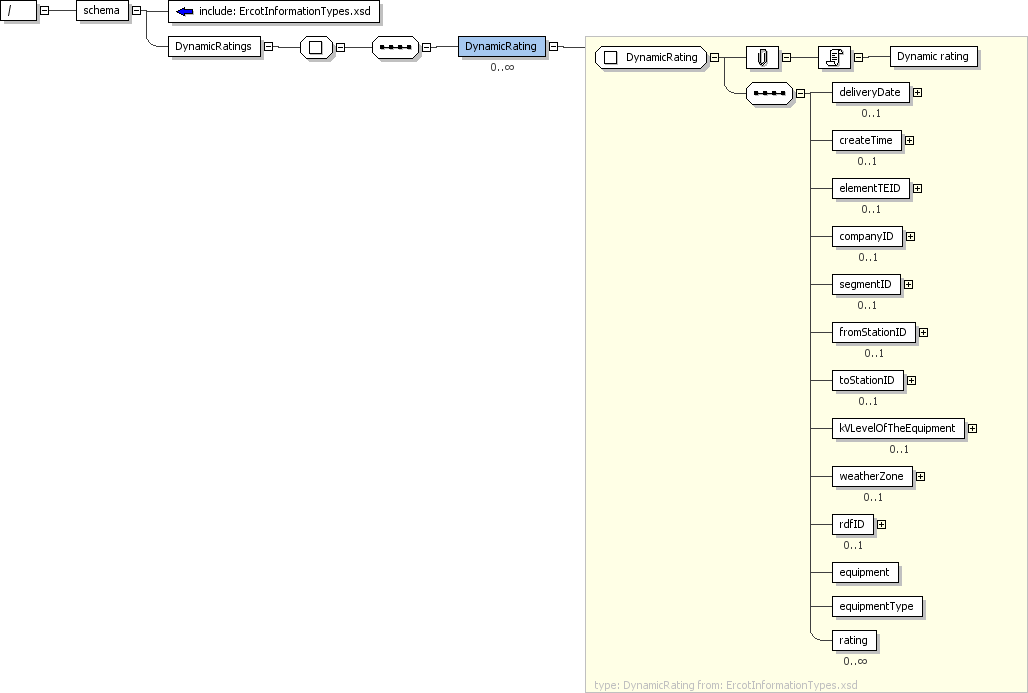


Figure 114 - DynamicRatings Container Structure

The ‘equipment’ tag identifies the name of the transmission element. The types of ratings that are currently defined include:

* Normal (in MVA)
  + The rating at which a Transmission Element can operate without reducing its normal life expectancy.
* Emergency (in MVA)
  + Two-hour MVA rating of a Transmission Element.
* 15-min (in MVA)
  + The 15-Minute MVA rating of a Transmission Element.

Note: createTime is dynamic ratings report created time.

The following is an XML example of payload:

<?xml version="1.0" encoding="UTF-8"?>

<DynamicRatings xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotInformation.xsd" xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<DynamicRating>

<deliveryDate>2006-05-04</deliveryDate>

<createTime>2006-05-04T18:13:51-06:00</createTime>

<elementTEID>7105</elementTEID>

<companyID>TESTQSE</companyID>

<segmentID>O</segmentID>

<fromStationID>XYZ</fromStationID>

<toStationID>TUV</toStationID>

<kVLevelOfTheEquipment>69</kVLevelOfTheEquipment>

<weatherZone>NORTH</weatherZone>

<rdfID>\_{00A00A0A-0AA0-0AA0-A0A0-00A0AA00000A}</rdfID>

<equipment>1990\_TST</equipment>

<equipmentType>LN</equipmentType>

<rating>

<ratingType>Normal</ratingType>

<ratingValue>48</ratingValue>

</rating>

<rating>

<ratingType>Emergency</ratingType>

<ratingValue>48</ratingValue>

</rating>

<rating>

<ratingType>15-min</ratingType>

<ratingValue>55</ratingValue>

</rating>

</DynamicRating>

</DynamicRatings>

### Load Ratio Share

This section describes interfaces used to retrieve the load ratio share for each QSE. This is a daily report and will return the full day data for the requested day.

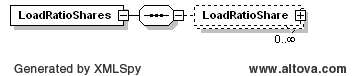
The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | LoadRatioShares |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/OperatingDate | *Day of the report* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | LoadRatioShares |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | *LoadRatioShares* |

The LoadRatioShares payload structure is described by the following diagram:



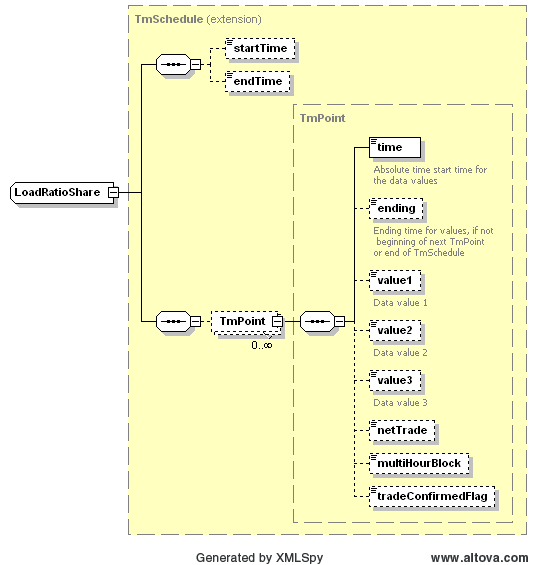


Figure 115 - LoadRatioShares Container Structure

The values of ‘value1’ would be used to identify the proportion assigned to the requesting QSE for each hour.

The following is an XML example:

<?xml version="1.0" encoding="UTF-8"?>

<LoadRatioShares xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotInformation.xsd" xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<LoadRatioShare>

<startTime>2007-07-25T00:00:00-06:00</startTime>

<endTime>2007-07-25T00:00:00-06:00</endTime>

<TmPoint>

<time>2007-07-25T00:00:00-06:00</time>

<ending>2007-07-25T00:00:00-06:00</ending>

<value1>4.14</value1>

</TmPoint>

</LoadRatioShare>

</LoadRatioShares>

### Aggregated Ancillary Service Offer Curves

This section describes interfaces used to retrieve aggregated ancillary service offer curves.

This interface will return the full day data for DAM or requested SASM run for the requested operating day.

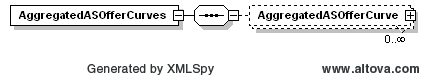
The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | AggregatedASOfferCurves |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/OperatingDate | *Day of interest* |
| Request/MarketType | *DAM or SASM* |
| Request/Option | *Optional: SASM ID (required for SASM)*  *Format: YYYYMMDDHHMMSS* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | AggregatedASOfferCurves |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | AggregatedASOfferCurves |

The payload structure is described by the following diagrams, where there is one curve for each AS type per hour:



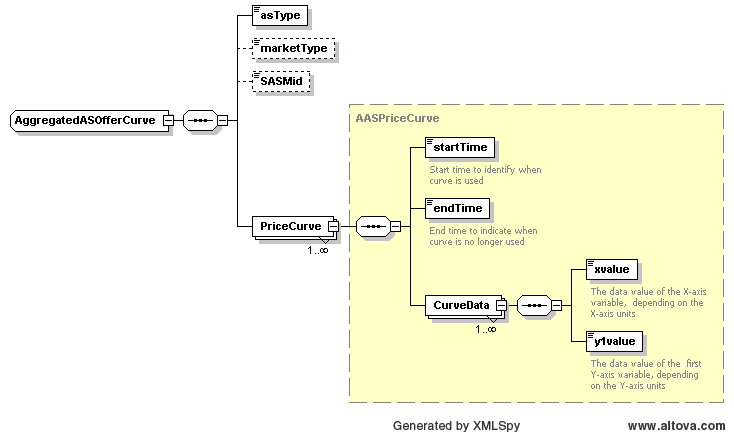


Figure 116 - AggregatedASOfferCurve Schedule

For each point on the AggregatedASOfferCurve, the xvalue identifies price value and y1value indicates megawatt level.

The following is an XML example:

<?xml version="1.0" encoding="UTF-8"?>

<ns1:AggregatedASOfferCurves xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotInformation.xsd">

<ns1:AggregatedASOfferCurve>

<ns1:asType>Off-Non-Spin</ns1:asType>

<ns1:marketType>DAM</ns1:marketType>

<ns1:PriceCurve>

<ns1:startTime>2009-06-11T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-11T01:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>1675.87</ns1:xvalue>

<ns1:y1value>7291890.8</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>1645.12</ns1:xvalue>

<ns1:y1value>729190.8</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>1685.17</ns1:xvalue>

<ns1:y1value>7456890.8</ns1:y1value>

</ns1:CurveData> </ns1:PriceCurve>

<ns1:PriceCurve>

<ns1:startTime>2009-06-11T01:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-11T02:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>1456.6</ns1:xvalue>

<ns1:y1value>75433456.2</ns1:y1value>

</ns1:CurveData>

</ns1:PriceCurve>

<ns1:PriceCurve>

<ns1:startTime>2009-06-11T02:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-11T03:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>1432.5</ns1:xvalue>

<ns1:y1value>76433456.5</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>1432.5</ns1:xvalue>

<ns1:y1value>76433456.5</ns1:y1value>

</ns1:CurveData>

</ns1:PriceCurve></ns1:AggregatedASOfferCurve>

.............................................

............................................

<ns1:AggregatedASOfferCurve>

<ns1:asType>Reg-Up</ns1:asType>

<ns1:marketType>DAM</ns1:marketType>

<ns1:PriceCurve>

<ns1:startTime>2009-06-11T23:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-12T00:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>2332.5</ns1:xvalue>

<ns1:y1value>18433457.3</ns1:y1value>

</ns1:CurveData>

</ns1:PriceCurve>

</ns1:AggregatedASOfferCurve>

</ns1:AggregatedASOfferCurves>

|  |
| --- |
| ***[NPRR863: Replace the example above with the following upon system implementation]***  <ns1:AggregatedASOfferCurves xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotInformation.xsd">  <ns1:AggregatedASOfferCurve>  <ns1:asType>Off-Non-Spin</ns1:asType>  <ns1:marketType>DAM</ns1:marketType>  <ns1:PriceCurve>  <ns1:startTime>2009-06-11T00:00:00-05:00</ns1:startTime>  <ns1:endTime>2009-06-11T01:00:00-05:00</ns1:endTime>  <ns1:CurveData>  <ns1:xvalue>1675.87</ns1:xvalue>  <ns1:y1value>7291890.8</ns1:y1value>  </ns1:CurveData>  <ns1:CurveData>  <ns1:xvalue>1645.12</ns1:xvalue>  <ns1:y1value>729190.8</ns1:y1value>  </ns1:CurveData>  <ns1:CurveData>  <ns1:xvalue>1685.17</ns1:xvalue>  <ns1:y1value>7456890.8</ns1:y1value>  </ns1:CurveData> </ns1:PriceCurve>  <ns1:PriceCurve>  <ns1:startTime>2009-06-11T01:00:00-05:00</ns1:startTime>  <ns1:endTime>2009-06-11T02:00:00-05:00</ns1:endTime>  <ns1:CurveData>  <ns1:xvalue>1456.6</ns1:xvalue>  <ns1:y1value>75433456.2</ns1:y1value>  </ns1:CurveData>  </ns1:PriceCurve>  <ns1:PriceCurve>  <ns1:startTime>2009-06-11T02:00:00-05:00</ns1:startTime>  <ns1:endTime>2009-06-11T03:00:00-05:00</ns1:endTime>  <ns1:CurveData>  <ns1:xvalue>1432.5</ns1:xvalue>  <ns1:y1value>76433456.5</ns1:y1value>  </ns1:CurveData>  <ns1:CurveData>  <ns1:xvalue>1432.5</ns1:xvalue>  <ns1:y1value>76433456.5</ns1:y1value>  </ns1:CurveData>  </ns1:PriceCurve></ns1:AggregatedASOfferCurve>  .............................................  ............................................  <ns1:AggregatedASOfferCurve>  <ns1:asType>Reg-Up</ns1:asType>  <ns1:marketType>DAM</ns1:marketType>  <ns1:PriceCurve>  <ns1:startTime>2009-06-11T23:00:00-05:00</ns1:startTime>  <ns1:endTime>2009-06-12T00:00:00-05:00</ns1:endTime>  <ns1:CurveData>  <ns1:xvalue>2332.5</ns1:xvalue>  <ns1:y1value>18433457.3</ns1:y1value>  </ns1:CurveData>  </ns1:PriceCurve>  </ns1:AggregatedASOfferCurve>  <ns1:AggregatedASOfferCurve>  <ns1:asType>RRSPF</ns1:asType>  <ns1:marketType>DAM</ns1:marketType>  <ns1:PriceCurve>  <ns1:startTime>2009-06-11T23:00:00-05:00</ns1:startTime>  <ns1:endTime>2009-06-12T00:00:00-05:00</ns1:endTime>  <ns1:CurveData>  <ns1:xvalue>1532.5</ns1:xvalue>  <ns1:y1value>16134457.2</ns1:y1value>  </ns1:CurveData>  </ns1:PriceCurve>  </ns1:AggregatedASOfferCurve>  </ns1:AggregatedASOfferCurves> |

### Ancillary Service Plan

This section describes interfaces used to retrieve the Ancillary Service Plan for the DAM market. The interface returns hourly values for the next 7 days (168 hours)

The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | ASServicePlan |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/EndTime | *End time of interest* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | ASServicePlan |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | *ASServicePlan* |

The payload structure is described by the following diagram:

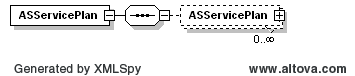


Figure 117 - ASServicePlan Container Structure

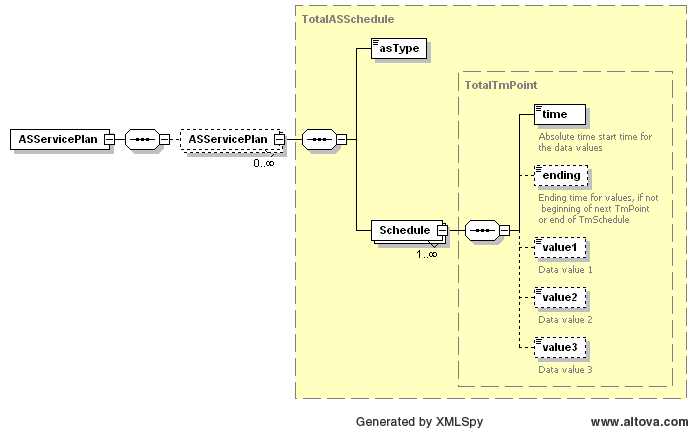


Figure 118 - ASSchedule Structure

The values of ‘value1’ identify the total scheduled MW capacity for the given AS type.

The following is an XML example:

<ns1:ASServicePlan xmlns:ns0 = "http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1 = "http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:ASServicePlan>

<ns1:asType>Reg-Up</ns1:asType>

<ns1:Schedule>

<ns1:time>2009-07-28T00:00:00-05:00</ns1:time>

<ns1:ending>2009-07-28T01:00:00-05:00</ns1:ending>

<ns1:value1>800</ns1:value1>

</ns1:Schedule>

<ns1:Schedule>

<ns1:time>2009-07-28T01:00:00-05:00</ns1:time>

<ns1:ending>2009-07-28T02:00:00-05:00</ns1:ending>

<ns1:value1>920</ns1:value1>

</ns1:Schedule>

………………….

<ns1:Schedule>

<ns1:time>2009-07-29T23:00:00-05:00</ns1:time>

<ns1:ending>2009-07-30T00:00:00-05:00</ns1:ending>

<ns1:value1>0</ns1:value1>

</ns1:Schedule>

</ns1:ASServicePlan>

</ns1:ASServicePlan>

### Startup and Shutdown Instructions

This section describes interfaces used to retrieve unit startup and shutdown instructions. The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | StartupShutdownInstructions |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/MarketType | *Market Type (HRUC or DRUC)* |
| Request/TradingDate | *Trading date* |
| Request/Option | *Optional: Hour Ending (Required for HRUC) \*\** |

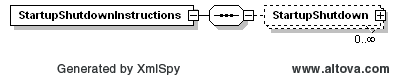
\*\*Note: The hour ending specified in the StartupShutdownInstruction request is the hour the report was \*generated\*, not the hour that the report is awarding. Therefore, the report returned can contain multiple hours as one HRUC run is capable of awarding more than one hour of the day.

To request for repeating hour on long day(DST change day), Hour Ending value should be 2\*.

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | StartupShutdownInstructions |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | StartupShutdownInstructions |

The payload structure is described by the following diagram:



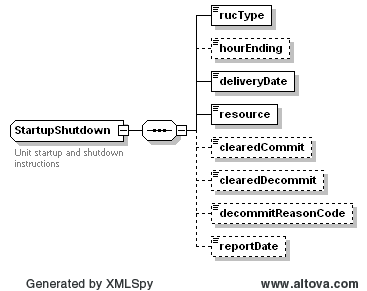


Figure 119 - StartupShutdownInstructions Container Structure

For each startup or shutdown instruction, the following elements are provided:

* RUC type (DRUC, HRUC)
* RUC Ending hour for the instructions. On long day(DST change day), hourEnding is represented as 2\* for repeating hour.
* Delivery date
* Resource name
* Cleared commit
* Cleared decommit
* Reason code for decommit
* Report Generation Time

The following is an XML example:

<?xml version="1.0" encoding="UTF-8"?>

<StartupShutdownInstructions xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<StartupShutdown>

<rucType>DRUC</rucType>

<hourEnding>6</hourEnding>

<deliveryDate>2007-07-25</deliveryDate>

   <resource>Resource1</resource>

<clearedCommit>true</clearedCommit>

<clearedDecommit>false</clearedDecommit>

<decommitReasonCode>50 characters of free form text</decommitReasonCode>

<reportDate>2007-07-25T11:00:00-06:00</reportDate>

</StartupShutdown>

</StartupShutdownInstructions>

### Total Ancillary Service Offers

This section describes interfaces used to retrieve total ancillary services offers for DAM Market. This interface will return the full day data for DAM market for the requesting day.

The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | TotalASOffers |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/OperatingDate | *Day of interest* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | TotalASOffers |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | TotalASOffers |

The payload structure is described by the following diagrams:

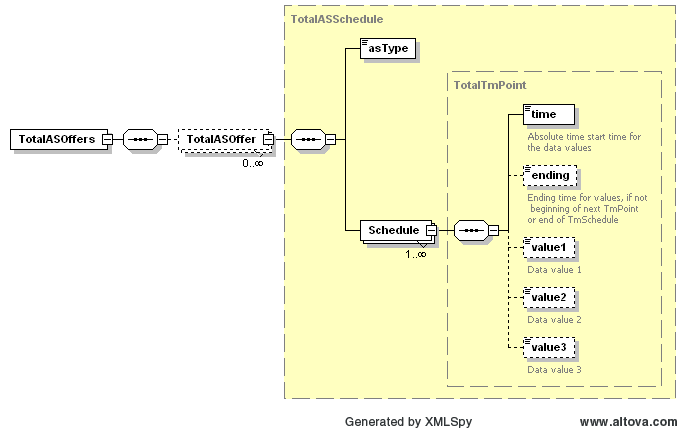


Figure 120 - ASOffers Structure

The values of ‘value1’ identify the total MW offers for the given AS type. There is no consideration given to linked offers.

The following is an XML example:

<ns1:TotalASOffers xmlns:ns0 = "http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1 = "http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:TotalASOffer>

<ns1:asType>Reg-Up</ns1:asType>

<ns1:Schedule>

<ns1:time>2009-07-29T00:00:00-05:00</ns1:time>

<ns1:ending>2009-07-29T01:00:00-05:00</ns1:ending>

<ns1:value1>330170</ns1:value1>

</ns1:Schedule>

<ns1:Schedule>

<ns1:time>2009-07-29T01:00:00-05:00</ns1:time>

<ns1:ending>2009-07-29T02:00:00-05:00</ns1:ending>

<ns1:value1>330170</ns1:value1>

</ns1:Schedule>

……….

</ns1:TotalASOffer>

</ns1:TotalASOffers>

|  |
| --- |
| ***[NPRR863: Replace the example above with the following upon system implementation]***  <ns1:TotalASOffers xmlns:ns0 = "http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1 = "http://www.ercot.com/schema/2007-06/nodal/ews">  <ns1:TotalASOffer>  <ns1:asType>RRSPF</ns1:asType>  <ns1:Schedule>  <ns1:time>2021-11-10T00:00:00-05:00</ns1:time>  <ns1:ending>2021-11-10T01:00:00-05:00</ns1:ending>  <ns1:value1>330170</ns1:value1>  </ns1:Schedule>  <ns1:Schedule>  <ns1:time>2021-11-10T01:00:00-05:00</ns1:time>  <ns1:ending>2021-11-10T02:00:00-05:00</ns1:ending>  <ns1:value1>330170</ns1:value1>  </ns1:Schedule>  ……….  </ns1:TotalASOffer>  </ns1:TotalASOffers> |

### Total DAM Energy \*\*\* Same as 4.3.10 - Market Totals \*\*\*

This section describes interfaces used to retrieve the total amount of energy bought or sold in the Day Ahead Market. This interface will return the full day data for DAM market for the requesting day.

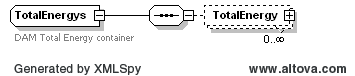
The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | TotalEnergys |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/option | *One of:*   * *EnergyBoughtInDAM* * *EnergySoldInDAM* |
| Request/OperatingDate | *Day of interest* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | TotalEnergys |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | TotalEnergy |

The payload structure is described by the following diagram:



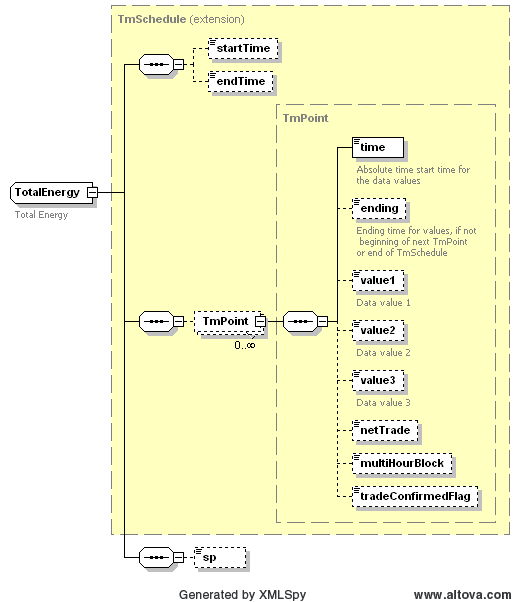


Figure 121 - TotalEnergy Structure

The Totals structure is based upon a CIM RegularIntervalSchedules, where the values of ‘value1’ are total megawatt values.

The following is an XML example:

<ns0:TotalEnergys xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns0:TotalEnergy>

<ns0:TmPoint>

<ns0:time>2009-06-13T00:00:00-05:00</ns0:time>

<ns0:ending>2009-06-13T01:00:00-05:00</ns0:ending>

<ns0:value1>0</ns0:value1>

</ns0:TmPoint>

<ns0:sp>DIB\_PUN1</ns0:sp>

</ns0:TotalEnergy>

<ns0:TotalEnergy>

<ns0:TmPoint>

<ns0:time>2009-06-13T00:00:00-05:00</ns0:time>

<ns0:ending>2009-06-13T01:00:00-05:00</ns0:ending>

<ns0:value1>0</ns0:value1>

</ns0:TmPoint>

<ns0:sp>DUKE\_CC1</ns0:sp>

</ns0:TotalEnergy>

<ns0:TotalEnergy>

<ns0:TmPoint>

<ns0:time>2009-06-13T00:00:00-05:00</ns0:time>

<ns0:ending>2009-06-13T01:00:00-05:00</ns0:ending>

<ns0:value1>0</ns0:value1>

</ns0:TmPoint>

<ns0:sp>EXN\_PUN1</ns0:sp>

</ns0:TotalEnergy>

<ns0:TotalEnergy>

<ns0:TmPoint>

<ns0:time>2009-06-13T00:00:00-05:00</ns0:time>

<ns0:ending>2009-06-13T01:00:00-05:00</ns0:ending>

<ns0:value1>37</ns0:value1>

</ns0:TmPoint>

<ns0:sp>DIB\_DIB\_G1</ns0:sp>

</ns0:TotalEnergy>

………………………..

<ns0:TotalEnergy>

<ns0:TmPoint>

<ns0:time>2009-06-13T23:00:00-05:00</ns0:time>

<ns0:ending>2009-06-14T00:00:00-05:00</ns0:ending>

<ns0:value1>0</ns0:value1>

</ns0:TmPoint>

<ns0:sp>FTR\_CC1</ns0:sp>

</ns0:TotalEnergy>

<ns0:TotalEnergy>

<ns0:TmPoint>

<ns0:time>2009-06-13T23:00:00-05:00</ns0:time>

<ns0:ending>2009-06-14T00:00:00-05:00</ns0:ending>

<ns0:value1>0</ns0:value1>

</ns0:TmPoint>

<ns0:sp>\_SL\_PUN1</ns0:sp>

</ns0:TotalEnergy>

</ns0:TotalEnergys>

### Weather Data

The purpose of this interface is to provide a query for fetching the Actual, historical, and forecast weather related information from EMS system. EMS gets its weather feeds directly from national weather service. This interface will return the full day data for the requesting day.

The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | WeatherInfos |
| Header/Source | MARKET PARTICIPANT ID |
| Header/UserID | ID of user |
| Request/OperatingDate | Day of Interest |
| Request/Option | Optional: Zone of interest  If not provided Info for all Zones is returned |

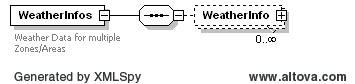
The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | WeatherInfos |
| Header/Source | ERCOT |
| Reply/ReplyCode | Reply code, success=OK, error=ERROR |
| Reply/Error | Error message, if error encountered |
| Payload | WeatherInfos |

A list of available zones are provided below. If request for no specific zones is submitted, weather information for all zones are returned.

1. Coast
2. East
3. FarWest
4. North
5. NorthCentral
6. SouthCentral
7. Southern
8. West

The structure of a reply for weather payload is shown by the following diagrams:



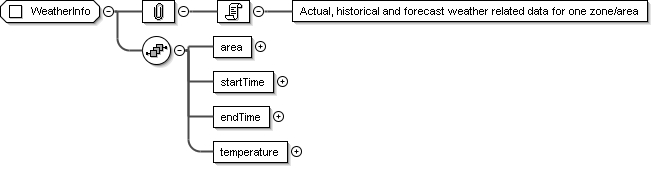


Figure 122 – Weather Information Container Structure

The following is an XML example for a response for forecast weather related information from EMS system

<?xml version="1.0" encoding="UTF-8"?>

<WeatherInfos xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotInformation2.xsd">

<WeatherInfo>

<area>North Austin</area>

<startTime>2008-01-01T00:00:00-06:00</startTime>

<endTime>2008-01-02T00:00:00-06:00</endTime>

<temperature>72</temperature>

</WeatherInfo>

<WeatherInfo>

<area>West Austin</area>

<startTime>2008-01-02T00:00:00-06:00</startTime>

<endTime>2008-01-03T00:00:00-06:00</endTime>

<temperature>75</temperature>

</WeatherInfo>

</WeatherInfos>

### Responsive Reserve Capacity – Please See Section 4.3.39, System Parameters

### Non-Spinning Reserve – Please See Section 4.3.39, System Parameters

### Undeployed Reg-Up and Reg-Down – Please See Section 4.3.39, System Parameters

### Available Capacity – Please See Section 4.3.39, System Parameters

### Wind-Powered Generation Resource Production Potential

The purpose of this interface is to provide a query for fetching Wind-Powered Generation Resource Production Potential (WGRPP) for each Wind Generation Resource (WGR) as well as WGRPP forecast for all WGRs. This interface will return the last hour data for the requesting hour-end.

The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | WGRPP |
| Header/Source | MARKET PARTICIPANT ID |
| Header/UserID | ID of user |
| Request/endTime | End time of interest |
| Request/ID | Optional: WGR identity  If No WGR is provided, WGRPP for all WGRs is returned. |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | WGRPP |
| Header/Source | ERCOT |
| Reply/ReplyCode | Reply code, success=OK |
| Reply/Error | Error message, if error encountered |
| Payload/ | WGRPP |

The following diagram defines the ForecastPayload structure in the WGRPP response

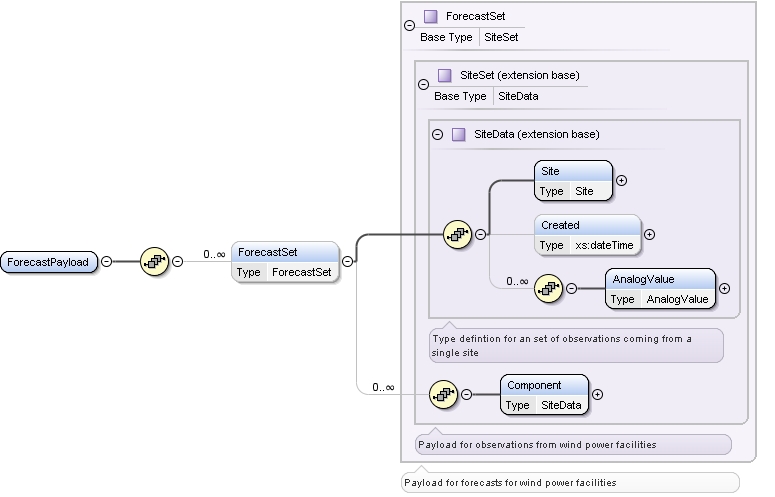


Figure 123 – WGRPP Structure

The following is an abbreviated XML example of a WGRPP response:

<ns1:ForecastPayload xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il"  
 xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">  
 <ns1:ForecastSet>  
 <ns1:Site duns="1234567890000" name="WIND\_1" qseid="QSE1">SITE1</ns1:Site>  
 <ns1:Created>2016-01-14T09:10:59-06:00</ns1:Created>  
 <ns1:AnalogValue statistic="MEAN" timeStamp="2016-01-14T10:00:00-06:00" type="WGRPP"  
 units="MW">5.63E1</ns1:AnalogValue>  
 <ns1:AnalogValue statistic="MEAN" timeStamp="2016-01-21T09:00:00-06:00" type="WGRPP"  
 units="MW">2.46E1</ns1:AnalogValue>  
 </ns1:ForecastSet>  
 <ns1:ForecastSet>  
 <ns1:Site duns="1234567890000" name="WIND\_2" qseid="QSE1">SITE2</ns1:Site>  
 <ns1:Created>2016-01-14T09:10:59-06:00</ns1:Created>  
 <ns1:AnalogValue statistic="MEAN" timeStamp="2016-01-14T10:00:00-06:00" type="WGRPP"  
 units="MW">7.65E1</ns1:AnalogValue>  
 <ns1:AnalogValue statistic="MEAN" timeStamp="2016-01-21T09:00:00-06:00" type="WGRPP"  
 units="MW">4.19E1</ns1:AnalogValue>  
 </ns1:ForecastSet>  
</ns1:ForecastPayload>

### Short-Term Wind Power Forecast

The purpose of this interface is to provide a query for fetching short-term wind power forecast (STWPF) for each WGR and STWPF forecast for all WGRs. This interface will return the last hour data for the requesting hour-end.

The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | *get* |
| Header/Noun | STWPF |
| Header/Source | *MARKET PARTICIPANT ID* |
| Header/UserID | *ID of user* |
| Request/endTime | *End time of interest* |
| Request/ID | *Optional: WGR identity*  *If No WGR provided then* STWPF *for all WGRs is returned* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | *reply* |
| Header/Noun | STWPF |
| Header/Source | *ERCOT* |
| Reply/ReplyCode | *Reply code, success=OK* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | *STWPF* |

The following diagram defines the ForecastPayload structure in the STWPF response:

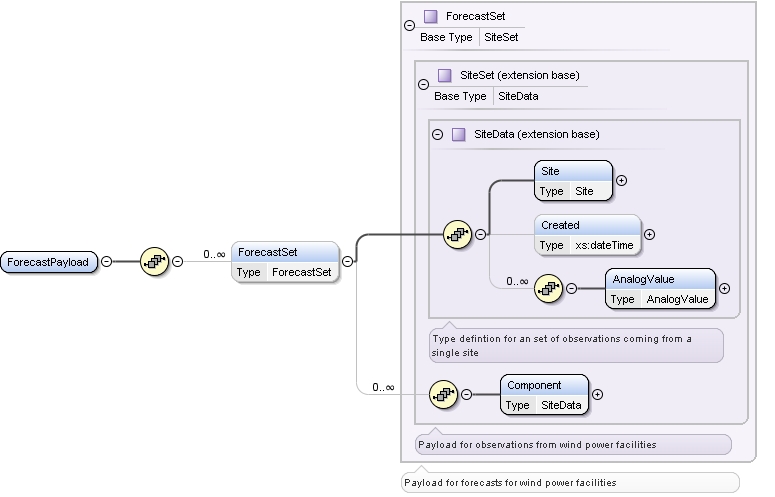


Figure 124 - STWPF Structure

The following is an abbreviated XML example of a STWPF response:

<ns1:ForecastPayload xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il"  
 xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">  
 <ns1:ForecastSet>  
 <ns1:Site duns="1234567890000" name="WIND\_1" qseid="QSE1">SITE1</ns1:Site>  
 <ns1:Created>2016-01-14T09:10:59-06:00</ns1:Created>  
 <ns1:AnalogValue statistic="MEAN" timeStamp="2016-01-14T10:00:00-06:00" type="STWPF"  
 units="MW">1.6E0</ns1:AnalogValue>  
 </ns1:ForecastSet>  
 <ns1:ForecastSet>  
 <ns1:Site duns="1234567890000" name="WIND\_2" qseid="QSE1">SITE2</ns1:Site>  
 <ns1:Created>2016-01-14T09:10:59-06:00</ns1:Created>  
 <ns1:AnalogValue statistic="MEAN" timeStamp="2016-01-14T10:00:00-06:00" type="STWPF"  
 units="MW">2.7E0</ns1:AnalogValue>  
 </ns1:ForecastSet>  
</ns1:ForecastPayload>

### Total ERCOT Load \*\*\* Same as 4.3.9 - System Load \*\*\*

This section describes interfaces used to retrieve the total amount of ERCOT load. The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | TotalLoad |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/startTime | *Start time* |
| Request/endTime | *End time* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | TotalLoad |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR* |
| Reply/Error | *Error message, if error encountered* |
| Payload | TotalLoad |

The payload structure is described by the following diagram:

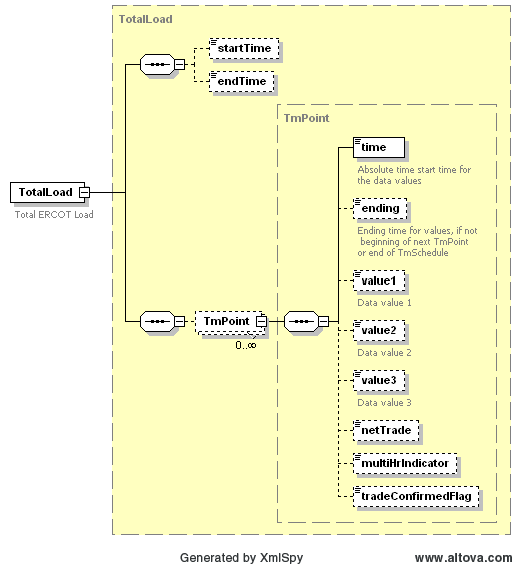


Figure 125 – Total Load Container Structure

The following is an XML example for total amount of ERCOT load:

<?xml version="1.0" encoding="UTF-8"?>

<TotalLoad xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotInformation2.xsd">

<startTime>2008-01-01T00:00:00-06:00</startTime>

<endTime>2008-01-02T00:00:00-06:00</endTime>

<TmPoint>

<time>2008-01-01T00:00:00-06:00</time>

<value1>15000</value1>

</TmPoint>

<TmPoint>

<time>2008-01-01T03:00:00-06:00</time>

<value1>16000</value1>

</TmPoint>

</TotalLoad>

### Total ERCOT Generation Operating Reserve – Please See Section 4.3.40, System Parameters

### UnconfirmedTrades

This section describes interfaces used to retrieve the list if trades that the submitting QSE have been identified as the counter party in a trade, but have not yet entered a matching energy, capacity, or ancillary services trade. If a matching trade is not entered by the close of the market, the trade is rejected.

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | UnconfirmedTrades |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/TradingDate | *Trading date* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | UnconfirmedTrades |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload | UnconfirmedTrades |

The payload structure is described by the following diagram:



Figure 126 – UnconfirmedTrades Container Structure

The details of the structures for ASTrades, CapacityTrades, and EnergyTrades are described in sections 3.3.5, 3.3.6, and 3.3.12, respectively. The following is an XML of an unconfirmed trade’s payload:

<UnconfirmedTrades>

<ASTrade>

<mRID>ACME.20080101.AST.<ASType>.<BuyerQSE>.<SellerQSE></mRID>

</ASTrade>

<CapacityTrade>

<mRID>XYZ.20080101.CT.<BuyerQSE>.<SellerQSE></mRID>

</CapacityTrade>

<EnergyTrade>

<mRID>ACME.20080101.ET.<SettlementPoint>.<BuyerQSE>.<SellerQSE></mRID>

</EnergyTrade>

</UnconfirmedTrades>

The mRID used in the message is the mRID for the trade that was submitted by the counterparty.

### ConfirmedTrades

This section describes interfaces used to retrieve the list if trades that the submitting QSE have been identified as the counter party in a trade and have received a matching energy, capacity, or ancillary services trade.

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | ConfirmedTrades |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/TradingDate | *Trading date* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | ConfirmedTrades |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload | ConfirmedTrades |

The payload structure is described by the following diagram:



Figure 127 – ConfirmedTrades Container Structure

The following is an XML of a confirmed trade’s payload:

<ConfirmedTrades>

<ASTrade>

<mRID>ACME.20080101.AST.<ASType>.<BuyerQSE>.<SellerQSE></mRID>

</ASTrade>

<CapacityTrade>

<mRID>XYZ.20080101. CT.<BuyerQSE>.<SellerQSE></mRID>

</CapacityTrade>

<EnergyTrade>

<mRID>ACME.20080101. ET.<SettlementPoint>.<BuyerQSE>.<SellerQSE></mRID>

</EnergyTrade>

</ConfirmedTrades>

The mRID used in the message is the mRID for the trade that was submitted by the counterparty.

### DAM Ancillary Service Offer Insufficiency Report

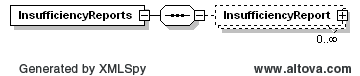
This section describes interfaces used to retrieve the DAM AS Offer Insufficiency Report. DAM AS Offer Insufficiency Report reflects available and insufficient AS Offer quantities. The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | InsufficiencyReport |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/TradingDate | *Trading date* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | InsufficiencyReport |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload | InsufficiencyReports |

The payload structure is described by the following diagram:



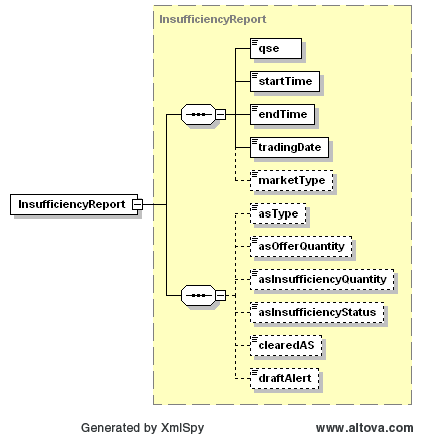


Figure 128 – DAM AS Offer Insufficiency Report Container Structure

The following is an XML example for DAM AS Offer Insufficiency Report:

<InsufficiencyReports xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotAwards.xsd" xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<InsufficiencyReport>

<qse>ERCOT</qse>

<startTime>2008-01-01T10:00:00-06:00</startTime>

<endTime>2008-01-01T12:00:00-06:00</endTime>

<tradingDate>2008-01-01</tradingDate>

<marketType>DAM</marketType>

<asType>Reg-Up</asType>

<asOfferQuantity>3.1</asOfferQuantity>

<asInsufficiencyQuantity>3.1</asInsufficiencyQuantity>

<asInsufficiencyStatus>3.1</asInsufficiencyStatus>

<clearedAS>String</clearedAS>

<draftAlert>String</draftAlert>

<InsufficiencyReport>

</InsufficiencyReports>

### End of Adjustment Period Results

This section describes the interface used to retrieve the End of Adjustment Period results. The result set will hold the Output Schedule Validation results (errors and warnings) for the specified trading date and the end hour.

The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | EndAdjPeriod |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/TradingDate | *Trading date* |
| Request/Option | *End of Adjustment Period Hour Ending* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | EndAdjPeriod |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload | BidSet/OutputSchedule |

The payload structure is described by the following diagram:



Figure 129 – End of Adjustment Period results: BidSet/OutputSchedule

The following is an XML example for End of Adjustment Period Results:

<BidSet xmlns="http://www.ercot.com/schema/2007-05/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<tradingDate>2008-01-01</tradingDate>

<OutputSchedule>

<mRID>ACME.20080101.OS.Resource1</mRID>

<status>ACCEPTED</status>

<error>

<ns1:severity>WARNING</ns1:severity>

<text>Five-minute intervals beginning 21:10 through interval 22:0 conflict with

existing Energy Offer Curve</text>

</error>

<error>

<ns1:severity>INFORMATIVE</ns1:severity>

<text>Successfully validated the ERCOT Output Schedule.</text>

</error>

<error>

<ns1:severity>WARNING</ns1:severity>

<text>The Output Schedule exists for hour ending 22 where resource status is OFF </text>

</error>

</OutputSchedule>

</BidSet>

### Two Hour Warning Results

This section describes the interface used to retrieve the two hour warning results. The result set will hold the Output Schedule Validation results (errors and warnings) for the specified trading date and the end hour.

The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | TwoHrNotif |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/TradingDate | *Trading date* |
| Request/Option | *End of Adjustment Period Hour Ending* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | TwoHrNotif |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload | BidSet/OutputSchedule |

The payload structure is described by the following diagram:



Figure 130 – Two Hour Warning results: BidSet/OutputSchedule

The following is an XML example for Two Hour Warning results:

<BidSet xmlns="http://www.ercot.com/schema/2007-05/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<tradingDate>2008-01-01</tradingDate>

<OutputSchedule>

<mRID>ACME.20080101.OS.Resource1</mRID>

<status>ACCEPTED</status>

<error>

<ns1:severity>WARNING</ns1:severity>

<text>Five-minute intervals beginning 21:10 through interval 22:0 conflict with

existing Energy Offer Curve</text>

</error>

<error>

<ns1:severity>INFORMATIVE</ns1:severity>

<text>Successfully validated the ERCOT Output Schedule.</text>

</error>

<error>

<ns1:severity>WARNING</ns1:severity>

<text>The Output Schedule exists for hour ending 22 where resource status is OFF </text>

</error>

</OutputSchedule>

</BidSet>

### DAM Phase II Validation Results

This section describes the interface used to retrieve the DAM Phase II Validation results. The result set will hold the requested Bid Type Validation results (bid cancellations only) for the specified trading date.

The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | P2ValidationSet |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/TradingDate | *Trading date* |
| Request/Option | *BidType of interest*  *See below for valid bid types* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | P2ValidationSet |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload | BidSet/[Bid Type] |

Valid Bid Types are:

| *Valid Bid Type* | *Description* |
| --- | --- |
| ASOffer | Ancillary Service Offer |
| ASTrade | Ancillary Service Trade |
| AVP | Availability Plan |
| CapacityTrade | Capacity Trade |
| COP | Current Operating Plan |
| CRR | PTP Obligation w/ Link to Option |
| EnergyBid | DAM Energy Bid |
| EnergyOnlyOffer | DAM Energy-Only Offer |
| EnergyTrade | Energy Trade |
| IncDecOffer | Inc Dec Energy Offer |
| OutputSchedule | Output Schedule |
| PTPObligation | PTP Obligation Bid |
| RTMEnergyBid | Real-Time Market Energy Bid |
| SelfArrangedAS | Self-Arranged AS |
| SelfSchedule | Self-Schedule |
| ThreePartOffer | Three Part Supply Offer |

The payload structure is described by the following diagram:

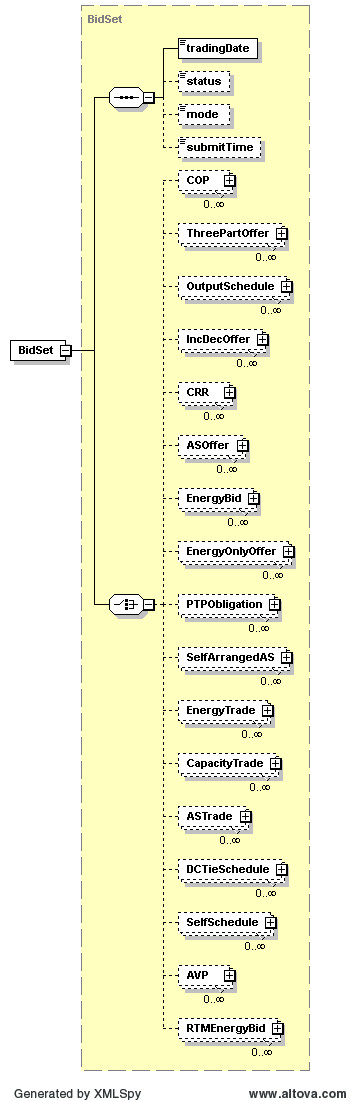


Figure 131 – DAM Phase II Validation Results: BidSet/[Bid Type]

The following is an XML example for DAM Phase II validation:

<BidSet xmlns="http://www.ercot.com/schema/2007-05/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

            <tradingDate>2008-02-19</tradingDate>

            <ThreePartOffer>

                        <mRID>ACME.20080101.TPO.DG\_BIOE\_2UNITS</mRID>

                        <status>CANCELED</status>

                                    <error>

<severity>ERROR</severity>

                                                <text>Validation of the Energy Three Part Offer failed.</text>

</error>

                                    <error>

<severity>ERROR</severity>

                                                <text>The data required for credit exposure calculation cannot be found</text>

</error>

            </ThreePartOffer>

</BidSet>

### System Parameters

This section describes the interface used to retrieve the most recent Energy management System (EMS) System Parameters. The result set will hold the near real time values available from EMS.

These Real-Time summaries are also available to ERCOT operators and all Market Participants using the MIS and ICCP, giving updates of calculations every ten seconds, which show the Real-Time total system amount of:

* Responsive Reserve Capacity from Generation Resources
* Responsive Reserve Capacity from Load Resources excluding Controllable Load Resources
* Responsive Reserve Capacity from Controllable Load Resources
* Unprocured Additional Capacity from Load Resources excluding Controllable Load Resources
* Responsive Reserve Capacity from Deployed Generation Resource and Controllable Load Resources
* Responsive Reserve Responsibility from Generation Resource
* Responsive Reserve Responsibility from Load Resources excluding Controllable Load Resources
* Responsive Reserve Responsibility from Controllable Load Resources
* Non-Spinning Reserve Capacity available from On-Line Generation Resources with Energy Offer Curves
* Non-Spinning Reserve Capacity available from undeployed Load Resources
* Non-Spinning Reserve Capacity available from Off-Line Generation Resources
* Non-Spinning Reserve Capacity available from Resources with Output Schedules
* Non-Spinning Reserve Responsibility from On-Line Generation Resources with Energy Offer Curves
* Non-Spinning Reserve Responsibility from On-Line Generation Resources with Output Schedules
* Non-Spinning Reserve Responsibility from Load Resources
* Non-Spinning Reserve Responsibility from Off-Line Generation Resources excluding Quick Start Generation Resources
* Non-Spinning Reserve Responsibility from Quick Start Generation Resources
* Undeployed Reg-Up and undeployed Reg-Down
* Deployed Reg-Up and deployed Reg-Down
* Regulation Responsibility Reg-Up and Reg-Down
* Available capacity from Controllable Load Resources available to decrease Base Points (energy consumption) in SCED
* Available capacity from Controllable Load Resources available to increase Base Points (energy consumption) in SCED
* Available capacity with Energy Offer Curves in the ERCOT System that can be used to increase Base Points in SCED
* Available capacity with Energy Offer Curves in the ERCOT System that can be used to decrease Base Points in SCED
* Available capacity without Energy Offer Curves in the ERCOT System that can be used to increase Base Points in SCED
* Available capacity without Energy Offer Curves in the ERCOT System that can be used to decrease Base Points in SCED
* Available capacity to increase Generation Resource Base Points in the next 5 minutes in SCED (HDL)
* Available capacity to decrease Generation Resource Base Points in the next 5 minutes in SCED (LDL)
* The ERCOT-wide Physical Responsive Capability (PRC)
* Real-Time On-Line Reserve Capacity
* Real-Time On-Line and Off-Line Reserve Capacity
* Total FFR computed by summing FFR MW from ONFFRRRS resource
* Total FFR Capacity computed by taking resource limits into account

The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | SystemParameters |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | SystemParameters |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload | SystemParameters |

The following is an XML example for System Parameters interface:

<ns1:SystemParameters xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:time>2016-11-08T00:34:23-06:00</ns1:time>

<ns1:rrcGenRes>1346.27062988281</ns1:rrcGenRes>

<ns1:rrcLoadResWoClr>1310.12280273438</ns1:rrcLoadResWoClr>

<ns1:rrcClr>0.0</ns1:rrcClr>

<ns1:nsrOnlineGenResWoEo>0.0</ns1:nsrOnlineGenResWoEo>

<ns1:nsrUndeployedLoadRes>0.0</ns1:nsrUndeployedLoadRes>

<ns1:nsrOfflineGenRes>937.5</ns1:nsrOfflineGenRes>

<ns1:nsrOfflineResWOS>0.0</ns1:nsrOfflineResWOS>

<ns1:undeployedRegUp>311.0</ns1:undeployedRegUp>

<ns1:undeployedRegDown>0.0</ns1:undeployedRegDown>

<ns1:capWEOIncreaseBP>0.0</ns1:capWEOIncreaseBP>

<ns1:capWEODecreaseBP>0.0</ns1:capWEODecreaseBP>

<ns1:capWOEOIncreaseBP>5531.263671875</ns1:capWOEOIncreaseBP>

<ns1:capWOEODecreaseBP>19995.287109375</ns1:capWOEODecreaseBP>

<ns1:prc>4330.8662109375</ns1:prc>

<ns1:capCLRIncreaseBP>0.0</ns1:capCLRIncreaseBP>

<ns1:capCLRDecreaseBP>0.0</ns1:capCLRDecreaseBP>

<ns1:rrcUnprocuredL>151.511703491211</ns1:rrcUnprocuredL>

<ns1:rrcDeployedGnCl>0.0</ns1:rrcDeployedGnCl>

<ns1:rrRespGn>1351.10095214844</ns1:rrRespGn>

<ns1:rrRespNcl>1348.10021972656</ns1:rrRespNcl>

<ns1:rrRespCl>0.0</ns1:rrRespCl>

<ns1:nsRespGnWEo>0.0</ns1:nsRespGnWEo>

<ns1:nsRespGnWOS>0.0</ns1:nsRespGnWOS>

<ns1:nsRespLd>10.0</ns1:nsRespLd>

<ns1:nsRespGnOff>1783.19995117187</ns1:nsRespGnOff>

<ns1:nsRespQsgr>0.0</ns1:nsRespQsgr>

<ns1:deployedRegUp>0.0</ns1:deployedRegUp>

<ns1:deployedRegDown>229.099990844727</ns1:deployedRegDown>

<ns1:regUpResp>311.0</ns1:regUpResp>

<ns1:regDnResp>229.099990844727</ns1:regDnResp>

<ns1:capGnIncreaseBp>2818.30590820312</ns1:capGnIncreaseBp>

<ns1:capGnDecreaseBp>7597.4189453125</ns1:capGnDecreaseBp>

<ns1:rtRvCapOn>8880.2197265625</ns1:rtRvCapOn>

<ns1:rtRvCapOnOff>12329.2197265625</ns1:rtRvCapOnOff>

<ns1:hslTEmr>91.0</ns1:hslTEmr>

<ns1:hslTOut>6215.57958984375</ns1:hslTOut>

<ns1:hslTOutl>372.873840332031</ns1:hslTOutl>

<ns1:ffrrt>0.0</ns1:ffrrt>

<ns1:ffrrrscp>0.0</ns1:ffrrrscp> </ns1:SystemParameters>

### RTD Indicative Base Points

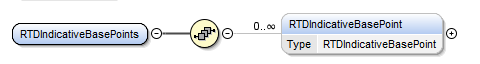
This section describes interfaces used to retrieve RTD Indicative Base Points. This report is posted for a given QSE and will contain the Base Points for every resource belonging only to that QSE for every interval in a particular RTD Study Period. The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | RTDIndicativeBasePoints |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/StartTime | *Start time of interest* |
| Request/EndTime | *End time of interest* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | RTDIndicativeBasePoints |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | RTDIndicativeBasePoints |

The structure of RTD Indicative Base Points are described by the following diagram:



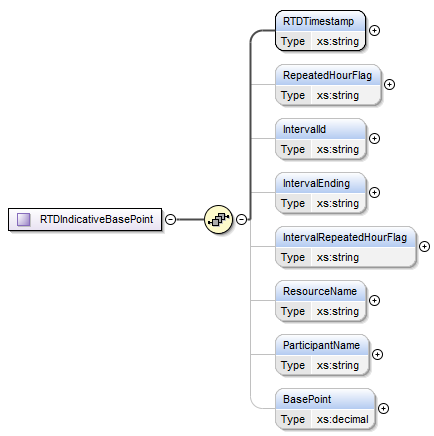


Figure 132 - RTDIndicativeBasePoints Container Structure

The following elements are used to report RTD Indicative Base Points:

* RTD Timestamp
* Repeated Hour Flag which indicates the extra hour on the DST long day
* Interval ID
* Interval Ending
* Interval Repeated Hour Flag
* Resource Name
* Participant Name
* Base Point

The following is an XML example:

<ns1:RTDIndicativeBasePoints xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:RTDIndicativeBasePoint>

<ns1:RTDTimestamp>03/30/2012 15:04:01</ns1:RTDTimestamp>

<ns1:RepeatedHourFlag>N</ns1:RepeatedHourFlag>

<ns1:IntervalId>1</ns1:IntervalId>

<ns1:IntervalEnding>03/30/2012 15:10:00</ns1:IntervalEnding>

<ns1:IntervalRepeatedHourFlag>N</ns1:IntervalRepeatedHourFlag>

<ns1:ResourceName>RES\_ABC1</ns1:ResourceName>

<ns1:ParticipantName>QABC</ns1:ParticipantName>

<ns1:BasePoint>29.7</ns1:BasePoint>

</ns1:RTDIndicativeBasePoint></ns1: RTDIndicativeBasePoints>

### RTD Indicative LMPs

This section describes interfaces used to retrieve RTD Indicative LMPs. This report will provide the final LMP at every settlement point for every interval in a particular RTD Study Period. The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | RTDIndicativeLMPs |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/StartTime | *Start time of interest* |
| Request/EndTime | *End time of interest* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | RTDIndicativeLMPs |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | RTDIndicativeLMPs |

The structure of RTD Indicative LMPs are described by the following diagram:

RTDpic1

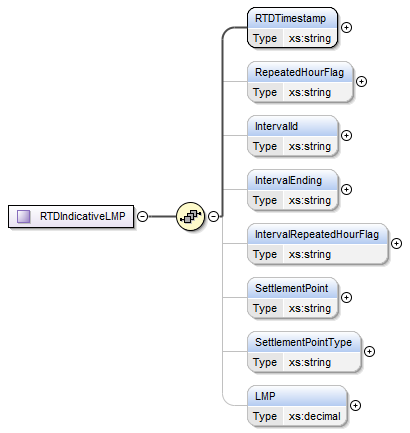


Figure 133 - RTDIndicativeLMPs Container Structure

The following elements are used to report RTD Indicative LMPs:

* RTD Timestamp
* Repeated Hour Flag which indicates the extra hour on the DST long day
* Interval ID
* Interval Ending
* Interval Repeated Hour Flag
* Settlement Point
* Settlement Point Type
* LMP

The following is an XML example:

<ns1:RTDIndicativeLMPs xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:RTDIndicativeLMP>

<ns1:RTDTimestamp>03/21/2012 16:19:01</ns1:RTDTimestamp>

<ns1:RepeatedHourFlag>N</ns1:RepeatedHourFlag>

<ns1:IntervalId>1</ns1:IntervalId>

<ns1:IntervalEnding>03/21/2012 16:25:00</ns1:IntervalEnding>

<ns1:IntervalRepeatedHourFlag>N</ns1:IntervalRepeatedHourFlag>

<ns1:SettlementPoint>STLP\_ALL</ns1:SettlementPoint>

<ns1:SettlementPointType>RN</ns1:SettlementPointType>

<ns1:LMP>17.73</ns1:LMP>

</ns1:RTDIndicativeLMP></ns1: RTDIndicativeLMPs>

### SCED ORDC Price Adders

This section describes interfaces used to retrieve SCED Total Available Reserves and Price Adders Based on ORDC. The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | SCEDORDCPriceAdders |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/StartTime | *Start time of interest* |
| Request/EndTime | *End time of interest* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | SCEDORDCPriceAdders |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | SCEDPriceAdderORDCs |

The structure of SCEDPriceAdderORDCs is described by the following diagram:

ErcotInformation_SCEDPriceAdderORDCs

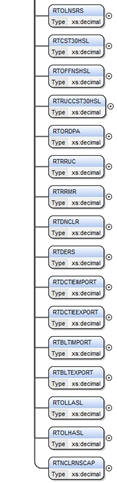
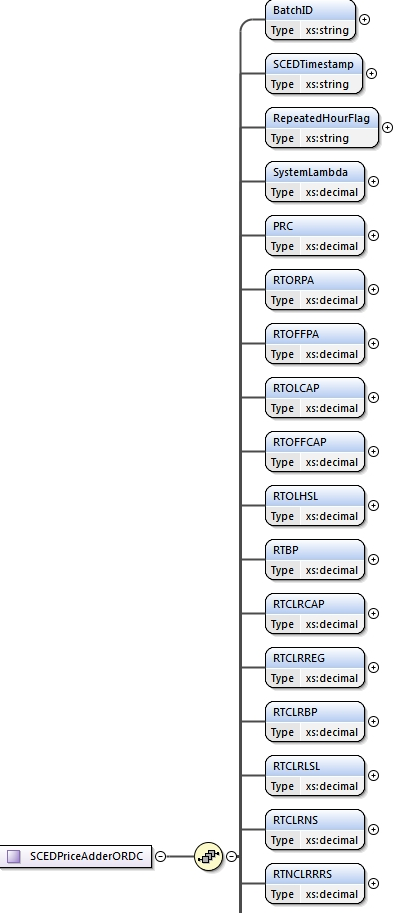


Figure 134 - SCEDPriceAdderORDC Container Structure

The following elements are used to report SCED Price Adder ORDC:

* BatchID
* SCEDTimestamp
* Repeated Hour Flag which indicates the extra hour on the DST long day
* System Lambda
* PRC
* RTORPA
* RTOFFPA
* RTOLCAP
* RTOFFCAP
* RTOLHSL
* RTBP
* RTCLRCAP
* RTCLRREG
* RTCLRBP
* RTCLRLSL
* RTCLRNS
* RTNCLRRRS
* RTOLNSRS
* RTCST30HSL
* RTOFFNSHSL
* RTRUCCST30HSL
* RTORDPA
* RTRRUC
* RTRRMR
* RTDNCLR
* RTDERS
* RTDCTIEIMPORT
* RTDCTIEEXPORT
* RTBLTIMPORT
* RTBLTEXPORT
* RTOLLASL
* RTOLHASL
* RTNCLRNSCAP

The following is an XML example:

<ns1:SCEDPriceAdderORDCs xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:SCEDPriceAdderORDC>

<ns1:BatchID>5146137</ns1:BatchID>

<ns1:SCEDTimestamp>06/04/2018 10:40:10</ns1:SCEDTimestamp>

<ns1:RepeatedHourFlag>N</ns1:RepeatedHourFlag>

<ns1:SystemLambda>8999.998</ns1:SystemLambda>

<ns1:PRC>4772.2</ns1:PRC>

<ns1:RTORPA>0.0</ns1:RTORPA>

<ns1:RTOFFPA>0.0</ns1:RTOFFPA>

<ns1:RTOLCAP>11698.87</ns1:RTOLCAP>

<ns1:RTOFFCAP>3625.47</ns1:RTOFFCAP>

<ns1:RTOLHSL>50337.16</ns1:RTOLHSL>

<ns1:RTBP>39766.7</ns1:RTBP>

<ns1:RTCLRCAP>0.0</ns1:RTCLRCAP>

<ns1:RTCLRREG>0.0</ns1:RTCLRREG>

<ns1:RTCLRBP>0.0</ns1:RTCLRBP>

<ns1:RTCLRLSL>0.0</ns1:RTCLRLSL>

<ns1:RTCLRNS>0.0</ns1:RTCLRNS>

<ns1:RTNCLRRRS>1389.73</ns1:RTNCLRRRS>

<ns1:RTOLNSRS>61.3</ns1:RTOLNSRS>

<ns1:RTCST30HSL>2681.46</ns1:RTCST30HSL>

<ns1:RTOFFNSHSL>882.7</ns1:RTOFFNSHSL>

<ns1:RTRUCCST30HSL>0.0</ns1:RTRUCCST30HSL>

<ns1:RTORDPA>0.0</ns1:RTORDPA>

<ns1:RTRRUC>0.0</ns1:RTRRUC>

<ns1:RTRRMR>0.0</ns1:RTRRMR>

<ns1:RTDNCLR>0.0</ns1:RTDNCLR>

<ns1:RTDERS>0.0</ns1:RTDERS>

<ns1:RTDCTIEIMPORT>0.0</ns1:RTDCTIEIMPORT>

<ns1:RTDCTIEEXPORT>0.0</ns1:RTDCTIEEXPORT>

<ns1:RTBLTIMPORT>0.0</ns1:RTBLTIMPORT>

<ns1:RTBLTEXPORT>0.0</ns1:RTBLTEXPORT>

<ns1:RTOLLASL>28084.18</ns1:RTOLLASL>

<ns1:RTOLHASL>57800.8</ns1:RTOLHASL>

<ns1:RTNCLRNSCAP>6800.0</ns1: RTNCLRNSCAP>

</ns1:SCEDPriceAdderORDC>

### RT Price Adder ORDC

This section describes interfaces used to retrieve Real-Time Reserve Prices Based on ORDC for 15-minute Settlement Interval. The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | RT15minORDCReservePrices |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/StartTime | *Start time of interest* |
| Request/EndTime | *End time of interest* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | RT15minORDCReservePrices |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | RTPriceAdderORDCs |

The structure of RTPriceAddersORDCs is described by the following diagram:

ErcotInformation_RTPriceAdderORDCs

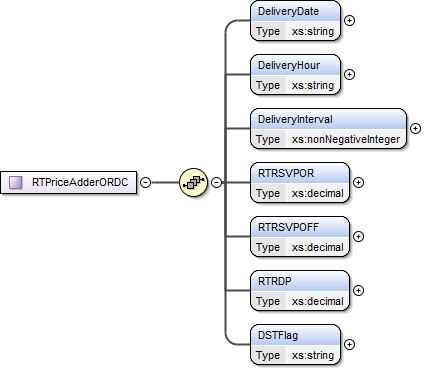


Figure 135 - RTPriceAdderORDC Container Structure

The following elements are used to report RT Price Adder ORDC:

* DeliveryDate
* DeliveryHour
* DeliveryInterval
* RTRSVPOR
* RTRSVPOFF
* RTRDP
* DSTFlag

The following is an XML example:

<ns1:RTPriceAdderORDCs xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:RTPriceAdderORDC>

<ns1:DeliveryDate>08/14/2014</ns1:DeliveryDate>

<ns1:DeliveryHour>9</ns1:DeliveryHour>

<ns1:DeliveryInterval>4</ns1:DeliveryInterval>

<ns1:RTRSVPOR>0.0</ns1:RTRSVPOR>

<ns1:RTRSVPOFF>0.0</ns1:RTRSVPOFF>

<ns1:RTRDP>0.0</ns1:RTRDP>

<ns1:DSTFlag>N</ns1:DSTFlag>

</ns1:RTPriceAdderORDC>

</ns1:RTPriceAdderORDCs>

### RTD Indicative ORDC Price Adders

This section describes interfaces used to retrieve RTD Indicative Total Available Reserves and Price Adders Based on ORDC. The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | RTDIndicativeORDCPriceAdders |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/StartTime | *Start time of interest* |
| Request/EndTime | *End time of interest* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | RTDIndicativeORDCPriceAdders |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | RTDPriceAdderORDCs |

The structure of RTDPriceAdderORDCs is described by the following diagram:

ErcotInformation_RTDPriceAdderORDCs

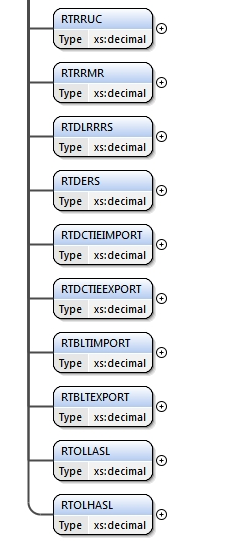
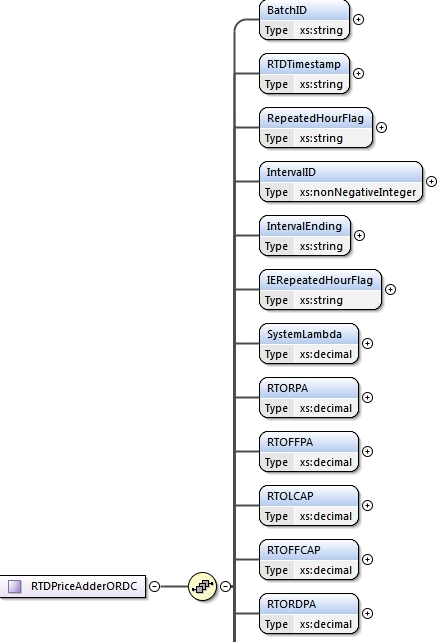


Figure 136 - RTDPriceAdderORDC Container Structure

The following elements are used to report RTD Price Adder ORDC:

* BatchID
* RTDTimestamp
* RepeatedHourFlag
* IntervalID
* IntervalEnding
* IERepeatedHourFlag
* SystemLambda
* RTORPA
* RTOFFPA
* RTOLCAP
* RTOFFCAP
* RTORDPA
* RTRRUC
* RTRRMR
* RTDLRRRS
* RTDERS
* RTDCTIEIMPORT
* RTDCTIEEXPORT
* RTBLTIMPORT
* RTBLTEXPORT
* RTOLLASL
* RTOLHASL

The following is an XML example:

<ns1:RTDPriceAdderORDCs xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:RTDPriceAdderORDC>

<ns1:BatchID>5146138</ns1:BatchID>

<ns1:RTDTimestamp>06/04/2018 10:40:01</ns1:RTDTimestamp>

<ns1:RepeatedHourFlag>N</ns1:RepeatedHourFlag>

<ns1:IntervalID>1</ns1:IntervalID>

<ns1:IntervalEnding>06/04/2018 10:45:00</ns1:IntervalEnding>

<ns1:IERepeatedHourFlag>N</ns1:IERepeatedHourFlag>

<ns1:SystemLambda>8999.9985</ns1:SystemLambda>

<ns1:RTORPA>0.0</ns1:RTORPA>

<ns1:RTOFFPA>0.0</ns1:RTOFFPA>

<ns1:RTOLCAP>11645.57</ns1:RTOLCAP>

<ns1:RTOFFCAP>3625.47</ns1:RTOFFCAP>

<ns1:RTORDPA>0.0</ns1:RTORDPA>

<ns1:RTRRUC>0.0</ns1:RTRRUC>

<ns1:RTRRMR>0.0</ns1:RTRRMR>

<ns1:RTDLRRRS>0.0</ns1:RTDLRRRS>

<ns1:RTDERS>0.0</ns1:RTDERS>

<ns1:RTDCTIEIMPORT>0.0</ns1:RTDCTIEIMPORT>

<ns1:RTDCTIEEXPORT>0.0</ns1:RTDCTIEEXPORT>

<ns1:RTBLTIMPORT>0.0</ns1:RTBLTIMPORT>

<ns1:RTBLTEXPORT>0.0</ns1:RTBLTEXPORT>

<ns1:RTOLLASL>28084.18</ns1:RTOLLASL>

<ns1:RTOLHASL>57800.8</ns1:RTOLHASL>

</ns1:RTDPriceAdderORDC>

### Photovoltaic Generation Resource Production Potential

The purpose of this interface is to provide a query for fetching Photovoltaic Generation Resource Production Potential (PVGRPP) for each Photovoltaic Generation Resource (PVGR) as well as PVGRPP forecast for all PVGRs. This interface will return the last hour data for the requesting hour-end.

The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | *get* |
| Header/Noun | PVGRPP |
| Header/Source | *MARKET PARTICIPANT ID* |
| Header/UserID | *ID of user* |
| Request/endTime | *End time of interest* |
| Request/ID | *Optional: PVGR identity*  *If No PVGR is provided, PVGRPP for all PVGRs is returned.* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | *reply* |
| Header/Noun | PVGRPP |
| Header/Source | *ERCOT* |
| Reply/ReplyCode | *Reply code, success=OK* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | *PVGRPP* |

The following diagram defines the ForecastSolarPayload structure of the PVGRPP response

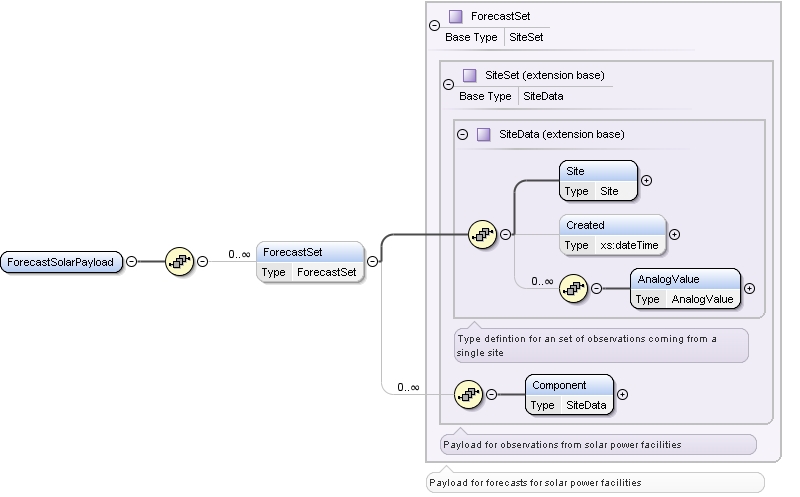


Figure 137 – PVGRPP Structure

The following is an abbreviated XML example for a PVGRPP response:

<ns1:ForecastSolarPayload xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il"  
 xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">  
 <ns1:ForecastSet>  
 <ns1:Site duns="1234567890000" name="SOLAR1" qseid="QSE1">SITE1</ns1:Site>  
 <ns1:Created>2015-12-18T14:00:01-06:00</ns1:Created>  
 <ns1:AnalogValue statistic="MEAN" timeStamp="2015-12-18T14:00:00-06:00" type="PVGRPP"  
 units="MW">0.0E0</ns1:AnalogValue>  
 <ns1:AnalogValue statistic="MEAN" timeStamp="2015-12-18T15:00:00-06:00" type="PVGRPP"  
 units="MW">0.0E0</ns1:AnalogValue>  
 </ns1:ForecastSet>  
</ns1:ForecastSolarPayload>

### Short-Term Photovoltaic Power Forecast

The purpose of this interface is to provide a query for fetching short-term photovoltaic power forecast (STPPF) for each PVGR and STPPF forecast for all PVGRs. This interface will return the last hour data for the requesting hour-end.

The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | *get* |
| Header/Noun | STPPF |
| Header/Source | *MARKET PARTICIPANT ID* |
| Header/UserID | *ID of user* |
| Request/endTime | *End time of interest* |
| Request/ID | *Optional: PVGR identity*  *If No PVGR provided then* STPPF *for all PVGRs is returned* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | *reply* |
| Header/Noun | STPPF |
| Header/Source | *ERCOT* |
| Reply/ReplyCode | *Reply code, success=OK* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | *STPPF* |

The following diagram defines the ForecastSolarPayload structure in the STPPF response:

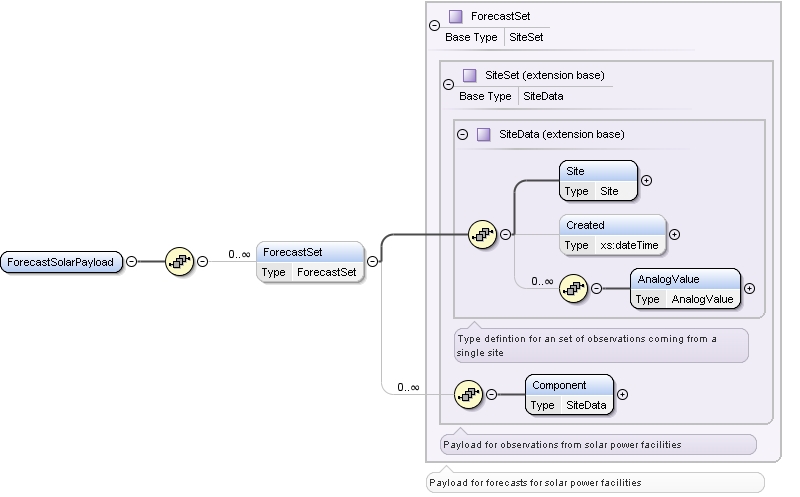


Figure 138 - STPPF Structure

The following is an abbreviated XML example of an STPPF response:

<ns1:ForecastSolarPayload xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il"  
 xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">  
 <ns1:ForecastSet>  
 <ns1:Site duns="1234567890000" name="SOLAR1" qseid="QSE1">SITE1</ns1:Site>  
 <ns1:Created>2015-12-18T14:00:01-06:00</ns1:Created>  
 <ns1:AnalogValue statistic="MEAN" timeStamp="2015-12-18T14:00:00-06:00" type="STPPF"  
 units="MW">0.0E0</ns1:AnalogValue>  
 <ns1:AnalogValue statistic="MEAN" timeStamp="2015-12-18T15:00:00-06:00" type="STPPF"  
 units="MW">0.0E0</ns1:AnalogValue>  
 </ns1:ForecastSet>  
</ns1:ForecastSolarPayload>

### Price Corrected SOTG/SODG 15-min Prices

This section describes interfaces used to retrieve price corrected SOTG/SODG 15 minute settlement meter prices. The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | SOGPricesPC |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/StartTime | *Start time of interest* |
| Request/EndTime | *End time of interest* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | SOGPricesPC |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | RTMPriceCorrectionSOGPRICES |

The structure of RTMPriceCorrectionSOGPRICES are described by the following diagram:



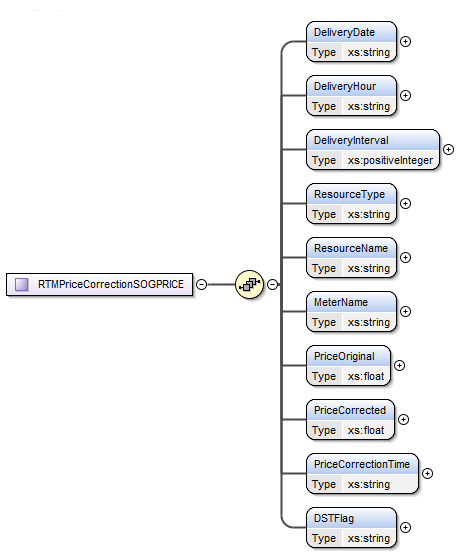


Figure 139 - RTMPriceCorrectionSOGPRICE Container Structure

The following elements are used to report price corrections for SOTG/SODG 15-min settlement meter prices:

* Delivery Date
* Delivery Hour
* Delivery Interval
* Resource Type
* Resource Name
* Meter Name
* Price Original
* Price Corrected
* Price Correction Time
* DST Flag

The following is an XML example:

<ns1:RTMPriceCorrectionSOGPRICES xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:RTMPriceCorrectionSOGPRICE>

<ns1:DeliveryDate>07/28/2021</ns1:DeliveryDate>

<ns1:DeliveryHour>11</ns1:DeliveryHour>

<ns1:DeliveryInterval>3</ns1:DeliveryInterval>

<ns1:ResourceType>SOTG</ns1:ResourceType>

<ns1:ResourceName>RES\_XXXX2</ns1:ResourceName>

<ns1:MeterName>METER0002</ns1:MeterName>

<ns1:PriceOriginal>38.22</ns1:PriceOriginal>

<ns1:PriceCorrected>38.16</ns1:PriceCorrected>

<ns1:PriceCorrectionTime>07/30/2021 16:00:00</ns1:PriceCorrectionTime>

<ns1:DSTFlag>N</ns1:DSTFlag>

</ns1:RTMPriceCorrectionSOGPRICE> <ns1:RTMPriceCorrectionSOGPRICE>

<ns1:DeliveryDate>07/28/2021</ns1:DeliveryDate>

<ns1:DeliveryHour>11</ns1:DeliveryHour>

<ns1:DeliveryInterval>3</ns1:DeliveryInterval>

<ns1:ResourceType>SOTG</ns1:ResourceType>

<ns1:ResourceName>RES\_XXXX2</ns1:ResourceName>

<ns1:MeterName>METER0003</ns1:MeterName>

<ns1:PriceOriginal>38.22</ns1:PriceOriginal>

<ns1:PriceCorrected>38.16</ns1:PriceCorrected>

<ns1:PriceCorrectionTime>07/30/2021 16:00:00</ns1:PriceCorrectionTime>

<ns1:DSTFlag>N</ns1:DSTFlag>

</ns1:RTMPriceCorrectionSOGPRICE></ns1:RTMPriceCorrectionSOGPRICES>

### Price Corrected SOTG/SODG LMPs

This section describes interfaces used to retrieve price corrected SOTG/SODG 5-min SCED LMPs. The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | SOGLMPsPC |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/StartTime | *Start time of interest* |
| Request/EndTime | *End time of interest* |

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | SOGLMPsPC |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | *RTMPriceCorrectionSOGLMPS* |

The structure of RTMPriceCorrectionSOGLMPS are described by the following diagram:



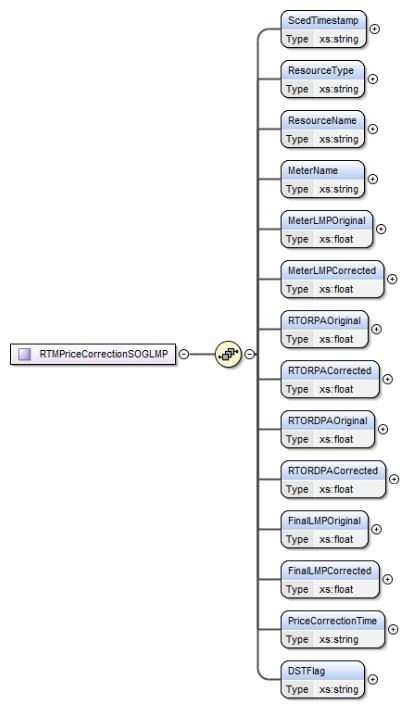


Figure 140 - RTMPriceCorrectionSOGLMP Container Structure

The following elements are used to report price corrections for SOTG/SODG 5-min SCED LMPs:

* SCED Timestamp
* Resource Type
* Resource Name
* Meter Name
* Meter LMP Original
* Meter LMP Corrected
* RTORPA Original
* RTORPA Corrected
* RTORDPA Original
* RTORDPA Corrected
* Final LMP Original
* Final LMP Corrected
* Price Correction Time
* DST Flag

The following is an XML example:

<RTMPriceCorrectionSOGLMPS>  
 <RTMPriceCorrectionSOGLMP>  
 <ScedTimestamp>07/28/2021 10:41:51</ScedTimestamp>  
 <ResourceType>SOTG</ResourceType>  
 <ResourceName>RES\_XXXX1</ResourceName>  
 <MeterName>METER0000</MeterName>  
 <MeterLMPOriginal>38.84</MeterLMPOriginal>  
 <MeterLMPCorrected>38.57</MeterLMPCorrected>  
 <RTORPAOriginal>0</RTORPAOriginal>  
 <RTORPACorrected>0</RTORPACorrected>  
 <RTORDPAOriginal>0</RTORDPAOriginal>  
 <RTORDPACorrected>0</RTORDPACorrected>  
 <FinalLMPOriginal>38.84</FinalLMPOriginal>  
 <FinalLMPCorrected>38.57</FinalLMPCorrected>  
 <PriceCorrectionTime>07/30/2021 16:00:00</PriceCorrectionTime>  
 <DSTFlag>N</DSTFlag>  
 </RTMPriceCorrectionSOGLMP>  
 <RTMPriceCorrectionSOGLMP>  
 <ScedTimestamp>07/28/2021 10:41:51</ScedTimestamp>  
 <ResourceType>SOTG</ResourceType>  
 <ResourceName>RES\_XXXX1</ResourceName>  
 <MeterName>METER0001</MeterName>  
 <MeterLMPOriginal>38.84</MeterLMPOriginal>  
 <MeterLMPCorrected>38.57</MeterLMPCorrected>  
 <RTORPAOriginal>0</RTORPAOriginal>  
 <RTORPACorrected>0</RTORPACorrected>  
 <RTORDPAOriginal>0</RTORDPAOriginal>  
 <RTORDPACorrected>0</RTORDPACorrected>  
 <FinalLMPOriginal>38.84</FinalLMPOriginal>  
 <FinalLMPCorrected>38.57</FinalLMPCorrected>  
 <PriceCorrectionTime>07/30/2021 16:00:00</PriceCorrectionTime>  
 <DSTFlag>N</DSTFlag>  
 </RTMPriceCorrectionSOGLMP>

</RTMPriceCorrectionSOGLMPS >

# Notifications

A key aspect of the design for external interfaces for Market Participants is to provide a simplified variant of the OASIS WS-Notifications standard for the definition of an ERCOT Notification interface. Each Market Participant using the external interface would be required to provide a listener interface for the receipt of notification messages, compliant with the interface provided by ERCOT.

## Interfaces Provided

The interfaces provided are based upon the WS-Notifications specification, although for use by ERCOT the interface has been simplified and augmented to fill one gap in the WS-Notification specification. For the purposes of use by Market Participants, there is only one interfaces of interest that is implemented by a Market Participant listener; this is the ‘Notify’ interface.

The ‘Notify’ interface is used as a means to asynchronously receive information by Market Participants from ERCOT. Specific examples of this information include:

* Notices
* Alerts
* BidSet acceptance or error reporting
* Awards
* Obligations
* Startup/Shutdown Instructions
* Outages

The following sequence diagram describes the typical/potential set of information exchanges related to Notifications. The actual flows would be different based on the type of notification:

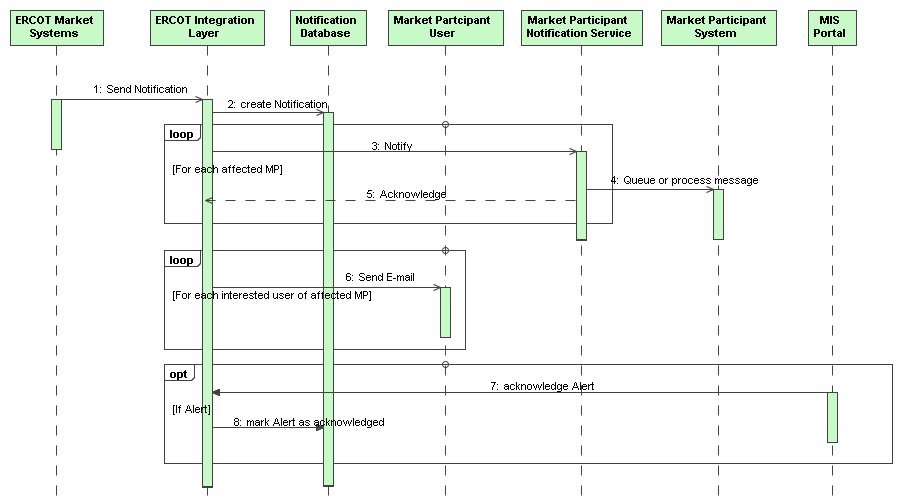


Figure 141 - Notification Sequence Diagram

## Interfaces Required

The contents of the any structure would be wrapped using the ResponseMessage structure defined in section 2.1.4, as this would allow necessary security information to be passed in a uniform manner. The following table describes the contents of the Notify message.

|  |  |
| --- | --- |
| Message Element | Value |
| NotificationMessage/Message/any | *One or more messages as defined using ERCOT common message structure* |

The following table describes the contents of the Acknowledge message.

|  |  |
| --- | --- |
| Message Element | Value |
| ReplyCode | *OK or ERROR* |
| TimeStamp | *Current time string* |

## Message Specifications

The WS-Notifications specification identifies a single message structure for notification messages. The payload that is to be delivered by the notification message is represented by the ‘any’ element in the following message structure, loosely coupling the definition of the payload from the definition of the container message structure defined by WS-Notifications thereby allowing WS-Notifications to be used with any XML data structure. ERCOT uses the SubscriptionReference, Topic and ProducerReference are not used.



Figure 142 - WS-Notifications Message Structure

It is important to note that the payloads for NotificationMessages are ALWAYS conveyed using the common message structure (e.g. Message, RequestMessage, or ResponseMessage) as defined in section 2.1 serving as the ‘any’ sub-structure in the NotificationMessage structure.

ERCOT uses a variant of WS-Notifications, where an ‘Acknowledge’ message is sent to acknowledge the receipt of a notification. The WSDL and XSD provided by ERCOT are also simplified to eliminate definitions for interfaces and structures not used by ERCOT. These are described in the appendix.

ERCOT will allow a Market Participant to specify up to two URLs. ERCOT will send notifications to port 443 of the specified URL using an HTTPS connection. ERCOT will sign the SOAP message, where the signature can be verified using the ERCOT public key. This will verify that the notification was sent by ERCOT and was not altered. The ‘Acknowledge’ message does not need to be signed.

### Notices and Alerts

The specific payload for an Alert or Notice is described in the following diagram. Event container is used to hold Alerts and Notices. While there may be many different types of Notifications, the primary difference between a Notice and an Alert is that there is an expectation that an Alert would be acknowledged:

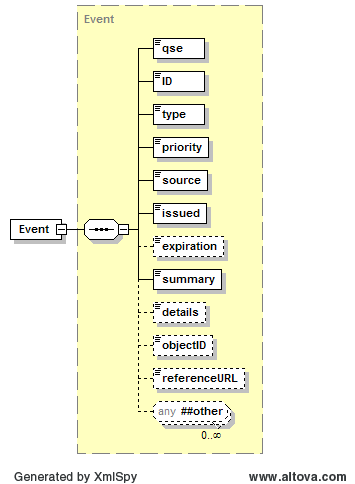


Figure 143 - Event Structure

Example:

<Event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotEvents.xsd" xmlns="http://www.ercot.com/schema/2007-06/nodal/ews">

            <qse>TXU</qse>

            <ID>CM-CAPTRD-NOTF</ID>

            <type>Reported Trade</type>

            <priority>High</priority>

            <source>MMS</source>

            <issued> 2008-10-17T12:53:41.115-05:00 </issued>

            <summary>Trade Submission update by TXU: Confirmed: YES, for Trade Date: 11/21/2008,  Buyer: TXU, Seller: Calpine</summary>

</Event>

#### MMS System-Generated Notices

The following notices are Market Management System (MMS) system-generated and will display on the MMS market-facing user interface on the notices page as well as on the MIS website’s Notices page. Additionally these events are sent out asynchronously to the Market Participants Listeners.

In addition to the protocol-required notices, the MMS system generates some additional notices that are documented below.

The asynchronous outgoing notification alerts will have the following verb/noun combination:

* verb=created
* noun=Alert

| # | Alert Text | Description | Type + Report | Priority | Audience |
| --- | --- | --- | --- | --- | --- |
| 1 | CM-ASCK-FAIL AS Responsibility Check failed for Trade Date: <MM/DD/YYYY>, Delivery Hour: <HH> for AS Type: <AS\_TYPE>. The Total COP is <XX> MW and the AS Difference = <XX> MW | ERCOT shall notify the QSE if the sum of the Ancillary Service capacity designated in the COP for each hour, by service type) is less than the QSE's Ancillary Service Supply Responsibility for each service type for that hour. If the QSE does not correct the deficiency within one hour after receiving the notice from ERCOT, then ERCOT shall follow the procedures outlined in Section 6.4.8.1, Evaluation, and Maintenance of Ancillary Service Capacity Sufficiency. | COP Warning | High | A QSE |
| 2 | DRUC-AWD-RPT Reports on DRUC Awards for operating date <MM/DD/YYYY> are available.  HRUC-AWD-RPT Reports on HRUC Awards for operating date <MM/DD/YYYY> hr <XX> are available. | ERCOT shall review the RUC-recommended Resource commitments to assess feasibility and shall make any changes that it considers necessary, in its sole discretion. ERCOT shall notify each QSE which of its Resources have been committed as a result of the RUC process. | RUC Commitment  Noun: StartupShutdownInstructions  Synch : 4.3.27  Async: 5.3.13 | Low | A QSE |
| 3 | DAM-ASOBL-RPT Reports on AS obligation for operating day <MM/DD/YYYY> are available | By 0600 of the Day-Ahead, ERCOT shall notify each QSE of its Ancillary Service Obligation for each service and for each hour of the Operating Day | QSE Data Available  Noun: ASObligations  Synch: 4.3.15  Async: 5.3.11 | Low | A QSE |
| 4 | CM-CAPTRD-NOTF Capacity Trade <Submission/Cancel> update by <PARTICIPANT\_NAME>: Confirmed: <NO/YES>, for Trade Date: <MM/DD/YYYY>, Buyer:<PARTICIPANT\_NAME, Seller:<PARTICIPANT\_NAME> | When a Capacity Trade is reported to ERCOT, ERCOT shall notify the counterparty to the trade. | Reported Trade  None: Confirmed and Unconfirmed Trades  Synch: 4.3.46,47  Async: 5.3.4 | High | counterparty to the trade. |
| 5 | CM-ENGTRD-NOTF Energy Trade <Submission/Cancel> update by <PARTICIPANT\_NAME>: Confirmed: <NO/YES>, for Trade Date: <MM/DD/YYYY>, Buyer:<PARTICIPANT\_NAME, Seller:<PARTICIPANT\_NAME>, Settlement Point: <SETTLEMENT\_POINT> | When an Energy Trade is reported to ERCOT, ERCOT shall notify the counterparty to the trade. | Reported Trade  Noun: Confirmed and Unconfirmed Trades  Synch: 4.3.46,47  Async: 5.3.4 | High | counterparty to the trade. |
| 6 | CM-ASTRD-NOTF AS <AS\_TYPE> Trade <Submission/Cancel> update by <PARTICIPANT\_NAME>: Confirmed: <NO/YES>, for Trade Date: <MM/DD/YYYY>, Buyer:<PARTICIPANT\_NAME>,Seller:<PARTICPANT\_NAME> | When an Ancillary Service Trade is reported to ERCOT, ERCOT shall notify the counterparty to the trade. | Reported Trade  Noun: Confirmed and Unconfirmed Trades  Synch 4.3.46,47  Async: 5.3.4 | High | counterparty to the trade. |
| 7 | SCED-ADJPER-FAIL End of Adjustment Period: Output Schedule Validation FAILED for Hour:<HH> and Date: <MM/DD/YYYY>,. Please find the details in the report <REPORT\_NAME>. | If a valid Energy Offer Curve or an Output Schedule does not exist for a Resource that has a status of On-Line at the end of the Adjustment Period, then ERCOT shall notify the QSE and set the Output Schedule equal to the then current telemetered output of the Resource until an Output Schedule or Energy Offer Curve is submitted in a subsequent Adjustment Period. | Submission Warning | Med | A QSE |
| 8 | DAM AWARDS:  DAM-ASAWD-RPT Reports on AS Awards for operating date <MM/DD/YYYY> are available.  DAM-TPAWD-RPT Reports on 3Part Awards for operating date <MM/DD/YYYY> are available.  DAM-ENGYAWD-RPT Reports on Energy Awards for operating date <MM/DD/YYYY> are available.  DAM-CRRAWD-RPT Reports on CRR Awards for operating date <MM/DD/YYYY> are available.  DAM-PTPAWD-RPT Reports on PTP Obligation Awards for operating date <MM/DD/YYYY> are available. | Cleared DAM transaction (e.g., the buyer and the seller) of the results of the DAM:  (a) Awarded Ancillary Service Offers, specifying Resource, MW, Ancillary Service Type, and price, for each hour of the awarded offer;  (b) Awarded energy offers from Three-Part Supply Offers and from DAM Energy-Only Offers, specifying Resource (except for DAM Energy-Only Offers), MWh, Settlement Point, and Settlement Point Price, for each hour of the awarded offer;  (c) Awarded DAM Energy Bids, specifying MWh, Settlement Point, and Settlement Point Price for each hour of the awarded bid;  (d) Awarded CRR Offers (PTP Options and PTP Options with Refund), specifying CRR identifier(s), number of CRRs in MW, source and sink Settlement Points, and price, for each Settlement Interval of the awarded offer; and  (e) Awarded PTP Obligation Bids, number of PTP Obligations in MW, source and sink Settlement Points, and price for each Settlement Interval of the awarded bid. | QSE Data Available  Nouns:  AwardSet  AwardedAS  AwardedCRR  AwardedEnergyBid  AwardedEnergyOffer  AwardedEnergyOnlyOffer  AwardedPTPObligation  Synch: 4.3.1- 4.3.7  Async: 5.3.5 - 5.3.11 | Low | A QSE |
| 9 | SCED-TWOHR-FAIL Two Hour Notification: Check FAILED due to missing Output Schedules/Energy Offer Curves for Resource: <RESOURCE\_NAME>, Hour: <HH> and Date: <MM/DD/YYYY>  SCED-TWOHR-FAIL Two Hour Notification: Output Schedule Validation FAILED for Hour: <HH> and Date: <MM/DD/YYYY>. Please find the details in the report MP\_S\_OUTSCHED\_TwoHrNotif\_<HH>. | Notify the QSE that an Energy Offer Curve or Output Schedule has not yet been submitted for a Resource as a reminder that one of the two must be submitted by the end of the Adjustment Period | Submission Warning | Med | A QSE |
| 10 | SCED-TELOS-NOTF Telemetered Output Schedule Notification: Output Schedule for Resource: <RESOURCE\_NAME> | If a valid Output Schedule does not exist for a Resource that has a status of On-Line Dynamically Scheduled Resource at the time of SCED execution, then ERCOT shall notify the QSE and set the Output Schedule equal to the telemetered output of the Resource until a revised Output Schedule is validated. | Submission Warning | Med | A QSE |
| 11 | CM-CONSCK-FAIL Consistency Check for Resource Status failed for Resource: <RESOURCE\_NAME>, Operating Date: <MM/DD/YYYY> , Operating Hour: <HH>, Cop Status: <COP\_STATUS>, Telemetered Status: <TEL\_STATUS> | Five minutes before the end of each hour, ERCOT shall identify inconsistencies between the telemetered Resource Status and the Resource Status stated in the COP for that Resource in the next hour. On detecting an inconsistency, ERCOT shall provide a notice of inconsistent Resource Status to the QSE using the Messaging System | COP Warning | Med | A QSE |
| 12 | SASM-STDYPER-NOTF SASM study period for SASM ID <*YYYYMMDDHHMMSS*> set from <MM/DD> hour <HH> to <MM/DD> hour <HH>. |  | Market Message | Low | All QSEs |
| 13 | SASM-ASOBL-RPT Reports on SASM Obligation for SASM ID <*YYYYMMDDHHMMSS*> from operating date <MM/DD> hour <HH> through operating date <MM/DD> hour <HH> are available. | Notify QSEs of any additional Ancillary Service Obligation, allocated to each LSE and aggregated to the QSE level. | Market Message  Noun: ASObligations  Synch: 4.3.15  Async: 5.3.11 | Low | A QSE |
| 14 | SASM-OPEN-NOTF SASM has opened for SASM ID <*YYYYMMDDHHMMSS*> for Self AS submission. SASM will close at <HH:MM>. | Notify all QSEs of intent to procure additional Ancillary Services. | Market Message | Low | All QSEs |
| 15 | SASM-ASAWD-RPT Reports on SASM Awards for SASM ID <*YYYYMMDDHHMMSS*> from operating date <MM/DD> hour <HH> through operating date <MM/DD/HH> are available. | Notify each QSE of its awarded Ancillary Service Offer quantities in each SASM, specifying Resource, Ancillary Service type, SASM MCPC, and first and last hours of the awarded offer. | Market Message  Noun: AwardedAS  Synch: 4.3.2  Async: 5.3.8 | Low | All QSEs |
| 16 | SASM-ASMCPC-RPT Reports for SASM AS MCPC for SASM ID <*YYYYMMDDHHMMSS*> from operating date <MM/DD> hour <HH> through operating date <MM/DD/HH> are available. | Notify QSEs that SASM MCPC reports are available | QSE Data Available | Low | All QSEs |
| 17 | DAM-ASINSUFF-NOTF Due to AS Insufficiency, DAM market for <MM/DD/YYYY> has reopened for AS offer resubmission. The DAM market will close today at <HH:MM>. | ERCOT shall declare an Ancillary Service insufficiency and issue an Alert under Section 6.5.9.3.3, Alert. | Watch | High | All QSEs |
| 18 | DAM-CLOSE-NOTF DAM has closed for <MM/DD/YYYY> | Notify QSEs that the DAM has closed | Market Message | Low | All QSEs |
| 19 | DAM-RECLOSE-NOTF DAM has closed due to AS Insufficiency for <MM/DD/YYYY>. | Notify QSEs that the DAM has closed again after reopening for AS insufficiency. | Market Message | Low | All QSEs |
| 20 | CM-OPHRCLOSE-NOTF Market has closed for <MM/DD/YYYY> hr <HH>. | Notify QSEs that the Market has closed | Market Message | Low | All QSEs |
| 21 | CM-MKTOPEN-NOTF Market has opened for <MM/DD/YYYY>. | Notify QSEs that the Market has opened | Market Message | Low | All QSEs |
| 22 | CM-ADJOPEN-NOTF Adjustment window has opened for <MM/DD/YYYY>. | Notify QSEs that the adjustment window has opened | Market Message | Low | All QSEs |
| 23 | CM-ADJCLOSE-NOTF Adjustment window has closed for <MM/DD/YYYY> hr <HH>. | Notify QSEs that the adjustment window has closed | Market Message | Low | All QSEs |
| 24 | DAM-P2VAL-NOTF Phase2 Validation has completed for the date: <MM/DD/YYYY> | Notify QSEs that the Phase 2 validation (for credit validation, etc) has completed. | Market Message | Low | All QSEs |
| 25 | DAM-P2VAL-ERR Phase2 Validation has failed for the date: <MM/DD/YYYY>, type: <DATA\_TYPE>. Please find the details in the report <REPORT\_NAME>. | Notify QSEs that the Phase 2 validation has failed. | Market Message  Noun: NONE  Synch: none  Async: none | Med | A QSE |
| 26 | DAM-ASINSUFF-RPT Reports on AS Insufficiency for operating date <MM/DD/YYYY> are available. | Notify QSEs that AS Insufficiency reports are available | QSE Data Available  Noun: InsufficiencyReport  Synch: 4.3.46  Async: 5.3.15 | Low | All QSEs |
| 27 | DAM-ASMCPC-RPT Reports on AS MCPC for operating date <MM/DD/YYYY> are available. | Notify QSEs that AS MCPC reports are available | QSE Data Available  Noun: MCPCs  Synch: 4.3.12  Async: none | Low | All QSEs |
| 28 | SASM-CLOSE-NOTF SASM has closed for Self AS submission. | Notify QSEs that SASM has closed for AS submission | Market Message | Low | All QSEs |
| 29 | CM-COPCK-NOTF Cop Update Check failed for Resource: <RESOURCE\_NAME>, Operating Date: <MM/DD/YYYY> , Operating Hour: <HH>, Cop Status: <XX> | Notify QSEs who do not have complete COPs for all hours of the current operating day and the following day | COP Warning | Med | A QSE |
| 30 | CM-ASM-NOTF AS\_TYPE: <AS\_TYPE>, RES\_NAME:<RESOURCE\_NAME>, DEPLOY\_MW: <XX.XX>, BEGIN\_TIME: <MM/DD/YYYY> <HH>24:MI:SS CST/CDT, END\_TIME: <MM/DD/YYYY> <HH>24:MI:SS CST/CDT, DURATION: <XX.X> Hrs | Notify QSE of AS deployment | Market Message | High | A QSE |
| 31 | CM-ASM-NOTF AS\_TYPE: ERS, QSE\_NAME:<QSE\_NAME>, RAMP\_TYPE: <10 or 30>, GROUP\_NUMBER: <1, 2, 3, etc.>, GROUP\_TYPE: <t or d>, WS\_TYPE: <WS or NON-WS> ,DEPLOY\_MW: <XX.XX>, BUSINESS\_HOUR: <TP1, TP2, TP3, TP4, TP5 or TP6>, BEGIN\_TIME: <YYYY-MM-DD HH24:MI:SS CST/CDT> | Notify QSE of AS deployment for ERS | Market Message | High | A QSE |
| 32 | CM-ASM-NOTF AS\_TYPE: <AS\_TYPE>, DEPLOY\_MW: <XX.XX>, BEGIN\_TIME: <MM/DD/YYYY HH24:MI:SS CST/CDT>, END\_TIME: <MM/DD/YYYY HH24:MI:SS CST/CDT>, DURATION: <XX.X> Hrs | Notify QSE of AS deployment Summary | Market Message | High | All QSEs |
| 33 | CM-ASM-NOTF ERCOT Emergency Response <Service-10/Service-30> is Being <Deployed/ Recalled> | ERS Recall and Deployment messages | Operational Information | High | All QSEs |
| 34 | CM-VDI-NOTF Verbal Dispatch Instructions for acknowledgement have been sent for the Resource: *<RESOURCE\_NAME>* | Confirm with a QSE that VDIs were sent for a resource | Market Message | Low | A QSE |
| 35 | CM-SWCAP-NOTF SWCAP has been changed for AS/ENGY to <XXXX.XX> | Notify QSEs that the system-wide offer cap has changed for AS, Energy, or both. | Market Message | High | All QSEs |
| 36 | CM-COPVAL-NOTF For CC Unit: UNIT\_NAME, there are two or more modes online for Date: <MM/DD/YYYY> | MMS hourly process that checks the COP for combined cycle configurations and will issue warning messages sent to the QSE if two or more configurations are marked as online in any one of the remaining hours of today and all the hours for tomorrow. | Market Message | High | A QSEs |
| 37 | CM-COPVAL-NOTF For resource: <RESOURCE\_NAME> the COP Status does not match with the other jointly owned units belonging to Physical <RESOURCE\_NAME> for Date: <MM/DD/YYYY> for one or more hours | MMS hourly process that issues warning messages to QSEs if two or more jointly owned units belonging to the same physical resource do not have the same Resource Status in the COP for any of the remaining hours of today and all the hours of tomorrow. | Market Message | High | A QSE |
| 38 | SCED-ADJPER-WARN End of Adjustment Period: Reported warning(s) for Hour: <HH> and Date: <MM/DD/YYYY> Please find the details in the report <REPORT\_NAME>. | MMS hourly process that checks for Output Schedule ramp rate and HSL/LSL violations for the hour interval that is at the end of the Adjustment Period. | Submission Warning | Med | A QSE |
| 39 | SCED-TWOHR-WARN Two Hour Notification: Reported warning(s) for Hour: <HH> and Date: <MM/DD/YYYY> Please find the details in the report <REPORT\_NAME>. | MMS hourly process that checks for Output Schedule ramp rate and HSL/LSL violations for the hour interval two hours from current time. | Submission Warning | Med | A QSE |
| 40 | CM-ASUND-FAIL-NOTF AS Undeliverable for: <raic\_resource\_name> on trade date: <tradedate> at delivery time <delivery\_hour, <HH>24:<delivery\_min> for AS Type: <as\_type>, RUC Type: <ruc\_type>, Violated MW: <violated\_mw> [of <cop\_as\_mw>]. | Notify QSE | Submission Warning | Med | A QSE |
| 41 | DRUC-CLOSE-NOTF DRUC has closed for <MM/DD/YYYY> | Notify QSEs that the DRUC has closed | Market Message | Low | All QSEs |
| 42 | DAM-DSRLDTEL-NOTF DSR Load Telemetry Notification: DSR Load Telemetry is lost for more than <MIN> mins | Notify QSE that their DSR Load signal has been lost, which results in suspension of DSR output schedule validation. (This alert will be generated as often as every 5 minutes). | Submission Warning | Med | A QSE |
| 43 | WRUC-COMP-NOTF WRUC has completed for mm/dd/yyyy. | Notify QSEs that the WRUC has completed. | Market Message | Low | All QSEs |
| 44 | SCED-COMP-NOTF  SCED has completed for MM/DD/YYYY HH<24>:MI:SS. | Notify QSEs that SCED has completed. | Market Message | Low | All QSEs |
| 45 | PRC\_CORR\_SCED | Notify QSEs that SCED price corrections are available. | Market Message | Low | All QSEs |
| 46 | PRC\_CORR\_DAM | Notify QSEs that DAM price corrections are available. | Market Message | Low | All QSEs |
| 47 | PRC\_CORR\_SASM | Notify QSEs that SASM price corrections are available. | Market Message | Low | All QSEs |
| 48 | RTD-COMP-NOTF RTD has completed for MM/DD/YYYY HH24:MI:SS | Notify QSEs that an RTD run has completed. | Market Message | Low | All QSEs |
| 49 | ETAG-NOTF DC Tie e-Tag <NERC\_TAG> for DC Tie <DC\_TIE\_NAME> on <DELIVERY\_DATE> with a total flow of <QTY> MWhs was processed by ERCOT for <QSE> whose PSE is <PSE>. | Certified Notice to QSE when a confirmed e-Tag is downloaded, cancelled, or curtailed by ERCOT’s systems. | Market Message | Low | All QSEs |
| 50 | CM-COP-UPD-NOTF COP HSL for <GEN\_RES> on trade date <MM/DD/YYYY> hour <HR> has been updated to <VALUE> | Notify QSE when a WGR’s or PVGR’s COP HSL value is updated with the forecast value. | Market Message | Med | A QSE |
| 51 | CM-SCED-NOTIF  <resource\_name> status <original\_status> overridden to <new\_status>, SCED at  <YYYY-MM-DD HH24:MI:SS> | Notify a QSE whose resource status has been overwritten by SCED. | Market Message | Med | A QSE |
| 52 | CM-SCED-NOTIF <resource\_name> buyback status: <Y or N>, set by SCED <YYYY-MM-DD HH24:MI:SS> | Notify a QSE of the BUYBACK status of the RUC block once SCED sets the BUYBACK\_FLAG (Y or N) for the RUC block. | Market Message | Med | A QSE |

Figure 144 - MMS System generated Alerts and Notifications

#### EMS System-Generated Notices

The following notices are Energy Management System (EMS) system-generated and will display on the MIS website’s Notices page. Additionally these notices are sent out asynchronously to the Market Participants(MP) Listeners.

When an alert clears (or becomes normal) in EMS, a subsequent alert will be generated with text (“-RETURN”) appended to original alert. This follow-up alert will also be displayed on the MIS website’s Notices page and delivered to MP listeners.

The following table identifies the Resource Limit Calculator (RLC) alarm ID, description of the requirement, notice type, priority, audience, and alert text. Please note that any alert with a \* in the description denotes that the resource has become undispatchable by SCED, and RLC will set HDL=LDL=MW Output.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | ALARMID | Description | Type | Priority | Audience | Text |
| 1 | RLCUHSL | The telemetered HSL <= 0. \* | Telemetry | High | A QSE | QSE: <QSE\_ID> UNIT: <ID> ALERT: TEL HSL BELOW 0  Note: <ID> 🡪 <SUBSTATION>-<UNIT#> |
| 2 | RLCULSL | The telemetered LSL < 0. \* | Telemetry | High | A QSE | QSE: <QSE\_ID> UNIT: <ID> ALERT: TEL LSL BELOW 0  Note: <ID> 🡪 <SUBSTATION>-<UNIT#> |
| 3 | RLCHSLAS | HSL - LSL < Sum of Reg-Up Responsibility, Reg-Down Responsibility, Responsive Responsibility and Non-Spin Responsibility. \* | Telemetry | High | A QSE | QSE: <QSE\_ID> UNIT: <ID> ALERT: HSL MINUS LSL BELOW SUM OF AS RESP  Note: <ID> 🡪 <SUBSTATION>-<UNIT#> |
| 4 | RLCSCEUR | SCED Up Ramp Rate <= 0. | Telemetry | High | A QSE | QSE: <QSE\_ID> UNIT: <ID> ALERT: SURAMP BELOW 0  Note: <ID> 🡪 <SUBSTATION>-<UNIT#> |
| 5 | RLCSCEDR | SCED Down Ramp Rate <= 0. | Telemetry | High | A QSE | QSE: <QSE\_ID> UNIT: <ID> ALERT:  SDRAMP BELOW 0  Note: <ID> 🡪 <SUBSTATION>-<UNIT#> |
| 6 | RLCCCNO | For Combined Cycle configuration number validation, set it to default or last good value not 0. \* | Telemetry | High | A QSE | QSE: <QSE\_ID> UNIT: <ID> ALERT:  INVALID CC CONFIG  Note: <ID> 🡪 <SUBSTATION>-<UNIT#> |
| 7 | RLCHLSL | Invalid HSL or LSL telemetry (quality suspect). \* | Telemetry | High | A QSE | QSE: <QSE\_ID> UNIT: <ID> ALERT:  INVALID TEL HSL OR LSL  Note: <ID> 🡪 <SUBSTATION>-<UNIT#> |
| 8 | RLCXGTBD | Invalid or suspect resource status codes. \* | Telemetry | High | A QSE | QSE: <QSE\_ID> UNIT: <ID> ALERT:  INVALID TEL RST  Note: <ID> 🡪 <SUBSTATION>-<UNIT#> |
| 9 | BCRSMTCH | Add a SCADA calculation to sum the Mw telemetry of transmission lines moving power from a generating plant.  Compare this total to the sum of the telemetered Mw output of the Resources at the same plant.  Alarm the QSE if the amount exceeds a threshold. | Telemetry | High | A QSE | ALERT :<RES\_ID> - TEL MW= <VALUE> TRANS SUM= <VALUE>  Note:  <RES\_ID> 🡪 <SUBSTATION> <UNIT#> |
| 10 | RLCSUSMW | If the HDL = LDL = Actual Generation and Resource Status is not OFF then alarm the QSE operator to verify data telemetry used by RLC. \* | Telemetry | High | A QSE | QSE: <QSE\_ID> UNIT: <ID> ALERT:  SUSPECT TEL HDL=LDL=MW |
| 11 | RLCRSTMW | If Resource Status is OFF, and Resource output Mw is greater than 2 Mw, then alarm incorrect status.  If Resource Status is ON, but MW < 0.1 MW, then alarm incorrect status.  RLC will set RST to “N/A” and set HDL=LDL=MW Output – SDRAMP \* 5. | Telemetry | High | A QSE | QSE: <QSE\_ID> UNIT: <ID> ALERT:  INCONSIST RST AND MW  Note: <ID> 🡪 <SUBSTATION>-<UNIT#> |
| 12 | URGDASRT  URRASRT  UNSRSRT | The Ancillary Responsibilities (Reg-Up, Reg-Down, Responsive and Non-Spin) < 0  RLC will set the AS responsibility to 0. | Telemetry | High | A QSE | QSE:[qse], SUBSTATION:[substn], UNIT:[unitname],REGUP AS RESP < 0  QSE:[qse], SUBSTATION:[substn], UNIT:[unitname],REGDWN AS RESP < 0  QSE:[qse], SUBSTATION:[substn], UNIT:[unitname],RESP AS RESP < 0  QSE:[qse], SUBSTATION:[substn], UNIT:[unitname],NSPIN AS RESP < 0 |
| 13 | URRASRS  UNSRSRSS | The Ancillary Schedules (Responsive and Non-Spin) < 0  RLC will set the AS responsibility to 0. | Telemetry | High | A QSE | QSE:[qse], SUBSTATION:[substn], UNIT:[unitname],RESP AS SCHED < 0  QSE:[qse], SUBSTATION:[substn], UNIT:[unitname],NSPIN AS SCHED < 0 |
| 14 | BCVOLLUN | ERCOT shall notify each QSE representing the Split Generation Resource of any errors in telemetry detected by the State Estimator. | Telemetry | High | A QSE | QSE:[qse], SUBSTATION:[substn], UNIT:[unitname],SE SGU ERROR MW=[123] TSUM=[1234], |
| 15 | BCABRANCH | Once transmission line and transformer Dynamic Ratings are retrieved, ERCOT shall compare the actual flow and state estimated flow calculation of MVA to the effective Transmission Element limit and, if an out-of-limit condition exists, ERCOT shall produce an overload notification | Telemetry | High | A TSP | ALERT: <SUBSTATION : EQUIPMENT IDENTIFICATION> <value>  TEL OVERLOAD  OR  ALERT: <SUBSTATION : EQUIPMENT IDENTIFICATION> <value>  SE OVERLOAD |
| 16 | UREGUP  UREGDN | The Reg-Up/Reg-Down Participation Factors < 0.  RLC will set the Reg-Up/Reg-Down Participation Factors to 0. | Telemetry | High | QSE | QSE:[qse], SUBSTATION:[substn], UNIT:[unitname],REGUP PF < 0  QSE:[qse], SUBSTATION:[substn], UNIT:[unitname],REGDOWN PF < 0  QSE:[qse], SUBSTATION:[substn], UNIT:[unitname],HSL > THAN CURCAP |
| 17 | RLCHCAP | HSL is higher than the net dependable capability | Telemetry | High | QSE | QSE:[qse], SUBSTATION:[substn], UNIT:[unitname],HSL > THAN CURCAP |
| 18 | LFUPDT | ERCOT System, Weather Zone, and Load Zone Load forecasts for the next seven days, by hour, and a message on update indicating any changes to the forecasts by means of the Messaging System; | Market Message | Low | All MPs | ERCOT 7 DAY LF UPDATED <Add Alert Time> |
| 19 | EMS-LDF-UPDATE-VERIFICATION | Notify QSEs of posting and availability of report NP4-159: Load Distribution Factors. | Market Message | Low | All MPs | EMS-LDF-UPDATE-VERIFICATION completed on <YYYY-MM-DD HH24:MI:SS>. The LDF Report will be available by <YYYY-MM-DD HH24:MI:SS> |

Figure 145 - EMS System generated Alerts and Notifications

#### NMMS System-Generated Notices

The following notices are Network Model Management System (NMMS) system-generated and they will display on the MIS website’s Notices page.  Additionally these notices are sent out asynchronously to the Market Participants Listeners.

The following table identifies the description, notice type, priority, audience, and text of the notices.

| # | Description | Type | Priority | Audience | Text |
| --- | --- | --- | --- | --- | --- |
| 1 | Notifies the market participants that NOMCR and SAMR status changes are posted to MIS. | Notification | Med | TSP | Auto forwarded by a Rule:  Status change for [SAMR/NOMCR/Interim Update] - Number: '[SAMR/NOMCR ID]', Name: '[Short Description]' to [New Status]  Below are the NOMCR details:  Market Participant Account       User          Energize Date      Short Description    Long Description    [MPName]                           [Username]    [Energize Date]   [Short Description]   [Long Description]  Last Modified Date  [LastModifiedDate] |

Figure 146 - NMMS System generated Alerts and Notifications

#### Operator notices

The following custom notices are issued by various operator groups. These notices will also be displayed on the MIS website’s Notices page.

* Operations notices created by control room operators
* Operations notices created by day-ahead market (DAM) operators
* Planning notices created by the planning team
* CRR notices created by the CRR team
* Settlements notices generated by Client Services

The API notices will have the following verb/noun combination:

* verb=created
* noun=Alert

When a notice gets cancelled, a follow-up notice will be sent with “ERCOT is cancelling the following notice: . . .”  text prepended to original notice. Following verb/noun combination are used for cancel notice:

* verb=canceled
* noun=Alert

| # | Description | Type | Priority | Audience | Text |
| --- | --- | --- | --- | --- | --- |
| 1 | SCED Failure | Watch | High | All MPs | ERCOT has declared an Emergency due to the failure of the SCED process, starting at [hh:mm:z]. |
| 2 | DC Tie Outage | Operational Information | Low | All MPs(Public) | The <DcTie Name> DC Tie will be unavailable from [mmmddyyyy] through [mmmddyyyy] due to a planned outage. |
| 3 | EMS Site Failover | OCN | Low | All MPs(Public) | At [hh:mm:z]on [mmm dd yyyy],ERCOT will perform a site failover of its Energy and Market Management Systems from [hh:mm:z]through [hh:mm:z].During this time, market communications will be unavailable for about 15 minutes, and real-time communications will be unavailable for about 5 minutes. All systems should re-establish communications automatically. This event will be preceded by a hot-line call shorty before the start of the failover. |
| 4 | Capacity Insufficiency | OCN | Medium | All MPs | ERCOT is issuing an OCN for a projected reserve capacity shortage for hours ending [hh:mm:z]through [hh:mm:z].ERCOT is requesting all QSEs to update their COPs. |
| 5 | Physical Responsive Capability < 3000 MW | Advisory | Medium | All MPs(Public) | ERCOT is issuing an Advisory due to Physical Responsive Capability being below 3000 MW |
| 6 | Physical Responsive Capability < 2500 MW | Watch | Medium | All MPs(Public) | ERCOT is issuing a Watch due to Physical Responsive Capability being below 2500 MW |
| 7 | EEA Level 1 | Emergency Notice | High | All MPs(Public) | At [hh:mm:z] ERCOT declared EEA Level 1 of the Energy Emergency Alert (EEA). |
| 8 | EEA Level 2A | Emergency Notice | High | All MPs(Public) | At [hh:mm:z] ERCOT declared EEA Level 2A of the Energy Emergency Alert (EEA). |
| 9 | Loss of ERCOTs RTA and SE 15 Minutes | Advisory | Medium | All MPs(Public) | ERCOT's RTCA and State Estimator have not solved in the last 15 minutes. Please monitor your own service area and notify ERCOT if you exceed the normal rating of your transmission elements. Continue to monitor voltages in your area and notify ERCOT of any forced line operations in your area. |
| 10 | West-North Stability Limit | Operational Information | Low | All MPs(Public) | ERCOT is constraining for the West-North stability limit. |
| 11 | North-Houston Voltage Stability Limit | Operational Information | Low | All MPs(Public) | ERCOT is constraining for the North-Houston voltage stability limit. |
| 12 | EEA Level 2B | Emergency Notice | High | All MPs(Public) | At [hh:mm:z]ERCOT declared EEA Level 2B of the Energy Emergency Alert (EEA). |
| 13 | EECP Level 3 | Emergency Notice | High | All MPs(Public) | At [hh:mm:z]ERCOT declared EEA Level 3 of the Energy Emergency Alert (EEA). |
| 14 | EEA Level 3 to EEA Level 2B | Emergency Notice | High | All MPs(Public) | At [hh:mm:z]ERCOT moved from EEA Level 3 to EEA Level 2B |
| 15 | EEA Level 2B to EEA Level 2A | Emergency Notice | High | All MPs(Public) | At [hh:mm:z]ERCOT moved from EEA Level 2B to EEA Level 2A. |
| 16 | EECP Step 2 to Step 1 | Emergency Notice | High | All MPs(Public) | At [hh:mm:z]ERCOT moved from EEA Level 2A to EEA Level 1. |
| 17 | Change in GTC | OCN | High | All MPs | Change in GTC found. Please review. |
| 18 | Free form notices | Operational Information or  OCN or  Advisory or  Watch or  Emergency or  AS Postings or  Violations | High/Medium/Low | All MPs(Public) / All QSEs | Custom text – actual text entered by operator based on notice. |
| 19 | DAM Extension | Watch | High | All QSEs | ERCOT has extended the DAM for Operating Day [mmm dd yyyy].until [hh:mm:z],due to <reason text>.Please < interim requirements> |
| 20 | DAM Invalid | Watch | High | All QSEs | ERCOT has declared the DAM process invalid for Operating Day [mmm dd yyyy]due to <reason text>. Please <required schedules and trades during the Adjustment Period.> |
| 21 | AS Insufficiency | Watch | High | All QSEs | FR3061 DAM-ASINSUFF-NOTF Due to AS Insufficiency, DAM market for MM/DD/YYYY has reopened for AS offer resubmission. The DAM market will close today at HH:MM.  Note: This notice is not automatically sent and requires the operator to “push a button” to send it. |
| 22 | Electrical Bus Removal | Planning | High | All MPs(public) | The Electrical Bus [bus name]contained within [hub bus name]has been removed due to reflected changes in the Transmission Grid as submitted by [NOMCR owner]. and is effective [mmm dd yyyy].The Electrical Bus will be excluded from pricing calculations after removal |
| 23 | CRR Auction Bid Submissions | CRR | High | All MPs(public) | Bid submissions for the [Month or Year][monthly or annual] CRR auction will be accepted starting on [mmm dd yyyy].Details are posted on ERCOT.com |
| 24 | Settlements Delay | Settlements | High | All MPs(Public) | The following data will not be available on the scheduled date: (List all that are appropriate\*).The [RTM or DAM]for Operating Day [mmm dd yyyy]will be resettled. View details.  \* List  • Each Settlement Invoice for the DAM  • Payments for the DAM  • Each Late Fee Invoice for the DAM  • Payments for DAM Late Fee Invoices  • Settlement Invoice for the RTM  • Payments for the Real-Time Market  • Late Fee Invoice for the RTM  • Payments for RTM Late Fee Invoices  • RTM Uplift Invoice  • Payments for RTM Uplift Invoices  • Congestion Revenue Rights (CRR) Auction Invoic  • Payments for CRR Auction Invoices  • CRR Auction Revenue Distribution Invoice  • Payments for CRR Auction Revenue Distribution Invoices  • Each CRR Balancing Account Invoic  • Payments for CRR Balancing Account Invoice  • Settlement and Billing Disputes for each scheduled Settlement Statement. |
| 25 | Resettlement Notice | Settlements | High | All MPs(Public) | The <Market> Statement for Operating Day <Operating Day> will be resettled  (This notice is undergoing a change to include additional fields rather than the View Details link.) |
| 26 | Delay of RTM True-Up | Settlements | High | All MPs(Public) | The True-Up Statement for Operating Day [ ]has been delayed. This delay is due to [ ].ERCOT expects to issue the Real-Time Market True-Up Statement on [ ]. |
| 27 | RTM True-Up Settlement Notice | Settlements | High | All MPs(Public) | The True-Up Statement for Operating Day [mmm dd yyyy]will be posted on [mmm dd yyyy]. |

Example Operator notice XMLs:

<?xml version="1.0" encoding="UTF-8"?>

<ResponseMessage xmlns="http://www.ercot.com/schema/2007-06/nodal/ews/message" xmlns:ResponseMessage="http://www.ercot.com/schema/2007-06/nodal/ews/message" xmlns:ns2="http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

<Header>

<Verb>created</Verb>

<Noun>Alert</Noun>

<ReplayDetection>

<Nonce>1585254487</Nonce>

<Created>2010-08-11T11:23:00-05:00</Created>

</ReplayDetection>

<Revision>1.19K</Revision>

<Source>ERCOT</Source>

<UserID>183529049$tfrank</UserID>

<MessageID>Watch</MessageID>

</Header>

<Reply>

<ReplyCode>OK</ReplyCode>

<Timestamp>2010-08-11T11:23:00-05:00</Timestamp>

</Reply>

<Payload>

<Event xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:Event="http://www.ercot.com/schema/2007-06/nodal/ews">

<qse>ALLMPs</qse>

<ID>2010-08-11T11:23:00-05:00</ID>

<type>Watch</type>

<priority>High</priority>

<source>183529049$tfrank</source>

<issued>2010-08-11T11:23:00-05:00</issued>

<summary>ERCOT is issuing a Watch due to adjusted responsive reserve being below 2500 MW.</summary>

</Event>

<format>XML</format>

</Payload>

</ResponseMessage>

<?xml version="1.0" encoding="UTF-8"?>

<ResponseMessage xmlns="http://www.ercot.com/schema/2007-06/nodal/ews/message" xmlns:ResponseMessage="http://www.ercot.com/schema/2007-06/nodal/ews/message" xmlns:ns2="http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

<Header>

<Verb>created</Verb>

<Noun>Alert</Noun>

<ReplayDetection>

<Nonce>-1309492847</Nonce>

<Created>2010-08-11T11:23:00-05:00</Created>

</ReplayDetection>

<Revision>1.19K</Revision>

<Source>ERCOT</Source>

<UserID>183529049$tfrank</UserID>

<MessageID>Emergency Notice</MessageID>

</Header>

<Reply>

<ReplyCode>OK</ReplyCode>

<Timestamp>2010-08-11T11:23:00-05:00</Timestamp>

</Reply>

<Payload>

<Event xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:Event="http://www.ercot.com/schema/2007-06/nodal/ews">

<qse>ALLQSEs</qse>

<ID>2010-08-11T11:23:00-05:00</ID>

<type>Emergency Notice</type>

<priority>High</priority>

<source>183529049$tfrank</source>

<issued>2010-08-11T11:23:00-05:00</issued>

<summary>ERCOT has declared the DAM process invalid for Operating Day Apr 09 2011 due to xxxxxxxxxxxxxx..</summary>

</Event>

<format>XML</format>

</Payload>

</ResponseMessage>

### Offer and Bid Set Acceptance

The purpose of this notification is to indicate the acceptance of a BidSet to a QSE. This is necessary as BidSets are validated some time after the initial submission.

A more detailed description of this process is described in section 3. The following is an example payload where all bids have been set to a PENDING state.

<BidSet xmlns="http://www.ercot.com/schema/2007-05/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<tradingDate>2008-01-01</tradingDate>

<ASOffer>

<mRID>ACME.20080101.ASO.<Resource1>.Reg-Up</mRID>

<externalId>XID12345</externalId>

<status>PENDING</status>

</ASOffer>

<ASOffer>

<mRID>ACME.20080101.ASO.<Resource2>.Reg-Up</mRID>

<externalId>XID12346</externalId>

<status>PENDING</status>

</ASOffer>

</BidSet>

### Offer and Bid Set Errors

The purpose of this message is to indicate that one or more bids in a BidSet had errors. Those bids in a BidSet without errors are otherwise accepted. Error may also indicate the rejection of bids or offers for reasons such as the failure to receive credit approval.

Examples of errors are described in section 3.4. The following is an XML example where some bids were not valid and therefore rejected.

<BidSet xmlns="http://www.ercot.com/schema/2007-05/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<tradingDate>2008-01-01</tradingDate>

<ASOffer>

<mRID>ACME.20080101.ASO.<Resource1>.Reg-Up</mRID>

<externalId>XID12360</externalId>

<status>PENDING</status>

</ASOffer>

<ASOffer>

<mRID>ACME.20080101.ASO.<Resource2>.Reg-Up</mRID>

<externalId>XID12361</externalId>

<status>ERROR</status>

<error><severity>ERROR</severity><text>Invalid AS Type</text></error>

</ASOffer>

</BidSet>

### Confirmed and Unconfirmed Trades

A notification is issued periodically to a QSE where they have been identified as the counter party in a trade, but have not yet entered a matching energy, capacity, or ancillary services trade. If a matching trade is not entered by the close of the market, the trade is rejected.

The following response message structure will be used for created Unconfirmed trade notification:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | Created |
| Header/Noun | UnconfirmedTrades |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Current System Timestamp* |
| Payload | UnconfirmedTrades |

The following structure is used for conveying unconfirmed trades:



Figure 147 - UnconfirmedTrades Container

The details of the structures for ASTrades, CapacityTrades, and EnergyTrades are described in sections 3.3.5, 3.3.6, and 3.3.12, respectively. The following is an XML of an unconfirmed trade’s payload:

<UnconfirmedTrades>

<ASTrade>

<mRID>ACME.20080101.AST.<ASType>.<BuyerQSE>.<SellerQSE></mRID>

</ASTrade>

<CapacityTrade>

<mRID>ACME.20080101.CT.<BuyerQSE>.<SellerQSE></mRID>

</CapacityTrade>

<EnergyTrade>

<mRID>ACME.20080101.ET.<SettlementPoint>.<BuyerQSE>.<SellerQSE></mRID>

</EnergyTrade>

</UnconfirmedTrades>

The mRID used in the message is the mRID for the trade that was submitted by the counterparty. Similarly to unconfirmed trades, the following figure describes the container for confirmed trades.



Figure 148 - Confirmed Trades

The following response message structure will be used for Confirmed trade notification:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | Created |
| Header/Noun | ConfirmedTrades |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Current System Timestamp* |
| Payload | ConfirmedTrades |

The following is an XML of a confirmed trade’s payload:

<ConfirmedTrades>

<ASTrade>

<mRID>ACME.20080101.AST.<ASType>.<BuyerQSE>.<SellerQSE></mRID>

</ASTrade>

<CapacityTrade>

<mRID>ACME.20080101.CT.<BuyerQSE>.<SellerQSE></mRID>

</CapacityTrade>

<EnergyTrade>

<mRID>ACME.20080101.ET.<SettlementPoint>.<BuyerQSE>.<SellerQSE></mRID>

</EnergyTrade>

</ConfirmedTrades>

Confirmed and Unconfirmed Trades – cancel update

A separate notification is issued to a QSE when a Confirmed/Unconfirmed trade is canceled by counter party.

The following response message structure will be used for cancelled Unconfirmed trade notification:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | canceled |
| Header/Noun | UnconfirmedTrades |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Current System Timestamp* |
| Payload | UnconfirmedTrades |

The following structure is used for conveying unconfirmed trades:



Figure 149 - UnconfirmedTrades Container

The details of the structures for ASTrades, CapacityTrades, and EnergyTrades are described in sections 3.3.5, 3.3.6, and 3.3.12, respectively. The following is an XML of an unconfirmed trade’s payload:

<UnconfirmedTrades>

<ASTrade>

<mRID>ACME.20080101.AST.<ASType>.<BuyerQSE>.<SellerQSE></mRID>

</ASTrade>

<CapacityTrade>

<mRID>ACME.20080101.CT.<BuyerQSE>.<SellerQSE></mRID>

</CapacityTrade>

<EnergyTrade>

<mRID>ACME.20080101.ET.<SettlementPoint>.<BuyerQSE>.<SellerQSE></mRID>

</EnergyTrade>

</UnconfirmedTrades>

The mRID used in the message is the mRID for the trade that was submitted by the counterparty. Similarly to unconfirmed trades, the following figure describes the container for confirmed trades.



Figure 150 - Confirmed Trades (canceled)

The following response message structure will be used for Confirmed trade cancel notification:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | canceled |
| Header/Noun | ConfirmedTrades |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Current System Timestamp* |
| Payload | ConfirmedTrades |

The following is an XML of a confirmed trade’s payload:

<ConfirmedTrades>

<ASTrade>

<mRID>ACME.20080101.AST.<ASType>.<BuyerQSE>.<SellerQSE></mRID>

</ASTrade>

<CapacityTrade>

<mRID>ACME.20080101.CT.<BuyerQSE>.<SellerQSE></mRID>

</CapacityTrade>

<EnergyTrade>

<mRID>ACME.20080101.ET.<SettlementPoint>.<BuyerQSE>.<SellerQSE></mRID>

</EnergyTrade>

</ConfirmedTrades>

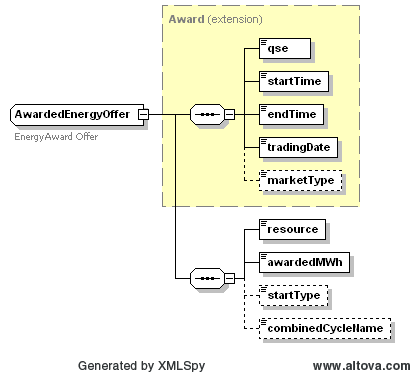
### Energy Offer Awards

The purpose of this message is to notify a QSE of DAM Energy Offer Awards.

The following response message structure will be used for Energy Offer Awards notification:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | created |
| Header/Noun | AwardedEnergyOffer |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Current System Timestamp* |
| Payload/AwardSet | *AwardedEnergyOffer* |

The payload structure of this message is shown by the following diagram:



The table found in sections 4.3.5 describes each AwardedEnergyOffer elements.

Figure 151 - AwardedEnergyOffer Structure

The following is an XML example:

<ns0:AwardedEnergyOffer>

<ns0:qse>LUMN</ns0:qse>

<ns0:startTime>2008-04-30T23:00:00-05:00</ns0:startTime>

<ns0:endTime>2008-05-01T00:00:00-05:00</ns0:endTime>

<ns0:tradingDate>2008-04-30</ns0:tradingDate>

<ns0:marketType>DAM</ns0: marketType >

<ns0:resource>Resource1</ns0:resource>

<awardedMWh>126.6</awardedMWh>

<startType>HOT</startType>

<combinedCycleName>CC1</combinedCycleName>

</ns0:AwardedEnergyOffer>

### DAM Energy-Only Offer Awards

The purpose of this message is to notify a QSE of DAM Energy-Only Offer Awards.

The following response message structure will be used for Energy Only Offer Awards notification:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | created |
| Header/Noun | AwardedEnergyOnlyOffer |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Current System Timestamp* |
| Payload/AwardSet | AwardedEnergyOnlyOffer |

The payload structure of this message is shown by the following diagram:

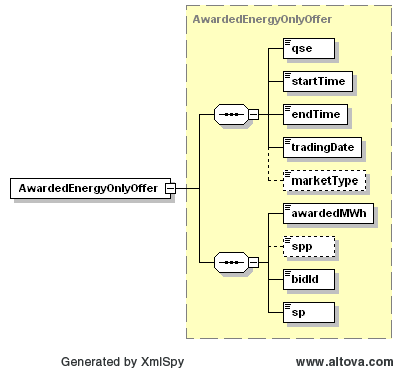


Figure 152 - AwardedEnergyOnlyOffer Structure

The table found in sections 4.3.6 describes each AwardedEnergyOnlyOffer elements.

The following is an XML example:

<ns0:AwardedEnergyOnlyOffer>

<ns0:qse>LUMN</ns0:qse>

<ns0:startTime>2008-04-30T00:00:00-05:00</ns0:startTime>

<ns0:endTime>2008-04-30T01:00:00-05:00</ns0:endTime>

<ns0:tradingDate>2008-04-30</ns0:tradingDate>

<ns0:marketType>DAM</ns0:marketType>

<ns0:awardedMWh>3</ns0:awardedMWh>

<ns0:bidId>456</ns0:bidId>

<ns0:sp>123</ns0:sp>

</ns0:AwardedEnergyOnlyOffer>

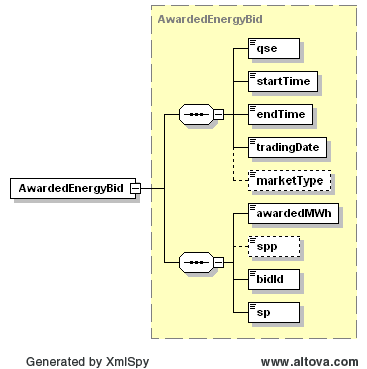
### DAM Energy Bid Award

The purpose of this message is to notify a QSE of DAM Energy Bid Awards.

The following response message structure will be used for Energy Bid Awards notification:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | created |
| Header/Noun | AwardedEnergyBid |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Current System Timestamp* |
| Payload/AwardSet | *AwardedEnergyBid* |

The payload structure of this message is shown by the following diagram:



The table found in sections 4.3.4 describes each AwardedEnergyBid elements.

Figure 153 - AwardedEnergyBid Structure

The following is an XML example:

<ns0:AwardedEnergyBid>

<ns0:qse>LUMN</ns0:qse>

<ns0:startTime>2008-04-30T00:00:00-05:00</ns0:startTime>

<ns0:endTime>2008-04-30T01:00:00-05:00</ns0:endTime>

<ns0:tradingDate>2008-04-30</ns0:tradingDate>

<ns0:marketType>DAM</ns0:marketType>

<ns0:awardedMWh>3</ns0:awardedMWh>

<ns0:bidId>1234</ns0:bidId>

<ns0:sp>1234</ns0:sp>

</ns0:AwardedEnergyBid>

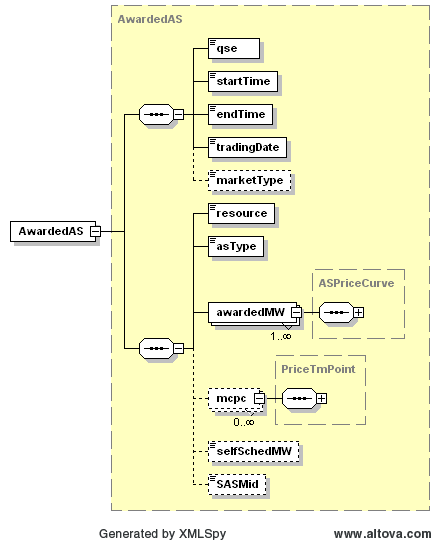
### Ancillary Service Awards

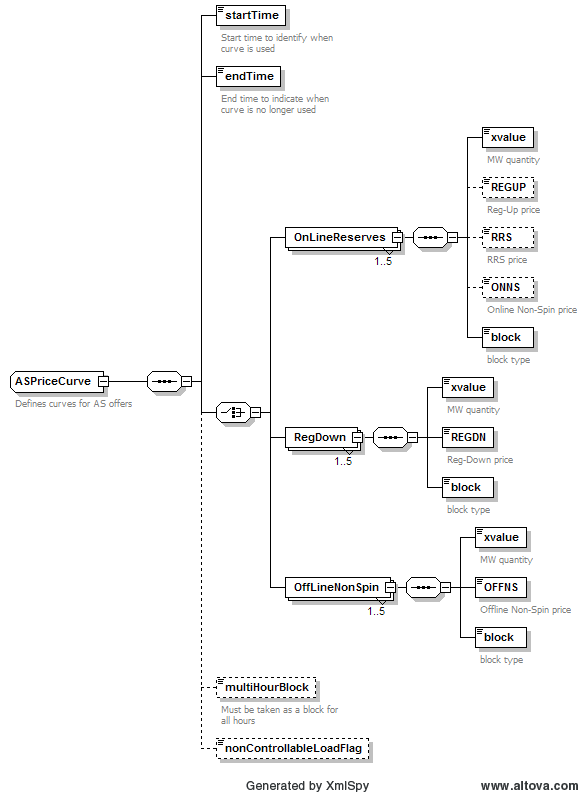
The purpose of this message is to notify a QSE of DAM or SASM Ancillary Services Awards.

The following response message structure will be used for Ancillary Service Awards notification:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | created |
| Header/Noun | AwardedAS |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Current System Timestamp* |
| Payload/AwardSet | *AwardedAS* |

The payload structure of this message is shown by the following diagram:





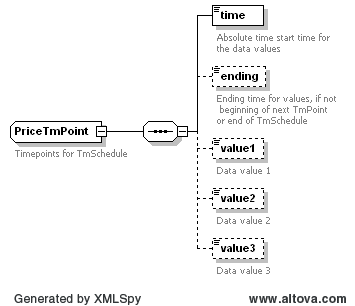


Figure 154 - AwardedAS Structure

The table found in sections 4.3.2 describes each AwardedAS elements.

The following is an XML example:

<ns0:AwardedAS>

<ns0:qse>LUMN</ns0:qse>

<ns0:startTime>2008-04-30T00:00:00-05:00</ns0:startTime>

<ns0:endTime>2008-04-30T01:00:00-05:00</ns0:endTime>

<ns0:tradingDate>2008-04-30</ns0:tradingDate>

<ns0:marketType>DAM</ns0: marketType >

<ns0:resource>Resource1</ns0:resource>

<ns0:asType>On-Non-Spin</ns0:asType>

<ns0:awardedMW>

<ns0:startTime>2008-04-30T00:00:00-05:00</ns0:startTime>

<ns0:endTime>2008-04-30T01:00:00-05:00</ns0:endTime>

<ns0:OnLineReserves>

<ns0:xvalue>0</ns0:xvalue>

<ns0:block>1</ns0:block>

</ns0:OnLineReserves>

<ns0:OnLineReserves>

<ns0:xvalue>0</ns0:xvalue>

<ns0:block>2</ns0:block>

</ns0:OnLineReserves>

<ns0:OnLineReserves>

<ns0:xvalue>0</ns0:xvalue>

<ns0:block>3</ns0:block>

</ns0:OnLineReserves>

<ns0:OnLineReserves>

<ns0:xvalue>0</ns0:xvalue>

<ns0:block>4</ns0:block>

</ns0:OnLineReserves>

<ns0:OnLineReserves>

<ns0:xvalue>0</ns0:xvalue>

<ns0:block>5</ns0:block>

</ns0:OnLineReserves>

</ns0:awardedMW>

<ns0:mcpc>

<time>2008-04-30T00:00:00-05:00</time>

<ending >2008-04-30T01:00:00-05:00></ending>

<value1>1234</value1>

</ns0:mcpc>

</ns0:AwardedAS>

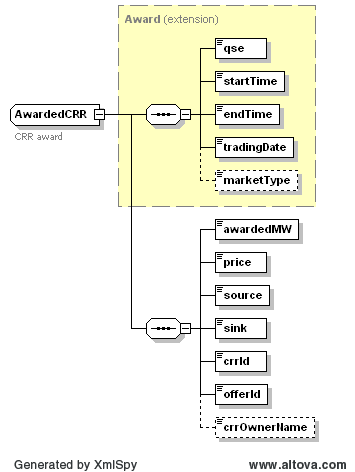
### CRR Awards

The purpose of this message is to notify a QSE of DAM CRR Awards.

The following response message structure will be used for CRR Awards notification:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | created |
| Header/Noun | AwardedCRR |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Current System Timestamp* |
| Payload/AwardSet | *AwardedCRR* |

The payload structure of this message is shown by the following diagram:



The table found in sections 4.3.3 describes each AwardedCRR elements.

Figure 155 - AwardedCRR Structure

The following is an XML example:

<ns0:AwardedCRR>

<ns0:qse>LUMN</ns0:qse>

<ns0:startTime>2008-04-30T00:00:00-05:00</ns0:startTime>

<ns0:endTime>2008-04-30T01:00:00-05:00</ns0:endTime>

<ns0:tradingDate>2008-04-30</ns0:tradingDate>

<ns0:marketType>DAM</ns0:marketType>

<ns0:awardedMW>3</ns0:awardedMW>

<ns0:price>0</ns0:price>

<ns0:source>BBSES\_UNIT1</ns0:source>

<ns0:sink>BBSES\_UNIT2</ns0:sink>

<ns0:crrId>12345</ns0:crrId>

<ns0:offerId>1234</ns0:offerId>

</ns0:AwardedCRR>

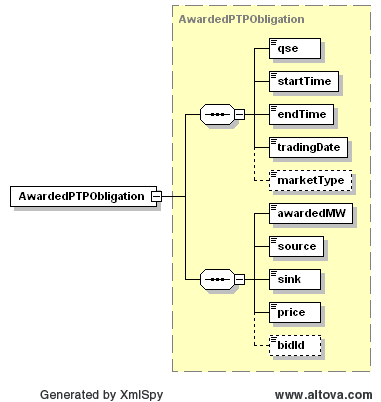
### PTP Obligation Awards

The purpose of this message is to notify a QSE of DAM PTP Obligation Awards.

The following response message structure will be used for PTP Obligation Awards notification:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | created |
| Header/Noun | AwardedPTPObligation |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Current System Timestamp* |
| Payload/AwardSet | AwardedPTPObligation |

The structure of this message is shown by the following diagram:



The table found in sections 4.3.7 describes each AwardedPTPObligation elements.

Figure 156 - AwardedPTPObligation Structure

The following is an XML example:

<ns0:AwardedPTPObligation>

<ns0:qse>LUMN</ns0:qse>

<ns0:startTime>2008-04-30T00:00:00-05:00</ns0:startTime>

<ns0:endTime>2008-04-30T01:00:00-05:00</ns0:endTime>

<ns0:tradingDate>2008-04-30</ns0:tradingDate>

<ns0:marketType>DAM</ns0: marketType >

<ns0:awardedMW>0</ns0:awardedMW>

<ns0:source>BBSES\_UNIT1</ns0:source>

<ns0:sink>LZ\_NORTH</ns0:sink>

<ns0:price>0</ns0:price>

<ns0:bidId>01</ns0:bidId>

</ns0:AwardedPTPObligation>

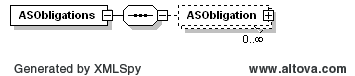
### Ancillary Service Obligations

The purpose of this notification message is to notify a QSE of DAM or SASM Ancillary Service Obligations.

The following response message structure will be used for Ancillary Service Obligations notification:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | created |
| Header/Noun | ASObligations |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Current System Timestamp* |
| Payload/ | ASObligations |

The structure of this message is shown by the following diagram:



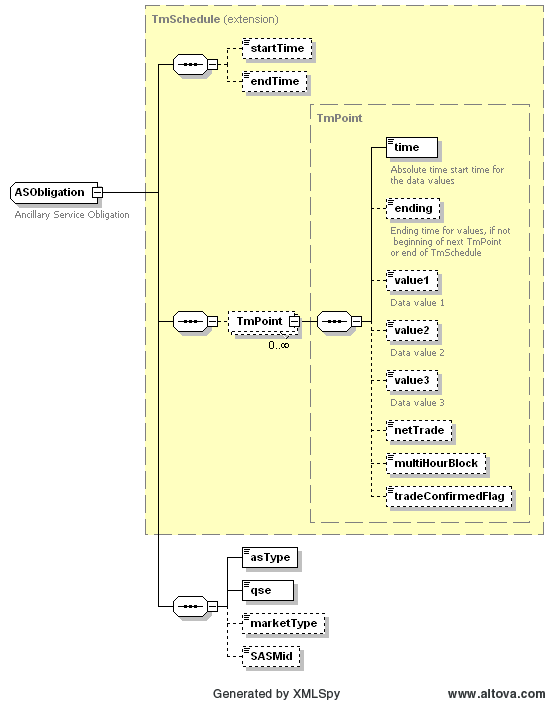


Figure 157 – Ancillary Service Obligation Structure

The following is an XML example:

<ASObligations>

<ASObligation>

<startTime>2008-01-01T00:00:00-06:00</startTime>

<endTime>2008-01-01T00:00:00-06:00</endTime>

<TmPoint>

<time>2008-01-01T00:00:00-06:00</time>

<ending>2008-01-01T00:00:00-06:00</ending>

<value1>3.1</value1>

</TmPoint>

<asType>Reg-Up</asType>

<qse>string</qse>

<marketType>SASM</marketType >

<SASMid>20091108122027</SASMid>

</ASObligation>

</ASObligations>

### Outage Notifications

The purpose of this notification is to notify market participants of outage state changes.

The following response message structure will be used for Outage State change notification:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | changed |
| Header/Noun | OutageSet |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Current System Timestamp* |
| Payload/ | OutageStateChange |

The following high-level payload structure is used to convey outage notifications.

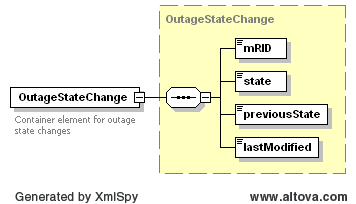


Figure 158 – Outage State Change Structure

The following is an XML example:

<OutageStateChange>

<mRID>TESTQSE.OTG.M1.Resource.XYZ121548</mRID>

<state>Cancl</state>

<previousState>Recvd</previousState>

<lastModified>2010-03-15</lastModified>

</OutageStateChange>

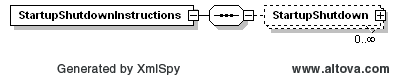
### Startup/Shutdown Instructions

The purpose of this notification is to notify market participants of startup and shutdown instructions.

The following response message structure will be used for StartupShutdown Instructions notification:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | created |
| Header/Noun | StartupShutdownInstructions |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Current System Timestamp* |
| Payload/ | StartupShutdownInstructions |

The following payload structure is used to convey these notifications.



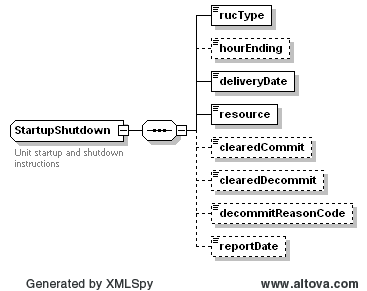


Figure 159 - StartupShutdownInstructions Structure

The following is an XML example:

<StartupShutdownInstructions xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<StartupShutdown>

<rucType>DRUC</rucType>

<hourEnding>6</hourEnding>

<deliveryDate>2007-07-25</deliveryDate>

<resource>Resource1</resource>

<clearedCommit>true</clearedCommit>

<clearedDecommit>false</clearedDecommit>

<decommitReasonCode>50 characters of free form text.</decommitReasonCode>

<reportDate>2007-07-25T11:00:00-06:00</reportDate>

</StartupShutdown>

</StartupShutdownInstructions>

Note : On long day(DST change day), hourEnding will be represented as 2\* for repeating hour.

### Wind Generation Forecast

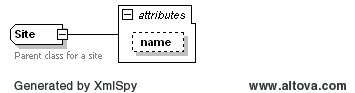
The purpose of this interface is to provide a notification for Wind Generation Resource Production Potential (WGRPP).

The purpose of Interface is to provide forecasts of Renewable Production Potential (RPP) for Wind Powered Generation Resources (WGR). ERCOT will provide forecasts for each WGR to the QSEs representing WGRs. The QSEs shall use the ERCOT provided forecasts for WGRs throughout the Day-Ahead and Operating Day for applicable markets and RUCs. Additionally the purpose is to allow WGRs to submit the LTWPF for their Resources to ERCOT and their respective QSE.

The following response message structure will be used for forecast notification:

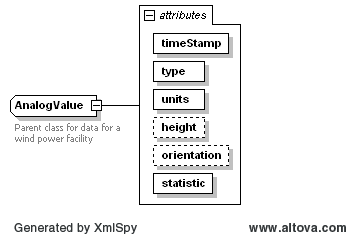
|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | create |
| Header/Noun | WindForecastData |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Current System Timestamp* |
| Payload/ | ForecastPayload |

The following diagram defines the structure of a site about which information must be exchanged by the ERCOT and Market Participants Notification listener using the Site tag:

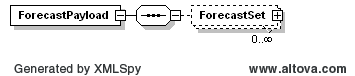


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *REQ* | *Data type* | *Description* | *Values* |
| Site | Y | String | Unique site ID | Prescribed at registration |
| *Attribute* | *REQ* | *Data type* | *Description* | *Values* |
| name | N | String | Verbose site ID | Prescribed at registration |

The following diagram defines the structure of Analog Values which must be exchanged by the ERCOT and Market Participants Notification Listener applications using the AnalogValue tag:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *REQ* | *Data type* | *Description* | *Values* |
| Analog Value | Y | AnalogValue | Value of data | Valid floating point value |
| *Attribute* | *REQ* | *Data type* | *Description* | *Values* |
| timeStamp | Y | dateTime | Time of data | Valid dateTime format |
| Value | Y | Float | Value of data | Valid floating point value |
| units | Y | String | Units of data | Enumeration (MW, m/s, Celsius) |
| height | N | Float | Height above ground level of data | Valid floating point value |
| orientation | N | Float | Directional orientation from true north of sensor | Valid floating point value between 0.0 and 359.99 inclusive. |
| statistic | Y | String | Statistic used to create data | Enumeration (AVG, MIN, MAX, SDV) |



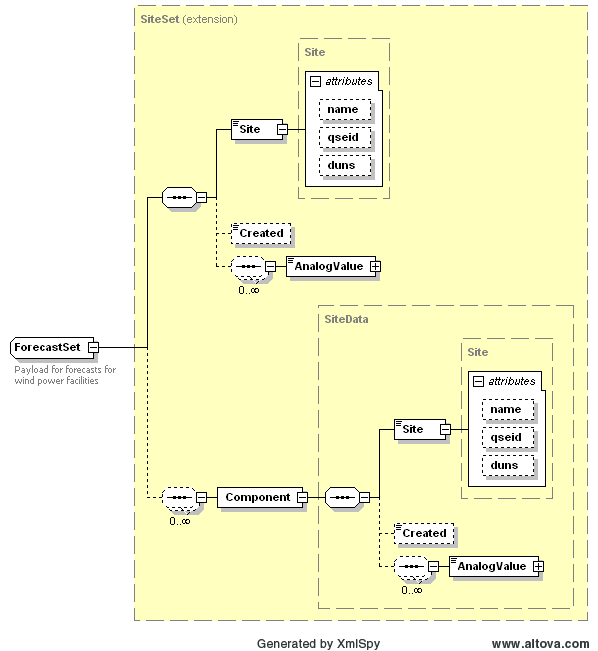


Figure 160 - Wind Generation Resource Production Potential Structure

Please note that the “Component” element is used only by the Wind Forecasting vendor to send the forecast information to ERCOT for all QSEs and it is not used to send the QSE specific forecast information to their listeners (see example below).

The following is an XML example for Forecast payload:

<ForecastPayload>

<ns1:ForecastSet>

<ns1:Site name="SITE1" qseid="TESTQSE1" duns="0012345676901">XXX</ns1:Site>

<ns1:Created>2010-04-27T15:14:01-05:00</ns1:Created>

<ns1:AnalogValue timeStamp="2010-04-27T16:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">4.4</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-27T16:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">0</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-27T17:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">5.7</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-27T17:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">0</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-27T18:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">5.5</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-27T18:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">0</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-27T19:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">6.6</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-27T19:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">0</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-27T20:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">7.7</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-27T20:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">0.3</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-27T21:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">15</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-27T21:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">7.6</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-27T22:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">23.1</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-27T22:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">15.7</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-27T23:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">23.3</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-27T23:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">16</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T00:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">25.9</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T00:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">18.5</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T01:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">31.4</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T01:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">24</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T02:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">34.7</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T02:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">27.3</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T03:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">39.4</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T03:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">32</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T04:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">42.3</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T04:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">34.9</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T05:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">46.4</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T05:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">39</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T06:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">44.3</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T06:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">36.9</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T07:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">45.6</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T07:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">38.2</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T08:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">44.7</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T08:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">37.3</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T09:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">37</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T09:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">29.6</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T10:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">34.4</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T10:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">27</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T11:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">31.2</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T11:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">23.8</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T12:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">28.4</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T12:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">21</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T13:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">27.1</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T13:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">19.7</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T14:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">26.7</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T14:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">19.3</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T15:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">27</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T15:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">19.6</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T16:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">27.6</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T16:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">20.2</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T17:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">28.6</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T17:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">21.3</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T18:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">29.4</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T18:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">22</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T19:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">32.1</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T19:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">24.7</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T20:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">37.7</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T20:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">30.3</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T21:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">41.2</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T21:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">33.8</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T22:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">44.1</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T22:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">36.7</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T23:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">48</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-28T23:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">40.6</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T00:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">51.4</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T00:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">44.1</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T01:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">55.5</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T01:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">48.1</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T02:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">61</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T02:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">53.6</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T03:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">64.6</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T03:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">57.2</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T04:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">63.4</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T04:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">56</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T05:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">61.1</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T05:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">53.7</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T06:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">60.5</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T06:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">53.1</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T07:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">59.7</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T07:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">52.3</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T08:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">57.6</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T08:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">50.2</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T09:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">55.4</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T09:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">48</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T10:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">47.7</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T10:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">40.4</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T11:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">41.8</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T11:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">34.4</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T12:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">38.9</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T12:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">31.5</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T13:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">37.5</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T13:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">30.1</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T14:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">42.9</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T14:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">35.5</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T15:00:00-05:00" type="STWPF" units="MW" statistic="MEAN">47.1</ns1:AnalogValue>

<ns1:AnalogValue timeStamp="2010-04-29T15:00:00-05:00" type="WGRPP" units="MW" statistic="MEAN">39.7</ns1:AnalogValue>

</ns1:ForecastSet><ForecastPayload>

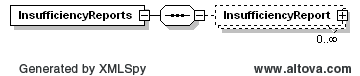
### DAM Ancillary Service Offer Insufficiency Report

Upon DAM market closure and before the start of the MMS optimization process, if there are not enough AS offers, the DAM AS Offer Insufficiency Report will be used to notify the market participants of available and insufficient AS Offer quantities.

The following response message structure will be used for DAM AS Offer Insufficiency Report notification:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | created |
| Header/Noun | InsufficiencyReport |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Current System Timestamp* |
| Payload/ | InsufficiencyReports |

The following payload structure is used to convey these notifications.



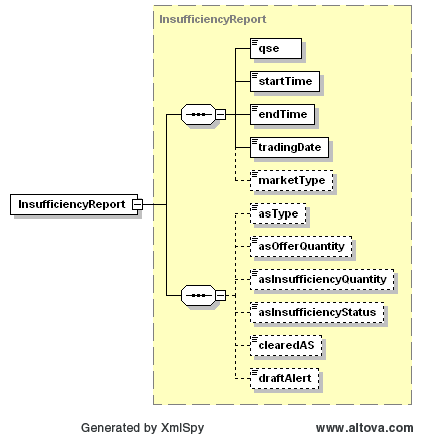


Figure 161 – Insufficiency Report Structure

The following is an XML example:

<InsufficiencyReports xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotAwards.xsd" xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<InsufficiencyReport>

<qse>ERCOT</qse>

<startTime>2008-01-01T10:00:00-06:00</startTime>

<endTime>2008-01-01T12:00:00-06:00</endTime>

<tradingDate>2008-01-01</tradingDate>

<marketType>DAM</marketType>

<asType>Reg-Up</asType>

<asOfferQuantity>3.1</asOfferQuantity>

<asInsufficiencyQuantity>3.1</asInsufficiencyQuantity>

<asInsufficiencyStatus>3.1</asInsufficiencyStatus>

<clearedAS>String</clearedAS>

<draftAlert>String</draftAlert>

<InsufficiencyReport></InsufficiencyReports>

### End of Adjustment Period Results

The purpose of this notification message is to notify End of Adjustment period results to a QSE. The result set will hold the Output Schedule Validation results (errors and warnings) for the specified trading date and the end hour.

The following response message structure will be used for End of Adjustment period results notification:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | created |
| Header/Noun | EndAdjPeriod |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Current System Timestamp* |
| Payload/ | BidSet/OutputSchedule |

The payload structure is described by the following diagram:



Figure 162 – End of Adjustment Period results: BidSet/OutputSchedule

The following is an XML example for End of Adjustment Period Results:

<BidSet xmlns="http://www.ercot.com/schema/2007-05/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<tradingDate>2008-01-01</tradingDate>

<OutputSchedule>

<mRID>ACME.20080101.OS.Resource1</mRID>

<status>ACCEPTED</status>

<error>

<ns1:severity>WARNING</ns1:severity>

<text>Five-minute intervals beginning 21:10 through interval 22:0 conflict with

existing Energy Offer Curve</text>

</error>

<error>

<ns1:severity>INFORMATIVE</ns1:severity>

<text>Successfully validated the ERCOT Output Schedule.</text>

</error>

<error>

<ns1:severity>WARNING</ns1:severity>

<text>The Output Schedule exists for hour ending 22 where resource status is OFF </text>

</error>

</OutputSchedule>

</BidSet>

### Two Hour Warning Results

The purpose of this notification message is to notify the two hour warning results to a QSE. The result set will hold the Output Schedule Validation results (errors and warnings) for the specified trading date and the end hour.

The following response message structure will be used for Two Hour Warning results notification:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | created |
| Header/Noun | TwoHrNotif |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Current System Timestamp* |
| Payload/ | BidSet/OutputSchedule |

The payload structure is described by the following diagram:



The following is an XML example for Two Hour Warning results:

<BidSet xmlns="http://www.ercot.com/schema/2007-05/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<tradingDate>2008-01-01</tradingDate>

<OutputSchedule>

<mRID>ACME.20080101.OS.Resource1</mRID>

<status>ACCEPTED</status>

<error>

<ns1:severity>WARNING</ns1:severity>

<text>Five-minute intervals beginning 21:10 through interval 22:0 conflict with

existing Energy Offer Curve</text>

</error>

<error>

<ns1:severity>INFORMATIVE</ns1:severity>

<text>Successfully validated the ERCOT Output Schedule.</text>

</error>

<error>

<ns1:severity>WARNING</ns1:severity>

<text>The Output Schedule exists for hour ending 22 where resource status is OFF </text>

</error>

</OutputSchedule>

</BidSet>

### DAM Phase II Validation Results

The purpose of this notification message is to notify DAM Phase II Validation results to a QSE.

The payload will hold the Bid Validation results (bid cancellations only) for the specified trading date.

The following response message structure will be used for Phase II validation results notification:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | canceled |
| Header/Noun | P2ValidationSet |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Current System Timestamp* |
| Payload/ | BidSet/<BidType>  <BidType> : one of the valid bid type like COP etc |

The payload structure is described by the following diagram:

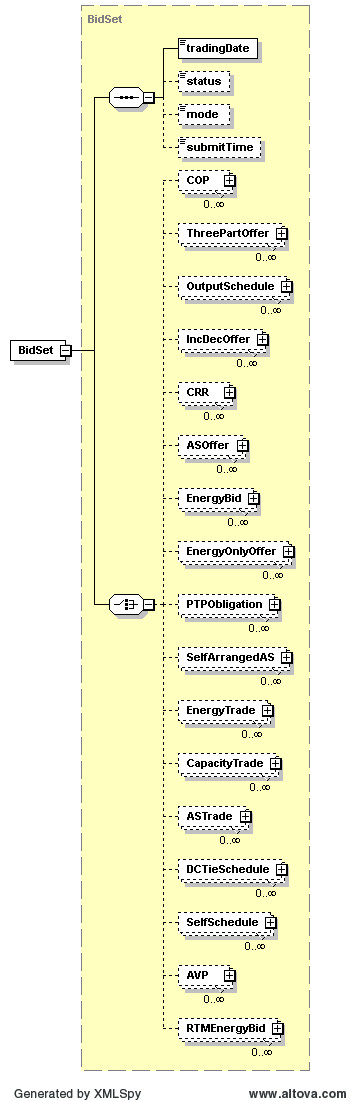


Figure 163 – DAM Phase II Validation Results: BidSet/[Bid Type]

The following is an XML example for DAM Phase II validation:

<BidSet xmlns="http://www.ercot.com/schema/2007-05/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

            <tradingDate>2008-02-19</tradingDate>

            <ThreePartOffer>

                        <mRID>ACME.20080101.TPO.DG\_BIOE\_2UNITS</mRID>

                        <status>CANCELED</status>

                                    <error>

<severity>ERROR</severity>

                                                <text>Validation of the Energy Three Part Offer failed.</text>

</error>

                                    <error>

<severity>ERROR</severity>

                                                <text>The data required for credit exposure calculation cannot be found</text>

</error>

            </ThreePartOffer>

</BidSet>

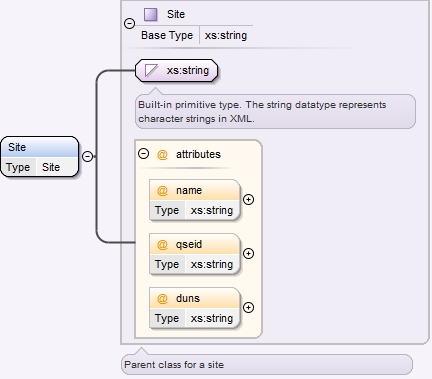
### Solar Generation Forecast

The purpose of this interface is to provide a notification for Photovoltaic Generation Resource Production Potential (PVGRPP), which is a forecast of Renewable Production Potential (RPP) for Photovoltaic Generation Resources (PVGR). ERCOT will provide forecasts for each PVGR to the QSEs representing PVGRs. The QSEs shall use the ERCOT provided forecasts for PVGRs throughout the Day-Ahead and Operating Day for applicable markets and RUCs.

The following response message structure will be used for forecast notification:

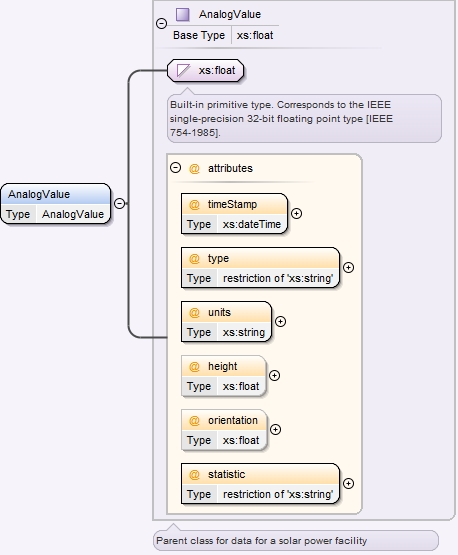
|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | create |
| Header/Noun | SolarForecastData |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Current System Timestamp* |
| Payload/ | ForecastSolarPayload |

The following diagram defines the structure of a site about which information must be exchanged by the ERCOT and Market Participants Notification listener using the Site tag:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *REQ* | *Data type* | *Description* | *Values* |
| Site | Y | String | Unique site ID | Prescribed at registration |
| *Attribute* | *REQ* | *Data type* | *Description* | *Values* |
| name | N | String | Verbose site ID | Prescribed at registration |
| qseid | N | String | QSE short name | Prescribed at registration |
| duns | N | String | QSE DUNS | DUNS Number |

The following diagram defines the structure of Analog Values which must be exchanged by the ERCOT and Market Participants Notification Listener applications using the AnalogValue tag:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *REQ* | *Data type* | *Description* | *Values* |
| Analog Value | Y | AnalogValue | Value of data | Valid floating point value |
| *Attribute* | *REQ* | *Data type* | *Description* | *Values* |
| timeStamp | Y | dateTime | Time of data | Valid dateTime format |
| Type | Y | String | Value of data | Enumeration (STPPF, PVGRPP, TE, PR, WS, WD, PWR, BPAVG) |
| Units | Y | String | Units of data | Enumeration (MW, m/s, Celsius) |
| Height | N | Float | Height above ground level of data | Not Applicable |
| orientation | N | Float | Directional orientation from true north of sensor | Not Applicable |
| Statistic | Y | String | Statistic used to create data | Enumeration (SAMPLE,MEAN,SD,ME,MAE,RMS) |

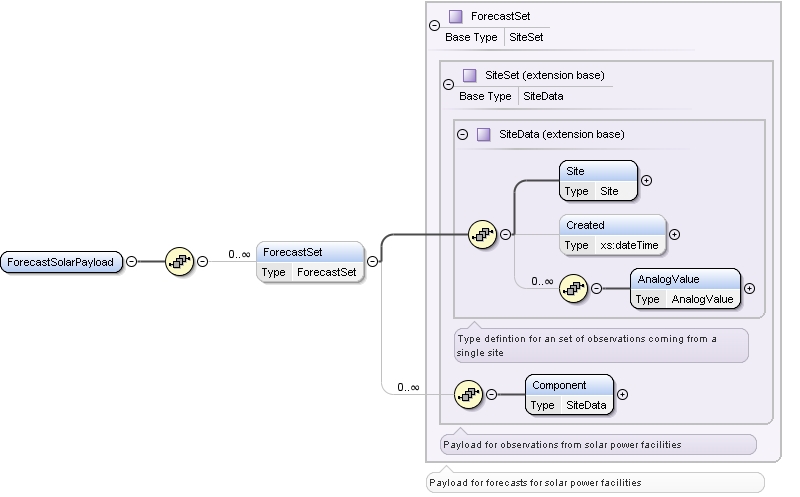


Figure 164 – Photovoltaic Generation Resource Production Potential Structure

The following is an abbreviated XML example for Solar Forecast payload:

<ns1:ForecastSolarPayload xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:ForecastSet>

                                <ns1:Site name="ABC\_DEF1" qseid="QSE1" duns="1234567890000">SITE1</ns1:Site>

                                <ns1:Created>2016-01-20T17:10:01-06:00</ns1:Created>

                                <ns1:AnalogValue timeStamp="2016-01-20T17:10:01-06:00" type="STPPF" units="MW" statistic="MEAN">4.4</ns1:AnalogValue>

                                <ns1:AnalogValue timeStamp="2016-01-20T17:10:01-06:00" type="PVGRPP" units="MW" statistic="MEAN">4.4</ns1:AnalogValue>

                                <ns1:AnalogValue timeStamp="2016-01-20T17:10:01-06:00" type="STPPF" units="MW" statistic="MEAN">0</ns1:AnalogValue>

                                <ns1:AnalogValue timeStamp="2016-01-20T17:10:01-06:00" type="PVGRPP" units="MW" statistic="MEAN">0</ns1:AnalogValue>

                                <ns1:AnalogValue timeStamp="2016-01-20T17:10:01-06:00" type="STPPF" units="MW" statistic="MEAN">0</ns1:AnalogValue>

                                <ns1:AnalogValue timeStamp="2016-01-20T17:10:01-06:00" type="PVGRPP" units="MW" statistic="MEAN">0</ns1:AnalogValue>

                                <ns1:AnalogValue timeStamp="2016-01-20T17:10:01-06:00" type="STPPF" units="MW" statistic="MEAN">0</ns1:AnalogValue>

                                <ns1:AnalogValue timeStamp="2016-01-20T17:10:01-06:00" type="PVGRPP" units="MW" statistic="MEAN">0</ns1:AnalogValue>

                                <ns1:AnalogValue timeStamp="2016-01-20T17:10:01-06:00" type="STPPF" units="MW" statistic="MEAN">1.7</ns1:AnalogValue>

                                <ns1:AnalogValue timeStamp="2016-01-20T17:10:01-06:00" type="PVGRPP" units="MW" statistic="MEAN">0</ns1:AnalogValue>

</ns1:ForecastSolarPayload>

# Outage Scheduling Interfaces

The purpose of this section is to describe interfaces related to the creation, update, query, and cancellation of outages. These interfaces would be used by QSEs and TPs, as needed to support integration with their system.

## Interfaces Provided

Interfaces are provided for the creation, update, query, and cancellation of outages by QSEs and TPs. The interfaces use a basic request/reply interaction pattern. The payload used for these interfaces is an ‘OutageSet’, which is a container that may contain zero, one or more ‘Outage’ objects.

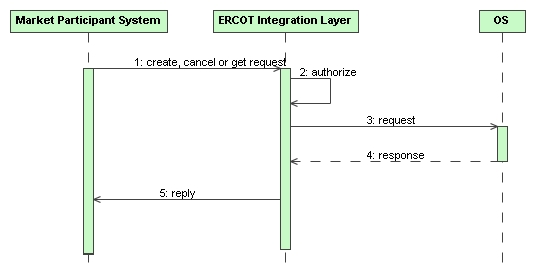


Figure 165 - Outage Scheduling Interface Sequence Diagram

Outages can be submitted by TP for transmission elements or by QSEs for resources.

There are also notifications sent to MPs when outages are created or changed. These are described in section 5.y.on elements, or by QSEs for resources. be returned.ustomers.t participant for a given tr

## Message Specifications

### Outage Creation

The request message for outage creation or update would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | create |
| Header/Noun | OutageSet |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Payload | *OutageSet* |

Figure 166 - Outage Creation Request

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | OutageSet |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload | *OutageSet* |

Figure 167 - Outage Creation Response

An Outage Scheduler mRID is constructed and returned in the payload of the response (included in ResourceOutage or Equipment Outage) as part of the OutageSet structure in the  following format

*mRID: <QSEID>.OTG.<outageType>.<outageCategory>.<outageIdent>*

Note: OutageIdent is assigned by Outage Scheduler Application.

The structure of the OutageSet payload is described by the following diagram. This leverages the structure of an Outage defined later in this section:

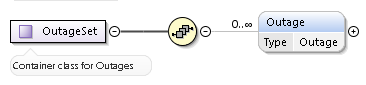


Figure 168 - OutageSet Structure

The high-level structure of an Outage record is described by the following diagram:

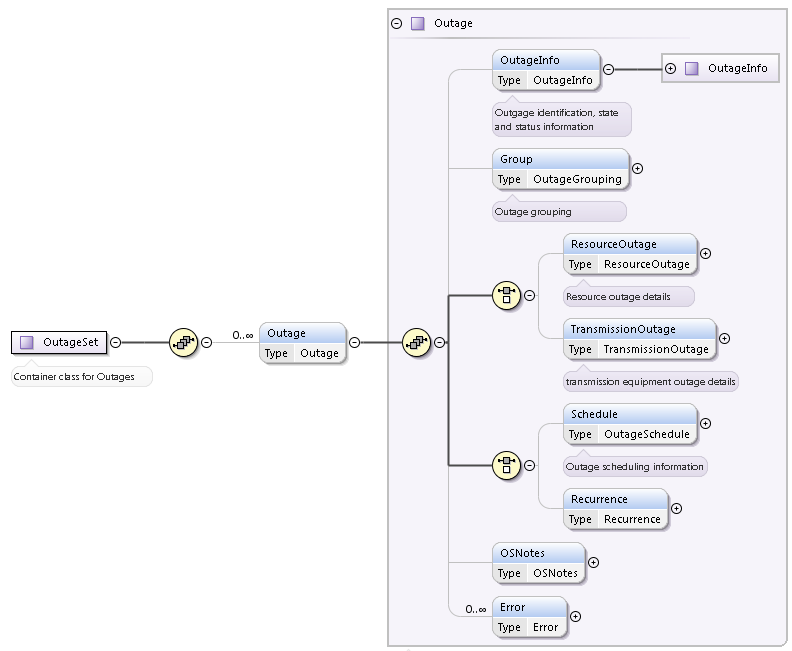


Figure 169 - High-level Outage Structure

|  |  |
| --- | --- |
|  |  |

Figure 170 - Outage Information

Within the OutageInfo substructure, the outage type, and state, status, and nature of work are identified.

Valid outage type codes include:

* FR  - Forced Outage
* M1 - Maintenance Level 1
* M2 - Maintenance Level 2
* M3 - Maintenance Level 3
* OP - Planned Outage Non-Reliability with Opportunity
* PL - Planned Outage Reliability
* RS - Forced Outage with Remedial Switching Action
* SM -  Simple Transmission Outage
* UE - Forced Outage with Unavoidable Extension
* FE - Forced Outage with Forced Extension
* RO - Rescheduled Outage

The valid state codes include:

* ENYS : Entered Not Yet Submitted
* Recvd: Received at ERCOT
* Study : Study
* Rejct : Rejected
* Accpt: Accepted
* Apprv: Approved
* Active: Active
* AExt: Active Extension
* PotOp: Potential Opportunity
* Withd: Withdrawn
* Cancl : Canceled
* PComp: Pending Completion
* ExpOp: Expired Opportunity
* CompEd: Completed - Editable
* CompNE: Completed - Not Editable

The valid status codes include:

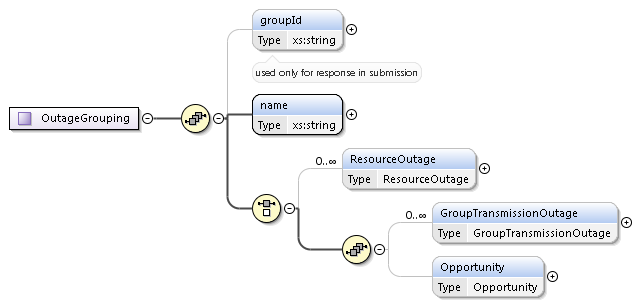
* ENYS: Entered Not Yet Submitted
* RatE: Received at ERCOT
* Accpt: Accepted
* Apprv: Approved
* Rejct: Rejected
* Study: Selected for Study
* Withd: Withdrawn
* ExpOp: ExpiredOpportunity
* Cancl: Canceled

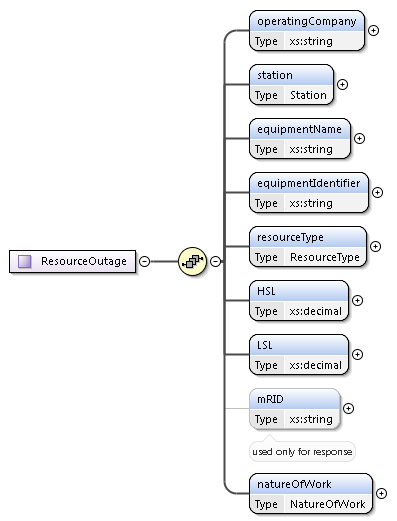
The valid Nature of Work codes for Resource Outages include:

* BS : Black Start Test
* BO : Boroscope
* CI : Combustion Inspection and Repair
* CW : Cooling Water Problems/Repairs
* EX : Exhaust Problems/Repairs
* FW : Feedwater Problems/Repairs
* FL : Fuel Limitation/Lack of Fuel Availability
* FP : Fuel Problems/Repairs
* GW : Gas Company Pipeline Work
* IP : Inlet Air Problems/Repairs
* LS : Lack of Steam Load Availability
* MO : Mothballed
* NE : New Equipment Energization
* OT : Other
* OV : Overhaul
* OE : Retirement of Old Equipment
* SE : Seasonal Resource
* TW : Steam Turbine Limited/Unavailable Due to CT/GT Work
* TL : Tube Leak
* VR : Turbine/Generator Vibration/Repair
* UN : Unknown

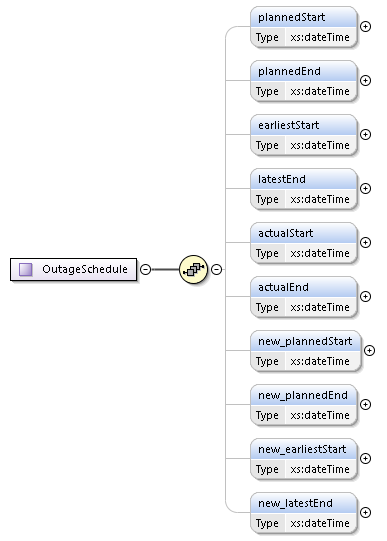
The valid Nature of Work codes for Transmission Outages include:

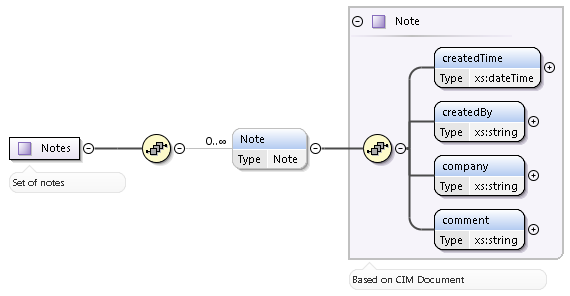
* AO : Associated Equipment Outage
* BM : Breaker Maintenance
* BR : Breaker Repair/Replace
* LM : Line Maintenance
* LR : Line Rebuild/Upgrade
* NE : New Equipment Energization
* OT : Other
* RW : Relay Work - See Notes
* HS : Repair Hot Spot
* DR : Repair/Replace Damaged Equipment
* RE : Replace Equipment
* OE : Retirement of Old Equipment
* SM : Switch Maintenance
* SR : Switch Repair/Replace
* TC : Telcom Work
* PR : Third Party Request
* TM : Transformer Maintenance
* TR : Transformer Repair/Replace
* TT : Transformer Testing
* UN : Unknown





|  |  |
| --- | --- |
|  |  |





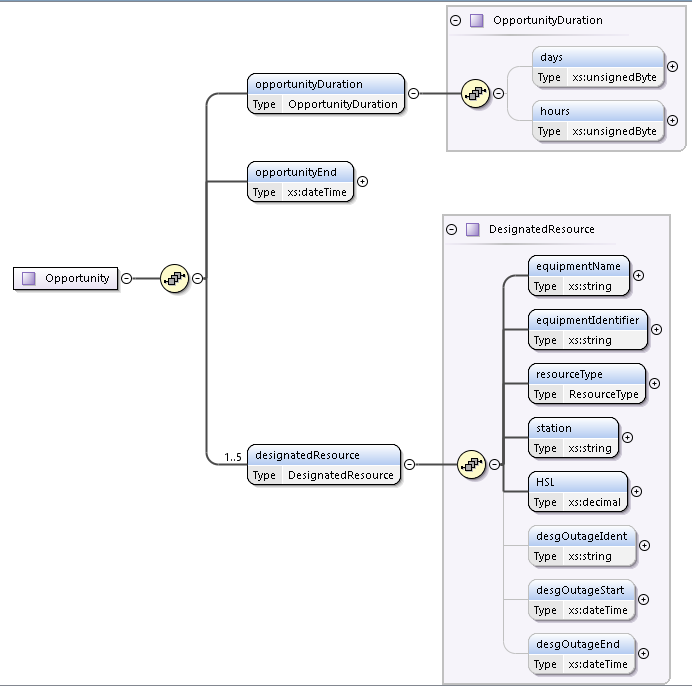


Figure 171 - Outage Grouping, Details, Schedule, Contacts, Notes, and Opportunity

As shown by the sub-structures, details are provided to reflect a resource outage or a transmission outage. For each resource, a name and resource type must be supplied. The resource is identified by its short name, and the resource type is one of ‘UN’, ‘LR’, ‘DGR’ or ‘DESR’. For each transmission equipment, a name and a transmission type must be supplied. The equipment is identified by its short name and the equipment type is one of ‘LN’, ‘DCLN’, ‘DSC’, ‘CB’, ‘XF’, ‘CP’, ‘SR’, ‘SVC’, ‘LD’, ‘SC’, ‘SCM’.

Notes provided may be types as ‘General’, ‘Supporting’ or ‘RASPS’.

On submission, the following table describes the items used for an Outage creation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Data type* | *Description* | *Values* |
| OutageInfo/outageType | Y | string | Type of Outage | FR,M1,M2,M3,OP,PL,  RS,SM,UE,FE, RO |
| OutageInfo/participant | N | string | Outage Participant |  |
| OutageInfo/versionId | N | string | It is automatically generated by the system, it is used for keeping an audit trail of all changes made to an Outage. |  |
| OutageInfo/state | Y | string | The State of the Outage, taken from ENYS, Recvd, Study, Rejct, Accpt, Apprv, Active, FExt, UExt, Withd, Cancl, ExpOp, CompEd, CompNE, PComp | ENSYS,Recvd,Study,Rejctd  Accptd,Active,FExt,UExt,Withdr  Cancld,PComp,ExpOp,CompEd  CompNE |
| OutageInfo/status | N | string | The Status of the Outage, taken from ENYS, RetE, Study, Rejct, Apprv, Accpt, Withd, Cancl, ExpOp . | ENYS,RatE,Accpt,Apprv,Rejct,  Study,Withd,Cancl,Pendg,Compl |
| OutageInfo/previousState | N | string | The previous value of the Outage State, used by the State Machine. | ENSYS,Recvd,Study,Rejctd  Accptd,Active,FExt,UExt,Withdr  Cancld,PComp, ExpOp ,CompEd  CompNE |
| OutageInfo/previousStatus | N | string | The previous value of the Outage Status, used by the State Machine. | ENYS,RatE,Accpt,Apprv,Rejct,  Study,Withd,Cancl,Pendg,Compl |
| OutageInfo/lastModifiedBy | N | string | Creator or responsible for last update |  |
| OutageInfo/warningFlag | N | Boolean | Warning Exists | true/false |
| OutageInfo/warningAcknowledge | N | Boolean | Warning Acknowledge | true/false |
| OutageInfo/warningMessage | N | string | Warnings are part of the xml response. The message indicates the issue to be resolved, they are fixed messages with integrated variables for customizing the message for the Outage. | The free-form text of the message. |
| OutageInfo/highImpactOutage | N | boolean | A Boolean indicating if the Outage includes equipment on the High Impact Transmission Elements (HITE) list. | Output only; Returned for single outages. For Group outages, see Group/GroupTransmissionOutage below. |
| OutageInfo/greater90Days | N | boolean | A Boolean indicating if the Outage is greater than 90 days in advance. | Output only; Returned for single outages. For Group outages, see Group/GroupTransmissionOutage below. |
| OutageInfo/Requestor/name | N | string | Contact name |  |
| OutageInfo/Requestor/userFullName | Y | string | Full name of requestor | UserFullName is required on a Submit, Update and Cancel. Note: the element is defined as optional in the XSD. |
| OutageInfo/Requestor/primaryContact | N | string | Primary Contact phone no. |  |
| OutageInfo/Requestor/SecondaryContact | N | string | Secondary Contact Phone No. |  |
| OutageInfo/Requestor/tertiaryContact | N | string | Tertiary Contact Phone No. |  |
| OutageInfo/requestDate | N | date | Request Date |  |
| OutageInfo/Disclaimer | Y | String | Outage disclaimer | Free-form text |
| OutageInfo/disclaimerAck | Y | Boolean | Outage disclaimer acknowledge flag | true/false |
| Group/groupId | N | string | This is the ability to group outages. A group outage has a Group ID. Each outage is still a separate outage with an outage ID as well. The Requestor can add to a group when creating the outage request. Once the outages have been submitted they can be removed from the group, creating an independent outage. Once the group has been submitted the requestor can’t add to it. The grouping will be used in the EMS-Outage Evaluation tool.  Outage grouping name | An Outage may be a member of an Outage Group. An Outage Group must have at least one and can have more than one Outage.  groupId Pattern  groupId will conform to the following three token pattern:  <ShortName>.OTG.<groupId>  groupId Example  For example, the XML document may contain the following groupId value:  <ns1:groupId>ShortName.OTG.1930</ns1:groupId>  Cancel Outage Reference  In this example, “1930” is the portion of the groupId token set that should be used when referring to this grouped outage in a cancel or get request. |
| Group/name | Y | string | Outage Group Name |  |
| Group/ResourceOutage/operatingCompany | Y | string | Group Resource Outage Operating Company Name |  |
| Group/ResourceOutage/station | Y | string | Group Resource Outage Station Name |  |
| Group/ResourceOutage/equipmentName | Y | string | This is the name of a Resource that is a Designated Resource for this Opportunity Outage. |  |
| Group/ResourceOutage/equipmentIdentifier | Y | string | RDFID used for uniquely identifying equipment | Example:  \_{123AE789-4477-4A1B-8D09-E41615AF38EC} |
| Group/ResourceOutage/resourceType | Y | string | This is the type of a Resource that is a Designated Resource for this Opportunity Outage | UN (Unit),LR (Load Resource), Distribution Generation Resource (DGR) and Distribution Energy Storage Resource (DESR) |
| Group/ResourceOutage/HSL | Y | Decimal | Group Outage Resource HSL |  |
| Group/ResourceOutage/LSL | Y | Decimal | Group Outage Resource LSL |  |
| Group/ResourceOutage/mRID | N | string | Unique identifier for an outage. | mRID is in the  format of <QSEID>.OTG.<outageType>.<outageCategory>.<outageIdent> |
| Group/ResourceOutage/natureOfWork | Y | string | This is a description of the nature of the work to be performed during the Outage. The allowable values for this field are stated above. |  |
| Group/GroupTransmissionOutage/OperatingCompany | Y | string |  |  |
| Group/ GroupTransmissionOutage /equipmentName | Y | string |  |  |
| Group/ GroupTransmissionOutage /equipmentIdentifier | Y | string | RDFID used for uniquely identifying equipment | Example:  \_{123AE789-4477-4A1B-8D09-E41615AF38EC} |
| Group/ GroupTransmissionOutage /transmissionType | Y | string | This is the type of Equipment subject to removal from service as a consequence of the Outage: e.g. CB, LINE, etc. |  |
| Group/ GroupTransmissionOutage /fromStation | Y | string | The name of the “From” Station in which an Equipment resides for all lines. |  |
| Group/ GroupTransmissionOutage /toStation | N | string | The name of the “To” Station in which an Equipment resides for all lines. |  |
| Group/ GroupTransmissionOutage /highImpactOutage | N | boolean | A Boolean indicating if the Outage includes equipment on the High Impact Transmission Elements (HITE) list. | Output only; Returned for group outages. For single outages, see OutageInfo above. |
| Group/ GroupTransmissionOutage /greater90Days | N | boolean | A Boolean indicating if the Outage is greater than 90 days in advance. | Output only; Returned for group outages. For single outages, see OutageInfo above. |
| Group/GroupTransmissionOutage/normalState | N | string | Equipment Normal State | Output only |
| Group/GroupTransmissionOutage/outageState | N | string | This field, which is pertinent (and required) only for switching devices, contains the state of the device when taken out of service: OPEN or CLOSED. This field is the opposite of the Normal State of the Equipment entity. | “C’,”O” or “” |
| Group/GroupTransmissionOutage/voltage | N | Decimal | Equipment Outage Voltage |  |
| Group/GroupTransmissionOutage/projectName | N | string | Equipment Outage NOMCR |  |
| Group/GroupTransmissionOutage/emergencyRestorationTime | Y | string | Eq. Outage Emer Restoration Time |  |
| Group/GroupTransmissionOutage/mRID | N | string | Outage Identity | mRID is in the  format of <QSEID>.OTG.<outageType>.<outageCategory>.<outageIdent> |
| Group/GroupTransmissionOutage/natureOfWork | Y | string | Eq. Outage Nature of Work |  |
| Group/Opportunity/opportunityDuration/days | N | unsigned Byte | For Opportunity Outages, the outage no. of days. | Positive no. |
| Group/Opportunity/opportunityDuration/hours | N | unsigned Byte | For Opportunity Outages, the outage no. of hours | Positive no. |
| Group/Opportunity/end | Y | DateTime | For Opportunity Outages, the outage end time | Date Time |
| Group/Opportunity/designatedResource/equipmentName | Y | string | This is the Resource name of a Resource that has been designated to an Opportunity |  |
| Group/Opportunity/designatedResource/equipmentIdentifier | Y | string | RDFID used for uniquely identifying equipment | Example:  \_{123AE789-4477-4A1B-8D09-E41615AF38EC} |
| Group/Opportunity/designatedResource/resourceType | Y | string | This is the Resource type of a Resource that has been designated to an Opportunity | UN (Unit),LR (Load Resource), Distribution Generation Resource (DGR) and Distribution Energy Storage Resource (DESR) |
| Group/Opportunity/designatedResource/station | Y | string | This is the station name of a Resource that has been designated to an Opportunity |  |
| Group/Opportunity/designatedResource/HSL | Y | Decimal |  |  |
| Group/Opportunity/desgOutageIdent | N | string | Designated Resource Outage Identifier corresponding to a valid TOO ( Transmission Opportunity Outage). | Not required for outage create transaction.  Outage Schedule will populate this field in OS query response |
| Group/Opportunity/desgOutageStart | N | DateTime | Designated Resource outage start dateTime | Not required for outage create transaction.  Outage Schedule will populate this field in OS query response |
| Group/Opportunity/desgOutageEnd | N | DateTime | Designated Resource outage End dateTime | Not required for outage create transaction.  Outage Schedule will populate this field in OS query response |
| ResourceOutage/operatingCompany | Y | string |  |  |
| ResourceOutage/station | Y | string | This is the station name of a Resource that is a Designated Resource for this Outage. |  |
| ResourceOutage/equipmentName | Y | string | This is the name of a Resource that is a Designated Resource for this Outage. |  |
| ResourceOutage/equipmentIdentifier | Y | string | RDFID used for uniquely identifying equipment | Example:  \_{123AE789-4477-4A1B-8D09-E41615AF38EC} |
| ResourceOutage/resourceType | N | string | This is the type of a Resource that is a Designated Resource for this Outage. | UN (Unit),LR (Load Resource), Distribution Generation Resource (DGR) and Distribution Energy Storage Resource (DESR) |
| ResourceOutage/HSL | Y | Decimal | Resource Outage HSL |  |
| ResourceOutage/LSL | Y | Decimal | Resource Outage LSL |  |
| ResourceOutage/mRID | N | string | Outage Identity | mRID is in the  format of <QSEID>.OTG.<outageType>.<outageCategory>.<outageIdent> |
| ResourceOutage/natureOfWork | Y | string | Resource Outage Nature of Work. . The allowable values for this field are stated above. |  |
| TransmissionOutage/operatingCompany | Y | string |  |  |
| TransmissionOutage/equipmentType | N | string | Allowed values for Transmission equipment types as stated above. |  |
| TransmissionOutage/equipmentName | Y | string | The name of the Equipment that can be the subject of an Outage. |  |
| TransmissionOutage/equipmentIdentifier | Y | string | RDFID used for uniquely identifying equipment | Example:  \_{123AE789-4477-4A1B-8D09-E41615AF38EC} |
| TransmissionOutage/transmissionType | Y | string | The Type of the Transmission Equipment that can be the subject of an Outage. |  |
| TransmissionOutage/fromStation | Y | string | This is the name of the Station in which an Equipment resides for all equipment types other than lines. For lines this refers to the “From” station |  |
| TransmissionOutage/toStation | N | string | This is the name of the “To” Station in which an Equipment resides for all lines. |  |
| TransmissionOutage/normalState | N | string | This field, which is pertinent only for switching devices, contains the normal state of the device: OPEN or CLOSED. This field is the same as the Normal State of the Equipment entity. | Output only |
| TransmissionOutage/outageState | N | string | This field, which is pertinent (and required) only for switching devices, contains the state of the device when taken out of service: OPEN or CLOSED. This field is the opposite of the Normal State of the Equipment entity. | “C’,”O” or “” |
| TransmissionOutage/voltage | N | Decimal | The voltage level at which the Equipment resides. For transformers it is the high side voltage level. |  |
| TransmissionOutage/projectName | N | String |  |  |
| TransmissionOutage/emergencyRestorationTime | Y | positiveInteger | This is the time in hours necessary to terminate the Outage and return the Equipment to service. |  |
| TransmissionOutage/mRID | N | string | Outage Identity | mRID is in the  format of <QSEID>.OTG.<outageType>.<outageCategory>.<outageIdent> |
| TransmissionOutage/natureOfWork | Y | string |  |  |
| TransmissionOutage/Opportunity/opportunityDuration/days | N | unsigned Byte | For Opportunity Outages, the outage no. of days. | Positive no. |
| TransmissionOutage/Opportunity/opportunityDuration/hours | N | unsigned Byte | For Opportunity Outages, the outage no. of hours | Positive no. |
| TransmissionOutage/Opportunity/opportunityEnd | Y | DateTime | For Opportunity Outages, the outage end time | Date Time |
| TransmissionOutage/Opportunity/designatedResource/equipmentName | Y | string | This is the Resource name of a Resource that has been designated to an Opportunity |  |
| TransmissionOutage/Opportunity/designatedResource/equipmentIdentifier | Y | String |  |  |
| TransmissionOutage/Opportunity/designatedResource/resourceType | Y | String | This is the Resource type of a Resource that has been designated to an Opportunity | UN (Unit) or LR (Load Resource), Distribution Generation Resource (DGR) and Distribution Energy Storage Resource (DESR) |
| TransmissionOutage/Opportunity/designatedResource/station | Y | String |  |  |
| TransmissionOutage/Opportunity/designatedResource/HSL | Y | Decimal |  |  |
| TransmissionOutage/Opportunity/designatedResource/desgOutageIdent | N | String | Designated Resource Outage Identifier corresponding to a valid TOO ( Transmission Opportunity Outage). | Not required for outage create transaction.  Outage Schedule will populate this field in OS query response |
| TransmissionOutage/Opportunity/designatedResource/desgOutageStart | N | dateTime | Designated Resource outage start dateTime | Not required for outage create transaction.  Outage Schedule will populate this field in OS query response |
| TransmissionOutage/Opportunity/designatedResource/desgOutageEnd | N | dateTime | Designated Resource outage End dateTime | Not required for outage create transaction.  Outage Schedule will populate this field in OS query response |
| Schedule/plannedSart | N | dateTime | This is the date/time at which the Outage is planned to start. | * Forced (Transmission and Resource) - Forced outages don’t have a Planned Start when created * Transmission Opportunity outages (TOO).  TOO’s doesn’t have a Planned Start or a Planned End when created |
| Schedule/plannedEnd | N | dateTime | This is the date/time at which the Outage is planned to end. |
| Schedule/earliestStart | N | dateTime | This is the earliest date/time at which the Outage may start. |
| Schedule/latestEnd | N | dateTime | This is the latest date/time at which the Outage may end. |
| Schedule/actualStart | N | dateTime | This is the actual date/time at which the Outage started. |
| Schedule/actualEnd | N | dateTime | This is the actual date/time at which the Outage ended. |
| Schedule/new\_plannedStart | N | dateTime | This is the proposed date/time at which an Unavoidable Extension may start. |
| Schedule/new\_plannedEnd | N | dateTime | This is the proposed date/time at which an Unavoidable Extension may end. |
| Schedule/new\_earliestStart | N | dateTime | This is the proposed earliest date/time at which a Resource Opportunity Outage may start. |
| Schedule/new\_latestEnd | N | dateTime | This is the proposed latest date/time at which a Resource Opportunity Outage may end. |
| Recurrence/datesList/dates/mRID | N | String | Outage Identity | mRID is in the  format of <QSEID>.OTG.<outageType>.<outageCategory>.<outageIdent> |
| Recurrence/datesList/dates/plannedStart | Y | Date | Planned start date of outage recurrence |  |
| Recurrence/datesList/dates/plannedEnd | Y | Date | Planned End date of outage recurrence |  |
| Recurrence/datesList/dates/earliestStart | Y | Date | Earliest start date of outage recurrence |  |
| Recurrence/datesList/dates/latestEnd | Y | Date | Latest Planned End date of outage recurrence |  |
| OSNotes/RequestorNotes/Note/createdTime | Y | string | There are three sections that the requestor can enter notes, Requestor Notes, Supporting Information, and Remedial Action or Special Protection System notes. Notes are required for some outage types but not all. They are not required when the outage is submitted but are required for some outages to be completed. There is another section for reviewer notes. This is for ERCOT to enter notes and the MP can only read them. |  |
| OSNotes/RequestorNotes/Note/createdBy | Y | string |  |  |
| OSNotes/RequestorNotes/Note/company | Y | string |  |  |
| OSNotes/RequestorNotes/Note/comment | Y | string |  |  |
| OSNotes/ReviewerNotes/Note/createdTime | Y | string |  |  |
| OSNotes/ReviewerNotes/Note/createdBy | Y | string |  |  |
| OSNotes/ReviewerNotes/Note/company | Y | string |  |  |
| OSNotes/ReviewerNotes/Note/comment | Y | string |  |  |
| OSNotes/SupportingNotes/Note/createdTime | Y | string |  |  |
| OSNotes/SupportingNotes/Note/createdBy | Y | string |  |  |
| OSNotes/SupportingNotes/Note/company | Y | string |  |  |
| OSNotes/SupportingNotes/Note/comment | Y | string |  |  |
| OSNotes/RASPSNotes/Note/createdTime | Y | string |  |  |
| OSNotes/RASPSNotes/Note/createdBy | Y | string |  |  |
| OSNotes/RASPSNotes/Note/company | Y | string |  |  |
| OSNotes/RASPSNotes/Note/comment | Y | string |  |  |
| Error/severity | N | string | Severity of Error Generated |  |
| Error/area | N | string | Area of generated error |  |
| Error/interval | N | string |  |  |
| Error/text | Y | string | Error Text |  |

Figure 172 – Outage Creation Elements

The following is an example of an OutageSet submission by a Market Participant:

<ns2:OutageSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:ns2="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns2:Outage>

<ns2:OutageInfo>

<ns2:outageType>PL</ns2:outageType>

<ns2:participant>TABC</ns2:participant>

<ns2:lastModified>2016-08-03T14:08:00-05:00</ns2:lastModified>

<ns2:lastModifiedBy>USER1</ns2:lastModifiedBy>

<ns2:Requestor>

<ns2:name>USER1</ns2:name>

<ns2:userFullName>Alex Smith</ns2:userFullName>

<ns2:primaryContact>512-555-5555</ns2:primaryContact>

<ns2:secondaryContact>512-555-5556</ns2:secondaryContact>

<ns2:tertiaryContact>512-555-5557</ns2:tertiaryContact>

</ns2:Requestor>

<ns2:requestDate>2016-08-03T14:09:00-05:00</ns2:requestDate>

<ns2:Disclaimer>temp disclaimer</ns2:Disclaimer>

<ns2:disclaimerAck>true</ns2:disclaimerAck>

</ns2:OutageInfo>

<ns2:TransmissionOutage>

<ns2:operatingCompany>TABC</ns2:operatingCompany>

<ns2:equipmentName>ABC\_123</ns2:equipmentName>

<ns2:equipmentIdentifier>\_{123AE789-4477-4A1B-8D09-E41615AF38EC}</ns2:equipmentIdentifier>

<ns2:transmissionType>LN</ns2:transmissionType>

<ns2:fromStation>STATION1</ns2:fromStation>

<ns2:normalState>C</ns2:normalState>

<ns2:outageState>O</ns2:outageState>

<ns2:voltage>6.9</ns2:voltage>

<ns2:projectName>project123</ns2:projectName>

<ns2:emergencyRestorationTime>1</ns2:emergencyRestorationTime>

<ns2:natureOfWork>OE</ns2:natureOfWork>

</ns2:TransmissionOutage>

<ns2:Schedule>

<ns2:plannedStart>2016-08-16T10:00:00</ns2:plannedStart>

<ns2:plannedEnd>2016-08-16T12:00:00</ns2:plannedEnd>

<ns2:earliestStart>2016-08-16T10:00:00</ns2:earliestStart>

<ns2:latestEnd>2016-08-16T14:00:00</ns2:latestEnd>

</ns2:Schedule>

<ns2:OSNotes>

<ns2:RequestorNotes>

<ns2:Note>

<ns2:createdTime>2016-04-11T11:00:00</ns2:createdTime>

<ns2:createdBy>ASmith</ns2:createdBy>

<ns2:company>TABC</ns2:company>

<ns2:comment>temp comment</ns2:comment>

</ns2:Note>

</ns2:RequestorNotes>

</ns2:OSNotes>

</ns2:Outage>

</ns2:OutageSet>

The following is an example response message for the above request:

<ns1:OutageSet xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il"  
 xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">  
 <ns1:Outage>  
 <ns1:OutageInfo>  
 <ns1:outageType>PL</ns1:outageType>  
 <ns1:participant>TABC</ns1:participant>  
 <ns1:versionId>1</ns1:versionId>  
 <ns1:state>Recvd</ns1:state>  
 <ns1:status>RatE</ns1:status>  
 <ns1:previousState>ENYS</ns1:previousState>

<ns1:highImpactOutage>true</ns1:highImpactOutage>  
            <ns1:greater90Days>false</ns1:greater90Days>  
 <ns1:Requestor>  
 <ns1:name>USER1</ns1:name>  
 <ns1:userFullName>Alex Smith</ns1:userFullName>  
 <ns1:primaryContact/>  
 <ns1:secondaryContact/>  
 <ns1:tertiaryContact>512-555-5557</ns1:tertiaryContact>  
 </ns1:Requestor>  
 <ns1:requestDate>2016-08-03T14:33:00-05:00</ns1:requestDate>  
 </ns1:OutageInfo>  
 <ns1:TransmissionOutage>  
 <ns1:operatingCompany>TABC</ns1:operatingCompany>  
 <ns1:equipmentType>LN</ns1:equipmentType>  
 <ns1:equipmentName> ABC\_123</ns1:equipmentName>  
 <ns1:equipmentIdentifier>\_{123AE789-4477-4A1B-8D09-E41615AF38EC}</ns1:equipmentIdentifier>  
 <ns1:transmissionType>LN</ns1:transmissionType>  
 <ns1:fromStation>STATION1</ns1:fromStation>  
 <ns1:toStation>STATION2</ns1:toStation>  
 <ns1:outageState>O</ns1:outageState>  
 <ns1:projectName>project123</ns1:projectName>  
 <ns1:emergencyRestorationTime>1</ns1:emergencyRestorationTime>  
 <ns1:mRID>TABC.OTG.PL.Transmission.ABC00118301</ns1:mRID>  
 <ns1:natureOfWork>OE</ns1:natureOfWork>  
 </ns1:TransmissionOutage>  
 <ns1:Schedule>  
 <ns1:plannedStart>2016-08-16T10:00:00-05:00</ns1:plannedStart>  
 <ns1:plannedEnd>2016-08-16T12:00:00-05:00</ns1:plannedEnd>  
 <ns1:earliestStart>2016-08-16T10:00:00-05:00</ns1:earliestStart>  
 <ns1:latestEnd>2016-08-16T14:00:00-05:00</ns1:latestEnd>  
 </ns1:Schedule>  
 <ns1:OSNotes>  
 <ns1:RequestorNotes>  
 <ns1:Note>  
 <ns1:createdTime>2016-04-11T11:00:00-05:00</ns1:createdTime>  
 <ns1:createdBy>ASmith</ns1:createdBy>  
 <ns1:company>TABC</ns1:company>  
 <ns1:comment>temp comment</ns1:comment>  
 </ns1:Note>  
 </ns1:RequestorNotes>  
 <ns1:ReviewerNotes/>  
 <ns1:SupportingNotes/>  
 <ns1:RASPSNotes/>  
 </ns1:OSNotes>  
 </ns1:Outage>  
</ns1:OutageSet>

The following is an example of a Group OutageSet Submission by a Market Participant:

<OutageSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews">  
 <Outage>  
 <OutageInfo>  
 <outageType>PL</outageType>  
 <participant>TABC</participant>  
 <lastModified>2016-08-04T15:24:00-05:00</lastModified>  
 <lastModifiedBy>asmith</lastModifiedBy>  
 <Requestor>  
 <name>2241</name>  
 <userFullName>Alex Smith</userFullName>  
 <tertiaryContact>512-555-1234</tertiaryContact>  
 </Requestor>  
 <requestDate>2016-08-04T15:24:00-05:00</requestDate>  
 <Disclaimer>Temp Disclaimer</Disclaimer>  
 <disclaimerAck>true</disclaimerAck>  
 </OutageInfo>  
 <Group>  
 <name>Grp2</name>  
 <GroupTransmissionOutage>  
 <operatingCompany>TABC</operatingCompany>  
 <equipmentName>ABC\_1234</equipmentName>  
 <equipmentIdentifier>\_{123AB3F7-11A4-4F87-A2C0-125F79F221B4}</equipmentIdentifier>  
 <transmissionType>DSC</transmissionType>  
 <fromStation>Station1</fromStation>  
 <normalState>C</normalState>  
 <outageState>O</outageState>  
 <voltage>69</voltage>  
 <projectName>Project2</projectName>  
 <emergencyRestorationTime>1</emergencyRestorationTime>  
 <natureOfWork>RE</natureOfWork>  
 </GroupTransmissionOutage>  
 <GroupTransmissionOutage>  
 <operatingCompany>TABC</operatingCompany>  
 <equipmentName>ABC\_2345</equipmentName>  
 <equipmentIdentifier>\_{A12BCD24-5968-45F3-9471-728032982EE3}</equipmentIdentifier>  
 <transmissionType>DSC</transmissionType>  
 <fromStation>Station1</fromStation>  
 <normalState>C</normalState>  
 <outageState>O</outageState>  
 <voltage>138</voltage>  
 <projectName>Project3</projectName>  
 <emergencyRestorationTime>1</emergencyRestorationTime>  
 <natureOfWork>RE</natureOfWork>  
 </GroupTransmissionOutage>  
 </Group>  
 <Schedule>  
 <plannedStart>2016-08-12T08:00:00-05:00</plannedStart>  
 <plannedEnd>2016-08-12T12:00:00-05:00</plannedEnd>  
 <earliestStart>2016-08-12T08:00:00-05:00</earliestStart>  
 <latestEnd>2016-08-12T13:00:00-05:00</latestEnd>  
 </Schedule>  
 </Outage>  
</OutageSet>

The following is an example response message for the above request:

<ns1:OutageSet xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:Outage>

<ns1:OutageInfo>

<ns1:outageType>PL</ns1:outageType>

<ns1:participant>TABC</ns1:participant>

<ns1:versionId>1</ns1:versionId>

<ns1:state>ENYS</ns1:state>

<ns1:status>RatE</ns1:status>

<ns1:previousState>ENYS</ns1:previousState>

<ns1:Requestor>

<ns1:name>2241</ns1:name>

<ns1:userFullName>Alex Smith</ns1:userFullName>

<ns1:primaryContact/>

<ns1:secondaryContact/>

<ns1:tertiaryContact>512-555-1234</ns1:tertiaryContact>

</ns1:Requestor>

<ns1:requestDate>2017-08-04T15:24:06-05:00</ns1:requestDate>

</ns1:OutageInfo>

<ns1:Group>

<ns1:groupId>TABC.OTG.167432</ns1:groupId>

<ns1:name>Grp2</ns1:name>

<ns1:GroupTransmissionOutage>

<ns1:operatingCompany>TABC</ns1:operatingCompany>

<ns1:equipmentName>ABC\_1234</ns1:equipmentName>

<ns1:equipmentIdentifier\_{123AB3F7-11A4-4F87-A2C0-125F79F221B4}</ns1:equipmentIdentifier>

<ns1:transmissionType>DSC</ns1:transmissionType>

<ns1:fromStation>Station1</ns1:fromStation>

<ns1:toStation>Station2</ns1:toStation>

<ns1:highImpactOutage>true</ns1:highImpactOutage>

<ns1:greater90Days>false</ns1:greater90Days>

<ns1:normalState>C</ns1:normalState>

<ns1:outageState>O</ns1:outageState>

<ns1:voltage>69</ns1:voltage>

<ns1:projectName>Project2</ns1:projectName>

<ns1:emergencyRestorationTime>1</ns1:emergencyRestorationTime>

<ns1:mRID>TABC.OTG.PL.Transmission.ABC00012345</ns1:mRID>

<ns1:natureOfWork>RE</ns1:natureOfWork>

</ns1:GroupTransmissionOutage>

<ns1:GroupTransmissionOutage>

<ns1:operatingCompany>TABC</ns1:operatingCompany>

<ns1:equipmentName>ABC\_2345</ns1:equipmentName>

<ns1:equipmentIdentifier>\_{A12BCD24-5968-45F3-9471-728032982EE3}</ns1:equipmentIdentifier>

<ns1:transmissionType>DSC</ns1:transmissionType>

<ns1:fromStation>Station1</ns1:fromStation>

<ns1:toStation>Station2</ns1:toStation>

<ns1:highImpactOutage>true</ns1:highImpactOutage>

<ns1:greater90Days>false</ns1:greater90Days>

<ns1:normalState>C</ns1:normalState>

<ns1:outageState>O</ns1:outageState>

<ns1:voltage>138</ns1:voltage>

<ns1:projectName>Project3</ns1:projectName>

<ns1:emergencyRestorationTime>1</ns1:emergencyRestorationTime>

<ns1:mRID>TABC.OTG.PL.Transmission.ABC00023456</ns1:mRID>

<ns1:natureOfWork>RE</ns1:natureOfWork>

</ns1:GroupTransmissionOutage>

</ns1:Group>

<ns1:Schedule>

<ns1:plannedStart>2016-08-12T08:00:00-05:00</ns1:plannedStart>

<ns1:plannedEnd>2016-08-12T12:00:00-05:00</ns1:plannedEnd>

<ns1:earliestStart>2016-08-12T08:00:00-05:00</ns1:earliestStart>

<ns1:latestEnd>2016-08-12T13:00:00-05:00</ns1:latestEnd>

</ns1:Schedule>

<ns1:OSNotes>

<ns1:RequestorNotes/>

<ns1:ReviewerNotes/>

<ns1:SupportingNotes/>

<ns1:RASPSNotes/>

</ns1:OSNotes>

</ns1:Outage>

</ns1:OutageSet>

### Outage Query

When getting Outages, an OutageSet container class is used to return zero or more outages that match the query criteria.

ERCOT will enforce a limitation on the number of outages that may be returned in response to an outage query request. The goal of this constraint is to protect the Nodal architecture and mitigate the risk of excessively large result sets negatively impacting the infrastructure and market operation. If a request arrives that discovers a larger number of outages than the enforced constraint allows, the response payload will not contain any outages. Instead, the response will contain a message indicating that the query exceeded the allowable number of outages. The response will further emphasize that the market participant should reduce the query duration to obtain an allowable number of outages. As market trials and Nodal testing continues, ERCOT will adjust the thresholds as necessary until the go-live target is observed and finalized.

The request message for querying outages would use the following message fields:

Please note

mRID format *<QSE ID>.OTG.<outageType>.<outageCategory>.<outageIdent>*

groupId format *<QSE ID>.OTG.<group\_id>*

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | OutageSet |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/StartTime | *Optional: start time of interest* |
| Request/EndTime | *Optional: end time of interest* |
| Request/ID | *Optional:*  *mRID of the outage to be queried  or groupId of the GroupOutage to be queried* |
| Payload | *Optional - for Ad-Hoc Queries:*  *OutageQuery* |

Figure 173 - OutageSet Query Request

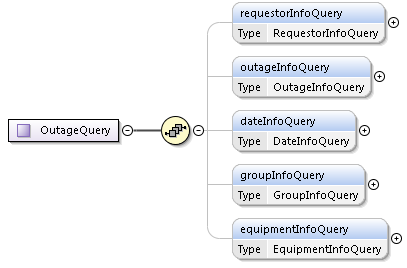
Please note:

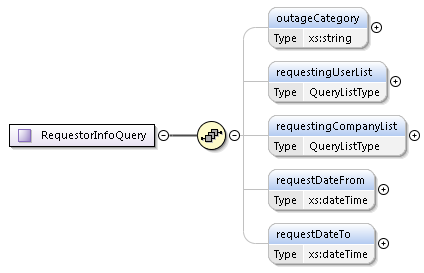
* If startTime and endTime are not specified, only active outages will be returned.Here startTime represents plannedStartFrom and endTime represents plannedEndTo.
* StartTime and endTime are optional but they should not be present when an mRID is sent.
* The mRID can be used to request details of a specific outage.
* The GroupID can be used to request outages that are part of a group.
* For Ad-Hoc Outage queries, OutageQuery (a container element for querying outages) can be included in the payload with select query elements populated.

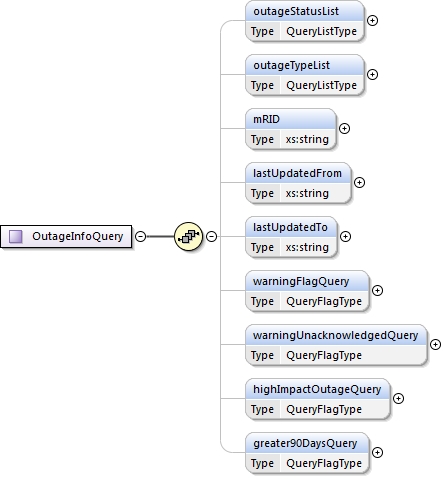
The corresponding response messages would use the following message fields:

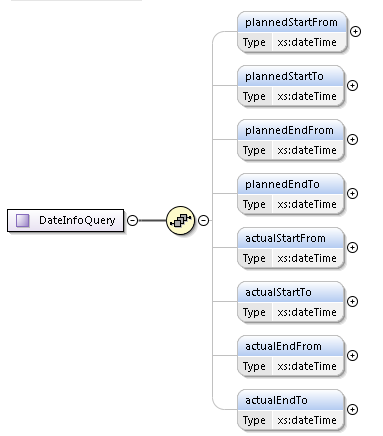
|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | OutageSet |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |
| Payload | OutageSet |

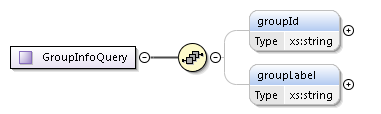
Figure 174 - OutageSet Query Response

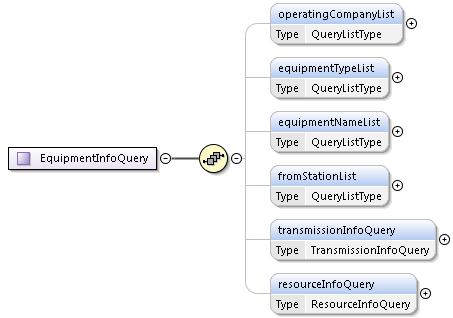


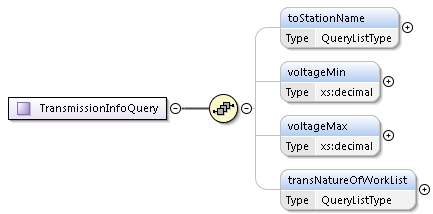












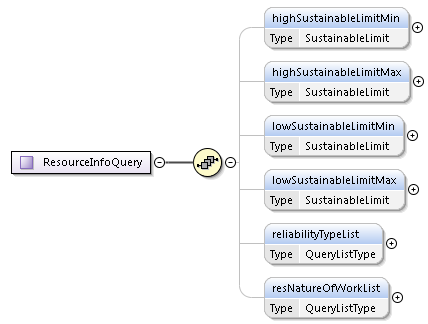


Figure 175 - OutageQuery Container Element structure

The following are XML examples for OutageQuery payload:

Example for querying Resource outages by requesting company:

<ns2:OutageQuery xmlns="http://www.ercot.com/schema/2007-06/nodal/ews"  
 xmlns:ns2="http://www.ercot.com/schema/2007-06/nodal/ews">  
 <ns2:requestorInfoQuery>  
 <ns2:outageCategory>Resource</ns2:outageCategory>  
 <ns2:requestingCompanyList>  
 <ns2:listItem>QABC</ns2:listItem>  
 </ns2:requestingCompanyList>  
 </ns2:requestorInfoQuery>  
</ns2:OutageQuery>

Example for querying outages with an *Accpt* status:

<OutageQuery xmlns="http://www.ercot.com/schema/2007-06/nodal/ews">  
 <requestorInfoQuery/>  
 <outageInfoQuery>  
 <outageStatusList>  
 <listItem>Accpt</listItem>  
 </outageStatusList>  
 </outageInfoQuery>  
</OutageQuery>

Example for querying an outage using the mRID:

<Request>  
 <ID>TLCRA.OTG.PL.Transmission.LCR00118301</ID>  
</Request>

### Outage Update

The request message for outage update would use the following message fields:

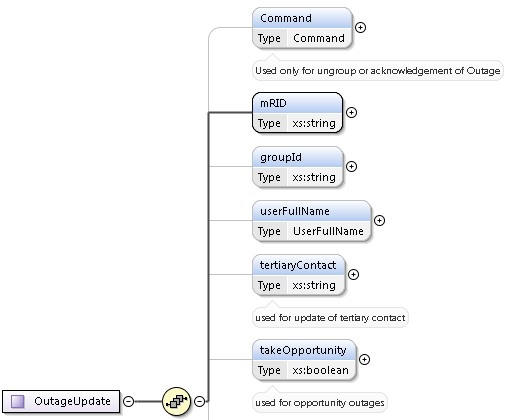
|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | Change |
| Header/Noun | OutageSet |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Payload | *OutageUpdate* |

Figure 176 - Outage Update Request

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | OutageSet |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |

Figure 177 - Outage Update Response



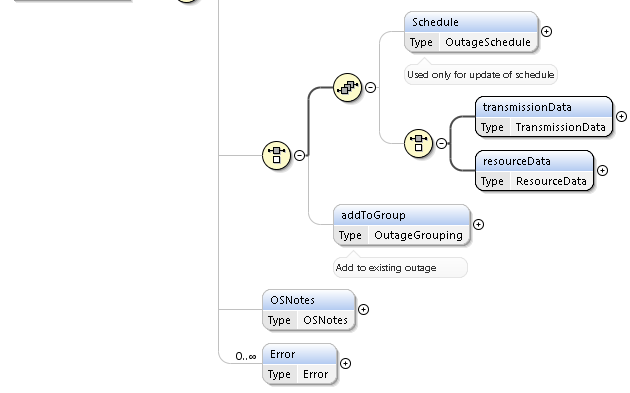


Figure 178 - Outage Update XML payload Schema Diagram

On Outage “change” requests, the following items are used:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Data type* | *Description* | *Values* |
| Command | N | string | Requestor Update | UNGROUP, ACKNOWLEDGE, |
| OutageUpdate/mRID | Y | string | Outage Identity | mRID is in the  format of <QSEID>.OTG.<outageType>.<outageCategory>.<outageIdent> |
| OutageUpdate/groupId | N | String | This is the id for a Group Outage |  |
| OutageUpdate/userFullName | Y | String |  | UserFullName is required on a Submit, Update and Cancel. Note: the element is defined as optional in the XSD. |
| OutageUpdate/tertiaryContact | N | string | Tertiary Contact Phone No. |  |
| OutageUpdate/takeOpportunity | N | boolean | For taking opportunity outage. i.e.  To accept the opportunity when a opportunity match exists. | true/false |
| Schedule/plannedSart | N | dateTime | This is the date/time at which the Outage is planned to start. |  |
| Schedule/plannedEnd | N | dateTime | This is the date/time at which the Outage is planned to end. |  |
| Schedule/earliestStart | N | dateTime | This is the earliest date/time at which the Outage may start. |  |
| Schedule/latestEnd | N | dateTime | This is the latest date/time at which the Outage may end. |  |
| Schedule/actualStart | N | dateTime | This is the actual date/time at which the Outage started. |  |
| Schedule/actualEnd | N | dateTime | This is the actual date/time at which the Outage ended. |  |
| Schedule/new\_plannedStart | N | dateTime | This is the proposed date/time at which an Unavoidable Extension may start. |  |
| Schedule/new\_plannedEnd | N | dateTime | This is the proposed date/time at which an Unavoidable Extension may end. |  |
| Schedule/new\_earliestStart | N | dateTime | This is the proposed earliest date/time at which a Resource Opportunity Outage may start. |  |
| Schedule/new\_latestEnd | N | dateTime | This is the proposed latest date/time at which a Resource Opportunity Outage may end. |  |
| transmissionData/emergencyRestorationTime | N | positiveInteger |  |  |
| transmissionData/transNatureOfWork | N | string |  |  |
| resourceData/highSustainableLimit | N | integer |  |  |
| resourceData/lowSustainableLimit | N | Integer |  |  |
| resourceData/resNatureOfWork | N | string |  |  |
| addToGroup/groupId | N | string |  |  |
| addToGroup/name | Y | string |  |  |
| ResourceOutage/operatingCompany | Y | string |  |  |
| ResourceOutage/station | Y | string | This is the station name of a Resource that is a Designated Resource for this Outage. |  |
| ResourceOutage/equipmentName | Y | string | This is the name of a Resource that is a Designated Resource for this Outage. |  |
| ResourceOutage/equipmentIdentifier | Y | string |  |  |
| ResourceOutage/resourceType | N | string | This is the type of a Resource that is a Designated Resource for this Outage. | UN (Unit), LR (Load Resource), Distribution Generation Resource (DGR) and Distribution Energy Storage Resource (DESR) |
| ResourceOutage/HSL | Y | Decimal | Resource Outage HSL |  |
| ResourceOutage/LSL | Y | Decimal | Resource Outage LSL |  |
| ResourceOutage/mRID | N | string | Outage Identity | mRID is in the  format of <QSEID>.OTG.<outageType>.<outageCategory>.<outageIdent> |
| ResourceOutage/natureOfWork | Y | string | Resource Outage Nature of Work. . The allowable values for this field are stated above. |  |
| GroupTransmissionOutage/operatingCompany | Y | string |  |  |
| GroupTransmissionOutage/equipmentName | Y | string | The name of the Equipment that can be the subject of an Outage. |  |
| GroupTransmissionOutage/equipmentIdentifier | Y | string |  |  |
| GroupTransmissionOutage/transmissionType | Y | string | The Type of the Transmission Equipment that can be the subject of an Outage. |  |
| GroupTransmissionOutage/fromStation | Y | string | This is the name of the Station in which an Equipment resides for all equipment types other than lines. For lines this refers to the “From” station |  |
| GroupTransmissionOutage/toStation | N | string | This is the name of the “To” Station in which an Equipment resides for all lines. |  |
| GroupTransmissionOutage/normalState | N | string | This field, which is pertinent only for switching devices, contains the normal state of the device: OPEN or CLOSED. This field is the same as the Normal State of the Equipment entity. | Output only |
| GroupTransmissionOutage/outageState | N | string | This field, which is pertinent (and required) only for switching devices, contains the state of the device when taken out of service: OPEN or CLOSED. This field is the opposite of the Normal State of the Equipment entity. | “C’,”O” or “” |
| GroupTransmissionOutage/voltage | N | Decimal | The voltage level at which the Equipment resides. For transformers it is the high side voltage level. |  |
| GroupTransmissionOutage/projectName | N | String |  |  |
| GroupTransmissionOutage/emergencyRestorationTime | Y | positiveInteger | This is the time in hours necessary to terminate the Outage and return the Equipment to service. |  |
| GroupTransmissionOutage/mRID | N | string | Outage Identity | mRID is in the  format of <QSEID>.OTG.<outageType>.<outageCategory>.<outageIdent> |
| GroupTransmissionOutage/natureOfWork | Y | string |  |  |
| Opportunity/opportunityDuration/days | N | unsigned Byte | For Opportunity Outages, the outage no. of days. | Positive no. |
| Opportunity/opportunityDuration/hours | N | unsigned Byte | For Opportunity Outages, the outage no. of hours | Positive no. |
| Opportunity/opportunityEnd | Y | DateTime | For Opportunity Outages, the outage end time | Date Time |
| Opportunity/designatedResource/equipmentName | Y | string | This is the Resource name of a Resource that has been designated to an Opportunity |  |
| Opportunity/designatedResource/equipmentIdentifier | Y | String |  |  |
| Opportunity/designatedResource/resourceType | Y | String | This is the Resource type of a Resource that has been designated to an Opportunity | UN (Unit),LR (Load Resource), Distribution Generation Resource (DGR) and Distribution Energy Storage Resource (DESR) |
| Opportunity/designatedResource/station | Y | String |  |  |
| Opportunity/designatedResource/HSL | Y | Decimal |  |  |
| Opportunity/designatedResource/desgOutageIdent | N | String | Designated Resource Outage Identifier corresponding to a valid TOO ( Transmission Opportunity Outage). | Not required for outage create transaction.  Outage Schedule will populate this field in OS query response |
| Opportunity/designatedResource/desgOutageStart | N | dateTime | Designated Resource outage start dateTime | Not required for outage create transaction.  Outage Schedule will populate this field in OS query response |
| Opportunity/designatedResource/desgOutageEnd | N | dateTime | Designated Resource outage End dateTime | Not required for outage create transaction.  Outage Schedule will populate this field in OS query response |
| OSNotes/RequestorNotes/Note/createdTime | Y | string | There are three sections that the requestor can enter notes, Requestor Notes, Supporting Information, and Remedial Action or Special Protection System notes. Notes are required for some outage types but not all. They are not required when the outage is submitted but are required for some outages to be completed. |  |
| OSNotes/RequestorNotes/Note/createdBy | Y | string |  |
| OSNotes/RequestorNotes/Note/company | Y | string |  |
| OSNotes/RequestorNotes/Note/comment | Y | string |  |
| OSNotes/SupportingNotes/Note/createdTime | Y | string |  |
| OSNotes/SupportingNotes/Note/createdBy | Y | string |  |
| OSNotes/SupportingNotes/Note/company | Y | string |  |
| OSNotes/SupportingNotes/Note/comment | Y | string |  |
| OSNotes/RASPSNotes/Note/createdTime | Y | string |  |
| OSNotes/RASPSNotes/Note/createdBy | Y | string |  |
| OSNotes/RASPSNotes/Note/company | Y | string |  |
| OSNotes/RASPSNotes/Note/comment | Y | string |  |
| Error/severity | N | string | Severity of Error Generated |  |
| Error/area | N | string | Area of generated error |  |
| Error/interval | N | string |  |  |
| Error/text | Y | string | Error Text |  |

The following is an example for an Update OutageSet request that updates schedule and emergency restoration time:

<OutageUpdate xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
 <mRID>TABC.OTG.PL.Transmission.ABC00118301</mRID>  
 <userFullName>Alex Smith</userFullName>  
 <Schedule>  
 <plannedStart>2016-08-16T11:00:00-05:00</plannedStart>  
 <plannedEnd>2016-08-16T13:00:00-05:00</plannedEnd>  
 <earliestStart>2016-08-16T11:00:00-05:00</earliestStart>  
 <latestEnd>2016-08-16T14:00:00-05:00</latestEnd>  
 </Schedule>  
 <transmissionData>  
 <emergencyRestorationTime>2</emergencyRestorationTime>  
 </transmissionData>  
 </OutageUpdate>

The following is an example for an Update OutageSet request that adds equipment to a group outage:

<OutageUpdate xmlns="http://www.ercot.com/schema/2007-06/nodal/ews"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
 <mRID>TABC.OTG.M1.Transmission.ABC00116294</mRID>  
 <userFullName>Alex Smith</userFullName>  
 <tertiaryContact>512-555-5555</tertiaryContact>  
 <addToGroup>  
 <groupId>TABC.OTG.166835</groupId>  
 <name>Grp1</name>  
 <GroupTransmissionOutage>  
 <operatingCompany>TABC</operatingCompany>  
 <equipmentName>AB\_2315</equipmentName>  
 <equipmentIdentifier>\_{ABCE0AC3-6111-4DCF-803F-27B372A3A081}</equipmentIdentifier>  
 <transmissionType>CB</transmissionType>  
 <fromStation>Station1</fromStation>  
 <normalState>0</normalState>  
 <outageState>O</outageState>  
 <voltage>0</voltage>  
 <projectName>Project1</projectName>  
 <emergencyRestorationTime>1</emergencyRestorationTime>  
 <natureOfWork>OT</natureOfWork>  
 </GroupTransmissionOutage>  
 </addToGroup>  
 <OSNotes>  
 <RequestorNotes>  
 <Note>  
 <createdTime>2016-03-10T11:13:51-06:00</createdTime>  
 <createdBy>Alex Smith</createdBy>  
 <company>TABC</company>  
 <comment>Adding equipment</comment>  
 </Note>  
 </RequestorNotes>  
 </OSNotes>  
 </OutageUpdate>

### Outage Cancellation

The request message for outage cancellation would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | Cancel |
| Header/Noun | OutageSet |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/ID | *mRID\* of the Outage to be canceled*  *Only one outage can be cancelled at a time* |
| Payload | *OutageCancel (Notes) [Optional]* |

Figure 179 - Outage Cancel Request

To cancel an Outage:

*<QSEID>.OTG.<outageType>.<outageCategory>.<outageIdent>.<cancelReasonCode>*

To cancel Outages by the group:

*<QSEID>.OTG.<outageType>.<outageCategory>.<outageIdent>.<groupId>.<cancelReasonCode>*

To cancel a single Outage in a group:

*<QSEID>.OTG.<outageType>.<outageCategory>.<outageIdent>.<groupId>.<cancelReasonCode>.TRUE*

* The Outage Scheduler cancellation reason code should be appended to the Outage Scheduler mRID as shown in the examples above.
* The <groupId> shown in the mRID of the second and third examples above corresponds to the third token of the groupId element returned in the Outage Creation XML response. For example, if the groupId returned in response to an Outage Creation request was *<ns1:groupId>ShortName.OTG.1930</ns1:groupId>*, the value placed in the Outage Cancel request for *<groupId>* should be *1930*. Refer to the table in Section 7.2.1 for additional information.
* When canceling a single outage within a group, the TRUE at the end of the mRID in the third example instructs Outage Scheduler to ungroup the outage.

|  |  |
| --- | --- |
| mRID CancelRequestCode Token | Outage Scheduler Cancel Request Reason Description |
| ERR | Cancel – ERROR |
| EXP | Cancel – Expired |
| RES1M | Cancel – Reschedule within 1 month |
| CWE | Cancel – Coordinating with ERCOT |
| EOP | Cancel – Emergency Operations |
| LCU | Cancel – Labor/Crew Unavailable |
| MUA | Cancel – Materials Unavailable |
| RDU | Cancel – Reschedule date unknown |
| RNS | Cancel – Reschedule within next season |
| WCD | Cancel – Weather Conditions |
| WNR | Cancel – Will not reschedule |

Outage cancel notes can be supplied using following XML elements (as specified in table) for an individual outage or a group outage. The notes specified in the table are optional and thus not required on a cancel outage request.  However, if these notes are included, they must be sent using the payload element in the request as defined by OutageCancel element in ErcotOutages.xsd.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Data type* | *Description* | *Values* |
| OutageCancel/userFullName | Y | string |  | UserFullName is required on a Submit, Update and Cancel. Note: the element is defined as optional in the XSD. |
| OutageCancel/OSNotes/RequestorNotes/Note/createdTime | N | string | There are three sections that the requestor can enter notes, Requestor Notes, Supporting Information, and Remedial Action or Special Protection System notes. |  |
| OutageCancel/OSNotes/RequestorNotes/Note/createdBy | N | string |  |
| OutageCancel/OSNotes/RequestorNotes/Note/company | N | string |  |
| OutageCancel/OSNotes/RequestorNotes/Note/comment | N | string |  |
| OutageCancel/OSNotes/SupportingNotes/Note/createdTime | N | string |  |
| OutageCancel/OSNotes/SupportingNotes/Note/createdBy | N | string |  |
| OutageCancel/OSNotes/SupportingNotes/Note/company | N | string |  |
| OutageCancel/OSNotes/SupportingNotes/Note/comment | N | string |  |
| OutageCancel/OSNotes/RASPSNotes/Note/createdTime | N | string |  |
| OutageCancel/OSNotes/RASPSNotes/Note/createdBy | N | string |  |
| OutageCancel/OSNotes/RASPSNotes/Note/company | N | string |  |
| OutageCancel/OSNotes/RASPSNotes/Note/comment | N | string |  |

The following is an example for a Cancel OutageSet request. This example includes the entire RequestMessage structure.

<RequestMessage xmlns="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<Header>

<Verb>cancel</Verb>

<Noun>OutageSet</Noun>

<ReplayDetection>

<Nonce>-123456789</Nonce>

<Created>2016-09-09T14:00:10.425-05:00</Created>

</ReplayDetection>

<Revision>1</Revision>

<Source>TABC</Source>

<UserID>asmith</UserID>

<MessageID>987654321</MessageID>

</Header>

<Request>

<ID>TABC.OTG.PL.Transmission.ABC00118441.WCD</ID>

</Request>

<Payload>

<ns2:OutageCancel xmlns="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:ns2="http://www.ercot.com/schema/2007-06/nodal/ews">

<userFullName>Alex Smith</userFullName>

<ns2:OSNotes>

<ns2:RequestorNotes>

<ns2:Note>

<ns2:createdTime>2016-09-09T14:00:10-05:00</ns2:createdTime>

<ns2:createdBy>asmith</ns2:createdBy>

<ns2:company>TABC</ns2:company>

<ns2:comment>example comment</ns2:comment>

</ns2:Note>

</ns2:RequestorNotes>

</ns2:OSNotes>

</ns2:OutageCancel>

</Payload>

</RequestMessage>

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | Reply |
| Header/Noun | OutageSet |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *Error message, if error encountered* |

Figure 180 - Outage Cancel Response

# Utility Interfaces

The purpose of this section is to describe interfaces that provide utility services that are supplemental to the interfaces described in previous sections.

## Interfaces Provided

The interfaces provided include the following:

* The means to status check the ERCOT web services
* The means to generate an mRID from a set of key values

Additional utility interfaces may be defined in the future as needed.

## Message Specifications

### Get mRID

The purpose of this interface is to provide the means to obtain an mRID value that corresponds to a given set of key values for a market product. The generated mRID values are structured according to the rules in section 2.3.4. Hours are not included in the generated mRID strings.

The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | mRID |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Payload | *A BidSet, OutageSet\*\*, ResParametersSet, or VDIs that provides the ‘key’ information required for a specific type of market product as needed to construct the desired mRID* |

Figure 181 - Get mRID Request

\*\*Please Note that Outage Ident, the 5th token of OutageSet mRID, is assigned by Outage Scheduler and will not be returned by this interface and will be reflected as following:

<QSEID>.OTG.<outageType>.<outageCategory>.[outageIdent]

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | mRID |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR* |
| Reply/Error | *Error message, if error encountered* |
| Reply/ID | *mRID string* |

Figure 182 - Get mRID Response

### System Status

The purpose of this interface is to provide the means to verify that a connection can be established with the ERCOT web services through a simple status check. The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | SystemStatus |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |

Figure 183 - SystemStatus Request

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | SystemStatus |
| Header/Source | ERCOT |
| Header/Revision | Identifies current version of interfaces |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Identifies current ERCOT local time* |

Figure 184 - SystemStatus Response

### Request Test Notification

The purpose of this interface is to provide the means to request that a test notification be sent back to the MP. The payload provided on the request will then be used as the notification payload. The following table describes the request parameters:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | create |
| Header/Noun | TestNotification |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Payload/any | *Any message payload* |

Figure 185 - TestNotification Request

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | TestNotification |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Identifies current ERCOT local time* |

Figure 186 - TestNotification Response

### Change Active Notification URL

The purpose of this interface is to provide the means to request that ERCOT send notifications to the defined PRIMARY or BACKUP URL. This interface does NOT allow a specific URL to be specified. The following table describes the request parameters:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | Change |
| Header/Noun | ActiveNotificationURL |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/Option | *See table below for a list of values* |

Figure 187 – Change ActiveNotificationURL Request

|  |  |
| --- | --- |
| Option Value | Description |
| Primary | Deprecated - will remove in a future release  Change Public URL to Primary |
| Backup | Deprecated - will remove in a future release  Change Public URL to Secondary |
| PublicPrimary | Change Public URL to Primary |
| PublicBackup | Change Public URL to Secondary |
| WANPrimary | Change WAN URL to Primary |
| WANBackup | Change WAN URL to Secondary |
| AllPrimary | Change both Public and WAN URLs to Primary |
| AllBackup | Change both Public and WAN URLs to Secondary |

Figure 188 – Accepted values for the Request/Option element

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | ActiveNotificationURL |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Identifies current ERCOT local time* |

Figure 189 – Change ActiveNotificationURL Response

### Get SASM ID List

The purpose of this interface is to provide the means to request that ERCOT send a list of SASM Market IDs for a given trading date. The following table describes the request parameters:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | Get |
| Header/Noun | SASMIDList |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/TradingDate | *Trading Date* |

Figure 190 – Get SASM ID Request

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | SASMIDList |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR* |
| Reply/Error | *Error message, if error encountered* |
| Reply/Timestamp | *Identifies current ERCOT local time* |
| Reply/ID | *SASM IDs found for the requested Trading Date*  *Format: YYYYMMDDHHMMSS* |

Figure 191 – Get SASM ID Response

# Reports

The interfaces provided include the following:

* The means to request the URLs to a set of reports for a given operating date using web services

The actual download of the documents themselves would continue to be done using the URL that is returned for each report.

## Message Specifications

### Get Reports

The purpose of this interface is to provide the means to obtain a list of URLs for reports related to a specific date range, and Report Id. The URLs can then be used for downloading the reports.

The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | Reports |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/StartTime | *Start Date & time of interest (Optional) \** |
| Request/EndTime | *End Date & time of interest (Optional) \** |
| Request/Option | *Report ID (Required). \** |

Note: \* Following are the valid query combinations for getReports

* + Option & (StartTime & EndTime) – All reports for the requested reportID and time duration.
  + Option & StartTime – All reports starting from StartTime till now (currentDateTime) for the requested reportID.
  + Option & EndTime – All available reports till EndTime for the requested reportID.
  + Only Option – All the available reports for the requested reportID.

Figure 192 - Reports Request

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | Reports |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR* |
| Reply/Error | *Error message, if error encountered* |
| Payload/Reports | *List of Report structures, where each Report identifies a number of data items including the URL where the actual report may be downloaded* |

Figure 193 - Reports Response

The following structure is used within the payload to convey a list of reports:

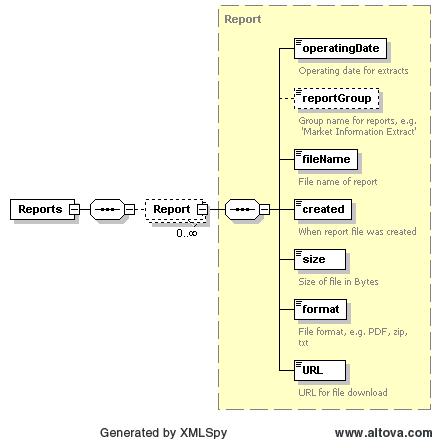


Figure 194 - Reports Container

|  |  |  |  |
| --- | --- | --- | --- |
| *Element* | *Data type* | *Description* | *Values* |
| operatingDate | date | Operating date for extracts |  |
| reportGroup | string | Group name for reports |  |
| filename | String | Report file name |  |
| Created | dateTime | Date of report creation. |  |
| Size | Integer | File size in Bytes |  |
| Format | string | Report file format |  |
| URL | string | URL for http download |  |

Example:

<ns1:Reports xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:Report>

<ns1:operatingDate>2009-12-16</ns1:operatingDate>

<ns1:reportGroup>60 Day Disclosure</ns1:reportGroup>

<ns1:fileName>rpt.00010029.0000000000000000.20091216.093316.dummy\_file.zip</ns1:fileName>

<ns1:created>2009-12-16T09:33:16</ns1:created>

<ns1:size>1412</ns1:size>

<ns1:format>zip</ns1:format>

<ns1:URL>https://<Host>:<Port>/<folder>…./?doclookupId=32423</ns1:URL>

</ns1:Report>

…

…

</ns1:Reports>

# Resource Parameter Transaction Service

Each Market Participant can provide directions (using a ResParametersSet) to ERCOT on any changes to resource parameters.  Any changes submitted become effective upon completion and validation of the submission for all applicable markets.  When submitted, ERCOT will validate the submission, reporting format or submission errors to Market Participants using notification messages.

## Interfaces Provided

The interfaces provide the means to create (i.e. change), get (i.e. query), and cancel (i.e. delete) resource parameters for the next trading date.  A single container class ‘ResParametersSet’ is used to hold a request for changing resource parameters within the Payload section of the message, where each of the requests may be of a different type.

The following diagram shows an example message sequence, using the ‘verb’ and ‘noun’ convention. Where this section focuses on the requests made by Market Participant systems to the ERCOT Nodal Web Services, the sequence diagram also includes notification messages sent from ERCOT to Market Participant Notification services (as described in section 5):



Figure 195 - Example ResParametersSet Sequence Diagram

The message sequence example shown involves the following steps:

1. Market participant sends a RequestMessage for ‘change ResParametersSet’ with an initial ResParametersSet in the payload to ERCOT.
2. In response to step 1, ERCOT performs a simple syntax scan and typically sends a ResponseMessage with ReplyCode=OK. In the response payload, each ResParametersSet will identify a ‘SUBMITTED’ status and an mRID value. (An alternative example could result in a reply of ‘ERRORS’ if the syntax check failed, in which case steps 3 and 4 would not occur). This reply is synchronous.
3. ERCOT validates the Resource Parameters Set request within the ResParametersSet. This could take several minutes. This processing is done asynchronously.
4. A notification message (using verb=changed) is sent to the notification interface provided by the Market Participant. The status of the Resource Parameters change requests within the ResParametersSet will indicate whether the request was PENDING/ACCEPTED or had ERRORS. This message will not include the complete ResParametersSet.

## Interfaces Required

### Querying for Resource Parameters Set

The following table describes the parameters used in the “get” request message (RequestMessage) for market information, noting that each transaction has a request and a response message. The verb “get” is used to query for ResParametersSet.

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | *get* |
| Header/Noun | ResParametersSet |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/ID | ResParametersSet mRID:  Full:QSEID.<ResParamatersTypeCode>.<resource>  Short:QSEID.<ResParamatersTypeCode> |

Figure 196 - Message: Request for a get ResParametersSet

The corresponding response messages (ResponseMessage) would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | ResParametersSet |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *May be any number of error message if the ReplyCode is ERROR* |
| Reply/Timestamp | *The time the submission was received by ERCOT* |
| Payload | ResParametersSet |

Figure 197 - Message: reply to get ResParametersSet request

In the cases of payloads that would otherwise exceed 1 megabyte, the payloads should be zipped, base64 encoded and stored within the ‘Payload/Compressed’ tag.

### Updating Resource Parameters Set

The following table describes the parameters used in the “change” request message (RequestMessage) for market transaction, noting that each transaction has a request and a response message. The verbs “change” is used to submit update requests to for ResParametersSet.

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | *change* |
| Header/Noun | ResParametersSet |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Payload | ResParametersSet (a single ResParametersSet may be used for update request messages |

Figure 198 - Message: Request for a change ResParametersSet

The corresponding response messages (ResponseMessage) would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | ResParametersSet |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *May be any number of error message if the ReplyCode is ERROR* |
| Reply/Timestamp | *The time the submission was received by ERCOT* |
| Payload | ResParametersSet, where the status of each Res Par Change Request within the ResParametersSet may be SUBMITTED, ACCEPTED, or ERRORS. Specific errors may be identified for each Res Parameter Change Request. |

Figure 199 - Message: reply to change ResParametersSet request

In the cases of payloads that would otherwise exceed 1 megabyte, the payloads should be zipped, base64 encoded and stored within the ‘Payload/Compressed’ tag.

For the purposes of ResParametersSet, the verbs create and change can be used interchangeably. Note that only one ResParametersSet “change” request is permitted for a given message.

### Canceling Resource Parameters Set

The following table describes the parameters used in the “cancel” request message (RequestMessage) for market transaction, noting that each transaction has a request and a response message. The verbs “cancel” is used to delete (cancel) a previously submitted “change” ResParametersSet.

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | *cancel* |
| Header/Noun | ResParametersSet |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/ID | ResParametersSet full mRID:  QSEID.<ResParamatersTypeCode>.<resource> |

Figure 200 - Message: Request for cancel a ResParametersSet

The corresponding response messages (ResponseMessage) would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | ResParametersSet |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *May be any number of error message if the ReplyCode is ERROR* |
| Reply/Timestamp | *The time the submission was received by ERCOT* |
| Payload | *ResParametersSet, where the status of each Res Par Change Request within the ResParametersSet may be CANCELED or ERRORS. Specific errors may be identified for each Res Parameter cancel Request.* |

Figure 201 - Message: reply ResParametersSet

In the cases of payloads that would otherwise exceed 1 megabyte, the payloads should be zipped, base64 encoded and stored within the ‘Payload/Compressed’ tag.

## Message Specifications

A ResParametersSet is the payload type used for the submission and query of Resource Parameter Set requests, and serves as a container for the different types of ResParametersSet requests that are submitted by a QSE.

An important note is that the ResParametersSet uses an XSD choice, requiring one type of ResParametersSet to be provided at a time.

In order to query a ResParametersSet, A ResParametersSet is sent using the ‘get’ verb, where the desired Resource Parameter requests for the specific ResParametersSet is identified.

When a ResParametersSet is returned by a ‘get’ request, the status value of each in the ResParametersSet is populated. Values could include:

* + SUBMITTED (to indicate submission, but no further processing has been completed)
  + ACCEPTED (to indicate successful validation and acceptance of the transaction)
  + ERRORS (to indicate that there are one or more errors for the transaction)

The following sub sections describe the structure of specific ResParameters types.

When submitting a ResParametersSet request using change, all properties for the ResParametersSet must be specified. When performing a get request, only those parameters that uniquely identify the ResParameters must be specified through the use of an mRID. The following diagram shows information commonly maintained for each type of ResParameters: The status for a given ResParameters may be SUBMITTED, ACCEPTED, or ERRORS. If the status is ERRORS, there may be one or more error stings identified.

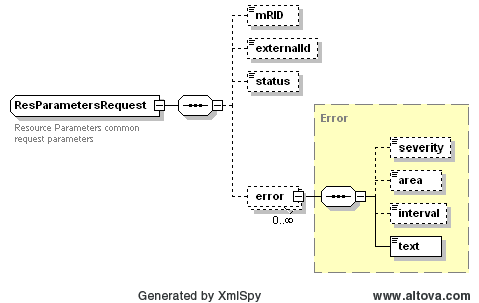


Figure 202 – ResParametersRequest Common Elements

An mRID is constructed and returned in the payload of the response as part of the ResParametersRequest structure in the following format

*QSEID.<ResParamatersTypeCode>.<resource>*

Where ResParamatersType can be:

|  |  |
| --- | --- |
| ResParamatersType | ResParamatersTypeCode |
| GenResourceParameters | GEN |
| ControllableLoadResource | CON |
| NonControllableLoadResource | NON |
| ResourceParameters | RES |

The mRID is not supplied for the initial submission of a bid, but must be supplied for query for a previously submitted ResParameters. When an mRID is specified for a ‘get’ request, it is supplied using a message request/ID tag.

Query by short-mRID is also supported in the following format (without the resource):

*QSEID.<ResParamatersTypeCode>*

Invoking queries on these web service (via “get”) using short mRID, returns Resource Parameters for all the resources that belong to the requesting QSE for a given ResParamatersType.

The ‘externalId’ may be populated by the QSE with an identifier of their choice. If supplied upon submission, the identifier will then be used in conjunction with notifications of acceptance or rejection due to errors.

### Generator Resource Parameters

The following diagram defines the structure of a Generator Resource Parameters that could be included within a ResParametersSet, using the GenResourceParameters tag:

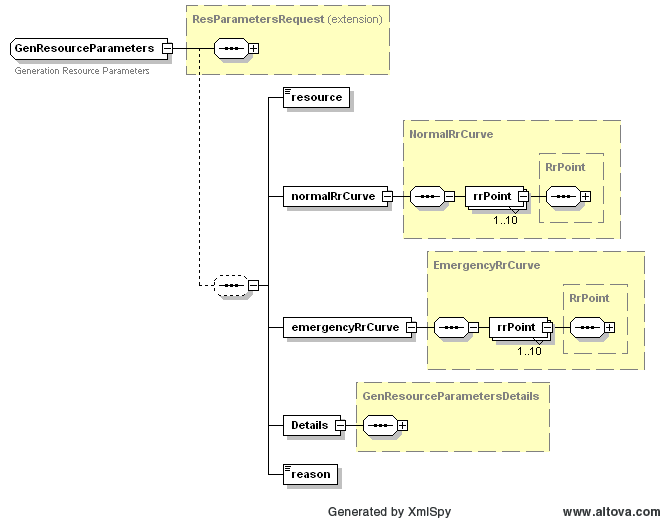
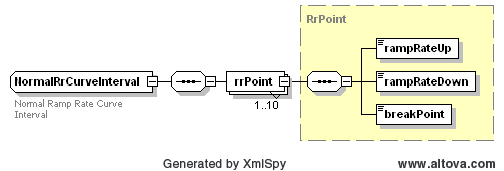


Figure 203 – Generator Resource Parameters Structure



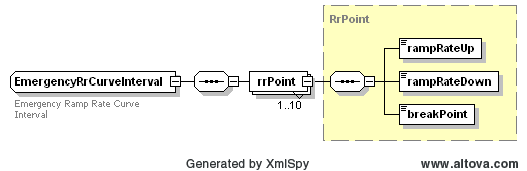


Figure 204 – Normal and Emergency Curve Structures

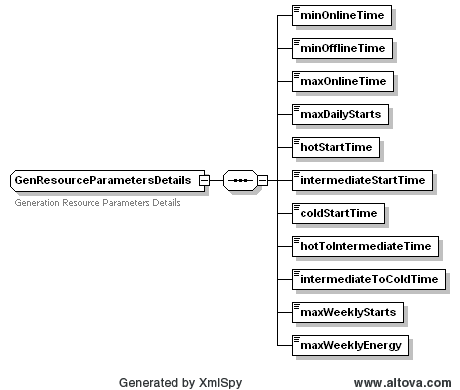


Figure 205 – Generator Resource Parameters Details Structures

The error tag is used to return one or more errors that may be the consequence of the failure of business or syntax validation rules for Generator Resource Parameter change submittals.

On submission, the following table describes the items used for a ResParametersSet change request.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| mRID | N | String | ERCOT assigned |  |
| externalId | N | String | External ID | QSE supplied |
| status | N | String | Return status | SUBMITTED, ACCEPTED, or ERRORS |
| Error/severity | N | String | Error if any | Error text |
| Error/area | N | String | Error if any | Error text |
| Error/interval | N | String | Error if any | Error text |
| Error/text | N | String | Error if any | Error text |
| resource | Y | String | Resource name |  |
| normalRrCurve/rrPoint/rampRateUp | Y | float | MW/min |  |
| normalRrCurve/rrPoint/rampRateDown | Y | float | MW/min |  |
| normalRrCurve/rrPoint/breakPoint | Y | float |  |  |
| emergencyRrCurve/rrPoint/rampRateUp | Y | float | MW/min |  |
| emergencyRrCurve/rrPoint/rampRateDown | Y | float | MW/min |  |
| emergencyRrCurve/rrPoint/breakPoint | Y | float |  |  |
| Details/minOnlineTime | Y | float |  | time in hours |
| Details/minOfflineTime | Y | float |  | time in hours |
| Details/maxOnlineTime | Y | float |  | time in hours |
| Details/maxDailyStarts | Y | Integer |  | time in hours |
| Details/hotStartTime | Y | float |  | time in hours |
| Details/intermediateStartTime | Y | float |  | time in hours |
| Details /coldStartTime | Y | float |  | time in hours |
| Details /hotToIntermediateTime | Y | float |  | time in hours |
| Details / intermediateToColdTime | Y | float |  | time in hours |
| Details /maxWeeklyStarts | Y | Integer |  |  |
| Details /maxWeeklyEnergy | Y | Integer | max amount of energy a Resource is allowed to produce per week | MWh value |
| Details /reason | Y | String | text reason for changing parameters | Description in plain text |

Figure 206 – GenResourceParameters Requirements

The following is an XML example of a GenResourceParameters:

<GenResourceParameters>

<mRID>String</mRID>

<externalId>String</externalId>

<status>SUBMITTED</status>

<error>

<severity>ERROR</severity>

<area>String</area>

<interval>String</interval>

<text>String</text>

</error>

<resource>String</resource>

<normalRrCurve>

<rrPoint>

<rampRateUp>0</rampRateUp>

<rampRateDown>0</rampRateDown>

<breakPoint>0</breakPoint>

</rrPoint>

</normalRrCurve>

<emergencyRrCurve>

<rrPoint>

<rampRateUp>0</rampRateUp>

<rampRateDown>0</rampRateDown>

<breakPoint>0</breakPoint>

</rrPoint>

</emergencyRrCurve>

<Details>

<minOnlineTime>0</minOnlineTime>

<minOfflineTime>0</minOfflineTime>

<maxOnlineTime>0</maxOnlineTime>

<maxDailyStarts>0</maxDailyStarts>

<hotStartTime>0</hotStartTime>

<intermediateStartTime>0</intermediateStartTime>

<coldStartTime>0</coldStartTime>

<hotToIntermediateTime>0</hotToIntermediateTime>

<intermediateToColdTime>0</intermediateToColdTime>

<maxWeeklyStarts>0</maxWeeklyStarts>

<maxWeeklyEnergy>0</maxWeeklyEnergy>

</Details>

<reason>String</reason>

</GenResourceParameters>

### Controllable Load Resource

The following diagram defines the structure of a Controllable Load Resource that could be included within a ResParametersSet, using the ControllableLoadResource tag:

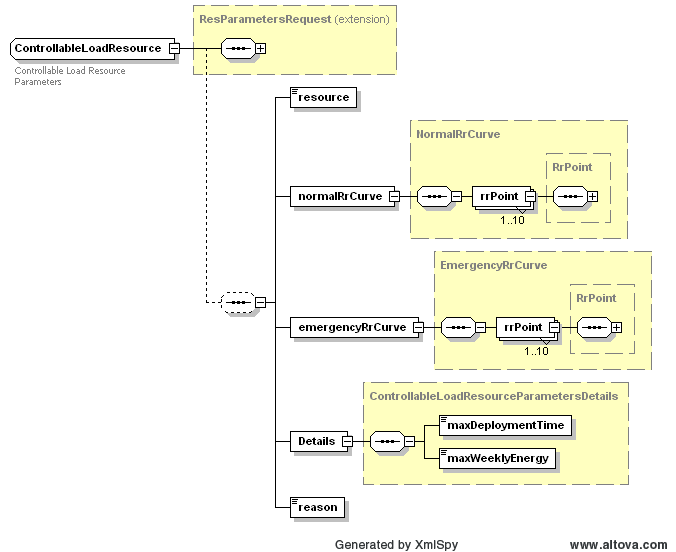
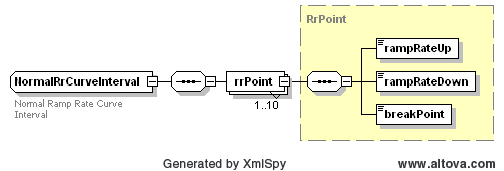


Figure 207 – Controllable Load Resource Structure



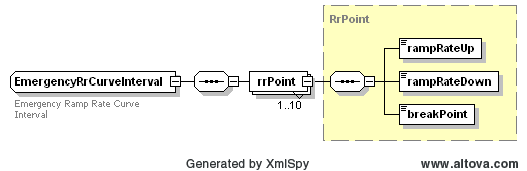


Figure 208 – Normal and Emergency Curve Structures

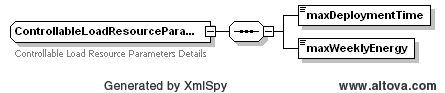


Figure 209 – Controllable Load Resource Parameters Details Structures

The error tag is used to return one or more errors that may be the consequence of the failure of business or syntax validation rules for Controllable Load Resource submittals.

On submission, the following table describes the items used for a Controllable Load Resource change request.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element | Req? | Datatype | Description | Values |
| mRID | N | String | ERCOT assigned |  |
| ExternalId | N | String | External ID | QSE supplied |
| status | N | String | Return status | SUBMITTED, ACCEPTED, or ERRORS |
| Error/severity | N | String | Error if any | Error text |
| Error/area | N | String | Error if any | Error text |
| Error/interval | N | String | Error if any | Error text |
| Error/text | N | String | Error if any | Error text |
| resource | N | String | Resource name |  |
| normalRrCurve/rrPoint/rampRateUp | Y | float | MW/min |  |
| normalRrCurve/rrPoint/rampRateDown | Y | float | MW/min |  |
| normalRrCurve/rrPoint/breakPoint | Y | float |  |  |
| emergencyRrCurve/rrPoint/rampRateUp | Y | float | MW/min |  |
| emergencyRrCurve/rrPoint/rampRateDown | Y | float | MW/min |  |
| emergencyRrCurve /rrPoint/breakPoint | Y | float |  |  |
| Details/maxDeploymentTime | Y | float |  | time in hours |
| Details/maxWeeklyEnergy | Y | Integer | max amount of energy a Resource is allowed to produce per week | MWh value |
| Details /reason | Y | String | text reason for changing parameters | Description in plain text |

Figure 210 – Controllable Load Resource Requirements

The following is an XML example of a Controllable Load Resource:

<ControllableLoadResource>

<mRID>String</mRID>

<externalId>String</externalId>

<status>SUBMITTED</status>

<error>

<severity>ERROR</severity>

<area>String</area>

<interval>String</interval>

<text>String</text>

</error>

<resource>String</resource>

<normalRrCurve>

<rrPoint>

<rampRateUp>0</rampRateUp>

<rampRateDown>0</rampRateDown>

<breakPoint>0</breakPoint>

</rrPoint>

</normalRrCurve>

<emergencyRrCurve>

<rrPoint>

<rampRateUp>0</rampRateUp>

<rampRateDown>0</rampRateDown>

<breakPoint>0</breakPoint>

</rrPoint>

</emergencyRrCurve>

<Details>

<maxDeploymentTime>0</maxDeploymentTime>

<maxWeeklyEnergy>0</maxWeeklyEnergy>

</Details>

<reason>String</reason>

</ControllableLoadResource>

### Non Controllable Load Resource

The following diagram defines the structure of a Non-Controllable Load Resource that could be included within a ResParametersSet, using the Non-ControllableLoadResource tag:

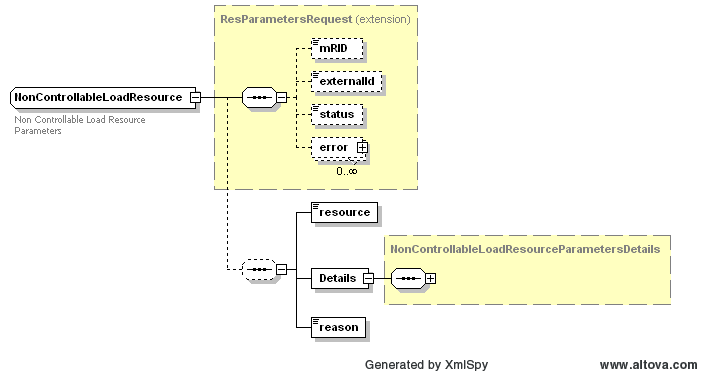


Figure 211 – Non Controllable Load Resource Structure

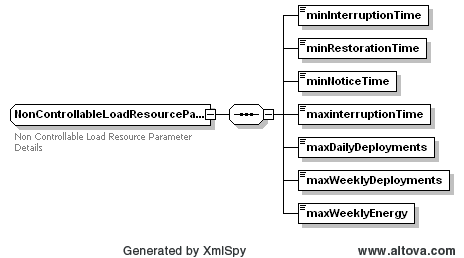


Figure 212 – Non Controllable Load Resource Parameters Details Structures

The error tag is used to return one or more errors that may be the consequence of the failure of business or syntax validation rules for NonControllableLoadResource submittals.

On submission, the following table describes the items used for a Non-Controllable Load Resource change request.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| mRID | N | String | ERCOT assigned |  |
| externalId | N | String | External ID | QSE supplied |
| status | N | String | Return status | SUBMITTED, ACCEPTED, or ERRORS |
| Error/severity | N | String | Error if any | Error text |
| Error/area | N | String | Error if any | Error text |
| Error/interval | N | String | Error if any | Error text |
| Error/text | N | String | Error if any | Error text |
| resource | N | String | Resource name |  |
| Details/minInterruptionTime | Y | float |  | time in hours |
| Details/minRestorationTime | Y | float |  | time in hours |
| Details/minNoticeTime | Y | float |  | time in hours |
| Details/maxInterruptionTime | Y | float |  | time in hours |
| Details/maxDailyDeployments | Y | Integer |  | Number |
| Details/maxWeeklyDeployments | Y | Integer |  | Number |
| Details /maxWeeklyEnergy | Y | Integer | max amount of energy a Resource is allowed to produce per week | MWh value |
| Details /reason | Y | String | text reason for changing parameters | Description in plain text |

Figure 213 – NonControllableLoadResource Requirements

The following is an XML example of a Non ControllableLoadResource:

<NonControllableLoadResource>

<mRID>String</mRID>

<externalId>String</externalId>

<status>SUBMITTED</status>

<error>

<severity>ERROR</severity>

<area>String</area>

<interval>String</interval>

<text>String</text>

</error>

<resource>String</resource>

<Details>

<minInterruptionTime>0</minInterruptionTime>

<minRestorationTime>0</minRestorationTime>

<minNoticeTime>0</minNoticeTime>

<maxInterruptionTime>0</maxInterruptionTime>

<maxDailyDeployments>0</maxDailyDeployments>

<maxWeeklyDeployments>0</maxWeeklyDeployments>

<maxWeeklyEnergy>0</maxWeeklyEnergy>

</Details>

<reason>String</reason>

</NonControllableLoadResource>

### Resource Parameters

The following diagram defines the structure of a Resource Parameters that could be included within a ResParametersSet, using the ResourceParameters tag:

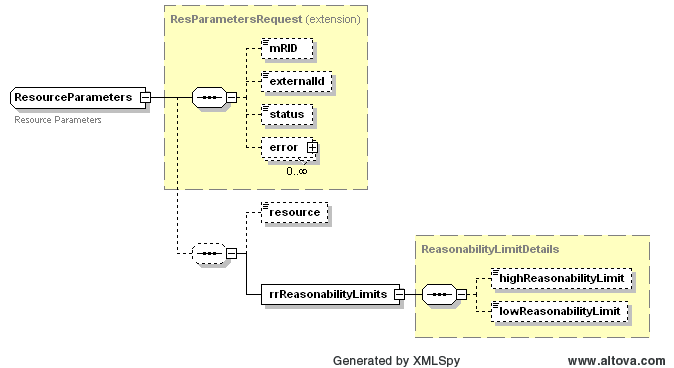


Figure 214 – Resource Parameters Structure

The error tag is used to return one or more errors that may be the consequence of the failure of business or syntax validation rules for Resource Parameter change submittals.

Please note that the Reasonability Limits are not changeable via this interface and this interface is only used for retrieval purposes. The following items are returned in a ResParametersSet “get” request.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| mRID | N | String | ERCOT assigned |  |
| externalId | N | String | External ID | QSE supplied |
| status | N | String | Return status | SUBMITTED, ACCEPTED, or ERRORS |
| Error/severity | N | String | Error if any | Error text |
| Error/area | N | String | Error if any | Error text |
| Error/interval | N | String | Error if any | Error text |
| Error/text | N | String | Error if any | Error text |
| resource | Y | String | Resource name |  |
| rrreasonabilityLimits/ highReasonabilityLimit | N | Float | High Reasonability Limit | Single Point Precision |
| rrreasonabilityLimits/ LowReasonabilityLimit | N | Float | Low Reasonability Limit | Single Point Precision |

Figure 215 – ResourceParameters Requirements

The following is an XML example of a ResourceParameters:

<ResourceParameters>

<mRID>String</mRID>

<externalId>String</externalId>

<status>SUBMITTED</status>

<error>

<severity>ERROR</severity>

<area>String</area>

<interval>String</interval>

<text>String</text>

</error>

<resource>String</resource>

<rrReasonabilityLimits>

<highReasonabilityLimit>3.1</highReasonabilityLimit>

<lowReasonabilityLimit>3.1</lowReasonabilityLimit>

</rrReasonabilityLimits>

</ResourceParameters>

# Verbal Dispatch Instructions

Each Market Participant can retrieve (via “get”) a Verbal Dispatch Instructions (VDIs) from ERCOT and acknowledge it (via “change”) VDIs request to ERCOT.

## Interfaces Provided

The interface provides the means to get (i.e. query) or acknowledge (i.e. change) Verbal Dispatch Instructions. A single container class ‘VDIs’ is used to hold a request for changing VDIs within the Payload section of the message.

The following diagram shows an example message sequence, using the ‘verb’ and ‘noun’ convention, where this section focuses on the requests made by Market Participant systems to the ERCOT Nodal Web Services. This transaction is a synchronous for “get” requests and asynchronous for “change” requests:



Figure 216 - Example Verbal Dispatch Instructions Sequence Diagram

The message sequence example shown involves the following steps:

1. Market participant sends a RequestMessage for ‘get ‘VDIs with a constructed mRID (explained in section 12.3 below).
2. In response to step 1, ERCOT performs a simple syntax scan and sends the request on to the Market Management System and passes the ResponseMessage with ReplyCode=OK back to the requester, holding the VDIs in response message payload. An alternative example could result in a reply of ‘ERRORS’ if the syntax checks failed. This reply is synchronous.
3. Market participant sends a RequestMessage for ‘change VDIs’ to ERCOT for a specific resource or “change VDIs” for multiple resources by having multiple VDIs filled in the payload.
4. In response to step 3, ERCOT performs a simple syntax scan and typically sends a ResponseMessage with ReplyCode=OK. In the response payload VDIs requests will identify a ‘SUBMITTED’ status. (An alternative example could result in a reply of ‘ERRORS’ if the syntax check failed, in which case steps 5 and 6 would not occur). This reply is synchronous.
5. ERCOT validates the VDIs request within the message payload. This processing is done asynchronously.
6. A notification message (using verb=reply) is sent to the notification interface provided by the Market Participant, stating “Acknowledged” in the reply part of the notification message.

## Interfaces Required

The following tables describes the parameters used in the request message (RequestMessage) for market information and transactions, noting that each transaction has a request and a response message. The verbs “get” and “change” is respectively used to retrieve and query Verbal Dispatch Instructions.

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | *get* |
| Header/Noun | VDIs |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Request/ID | mRID (*QSEID.VDI.<resource>)*  See section 12.3 |

Figure 217 – Request message: get VDIs

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | VDIs |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *May be any number of error message if the ReplyCode is ERROR* |
| Reply/Timestamp | *The time the submission was received by ERCOT* |
| Payload | Unacknowledged VDIs |

Figure 218 – Reply Message: reply to get VDIs

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | *change* |
| Header/Noun | VDIs |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Payload | VDIs (VDIs to be acknowledged) |

Figure 219 – Request Message: change VDI

The corresponding response messages (ResponseMessage) would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | VDIs |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR or FATAL* |
| Reply/Error | *May be any number of error message if the ReplyCode is ERROR* |
| Reply/Timestamp | *The time the submission was received by ERCOT* |
| Payload | VDIs, where the status of each VDI Change Request within the payload may be ACKNOWLEDGED, or ERRORS. Specific errors may be identified for each VDI Change Request. |

Figure 220 – Reply Message: reply VDIs

In the cases of payloads that would otherwise exceed 1 megabyte, the payloads should be zipped, base64 encoded and stored within the ‘Payload/Compressed’ tag.

For the purposes of VDIs, the verbs create and change can be used interchangeably.

## Message Specifications

VDIs is the payload type used for the submission of Verbal Dispatch Instructions requests and serves as a container for submitting an acknowledgement of a VDIs request that are submitted by a QSE.

In order to query VDIs, A request is sent using the ‘get’ verb, where the desired VDIs change requests for the specific resource is identified.

Once a VDI is acknowledged, the queries will no longer return that VDI. If all the VDIs have been acknowledged, the query will not return anything.

The following subsection describes the structure of Verbal Dispatch Instructions.

When submitting a VDI request using change, all properties for that VDI must be specified. When performing a get request, only those parameters that uniquely identify the VDI must be specified through the use of an mRID.

An mRID should be manually constructed and used in the Message Request as part of the VDIs “get” requests in the following format

*QSEID.VDI.<resource>*

An mRID is not supplied for the submission of a VDI change (acknowledge) request, but must be supplied for query for a particular resource. When an mRID is specified for a ‘get’ request, it is supplied using a message request/ID tag.

Query by short-mRID is also supported in the following format (without the resource):

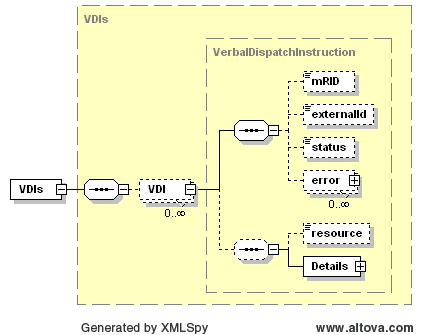
*QSEID.VDI*

Invoking queries (via “get”) using short mRID, returns VDIs for all the resources that belong to the requesting QSE.

The ‘externalId’ may be populated by the QSE with an identifier of their choice. If supplied upon submission, the identifier will then be used in conjunction with notifications of acceptance or rejection due to errors.

### Verbal Dispatch Instructions Structure

The following diagram defines the structure of a Verbal Dispatch Instructions:



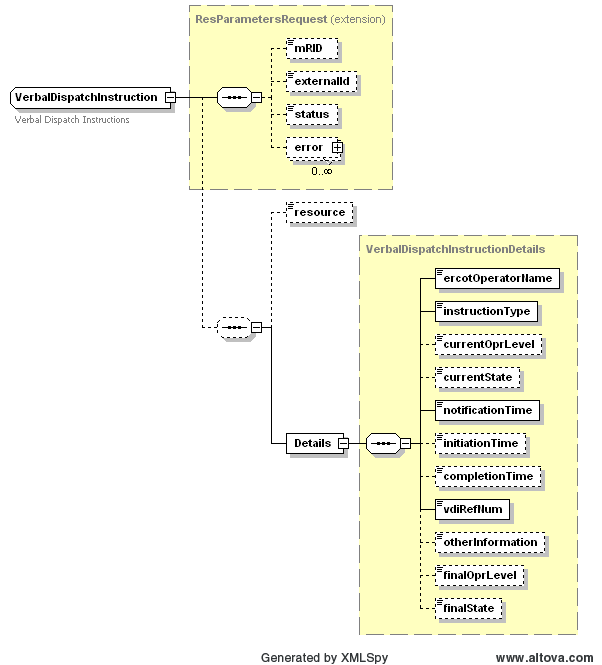


Figure 221 – VerbalDispatchInstruction Structure

The error tag is used to return one or more errors that may be the consequence of the failure of business or syntax validation rules for VerbalDispatchInstruction submittals.

On submission, the following table describes the items used for a VDI change request.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| mRID | N | String | ERCOT assigned |  |
| externalId | N | String | External ID | QSE supplied |
| status | N | String | Return status | SUBMITTED, ACKNOWLEDGED, or ERRORS |
| Error/severity | N | String | Error if any | Error text |
| Error/area | N | String | Error if any | Error text |
| Error/interval | N | String | Error if any | Error text |
| Error/text | N | String | Error if any | Error text |
| resource | N | String | Resource name | text |
| Details/ercotOperartorName | Y | String |  | text |
| Details/instructionType | Y | String | description/category of instruction | free text field |
| Details/currentOprLevel | N | Float | Current MW value for Resource is to be deployed to | MW value to one decimal |
| Details/currentState | N | String | current state of the Resource, | off/on, etc.  free text field |
| Details/notificationTime | Y | DateTime |  | DateTime |
| Details/initiationTime | N | DateTime |  | DateTime |
| Details/completionTime | N | DateTime |  | DateTime |
| Details/vdiRefNum | Y | string | VDI reference | Text |
| Details/otherInformation | N | string | Other Information | Text |
| Details/finalOprlevel | N | Float | Final MW value for Resource is to be deployed to | MW value to one decimal |
| Detail/finalState | N | string | final state of the Resource, | off/on, etc.  free text field |

Figure 222 – VerbalDispatchInstruction Requirements

IntrustionType values

|  |  |
| --- | --- |
| Instruction Type | Instruction Description |
| AB\_PQ\_LEV | STAY AT/ABOVE X MW/MVAR LEVEL |
| ALT\_FUEL | BEGIN ALTERNATIVE FUEL USAGE |
| AT\_PQ\_LEV | STAY AT X MW/MVAR LEVEL |
| AT\_PRE\_BP | BE AT PREVIOUS BP |
| AT\_VOL\_LEV | HOLD X VOLTAGE LEVEL |
| BE\_PQ\_LEV | STAY AT/BELOW X MW/MVAR LEVEL |
| BLT | IMPLEMENT BLT DURING EMERGENCY |
| CANNOT\_RET | CAN NOT RETURN EARLY FROM OUTAGE |
| CAN\_COM | CANCEL RUC COMMIT |
| COMMIT | COMMIT |
| CONS\_FREQ | OPERATE AT CONSTANT FREQUENCY |
| DC\_TIE | PROVIDE EMERGENCY POWER FROM DC TIE |
| DECOMMIT | DECOMMIT |
| OOME | FLEET OOME |
| OTHER\_QSE | OTHER FOR QSE |
| OTHER\_RES | OTHER FOR RESOURCE |
| OTHER\_TSP | OTHER FOR TSP |
| OTH\_ALLQSE | OTHER FOR ALL QSE |
| OTH\_ALLTSP | OTHER FOR ALL TSP |
| PROT\_UNIT | PROTECT UNIT AGAIN FREEZING CONDITIONSE |
| RAP\_MP | IMPLEMENT RAP/MP FOR LINE/BREAKER |
| REACT\_DEV | ENERGIZE/DE\_ENERGIZE REACTIVE DEVICE |
| RRS\_1\_2 | RRS DEPLOYMENT FROM GROUP 1 & 2 |
| RRS\_GR\_1 | RRS DEPLOYMENT FROM GROUP 1 |
| RRS\_GR\_2 | RRS DEPLOYMENT FROM GROUP 2 |
| RRS\_NCLR | RRS DEPLOYMENT FROM NON-CONTROLLABLE LR |
| SHED\_BLCLD | SHED X MW LOAD IN 100 MW BLOCKS |
| SHED\_LOAD | SHED X MW LOAD IN SPECIFIC AREA/SUBSTATION |
| TAP\_CHG | TAP TRANSFORMER UP/DOWN |

The following is an XML example of a VDI:

<VDIs>

<VDI>

<mRID>QSE.VDI.Resource1</mRID>

<externalId></externalId>

<status>SUBMITTED</status>

<error>

<severity>INFORMATIVE</severity>

<text> Successfully retrieved the ERCOT VERBAL\_DISP\_INSTRUCT. </text>

</error>

<resource>Resource1</resource>

<Details>

<ercotOperatorName>Tester1</ercotOperatorName>

<instructionType>RAP\_MP</instructionType>

<currentOprLevel>3.1</currentOprLevel>

<currentState>OPEN</currentState>

<notificationTime>2009-09-24T15:05:31-05:00</notificationTime>

<initiationTime>2009-09-24T15:05:34-05:00</initiationTime>

<completionTime>2009-09-24T15:05:36-05:00</completionTime>

<vdiRefNum>689</vdiRefNum>

<otherInformation>Unacknowledged</otherInformation>

<finalState>CLOSE</finalState>

</Details>

<VDI>

<VDIs>

# Get Notifications

This interface is used to retrieve notifications for previously submitted market transactions for the nouns BidSet, ResParametersSet and VDIs. The notifications in question are those notifications that were delivered asynchronously to market participant listeners as the result of create or change transactions for the aforementioned nouns (BidSet, ResParametersSet and VDIs).  The interface is synchronous in nature and subject to the below specifications.

Key Specifications:

* Service supports querying historical notifications upto 4 days old and is based on transaction submitted time. “Transaction submitted time” represents the time when the original create or change transaction was submitted to ERCOT.  It is not the TradingDate on the submission nor the StartTime/EndTime of an interval in the submission.
* A single query cannot span more than 24hours. Specifically, the StartTime and EndTime sent on the query may not exceeded 24hours.
* Notifications can be retrieved using combinations of following query parameters
  + - 1. mRID(s)
      2. BidType
      3. Bid Processed Status (ACCEPTED/ERROR)

StartTime and EndTime, which represent the submitted times that the query will span, are required with above parameter combinations.

A maximum of 1000 notifications will be returned from a single request.  This threshold will also be subject to payload size limitations.  If the resulting response exceeds 3MB after compression, an error will be returned indicating that the query parameters need to be refined (i.e. further constrained).

If a query by MRID is received and ERCOT determines that the MRID in question exists within a multi-bid notification (i.e. a notification that has two or more Bids and thus potentially two or more MRIDs), the entire multi-bid notification will be returned in the response.  ERCOT will not filter the notification and only return the bid that corresponds with the MRID in question.

Similarly, if a query by Bid Processed Status is received and ERCOT determines that a single notification has both accepted and rejected bids, the entire notification will be returned.  ERCOT will not filter the notification and only return the bid that corresponds with the Bid Processed Status in question.

* Following are the valid query parameter combinations
  + getNotifications by BidType
    - StartTime & EndTime
    - BidType
    - Bid Processed Status (optional)
  + getNotifications by mRID
    - StartTime & EndTime
    - One or more mRIDs
    - Bid Processed Status (optional)

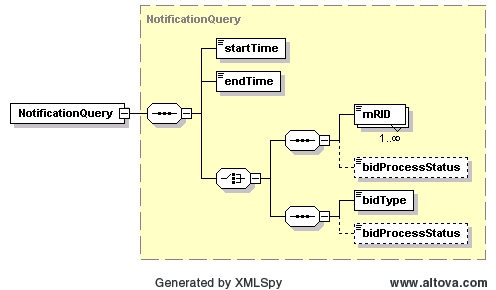
## Message Specifications

### Get Notifications

The request message would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | get |
| Header/Noun | BidSetNotifications/ ResParameterSetNotifications/ VDIsNotifications |
| Header/Source | *Market participant ID* |
| Header/UserID | *ID of user* |
| Payload/any | *Get Notification request Payload[Query payload should be mapped to any element}* |

Request Query Payload is as follows



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Element* | *Req?* | *Datatype* | *Description* | *Values* |
| startTime | Y | dateTime | Start date time of query | Valid date time |
| endTime | Y | dateTime | End date time of query | Valid date time |
| mRID | Y | String | mRID to query for notification.  - This element is required only for mRID based query. |  |
| bidProcessStatus | N | String | Bid Process Status (ACCEPTED or ERROR) |  |
| bidType | Y | String [ enumerate list] | BidType (refer to below list for valid bid types)  - This element is required only for BidType based query. |  |

Valid BidTypes for Noun =” BidSetNotifications”

| Transaction type | BidType Value |
| --- | --- |
| ASOffer | ASO |
| ASTrade | AST |
| CapacityTrade | CT |
| COP | COP |
| PTP Obligation w/ Link to Option | CRR |
| EnergyBid | EB |
| EnergyOnlyOffer | EOO |
| EnergyTrade | ET |
| IncDecOffer | IDO |
| OutputSchedule | OS |
| PTPObligation | PTP |
| SelfArrangedAS | SAA |
| SelfSchedule | SS |
| ThreePartOffer | TPO |
| AvailabilityPlan | AVP |
| RTMEnergyBid | REB |
| ExceptionalFuelCost | EFC |

For Noun=ResParameterSetNotifications

|  |  |
| --- | --- |
| Transaction type | BidType Value |
| GenResourceParameters | GEN |
| ControllableLoadResource | CON |
| NonControllableLoadResource | NON |
| ResourceParameters | RES |

For Noun=VDIsNotifications

|  |  |
| --- | --- |
| Transaction type | BidType Value |
| VDIs | VDI |

Valid mRID formats for getNotifications by mRID(s)

For noun=BidSetNotifications

| *BidType* | *mRID format* |
| --- | --- |
| ASOffer | QSEID.Date.ASO.resource.asType |
| ASTrade | QSEID.Date.AST.asType.buyer.seller |
| CapacityTrade | QSEID.Date.CT.buyer.seller |
| COP | QSEID.Date.COP.resource |
| PTP Obligation w/ Link to Option | QSEID.Date.CRR.crrId.offerid.crrAHId.source.sink |
| EnergyBid | QSEID.Date.EB.sp.bidId |
| EnergyOnlyOffer | QSEID.Date.EOO.sp.bidId |
| EnergyTrade | QSEID.Date.ET.sp.buyer.seller |
| IncDecOffer | QSEID.Date.IDO.resource.type |
| OutputSchedule | QSEID.Date.OS.resource |
| PTPObligation | QSEID.Date.PTP.bidid.source.sink |
| SelfArrangedAS | QSEID.Date.SAA.asType |
| SelfSchedule | QSEID.Date.SS.source.sink |
| ThreePartOffer | QSEID.Date.TPO.resource |
| AvailabilityPlan | QSEID.Date.AVP.resource.avpType |
| RTMEnergyBid | QSEID.Date.REB.resource |
| ExceptionFuelCost | QSEID.Date.EFC.resource |

For noun=ResParameterSetNotifications

| *BidType* | *mRID format* |
| --- | --- |
| GenResourceParameters | *QSEID.GEN.<resource>* |
| ControllableLoadResource | *QSEID.CON.<resource>* |
| NonControllableLoadResource | *QSEID.NON.<resource>* |
| ResourceParameters | *QSEID.RES.<resource>* |

Figure 223 – getNotifications Request

Example1 - Request Message (getNotification By mRID)

<?xml version="1.0" encoding="UTF-8"?>

<ns0:RequestMessage xmlns:mes="http://www.ercot.com/schema/2007-06/nodal/ews/message"

xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<ns0:Header>

<ns0:Verb>get</ns0:Verb>

<ns0:Noun>BidSetNotifications</ns0:Noun>

<ns0:ReplayDetection>

<ns0:Nonce>6212412181229622636116662D3410</ns0:Nonce>

<ns0:Created>2010-01-20T13:24:00.889-06:00</ns0:Created>

</ns0:ReplayDetection>

<ns0:Revision>1.0</ns0:Revision>

<ns0:Source>TESTQSE</ns0:Source>

<ns0:UserID>USER1</ns0:UserID>

<ns0:MessageID>62313564326161632D636166662D3430</ns0:MessageID>

<ns0:Comment/>

</ns0:Header>

<ns0:Payload>

<NotificationQuery xmlns="http://www.ercot.com/schema/2007-06/nodal/ews">

<startTime>2010-01-15T00:00:00-06:00</startTime>

<endTime>2010-01-15T04:00:00-06:00</endTime>

<mRID>TESTQSE.20100116.EB.XYZ.123456</mRID>

<mRID>TESTQSE.20100122.SAA.Reg-Up</mRID>

</NotificationQuery>

</ns0:Payload>

</ns0:RequestMessage>

Example2 - Request Message (getNotification By BidType)

<?xml version="1.0" encoding="UTF-8"?>

<ns0:RequestMessage xmlns:mes="http://www.ercot.com/schema/2007-06/nodal/ews/message"

xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<ns0:Header>

<ns0:Verb>get</ns0:Verb>

<ns0:Noun>ResParameterSetNotifications</ns0:Noun>

<ns0:ReplayDetection>

<ns0:Nonce>6212412181229622636116662D3410</ns0:Nonce>

<ns0:Created>2010-01-20T13:24:00.889-06:00</ns0:Created>

</ns0:ReplayDetection>

<ns0:Revision>1.0</ns0:Revision>

<ns0:Source>TESTQSE</ns0:Source>

<ns0:UserID>USER1</ns0:UserID>

<ns0:MessageID>62313564326161632D636166662D3430</ns0:MessageID>

<ns0:Comment/>

</ns0:Header>

<ns0:Payload>

<NotificationQuery xmlns="http://www.ercot.com/schema/2007-06/nodal/ews">

<startTime>2010-01-15T00:00:00-06:00</startTime>

<endTime>2010-01-15T04:00:00-06:00</endTime>

<bidType>GEN</bidType>

<bidProcessStatus>ACCEPTED</bidProcessStatus>

</NotificationQuery>

</ns0:Payload>

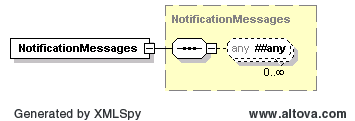
</ns0:RequestMessage>

The corresponding response messages would use the following message fields:

|  |  |
| --- | --- |
| Message Element | Value |
| Header/Verb | reply |
| Header/Noun | BidSetNotifications/ ResParameterSetNotifications/ VDIsNotifications |
| Header/Source | ERCOT |
| Reply/ReplyCode | *Reply code, success=OK, error=ERROR* |
| Reply/Error | *Error message, if error encountered* |
| Payload/ | *List of notifications in ResponseMessage format.(Refer to examples and Payload schema for more details)* |

Figure 224 – getNotifications Response

The following structure is used within the payload to convey a list of notifications:



Notifications (ResponseMessages) will be mapped to the any element under NotificationMessages.

Figure 225 – Notifications Container

<?xml version="1.0" encoding="UTF-8"?>

<NotificationMessages xmlns="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:ResponseMessage xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<ns1:Header>

<ns1:Verb>changed</ns1:Verb>

<ns1:Noun>BidSet</ns1:Noun>

<ns1:ReplayDetection>

<ns1:Nonce>69bfbe3aee98b34b659f309750944ba1</ns1:Nonce>

<ns1:Created>2010-01-20T14:24:53.84-06:00</ns1:Created>

</ns1:ReplayDetection>

<ns1:Revision>1.19E</ns1:Revision>

<ns1:Source>ERCOT</ns1:Source>

<ns1:UserID>USER1@TESTQSE</ns1:UserID>

<ns1:MessageID>62313564326161632D636166662D3430</ns1:MessageID>

<ns1:Comment>WBtSU7bT</ns1:Comment>

</ns1:Header>

<ns1:Reply>

<ns1:ReplyCode>OK</ns1:ReplyCode>

<ns1:Timestamp>2010-01-20T14:24:53.516-06:00</ns1:Timestamp>

</ns1:Reply>

<ns1:Payload>

<ns2:BidSet xmlns:ns2="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns2:tradingDate>2010-01-23</ns2:tradingDate>

<ns2:submitTime>2010-01-20T14:24:51.063-06:00</ns2:submitTime>

<ns2:EnergyOnlyOffer>

<ns2:mRID>TESTQSE.20100123.EOO.XYZ.15522</ns2:mRID>

<ns2:externalId/>

<ns2:status>ERRORS</ns2:status>

<ns2:error>

<ns2:severity>ERROR</ns2:severity>

<ns2:text>Validation of the Energy Only Offer Or Bid failed.</ns2:text>

</ns2:error>

<ns2:error>

<ns2:severity>ERROR</ns2:severity>

<ns2:text>The OFFER overlaps an existing multi-hour block OFFER with start hour ending 2 end

hour ending 2</ns2:text>

</ns2:error>

<ns2:error>

<ns2:severity>ERROR</ns2:severity>

<ns2:text>The OFFER overlaps an existing multi-hour block OFFER with start hour ending 4 end

hour ending 4</ns2:text>

</ns2:error>

</ns2:EnergyOnlyOffer>

</ns2:BidSet>

<ns1:format>XML</ns1:format>

</ns1:Payload>

</ns1:ResponseMessage>

<ns1:ResponseMessage xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<ns1:Header>

<ns1:Verb>created</ns1:Verb>

<ns1:Noun>BidSet</ns1:Noun>

<ns1:ReplayDetection>

<ns1:Nonce>158d14f627135430484f6b71cf9795dc</ns1:Nonce>

<ns1:Created>2010-01-20T14:27:17.918-06:00</ns1:Created>

</ns1:ReplayDetection>

<ns1:Revision>1.19E</ns1:Revision>

<ns1:Source>ERCOT</ns1:Source>

<ns1:UserID>USER1@TESTQSE</ns1:UserID>

<ns1:MessageID>12345678901234567</ns1:MessageID>

<ns1:Comment>W3LSU7bl</ns1:Comment>

</ns1:Header>

<ns1:Reply>

<ns1:ReplyCode>OK</ns1:ReplyCode>

<ns1:Timestamp>2010-01-20T14:27:17.834-06:00</ns1:Timestamp>

</ns1:Reply>

<ns1:Payload>

<ns2:BidSet xmlns:ns2="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns2:tradingDate>2010-01-22</ns2:tradingDate>

<ns2:submitTime>2010-01-20T14:27:16.802-06:00</ns2:submitTime>

<ns2:OutputSchedule>

<ns2:mRID>TESTQSE.20100122.OS.XYZ</ns2:mRID>

<ns2:externalId/>

<ns2:status>ACCEPTED</ns2:status>

<ns2:error>

<ns2:severity>INFORMATIVE</ns2:severity>

<ns2:text>Successfully processed the ERCOT Output Schedule.</ns2:text>

</ns2:error>

<ns2:error>

<ns2:severity>WARNING</ns2:severity>

<ns2:text>No COP entry submitted for hour ending 5</ns2:text>

</ns2:error>

</ns2:OutputSchedule>

</ns2:BidSet>

<ns1:format>XML</ns1:format>

</ns1:Payload>

</ns1:ResponseMessage>

<ns1:ResponseMessage xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<ns1:Header>

<ns1:Verb>created</ns1:Verb>

<ns1:Noun>BidSet</ns1:Noun>

<ns1:ReplayDetection>

<ns1:Nonce>9ab09b0a37be965f68bb47759c149b37</ns1:Nonce>

<ns1:Created>2010-01-20T14:37:52.793-06:00</ns1:Created>

</ns1:ReplayDetection>

<ns1:Revision>1.19E</ns1:Revision>

<ns1:Source>ERCOT</ns1:Source>

<ns1:UserID>USER1@TESTQSE</ns1:UserID>

<ns1:MessageID>9ab09b0a37be965f68bb47759c149b37</ns1:MessageID>

<ns1:Comment>WxtSU7bv</ns1:Comment>

</ns1:Header>

<ns1:Reply>

<ns1:ReplyCode>OK</ns1:ReplyCode>

<ns1:Timestamp>2010-01-20T14:37:52.577-06:00</ns1:Timestamp>

</ns1:Reply>

<ns1:Payload>

<ns2:BidSet xmlns:ns2="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns2:tradingDate>2010-01-22</ns2:tradingDate>

<ns2:submitTime>2010-01-20T14:37:50.602-06:00</ns2:submitTime>

<ns2:IncDecOffer>

<ns2:mRID>TESTQSE.20100122.IDO.XYZ.INC</ns2:mRID>

<ns2:externalId/>

<ns2:status>ERRORS</ns2:status>

<ns2:error>

<ns2:severity>ERROR</ns2:severity>

<ns2:text>Validation of the Incremental Decremental Offer failed.</ns2:text>

</ns2:error>

<ns2:error>

<ns2:severity>ERROR</ns2:severity>

<ns2:text>The MW Quantities and Price in the PQ Curve are same between points 1 and 2 for

hours ending 1 to 1</ns2:text>

</ns2:error>

<ns2:error>

<ns2:severity>ERROR</ns2:severity>

<ns2:text>The first quantity 12 in the pq\_curve element must be equal to low reasonability

limit 255 for hours ending 1 to 1</ns2:text>

</ns2:error>

<ns2:error>

<ns2:severity>ERROR</ns2:severity>

<ns2:text>The MW Quantities and Price in the PQ Curve are same between points 1 and 2 for

hours ending 2 to 2</ns2:text>

</ns2:error>

</ns2:IncDecOffer>

</ns2:BidSet>

<ns1:format>XML</ns1:format>

</ns1:Payload>

</ns1:ResponseMessage>

</NotificationMessages>

# Appendix A: WS-Notifications

The OASIS WS-Notifications specification can be obtained from <http://www.oasis-open.org>. The Notification interface to be used by ERCOT involves several modification to the WS-Notification standard. These include:

* The use of an Acknowledge message in order to improve reliability
* Reduction of the interface into a single XSD and a simplified WSDL

The key portions of WS-Notifications that are relevant to this specification are provided here for convenience. The following is the subset of the XML Schema for WS-Notifications that are relevant to this design.

<?xml version="1.0" encoding="UTF-8"?>

<!-- edited with XMLSpy v2007 sp2 (http://www.altova.com) by Scott Neumann (UISOL) -->

<!--

This Schema represents a revision to the OASIS WS-Notification standard for use by

ERCOT Market participants. This revision is intended to:

1. Reduce complexity by eliminating unused interfaces and references to external XSDs

2. Improve reliability through the addition of a positive acknowledgement message after

receipt of a notification message

Copyright (C) OASIS Open (2004-2006). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to OASIS, except as needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it into languages other than English.

-->

<xsd:schema xmlns:wsnt="http://www.ercot.com/schema/2007-06/nodal/notification" xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.ercot.com/schema/2007-06/nodal/notification" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- ================== Message Helper Types ===================== -->

<xsd:complexType name="NotificationMessageHolderType">

<xsd:sequence>

<xsd:element name="Message">

<xsd:complexType>

<xsd:sequence>

<xsd:any namespace="##any"

processContents="lax"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="NotificationMessage"

type="wsnt:NotificationMessageHolderType"/>

<!-- ========== Message Types for NotificationConsumer =========== -->

<xsd:element name="Notify">

<xsd:complexType>

<xsd:sequence>

<xsd:element ref="wsnt:NotificationMessage"

maxOccurs="unbounded"/>

<xsd:any namespace="##other" processContents="lax"

minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="Acknowledge">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="ReplyCode" type="xsd:string"

minOccurs="0"/>

<xsd:element name="Timestamp" type="xsd:dateTime"

minOccurs="0"/>

<xsd:any namespace="##other" processContents="lax"

minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="Fault">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="FaultCode" type="xsd:string"

minOccurs="0"/>

<xsd:element name="Timestamp" type="xsd:dateTime"

minOccurs="0"/>

<xsd:any namespace="##other" processContents="lax"

minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

The following WSDL represents the portion of WS-Notifications that is used for the receipt of Notification messages.

<?xml version="1.0" encoding="utf-8"?>

<!--

This Schema represents a revision to the OASIS WS-Notification standard for use by

ERCOT Market participants. This revision is intended to:

1. Reduce complexity by eliminating unused interfaces and references to external XSDs

2. Improve reliability through the addition of a positive acknowledgement message after

receipt of a notification message

Copyright (C) OASIS Open (2004-2006). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to OASIS, except as needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it into languages other than English.

-->

<wsdl:definitions name="WS-BaseNotification"

targetNamespace="http://www.ercot.com/wsdl/2007-06/nodal/notification"

xmlns:wsntw="http://www.ercot.com/wsdl/2007-06/nodal/notification"

xmlns:wsnt="http://www.ercot.com/schema/2007-06/nodal/notification"

xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">

<!-- ===================== Types Definitions ====================== -->

<wsdl:types>

<xsd:schema>

<xsd:import

namespace="http://www.ercot.com/schema/2007-06/nodal/notification"

schemaLocation="Notification.xsd"/>

</xsd:schema>

</wsdl:types>

<!-- ================ NotificationConsumer::Notify ================

Notify(

NotificationMessage

(SubscriptionReference, TopicExpression, ProducerReference,

Message)\*

returns: Ack

-->

<wsdl:message name="Notify">

<wsdl:part name="Notify" element="wsnt:Notify"/>

</wsdl:message>

<wsdl:message name="Acknowledge">

<wsdl:part name="Acknowledge" element="wsnt:Acknowledge"/>

</wsdl:message>

<wsdl:message name="Fault">

<wsdl:part name="Fault" element="wsnt:Fault"/>

</wsdl:message>

<!-- ========= NotificationConsumer PortType Definition =========== -->

<wsdl:portType name="NotificationConsumer">

<wsdl:operation name="Notify">

<wsdl:input message="wsntw:Notify" />

<wsdl:output message="wsntw:Acknowledge" />

<wsdl:fault message="wsntw:Fault" name="fault1"/>

</wsdl:operation>

</wsdl:portType>

</wsdl:definitions>

# Appendix B: WSDL for Market Requests

This WSDL uses a set of operations for servicing all market requests, related to bidding information requests and alert acknowledgements.

<?xml version = "1.0" encoding = "UTF-8"?>  
<wsdl:definitions attributeFormDefault="unqualified" elementFormDefault="qualified"  
 targetNamespace="http://www.ercot.com/wsdl/2007-05/nodal/ewsAbstract"  
 xmlns:ns="http://www.ercot.com/wsdl/2007-05/nodal/ewsAbstract"  
 xmlns:ns2="http://www.ercot.com/schema/2007-05/nodal/ews"  
 xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"  
 xmlns:xsd="http://www.w3.org/2001/XMLSchema">  
 <wsdl:import location="Message.xsd" namespace="http://www.ercot.com/schema/2007-05/nodal/ews"/>  
 <wsdl:types>  
 <xsd:schema targetNamespace="http://www.ercot.com/wsdl/2007-05/nodal/ewsAbstract">  
 <xsd:include schemaLocation="Message.xsd"/>  
 </xsd:schema>  
 </wsdl:types>  
 <wsdl:message name="RequestMessage">  
 <wsdl:part element="ns2:RequestMessage" name="Message"/>  
 </wsdl:message>  
 <wsdl:message name="ResponseMessage">  
 <wsdl:part element="ns2:ResponseMessage" name="Message"/>  
 </wsdl:message>  
 <wsdl:message name="FaultMessage">  
 <wsdl:part element="ns2:FaultMessage" name="part1"/>  
 </wsdl:message>  
 <wsdl:portType name="Operations">  
 <wsdl:operation name="MarketTransactions">  
 <wsdl:input message="ns:RequestMessage"/>  
 <wsdl:output message="ns:ResponseMessage"/>  
 <wsdl:fault message="ns:FaultMessage" name="fault1"/>  
 </wsdl:operation>  
 <wsdl:operation name="MarketInfo">  
 <wsdl:input message="ns:Request"/>  
 <wsdl:output message="ns:Response"/>  
 <wsdl:fault message="ns:FaultMessage" name="fault1"/>  
 </wsdl:operation>  
 <wsdl:operation name="Alerts">  
 <wsdl:input message="ns:Request"/>  
 <wsdl:output message="ns:Response"/>  
 <wsdl:fault message="ns:FaultMessage" name="fault1"/>  
 </wsdl:operation>  
 </wsdl:portType>  
 <wsdl:binding name="NodalSOAP" type="ns:Operations">  
 <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>  
 <wsdl:operation name="MarketTransactions">  
 <soap:operation soapAction="http://www.ercot.com/Nodal/MarketTransactions"/>  
 <wsdl:input>  
 <soap:body use="literal"/>  
 </wsdl:input>  
 <wsdl:output>  
 <soap:body use="literal"/>  
 </wsdl:output>  
 <wsdl:fault name="fault1">  
 <soap:body use="literal"/>  
 </wsdl:fault>  
 </wsdl:operation>  
 <wsdl:operation name="MarketInfo">  
 <soap:operation soapAction="http://www.ercot.com/Nodal/MarketInfo"/>  
 <wsdl:input>  
 <soap:body use="literal"/>  
 </wsdl:input>  
 <wsdl:output>  
 <soap:body use="literal"/>  
 </wsdl:output>  
 <wsdl:fault name="fault1">  
 <soap:body use="literal"/>  
 </wsdl:fault>  
 </wsdl:operation>  
 <wsdl:operation name="Alerts">  
 <soap:operation soapAction="http://www.ercot.com/Nodal/Alerts"/>  
 <wsdl:input>  
 <soap:body use="literal"/>  
 </wsdl:input>  
 <wsdl:output>  
 <soap:body use="literal"/>  
 </wsdl:output>  
 <wsdl:fault name="fault1">  
 <soap:body use="literal"/>  
 </wsdl:fault>  
 </wsdl:operation>  
 </wsdl:binding>  
 <wsdl:service name="Nodal">  
 <wsdl:port binding="ns:NodalSOAP" name="NodalSOAP">  
 <soap:address location="http://www.ercot.com/Nodal/"/>  
 </wsdl:port>  
 </wsdl:service>  
</wsdl:definitions>

# Appendix C: XML Schemas for Message and Payload Definitions

The following diagram describes the relationships between the XML Schemas defined by this document for incoming messages:



The following diagram describes the relationships between the XML Schemas defined by this document for outgoing messages:



# Appendix D: Annotated SOAP Message

An annotated example of a SOAP message follows, which includes the required security elements. The example is divided into sections and marked by a number, e.g. 1⇒. Following the SOAP message are explanatory notes describing each numbered section. In order to improve readability, some attributes that have URI values have been shortened.

The example does not include any namespace declarations. The table below lists the namespaces used in this example.

|  |  |  |
| --- | --- | --- |
| Prefix | Description | Namespace |
| SOAP-ENV | SOAP Envelop | http://schemas.xmlsoap.org/soap/envelope/ |
| ds | XML Digital Signature | http://www.w3.org/2000/09/xmldsig# |
| wsse | Web Service Security Extensions | http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd |
| wsu | Web Services Security Utility | http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd |
| xsd | XML Schema | http://www.w3.org/2001/XMLSchema |
| msg | An private namespace for specifying elements and attributes that are specific to ERCOT Web Services | TBD |

1⇒

<SOAP-ENV:Envelope ">

<SOAP-ENV:Header ">

2⇒

<wsse:Security SOAP-ENV:mustUnderstand="1">

<wsse:BinarySecurityToken

EncodingType="…#Base64Binary"

ValueType="…#X509v3"

wsu:Id="CertId-1776694">

MIIDDDCCAfSgAwIBAgIQM6YEf7FVYx/tZyEXgVComTANBgkqhkiG9w0 DAVPQVNJUzEeMBwGA1UEAwwVT0FTSVMgSW50ZXJvcCBUZXN0IENBMB4DTE4MDMxOTIzNTk1OVowQjEOMAwGA1UECgwFT0FTSVMxIDAeBgNVBAsVGVzdCBDZXJ0MQ4wDAYDVQQDDAVBbGljZTCBnzANBgkqhkiG9w0BAQE9By1VYo0aHrkKCNT4DkIgPL/SgahbeKdGhrbu3K2XG7arfD9tqIBIKMyvq+mUnMpNcKnLXLOjkTmMCqDYbbkehJlXPnaWLzve+mW0pJdPxtf3r sZKT8DN5Kyz+EZsCAwEAAaOBkzCBkDAJBgNVHRMEAjAAMDMGA1UdHwQaW50ZXJvcC5iYnRlc3QubmV0L2NybC9jYS5jcmwwDgYDVR0PAQH/BAQ4l0TUHZ1QV3V2QtlLNDm+PoxiDAfBgNVHSMEGDAWgBTAnSj8wes1oR3 JTw==

</wsse:BinarySecurityToken>

3⇒

<ds:Signature>

<ds:SignedInfo>

  <ds:CanonicalizationMethod

Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />

  <ds:SignatureMethod

Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />

4⇒

<ds:Reference URI="#id-1464350">

<ds:Transforms>

<ds:Transform

Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />

  </ds:Transforms>

  <ds:DigestMethod

Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />

  <ds:DigestValue">

1JmC1C0FrlPB42xfFKolgaCew5k=

</ds:DigestValue>

  </ds:Reference>

5⇒

<ds:Reference URI="#id-13498124">

<ds:Transforms>

<ds:Transform

Algorithm="…xml-exc-c14n#/>

<wsse:TransformationParameters">

<ds:CanonicalizationMethod

Algorithm="...xml-exc-c14n#"/>

  </wsse:TransformationParameters>

  </ds:Transform>

  </ds:Transforms>

  <ds:DigestMethod

Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />

  <ds:DigestValue>

sEaCJjrObpDVfM8zuabwQdBGKbY=

</ds:DigestValue>

  </ds:Reference>

6⇒

  </ds:SignedInfo>

 <ds:SignatureValue">

H1b7jH2bHpbrzJXkFS0msdUYycDMH4n6m4oTRtbo4Yk35/JzGcuwUYZ3

AwPcnqmcP5ROshjJparaPGuvQhbFR7zCxet2aoawJFWgG8jIeuDZDE8y6n+kbBzxadF2tGN8/nH6IlKg0+onD09i81rPHDAa 2kstCclX2NDet1Rnmfs=

</ds:SignatureValue>

7⇒

<ds:KeyInfo>

<wsse:SecurityTokenReference wsu:Id="id-13498124">

  <wsse:Reference URI="#CertId-1776694" ValueType="…#X509v3" />

  </wsse:SecurityTokenReference>

</ds:KeyInfo>

</ds:Signature>

</wsse:Security>

</SOAP-ENV:Header>

8⇒

<SOAP-ENV:Body wsu:Id="id-1464350" >

<msg:Message>

<msg:Header>

<msg:Verb>*verb*</msg:Verb>

<msg:Noun>*noun*</msg:Noun>

<msg:ReplayDetection>

<wsu:Created>

2006-11-29T20:05:55.022Z

</wsu:Created>

<wsse:Nonce EncodingType="…#Base64Binary">

75753793-50c2-455b-a9b3-123cb26474e7

</wsse:Nonce>

</msg:ReplayDetection>

<msg:Revision>1</msg:Revision>

<msg:Source>*market participant ID*</msg:Source>

</msg:Header>

.

.

.

</msg:Message>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

1⇒

This section starts the SOAP envelope and the SOAP header

2⇒

This section starts the Web Services Security Extensions, which includes security tokens and the digital signature. The first element in this section is the signer’s X.509 certificate, which is encoded in Base 64 binary. Note the identification of this element (CertId-1776694). The ID is later used in section 7⇒ to reference this certificate. Note that there is only one certificate in this message. The message verifier must ensure that the certificate chains to a trusted root.

3⇒

This section starts the digital signature block of the SOAP message. The signature is computed using the SHA-1 hash algorithm with RSA encryption.

4⇒

This section designates the first of two objects that are signed. This one points to the entire message body (#id-1464350), which is specified in section 8⇒ . The hashing algorithm is SHA-1.

5⇒

This section designates the second of two objects that are signed. This one points to the reference to the certificate (#STRId-13498124), which is specified in section 7⇒ .

6⇒

This section specifies the value of the signature. That is, the SHA-1 hash of references to sections 7⇒ and 8⇒ and the encryption of this hash using the signer’s private key.

7⇒

This section designates a reference to the signer’s certificate. In this case, the certificate is embedded in this SOAP message, and is referenced via the ID #CertId-1776694. This ID instructs the message verifier to get the certificate from section 2⇒of this SOAP message.

8⇒

This section starts the SOAP message body. It is designated using ID id-1464350, which is referenced as a signed element in section 2⇒. Note that the message body includes an element called ReplayDetection, which consists of a timestamp indicating when the message was signed and a unique number (the nonce). These two elements help detect and prevent replay attacks. The rest of the message body (i.e., the business transaction) is not shown.

# Appendix E: SOAP Examples

The purpose of this appendix is to provide some complete SOAP examples for reference purposes. Examples are provided for requests, replies and notifications.

SOAP Request Example for ThreePartOffer

<ns0:RequestMessage xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<ns0:Header>

<ns0:Verb>create</ns0:Verb>

<ns0:Noun>BidSet</ns0:Noun>

<ns0:ReplayDetection>

<ns0:Nonce>2b240d2d95f2aaa3ef7e552c965d512b</ns0:Nonce>

<ns0:Created>2009-08-06T16:13:25.685-05:00</ns0:Created>

</ns0:ReplayDetection>

<ns0:Revision>1.0</ns0:Revision>

<ns0:Source>QSE1</ns0:Source>

<ns0:UserID>USER1</ns0:UserID>

<ns0:Comment>Sample message template</ns0:Comment>

</ns0:Header>

<ns0:Request/>

<ns0:Payload>

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2009-08-06</ns1:tradingDate>

<ns1:ThreePartOffer>

<ns1:status>ACCEPTED</ns1:status>

<ns1:expirationTime>2009-08-08T00:00:00-05:00</ns1:expirationTime>

<ns1:resource>Resource1</ns1:resource>

<ns1:EocFipFop>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>0</ns1:fopPercent>

</ns1:EocFipFop>

<ns1:SuMeFipFop>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>0</ns1:fopPercent>

</ns1:SuMeFipFop>

<ns1:StartupCost>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:hot>131</ns1:hot>

<ns1:intermediate>152</ns1:intermediate>

<ns1:cold>174</ns1:cold>

</ns1:StartupCost>

<ns1:MinimumEnergy>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:cost>0</ns1:cost>

</ns1:MinimumEnergy>

<ns1:EnergyOfferCurve>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>0</ns1:xvalue>

<ns1:y1value>130.98</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>56</ns1:xvalue>

<ns1:y1value>134.51</ns1:y1value>

</ns1:CurveData>

</ns1:EnergyOfferCurve>

</ns1:ThreePartOffer>

</ns1:BidSet>

</ns0:Payload>

</ns0:RequestMessage>

SOAP Request Example for ASOffer

<ns0:RequestMessage xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<ns0:Header>

<ns0:Verb>create</ns0:Verb>

<ns0:Noun>BidSet</ns0:Noun>

<ns0:ReplayDetection>

<ns0:Nonce>ec1e821fc3c82f6e8a761c70a77ebafb</ns0:Nonce>

<ns0:Created>2009-08-06T16:12:39.239-05:00</ns0:Created>

</ns0:ReplayDetection>

<ns0:Revision>1.0</ns0:Revision>

<ns0:Source>QSE1</ns0:Source>

<ns0:UserID>USER1</ns0:UserID>

<ns0:Comment>Sample message template</ns0:Comment>

</ns0:Header>

<ns0:Request/>

<ns0:Payload>

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2009-08-06</ns1:tradingDate>

<ns1:ASOffer>

<ns1:status>ACCEPTED</ns1:status>

<ns1:expirationTime>2009-08-08T00:00:00-05:00</ns1:expirationTime>

<ns1:resource>Resource1</ns1:resource>

<ns1:asType>Off-Non-Spin</ns1:asType>

<ns1:ASPriceCurve>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:OffLineNonSpin>

<ns1:xvalue>33</ns1:xvalue>

<ns1:OFFNS>6.5</ns1:OFFNS>

<ns1:block>VARIABLE</ns1:block>

</ns1:OffLineNonSpin>

<ns1:multiHourBlock>true</ns1:multiHourBlock>

</ns1:ASPriceCurve>

</ns1:ASOffer>

</ns1:BidSet>

</ns0:Payload>

</ns0:RequestMessage>

SOAP Request Example for EnergyTrade

<ns0:RequestMessage xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<ns0:Header>

<ns0:Verb>create</ns0:Verb>

<ns0:Noun>BidSet</ns0:Noun>

<ns0:ReplayDetection>

<ns0:Nonce>342d41c37d8ea35b43c274b9601afe39</ns0:Nonce>

<ns0:Created>2009-08-06T16:13:28.46-05:00</ns0:Created>

</ns0:ReplayDetection>

<ns0:Revision>1.0</ns0:Revision>

<ns0:Source>QSE1</ns0:Source>

<ns0:UserID>USER1</ns0:UserID>

<ns0:Comment>Sample message template</ns0:Comment>

</ns0:Header>

<ns0:Request/>

<ns0:Payload>

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2009-08-06</ns1:tradingDate>

<ns1:EnergyTrade>

<ns1:status>ACCEPTED</ns1:status>

<ns1:buyer>QSE1</ns1:buyer>

<ns1:seller>QSE2</ns1:seller>

<ns1:sp>HB\_NORTH</ns1:sp>

<ns1:EnergySchedule>

<ns1:TmPoint>

<ns1:time>2009-08-06T07:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T07:15:00-05:00</ns1:ending>

<ns1:value1>25</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T07:15:00-05:00</ns1:time>

<ns1:ending>2009-08-06T07:30:00-05:00</ns1:ending>

<ns1:value1>25</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T07:30:00-05:00</ns1:time>

<ns1:ending>2009-08-06T07:45:00-05:00</ns1:ending>

<ns1:value1>25</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T07:45:00-05:00</ns1:time>

<ns1:ending>2009-08-06T08:00:00-05:00</ns1:ending>

<ns1:value1>25</ns1:value1>

</ns1:TmPoint>

</ns1:EnergySchedule>

</ns1:EnergyTrade>

</ns1:BidSet>

</ns0:Payload>

</ns0:RequestMessage>

SOAP Request Example for SelfArrangedAS

<ns0:RequestMessage xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<ns0:Header>

<ns0:Verb>create</ns0:Verb>

<ns0:Noun>BidSet</ns0:Noun>

<ns0:ReplayDetection>

<ns0:Nonce>e0fce1acb4f4a09f93f33aed74879455</ns0:Nonce>

<ns0:Created>2009-08-06T16:13:24.251-05:00</ns0:Created>

</ns0:ReplayDetection>

<ns0:Revision>1.0</ns0:Revision>

<ns0:Source>QSE1</ns0:Source>

<ns0:UserID>USER1</ns0:UserID>

<ns0:Comment>Sample message template</ns0:Comment>

</ns0:Header>

<ns0:Request/>

<ns0:Payload>

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2009-08-06</ns1:tradingDate>

<ns1:SelfArrangedAS>

<ns1:status>ACCEPTED</ns1:status>

<ns1:asType>Non-Spin</ns1:asType>

<ns1:CapacitySchedule>

<ns1:TmPoint>

<ns1:time>2009-08-06T15:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T16:00:00-05:00</ns1:ending>

<ns1:value1>2</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T16:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T17:00:00-05:00</ns1:ending>

<ns1:value1>2</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T17:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T18:00:00-05:00</ns1:ending>

<ns1:value1>2</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T18:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T19:00:00-05:00</ns1:ending>

<ns1:value1>2</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T19:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T20:00:00-05:00</ns1:ending>

<ns1:value1>2</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T20:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T21:00:00-05:00</ns1:ending>

<ns1:value1>2</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T21:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T22:00:00-05:00</ns1:ending>

<ns1:value1>2</ns1:value1>

</ns1:TmPoint>

<ns1:rrs\_values/>

</ns1:CapacitySchedule>

</ns1:SelfArrangedAS>

</ns1:BidSet>

</ns0:Payload>

</ns0:RequestMessage>

SOAP Request Example for EnergyOnlyOffer

<ns0:RequestMessage xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<ns0:Header>

<ns0:Verb>create</ns0:Verb>

<ns0:Noun>BidSet</ns0:Noun>

<ns0:ReplayDetection>

<ns0:Nonce>fcc6a592b67e3133a2e2f0cc91569924</ns0:Nonce>

<ns0:Created>2009-08-06T16:13:21.243-05:00</ns0:Created>

</ns0:ReplayDetection>

<ns0:Revision>1.0</ns0:Revision>

<ns0:Source>QSE1</ns0:Source>

<ns0:UserID>USER1</ns0:UserID>

<ns0:Comment>Sample message template</ns0:Comment>

</ns0:Header>

<ns0:Request/>

<ns0:Payload>

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2009-08-06</ns1:tradingDate>

<ns1:EnergyOnlyOffer>

<ns1:status>ACCEPTED</ns1:status>

<ns1:expirationTime>2009-08-06T01:00:00-05:00</ns1:expirationTime>

<ns1:sp>DC\_L</ns1:sp>

<ns1:bidID>100</ns1:bidID>

<ns1:EnergyOfferCurve>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:curveStyle>VARIABLE</ns1:curveStyle>

<ns1:CurveData>

<ns1:xvalue>20</ns1:xvalue>

<ns1:y1value>34</ns1:y1value>

</ns1:CurveData>

<ns1:multiHourBlock>true</ns1:multiHourBlock>

</ns1:EnergyOfferCurve>

</ns1:EnergyOnlyOffer>

<ns1:EnergyOnlyOffer>

<ns1:status>ACCEPTED</ns1:status>

<ns1:expirationTime>2009-08-06T01:00:00-05:00</ns1:expirationTime>

<ns1:sp>LZ\_WEST</ns1:sp>

<ns1:bidID>103</ns1:bidID>

<ns1:EnergyOfferCurve>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:curveStyle>FIXED</ns1:curveStyle>

<ns1:CurveData>

<ns1:xvalue>33</ns1:xvalue>

<ns1:y1value>41</ns1:y1value>

</ns1:CurveData>

<ns1:multiHourBlock>true</ns1:multiHourBlock>

</ns1:EnergyOfferCurve>

</ns1:EnergyOnlyOffer>

<ns1:EnergyOnlyOffer>

<ns1:status>ACCEPTED</ns1:status>

<ns1:expirationTime>2009-08-06T01:00:00-05:00</ns1:expirationTime>

<ns1:sp>HB\_BUSAVG</ns1:sp>

<ns1:bidID>101</ns1:bidID>

<ns1:EnergyOfferCurve>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:curveStyle>VARIABLE</ns1:curveStyle>

<ns1:CurveData>

<ns1:xvalue>76</ns1:xvalue>

<ns1:y1value>33</ns1:y1value>

</ns1:CurveData>

<ns1:multiHourBlock>true</ns1:multiHourBlock>

</ns1:EnergyOfferCurve>

</ns1:EnergyOnlyOffer>

<ns1:EnergyOnlyOffer>

<ns1:status>ACCEPTED</ns1:status>

<ns1:expirationTime>2009-08-06T01:00:00-05:00</ns1:expirationTime>

<ns1:sp>HLSES\_UNIT3</ns1:sp>

<ns1:bidID>102</ns1:bidID>

<ns1:EnergyOfferCurve>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:curveStyle>VARIABLE</ns1:curveStyle>

<ns1:CurveData>

<ns1:xvalue>34</ns1:xvalue>

<ns1:y1value>36</ns1:y1value>

</ns1:CurveData>

<ns1:multiHourBlock>true</ns1:multiHourBlock>

</ns1:EnergyOfferCurve>

</ns1:EnergyOnlyOffer>

</ns1:BidSet>

</ns0:Payload>

</ns0:RequestMessage>

SOAP Request Example for PTP Obligation w/ Link to Option

<ns0:RequestMessage xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<ns0:Header>

<ns0:Verb>create</ns0:Verb>

<ns0:Noun>BidSet</ns0:Noun>

<ns0:ReplayDetection>

<ns0:Nonce>f94df40917e8a3edb699f46899563706</ns0:Nonce>

<ns0:Created>2009-08-06T16:13:16.256-05:00</ns0:Created>

</ns0:ReplayDetection>

<ns0:Revision>1.0</ns0:Revision>

<ns0:Source>QSE1</ns0:Source>

<ns0:UserID>USER1</ns0:UserID>

<ns0:Comment>Sample message template</ns0:Comment>

</ns0:Header>

<ns0:Request/>

<ns0:Payload>

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2009-08-06</ns1:tradingDate>

<ns1:CRR>

<ns1:status>ACCEPTED</ns1:status>

<ns1:crrId>CRR\_246</ns1:crrId>

<ns1:offerId>100</ns1:offerId>

<ns1:crrAccountHolderId>XSTEC</ns1:crrAccountHolderId>

<ns1:source>DC\_E</ns1:source>

<ns1:sink>DC\_N</ns1:sink>

<ns1:CapacitySchedule>

<ns1:TmPoint>

<ns1:time>2009-08-06T06:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T22:00:00-05:00</ns1:ending>

<ns1:value1>20</ns1:value1>

<ns1:multiHourBlock>true</ns1:multiHourBlock>

</ns1:TmPoint>

</ns1:CapacitySchedule>

<ns1:MinimumReservationPrice>

<ns1:startTime>2009-08-06T06:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T22:00:00-05:00</ns1:endTime>

<ns1:price>12</ns1:price>

</ns1:MinimumReservationPrice>

<ns1:NOIEPeakLoadForecast>800</ns1:NOIEPeakLoadForecast>

</ns1:CRR>

<ns1:CRR>

<ns1:status>ACCEPTED</ns1:status>

<ns1:crrId>CRR\_247</ns1:crrId>

<ns1:offerId>100</ns1:offerId>

<ns1:crrAccountHolderId>XSTEC</ns1:crrAccountHolderId>

<ns1:source>HB\_HOUSTON</ns1:source>

<ns1:sink>LZ\_HOUSTON</ns1:sink>

<ns1:CapacitySchedule>

<ns1:TmPoint>

<ns1:time>2009-08-06T06:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T22:00:00-05:00</ns1:ending>

<ns1:value1>36</ns1:value1>

<ns1:multiHourBlock>true</ns1:multiHourBlock>

</ns1:TmPoint>

</ns1:CapacitySchedule>

<ns1:MinimumReservationPrice>

<ns1:startTime>2009-08-06T06:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T22:00:00-05:00</ns1:endTime>

<ns1:price>13</ns1:price>

</ns1:MinimumReservationPrice>

<ns1:NOIEPeakLoadForecast>800</ns1:NOIEPeakLoadForecast>

</ns1:CRR>

<ns1:CRR>

<ns1:status>ACCEPTED</ns1:status>

<ns1:crrId>CRR\_248</ns1:crrId>

<ns1:offerId>100</ns1:offerId>

<ns1:crrAccountHolderId>XSTEC</ns1:crrAccountHolderId>

<ns1:source>LZ\_SOUTH</ns1:source>

<ns1:sink>DC\_S</ns1:sink>

<ns1:CapacitySchedule>

<ns1:TmPoint>

<ns1:time>2009-08-06T22:00:00-05:00</ns1:time>

<ns1:ending>2009-08-07T00:00:00-05:00</ns1:ending>

<ns1:value1>20</ns1:value1>

<ns1:multiHourBlock>false</ns1:multiHourBlock>

</ns1:TmPoint>

</ns1:CapacitySchedule>

<ns1:MinimumReservationPrice>

<ns1:startTime>2009-08-06T22:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:price>17</ns1:price>

</ns1:MinimumReservationPrice>

<ns1:NOIEPeakLoadForecast>800</ns1:NOIEPeakLoadForecast>

</ns1:CRR>

</ns1:BidSet>

</ns0:Payload>

</ns0:RequestMessage>

SOAP Request Example for ASTrade

<ns0:RequestMessage xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<ns0:Header>

<ns0:Verb>create</ns0:Verb>

<ns0:Noun>BidSet</ns0:Noun>

<ns0:ReplayDetection>

<ns0:Nonce>524d992a5ccfd673fc38cf946b48cdd2</ns0:Nonce>

<ns0:Created>2009-08-06T16:13:27.431-05:00</ns0:Created>

</ns0:ReplayDetection>

<ns0:Revision>1.0</ns0:Revision>

<ns0:Source>QSE1</ns0:Source>

<ns0:UserID>USER1</ns0:UserID>

<ns0:Comment>Sample message template</ns0:Comment>

</ns0:Header>

<ns0:Request/>

<ns0:Payload>

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2009-08-06</ns1:tradingDate>

<ns1:ASTrade>

<ns1:status>ACCEPTED</ns1:status>

<ns1:buyer>QSE1</ns1:buyer>

<ns1:seller>QSE2</ns1:seller>

<ns1:asType>Reg-Up</ns1:asType>

<ns1:ASSchedule>

<ns1:TmPoint>

<ns1:time>2009-08-06T06:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T07:00:00-05:00</ns1:ending>

<ns1:value1>5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T07:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T08:00:00-05:00</ns1:ending>

<ns1:value1>5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T08:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T09:00:00-05:00</ns1:ending>

<ns1:value1>5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T09:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T10:00:00-05:00</ns1:ending>

<ns1:value1>5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T10:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T11:00:00-05:00</ns1:ending>

<ns1:value1>5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T11:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T12:00:00-05:00</ns1:ending>

<ns1:value1>5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T12:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T13:00:00-05:00</ns1:ending>

<ns1:value1>5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T13:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T14:00:00-05:00</ns1:ending>

<ns1:value1>5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T14:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T15:00:00-05:00</ns1:ending>

<ns1:value1>5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T15:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T16:00:00-05:00</ns1:ending>

<ns1:value1>5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T16:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T17:00:00-05:00</ns1:ending>

<ns1:value1>5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T17:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T18:00:00-05:00</ns1:ending>

<ns1:value1>5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T18:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T19:00:00-05:00</ns1:ending>

<ns1:value1>5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T19:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T20:00:00-05:00</ns1:ending>

<ns1:value1>5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T20:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T21:00:00-05:00</ns1:ending>

<ns1:value1>5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T21:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T22:00:00-05:00</ns1:ending>

<ns1:value1>5</ns1:value1>

</ns1:TmPoint>

</ns1:ASSchedule>

</ns1:ASTrade>

</ns1:BidSet>

</ns0:Payload>

</ns0:RequestMessage>

SOAP Request Example for CapacityTrade

<ns0:RequestMessage xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<ns0:Header>

<ns0:Verb>create</ns0:Verb>

<ns0:Noun>BidSet</ns0:Noun>

<ns0:ReplayDetection>

<ns0:Nonce>571909e70d48f3ef168913023643e723</ns0:Nonce>

<ns0:Created>2009-08-06T16:13:27.867-05:00</ns0:Created>

</ns0:ReplayDetection>

<ns0:Revision>1.0</ns0:Revision>

<ns0:Source>QSE1</ns0:Source>

<ns0:UserID>USER1</ns0:UserID>

<ns0:Comment>Sample message template</ns0:Comment>

</ns0:Header>

<ns0:Request/>

<ns0:Payload>

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2009-08-06</ns1:tradingDate>

<ns1:CapacityTrade>

<ns1:status>ACCEPTED</ns1:status>

<ns1:buyer>QSE1</ns1:buyer>

<ns1:seller>QSE2</ns1:seller>

<ns1:CapacitySchedule>

<ns1:TmPoint>

<ns1:time>2009-08-06T06:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T07:00:00-05:00</ns1:ending>

<ns1:value1>50</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T07:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T08:00:00-05:00</ns1:ending>

<ns1:value1>50</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T08:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T09:00:00-05:00</ns1:ending>

<ns1:value1>50</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T09:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T10:00:00-05:00</ns1:ending>

<ns1:value1>50</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T10:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T11:00:00-05:00</ns1:ending>

<ns1:value1>50</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T11:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T12:00:00-05:00</ns1:ending>

<ns1:value1>50</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T12:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T13:00:00-05:00</ns1:ending>

<ns1:value1>50</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T13:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T14:00:00-05:00</ns1:ending>

<ns1:value1>50</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T14:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T15:00:00-05:00</ns1:ending>

<ns1:value1>50</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T15:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T16:00:00-05:00</ns1:ending>

<ns1:value1>50</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T16:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T17:00:00-05:00</ns1:ending>

<ns1:value1>50</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T17:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T18:00:00-05:00</ns1:ending>

<ns1:value1>50</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T18:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T19:00:00-05:00</ns1:ending>

<ns1:value1>50</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T19:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T20:00:00-05:00</ns1:ending>

<ns1:value1>50</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T20:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T21:00:00-05:00</ns1:ending>

<ns1:value1>50</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-08-06T21:00:00-05:00</ns1:time>

<ns1:ending>2009-08-06T22:00:00-05:00</ns1:ending>

<ns1:value1>50</ns1:value1>

</ns1:TmPoint>

</ns1:CapacitySchedule>

</ns1:CapacityTrade>

</ns1:BidSet>

</ns0:Payload>

</ns0:RequestMessage>

SOAP Request Example for COP

<ns0:RequestMessage xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<ns0:Header>

<ns0:Verb>create</ns0:Verb>

<ns0:Noun>BidSet</ns0:Noun>

<ns0:ReplayDetection>

<ns0:Nonce>c65c9ef7b6743554252b80991c7bdee8</ns0:Nonce>

<ns0:Created>2009-08-06T16:13:07.45-05:00</ns0:Created>

</ns0:ReplayDetection>

<ns0:Revision>1.0</ns0:Revision>

<ns0:Source>QSE1</ns0:Source>

<ns0:UserID>USER1</ns0:UserID>

<ns0:Comment>Sample message template</ns0:Comment>

</ns0:Header>

<ns0:Request/>

<ns0:Payload>

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2009-08-06</ns1:tradingDate>

<ns1:COP>

<ns1:status>ACCEPTED</ns1:status>

<ns1:resource>Resource</ns1:resource>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T01:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T01:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T02:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T02:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T03:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T03:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T04:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T04:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T05:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T05:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T06:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T06:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T07:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T07:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T08:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T08:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T09:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T09:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T10:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T10:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T11:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T11:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T12:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T12:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T13:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T13:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T14:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T14:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T15:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T15:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T16:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T16:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T17:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T17:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T18:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T18:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T19:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T19:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T20:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T20:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T21:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T21:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T22:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T22:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T23:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:ResourceStatus>

<ns1:startTime>2009-08-06T23:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:operatingMode>ONRL</ns1:operatingMode>

</ns1:ResourceStatus>

<ns1:Limits>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T01:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T01:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T02:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T02:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T03:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T03:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T04:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T04:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T05:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T05:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T06:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T06:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T07:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T07:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T08:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T08:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T09:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T09:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T10:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T10:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T11:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T11:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T12:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T12:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T13:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T13:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T14:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T14:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T15:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T15:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T16:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T16:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T17:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T17:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T18:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T18:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T19:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T19:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T20:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T20:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T21:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T21:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T22:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T22:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T23:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:Limits>

<ns1:startTime>2009-08-06T23:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:hsl>4</ns1:hsl>

<ns1:lsl>0</ns1:lsl>

<ns1:hel>4</ns1:hel>

<ns1:lel>0</ns1:lel>

</ns1:Limits>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T01:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T01:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T02:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T02:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T03:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T03:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T04:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T04:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T05:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T05:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T06:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T06:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T07:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T07:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T08:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T08:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T09:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T09:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T10:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T10:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T11:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T11:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T12:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T12:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T13:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T13:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T14:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T14:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T15:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T15:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T16:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T16:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T17:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T17:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T18:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T18:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T19:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T19:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T20:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T20:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T21:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T21:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T22:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T22:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T23:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

<ns1:ASCapacity>

<ns1:startTime>2009-08-06T23:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:regUp>0</ns1:regUp>

<ns1:regDown>0</ns1:regDown>

<ns1:rrs>2</ns1:rrs>

<ns1:nonSpin>0</ns1:nonSpin>

</ns1:ASCapacity>

</ns1:COP>

</ns1:BidSet>

</ns0:Payload>

</ns0:RequestMessage>

SOAP Request Example for EnergyBid

<ns0:RequestMessage xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<ns0:Header>

<ns0:Verb>create</ns0:Verb>

<ns0:Noun>BidSet</ns0:Noun>

<ns0:ReplayDetection>

<ns0:Nonce>2307fd1a9063ce94a9da6557fc5a0429</ns0:Nonce>

<ns0:Created>2009-08-06T16:13:19.934-05:00</ns0:Created>

</ns0:ReplayDetection>

<ns0:Revision>1.0</ns0:Revision>

<ns0:Source>QSE1</ns0:Source>

<ns0:UserID>USER1</ns0:UserID>

<ns0:Comment>Sample message template</ns0:Comment>

</ns0:Header>

<ns0:Request/>

<ns0:Payload>

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2009-08-06</ns1:tradingDate>

<ns1:EnergyBid>

<ns1:status>ACCEPTED</ns1:status>

<ns1:expirationTime>2009-08-06T01:00:00-05:00</ns1:expirationTime>

<ns1:sp>HB\_HOUSTON</ns1:sp>

<ns1:bidID>101</ns1:bidID>

<ns1:PriceCurve>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:curveStyle>VARIABLE</ns1:curveStyle>

<ns1:CurveData>

<ns1:xvalue>110</ns1:xvalue>

<ns1:y1value>20</ns1:y1value>

</ns1:CurveData>

<ns1:multiHourBlock>true</ns1:multiHourBlock>

</ns1:PriceCurve>

</ns1:EnergyBid>

<ns1:EnergyBid>

<ns1:status>ACCEPTED</ns1:status>

<ns1:expirationTime>2009-08-06T01:00:00-05:00</ns1:expirationTime>

<ns1:sp>LZ\_NORTH</ns1:sp>

<ns1:bidID>103</ns1:bidID>

<ns1:PriceCurve>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:curveStyle>VARIABLE</ns1:curveStyle>

<ns1:CurveData>

<ns1:xvalue>65</ns1:xvalue>

<ns1:y1value>47</ns1:y1value>

</ns1:CurveData>

<ns1:multiHourBlock>false</ns1:multiHourBlock>

</ns1:PriceCurve>

</ns1:EnergyBid>

<ns1:EnergyBid>

<ns1:status>ACCEPTED</ns1:status>

<ns1:expirationTime>2009-08-06T01:00:00-05:00</ns1:expirationTime>

<ns1:sp>LZ\_WEST</ns1:sp>

<ns1:bidID>104</ns1:bidID>

<ns1:PriceCurve>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:curveStyle>VARIABLE</ns1:curveStyle>

<ns1:CurveData>

<ns1:xvalue>60</ns1:xvalue>

<ns1:y1value>42</ns1:y1value>

</ns1:CurveData>

<ns1:multiHourBlock>true</ns1:multiHourBlock>

</ns1:PriceCurve>

</ns1:EnergyBid>

<ns1:EnergyBid>

<ns1:status>ACCEPTED</ns1:status>

<ns1:expirationTime>2009-08-06T01:00:00-05:00</ns1:expirationTime>

<ns1:sp>HB\_NORTH</ns1:sp>

<ns1:bidID>102</ns1:bidID>

<ns1:PriceCurve>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:curveStyle>VARIABLE</ns1:curveStyle>

<ns1:CurveData>

<ns1:xvalue>70</ns1:xvalue>

<ns1:y1value>27</ns1:y1value>

</ns1:CurveData>

<ns1:multiHourBlock>true</ns1:multiHourBlock>

</ns1:PriceCurve>

</ns1:EnergyBid>

<ns1:EnergyBid>

<ns1:status>ACCEPTED</ns1:status>

<ns1:expirationTime>2009-08-06T01:00:00-05:00</ns1:expirationTime>

<ns1:sp>CEL\_CELANEG1</ns1:sp>

<ns1:bidID>100</ns1:bidID>

<ns1:PriceCurve>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:curveStyle>FIXED</ns1:curveStyle>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>38</ns1:y1value>

</ns1:CurveData>

<ns1:multiHourBlock>true</ns1:multiHourBlock>

</ns1:PriceCurve>

</ns1:EnergyBid>

<ns1:EnergyBid>

<ns1:status>ACCEPTED</ns1:status>

<ns1:expirationTime>2009-08-06T01:00:00-05:00</ns1:expirationTime>

<ns1:sp>SRB\_SRB\_GT12</ns1:sp>

<ns1:bidID>105</ns1:bidID>

<ns1:PriceCurve>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:curveStyle>FIXED</ns1:curveStyle>

<ns1:CurveData>

<ns1:xvalue>21</ns1:xvalue>

<ns1:y1value>35</ns1:y1value>

</ns1:CurveData>

<ns1:multiHourBlock>false</ns1:multiHourBlock>

</ns1:PriceCurve>

</ns1:EnergyBid>

</ns1:BidSet>

</ns0:Payload>

</ns0:RequestMessage>

SOAP Request Example for IncDecOffer

<ns0:RequestMessage xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<ns0:Header>

<ns0:Verb>create</ns0:Verb>

<ns0:Noun>BidSet</ns0:Noun>

<ns0:ReplayDetection>

<ns0:Nonce>dbd7362a7fce310c3c6e8a8d779896a6</ns0:Nonce>

<ns0:Created>2009-08-06T16:13:22.116-05:00</ns0:Created>

</ns0:ReplayDetection>

<ns0:Revision>1.0</ns0:Revision>

<ns0:Source>QSE1</ns0:Source>

<ns0:UserID>USER1</ns0:UserID>

<ns0:Comment>Sample message template</ns0:Comment>

</ns0:Header>

<ns0:Request/>

<ns0:Payload>

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2009-08-06</ns1:tradingDate>

<ns1:IncDecOffer>

<ns1:status>ACCEPTED</ns1:status>

<ns1:expirationTime>2009-08-08T00:00:00-05:00</ns1:expirationTime>

<ns1:resource>Resource1</ns1:resource>

<ns1:FipFop>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T01:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T01:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T02:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T02:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T03:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T03:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T04:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T04:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T05:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T05:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T06:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T06:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T07:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T07:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T08:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T08:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T09:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T09:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T10:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T10:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T11:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T11:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T12:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T12:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T13:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T13:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T14:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T14:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T15:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T15:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T16:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T16:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T17:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T17:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T18:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T18:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T19:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T19:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T20:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T20:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T21:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T21:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T22:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T22:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T23:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:FipFop>

<ns1:startTime>2009-08-06T23:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:fipPercent>0</ns1:fipPercent>

<ns1:fopPercent>100</ns1:fopPercent>

</ns1:FipFop>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T00:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T01:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T01:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T02:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T02:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T03:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T03:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T04:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T04:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T05:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T05:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T06:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T06:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T07:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T07:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T08:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T08:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T09:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T09:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T10:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T10:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T11:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T11:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T12:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T12:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T13:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T13:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T14:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T14:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T15:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T15:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T16:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T16:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T17:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T17:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T18:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T18:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T19:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T19:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T20:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T20:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T21:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T21:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T22:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T22:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-06T23:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

<ns1:IncrementalCurve>

<ns1:startTime>2009-08-06T23:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-08-07T00:00:00-05:00</ns1:endTime>

<ns1:CurveData>

<ns1:xvalue>5.4</ns1:xvalue>

<ns1:y1value>43</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>30</ns1:xvalue>

<ns1:y1value>57</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>54</ns1:xvalue>

<ns1:y1value>58</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>79</ns1:xvalue>

<ns1:y1value>77</ns1:y1value>

</ns1:CurveData>

<ns1:CurveData>

<ns1:xvalue>108.5</ns1:xvalue>

<ns1:y1value>78</ns1:y1value>

</ns1:CurveData>

</ns1:IncrementalCurve>

</ns1:IncDecOffer>

</ns1:BidSet>

</ns0:Payload>

</ns0:RequestMessage>

SOAP Request Example for OutputSchedule

<ns0:RequestMessage xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<ns0:Header>

<ns0:Verb>create</ns0:Verb>

<ns0:Noun>BidSet</ns0:Noun>

<ns0:ReplayDetection>

<ns0:Nonce>f4bccb1bce41a31e75d214a11f7278a8</ns0:Nonce>

<ns0:Created>2009-07-20T16:24:25.327-05:00</ns0:Created>

</ns0:ReplayDetection>

<ns0:Revision>1.0</ns0:Revision>

<ns0:Source>REPS</ns0:Source>

<ns0:UserID>USER1</ns0:UserID>

<ns0:Comment>Sample message template</ns0:Comment>

</ns0:Header>

<ns0:Request/>

<ns0:Payload>

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2009-06-25</ns1:tradingDate>

<ns1:OutputSchedule>

<ns1:status>ACCEPTED</ns1:status>

<ns1:resource>Resource1</ns1:resource>

<ns1:EnergySchedule>

<ns1:TmPoint>

<ns1:time>2009-06-25T00:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T00:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T00:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T00:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T00:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T00:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T00:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T00:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T00:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T00:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T00:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T00:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T00:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T00:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T00:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T00:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T00:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T00:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T00:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T00:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T00:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T00:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T00:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T01:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T01:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T01:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T01:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T01:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T01:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T01:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T01:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T01:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T01:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T01:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T01:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T02:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T02:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T02:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T02:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T02:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T02:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T02:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T02:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T02:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T02:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T02:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T02:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T03:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T03:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T03:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T03:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T03:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T03:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T03:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T03:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T03:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T03:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T03:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T03:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T03:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T03:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T03:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T03:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T03:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T03:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T03:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T03:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T03:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T03:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T03:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T03:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T04:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T04:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T04:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T04:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T04:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T04:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T04:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T04:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T04:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T04:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T04:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T04:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T04:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T04:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T04:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T04:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T04:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T04:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T04:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T04:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T04:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T04:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T04:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T04:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T05:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T05:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T05:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T05:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T05:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T05:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T05:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T05:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T05:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T05:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T05:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T05:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T05:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T05:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T05:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T05:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T05:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T05:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T05:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T05:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T05:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T05:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T05:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T05:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T06:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T06:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T06:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T06:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T06:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T06:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T06:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T06:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T06:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T06:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T06:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T06:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T06:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T06:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T06:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T06:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T06:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T06:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T06:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T06:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T06:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T06:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T06:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T06:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T07:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T07:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T07:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T07:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T07:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T07:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T07:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T07:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T07:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T07:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T07:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T07:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T07:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T07:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T07:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T07:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T07:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T07:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T07:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T07:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T07:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T07:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T07:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T07:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T08:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T08:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T08:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T08:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T08:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T08:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T08:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T08:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T08:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T08:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T08:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T08:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T08:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T08:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T08:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T08:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T08:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T08:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T08:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T08:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T08:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T08:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T08:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T08:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T09:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T09:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T09:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T09:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T09:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T09:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T09:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T09:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T09:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T09:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T09:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T09:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T09:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T09:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T09:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T09:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T09:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T09:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T09:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T09:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T09:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T09:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T09:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T09:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T10:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T10:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T10:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T10:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T10:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T10:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T10:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T10:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T10:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T10:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T10:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T10:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T10:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T10:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T10:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T10:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T10:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T10:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T10:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T10:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T10:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T10:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T10:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T10:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T11:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T11:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T11:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T11:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T11:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T11:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T11:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T11:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T11:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T11:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T11:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T11:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T11:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T11:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T11:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T11:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T11:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T11:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T11:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T11:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T11:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T11:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T11:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T11:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T12:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T12:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T12:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T12:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T12:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T12:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T12:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T12:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T12:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T12:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T12:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T12:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T12:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T12:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T12:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T12:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T12:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T12:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T12:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T12:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T12:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T12:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T12:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T12:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T13:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T13:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T13:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T13:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T13:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T13:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T13:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T13:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T13:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T13:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T13:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T13:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T13:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T13:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T13:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T13:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T13:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T13:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T13:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T13:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T13:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T13:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T13:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T13:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T14:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T14:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T14:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T14:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T14:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T14:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T14:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T14:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T14:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T14:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T14:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T14:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T14:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T14:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T14:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T14:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T14:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T14:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T14:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T14:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T14:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T14:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T14:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T14:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T15:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T15:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T15:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T15:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T15:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T15:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T15:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T15:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T15:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T15:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T15:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T15:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T15:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T15:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T15:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T15:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T15:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T15:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T15:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T15:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T15:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T15:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T15:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T15:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T16:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T16:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T16:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T16:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T16:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T16:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T16:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T16:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T16:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T16:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T16:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T16:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T16:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T16:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T16:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T16:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T16:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T16:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T16:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T16:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T16:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T16:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T16:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T16:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T17:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T17:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T17:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T17:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T17:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T17:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T17:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T17:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T17:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T17:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T17:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T17:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T17:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T17:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T17:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T17:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T17:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T17:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T17:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T17:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T17:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T17:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T17:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T17:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T18:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T18:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T18:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T18:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T18:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T18:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T18:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T18:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T18:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T18:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T18:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T18:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T18:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T18:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T18:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T18:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T18:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T18:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T18:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T18:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T18:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T18:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T18:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T18:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T19:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T19:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T19:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T19:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T19:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T19:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T19:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T19:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T19:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T19:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T19:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T19:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T19:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T19:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T19:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T19:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T19:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T19:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T19:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T19:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T19:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T19:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T19:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T19:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T20:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T20:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T20:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T20:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T20:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T20:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T20:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T20:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T20:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T20:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T20:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T20:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T20:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T20:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T20:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T20:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T20:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T20:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T20:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T20:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T20:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T20:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T20:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T20:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T21:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T21:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T21:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T21:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T21:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T21:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T21:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T21:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T21:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T21:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T21:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T21:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T21:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T21:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T21:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T21:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T21:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T21:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T21:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T21:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T21:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T21:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T21:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T21:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T22:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T22:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T22:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T22:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T22:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T22:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T22:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T22:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T22:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T22:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T22:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T22:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T22:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T22:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T22:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T22:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T22:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T22:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T22:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T22:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T22:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T22:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T22:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T22:55:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T23:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:05:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T23:05:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:10:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T23:10:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:15:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T23:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:20:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T23:20:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:25:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T23:25:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:30:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T23:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:35:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T23:35:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:40:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T23:40:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:45:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T23:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:50:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T23:50:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:55:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T23:55:00-05:00</ns1:time>

<ns1:ending>2009-06-26T00:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

</ns1:TmPoint>

</ns1:EnergySchedule>

</ns1:BidSet>

</ns0:Payload>

</ns0:RequestMessage>

SOAP Request Example for PTPObligation

<ns0:RequestMessage xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<ns0:Header>

<ns0:Verb>create</ns0:Verb>

<ns0:Noun>BidSet</ns0:Noun>

<ns0:ReplayDetection>

<ns0:Nonce>932b24d36891f6cc87a9962658da1764</ns0:Nonce>

<ns0:Created>2009-07-20T16:23:17.001-05:00</ns0:Created>

</ns0:ReplayDetection>

<ns0:Revision>1.0</ns0:Revision>

<ns0:Source>STRT</ns0:Source>

<ns0:UserID>USER1</ns0:UserID>

<ns0:Comment>Sample message template</ns0:Comment>

</ns0:Header>

<ns0:Request/>

<ns0:Payload>

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2009-06-25</ns1:tradingDate>

<ns1:PTPObligation>

<ns1:status>ACCEPTED</ns1:status>

<ns1:source>R\_LOSTPI\_2</ns1:source>

<ns1:sink>BTRCK\_E\_2</ns1:sink>

<ns1:bidId>101</ns1:bidId>

<ns1:CapacitySchedule>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:00:00-05:00</ns1:time>

<ns1:ending>2009-06-26T00:00:00-05:00</ns1:ending>

<ns1:value1>42</ns1:value1>

<ns1:multiHourBlock>true</ns1:multiHourBlock>

</ns1:TmPoint>

</ns1:CapacitySchedule>

<ns1:MaximumPrice>

<ns1:startTime>2009-06-25T02:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-26T00:00:00-05:00</ns1:endTime>

<ns1:price>20</ns1:price>

</ns1:MaximumPrice>

</ns1:PTPObligation>

<ns1:PTPObligation>

<ns1:status>ACCEPTED</ns1:status>

<ns1:source>R\_COLETO\_1</ns1:source>

<ns1:sink>CTL\_GT\_102</ns1:sink>

<ns1:bidId>102</ns1:bidId>

<ns1:CapacitySchedule>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:00:00-05:00</ns1:ending>

<ns1:value1>51</ns1:value1>

<ns1:multiHourBlock>false</ns1:multiHourBlock>

</ns1:TmPoint>

</ns1:CapacitySchedule>

<ns1:MaximumPrice>

<ns1:startTime>2009-06-25T02:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-25T23:00:00-05:00</ns1:endTime>

<ns1:price>26</ns1:price>

</ns1:MaximumPrice>

</ns1:PTPObligation>

<ns1:PTPObligation>

<ns1:status>ACCEPTED</ns1:status>

<ns1:source>FALCON\_E\_2</ns1:source>

<ns1:sink>R\_SWCOG\_1</ns1:sink>

<ns1:bidId>103</ns1:bidId>

<ns1:CapacitySchedule>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:00:00-05:00</ns1:time>

<ns1:ending>2009-06-26T00:00:00-05:00</ns1:ending>

<ns1:value1>43</ns1:value1>

<ns1:multiHourBlock>true</ns1:multiHourBlock>

</ns1:TmPoint>

</ns1:CapacitySchedule>

<ns1:MaximumPrice>

<ns1:startTime>2009-06-25T02:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-26T00:00:00-05:00</ns1:endTime>

<ns1:price>20</ns1:price>

</ns1:MaximumPrice>

</ns1:PTPObligation>

<ns1:PTPObligation>

<ns1:status>ACCEPTED</ns1:status>

<ns1:source>AUTO\_CBEC2</ns1:source>

<ns1:sink>MGSES\_E\_2</ns1:sink>

<ns1:bidId>104</ns1:bidId>

<ns1:CapacitySchedule>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:00:00-05:00</ns1:ending>

<ns1:value1>50</ns1:value1>

<ns1:multiHourBlock>false</ns1:multiHourBlock>

</ns1:TmPoint>

</ns1:CapacitySchedule>

<ns1:MaximumPrice>

<ns1:startTime>2009-06-25T02:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-25T23:00:00-05:00</ns1:endTime>

<ns1:price>17</ns1:price>

</ns1:MaximumPrice>

</ns1:PTPObligation>

<ns1:PTPObligation>

<ns1:status>ACCEPTED</ns1:status>

<ns1:source>PEARSALLSW9</ns1:source>

<ns1:sink>R\_HLSES\_1</ns1:sink>

<ns1:bidId>105</ns1:bidId>

<ns1:CapacitySchedule>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:00:00-05:00</ns1:ending>

<ns1:value1>54</ns1:value1>

<ns1:multiHourBlock>false</ns1:multiHourBlock>

</ns1:TmPoint>

</ns1:CapacitySchedule>

<ns1:MaximumPrice>

<ns1:startTime>2009-06-25T01:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-25T23:00:00-05:00</ns1:endTime>

<ns1:price>8</ns1:price>

</ns1:MaximumPrice>

</ns1:PTPObligation>

<ns1:PTPObligation>

<ns1:status>ACCEPTED</ns1:status>

<ns1:source>TNDOW\_COGN1</ns1:source>

<ns1:sink>ISSUE\_38</ns1:sink>

<ns1:bidId>106</ns1:bidId>

<ns1:CapacitySchedule>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:00:00-05:00</ns1:ending>

<ns1:value1>48</ns1:value1>

<ns1:multiHourBlock>true</ns1:multiHourBlock>

</ns1:TmPoint>

</ns1:CapacitySchedule>

<ns1:MaximumPrice>

<ns1:startTime>2009-06-25T01:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-25T23:00:00-05:00</ns1:endTime>

<ns1:price>27</ns1:price>

</ns1:MaximumPrice>

</ns1:PTPObligation>

<ns1:PTPObligation>

<ns1:status>ACCEPTED</ns1:status>

<ns1:source>ISSUE\_8</ns1:source>

<ns1:sink>R\_SANDHSYD\_1</ns1:sink>

<ns1:bidId>107</ns1:bidId>

<ns1:CapacitySchedule>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:00:00-05:00</ns1:ending>

<ns1:value1>48</ns1:value1>

<ns1:multiHourBlock>true</ns1:multiHourBlock>

</ns1:TmPoint>

</ns1:CapacitySchedule>

<ns1:MaximumPrice>

<ns1:startTime>2009-06-25T02:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-25T23:00:00-05:00</ns1:endTime>

<ns1:price>18</ns1:price>

</ns1:MaximumPrice>

</ns1:PTPObligation>

<ns1:PTPObligation>

<ns1:status>ACCEPTED</ns1:status>

<ns1:source>DECKER\_E\_1</ns1:source>

<ns1:sink>PSA\_PSA\_G5</ns1:sink>

<ns1:bidId>108</ns1:bidId>

<ns1:CapacitySchedule>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:00:00-05:00</ns1:ending>

<ns1:value1>32</ns1:value1>

<ns1:multiHourBlock>true</ns1:multiHourBlock>

</ns1:TmPoint>

</ns1:CapacitySchedule>

<ns1:MaximumPrice>

<ns1:startTime>2009-06-25T02:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-25T23:00:00-05:00</ns1:endTime>

<ns1:price>28</ns1:price>

</ns1:MaximumPrice>

</ns1:PTPObligation>

<ns1:PTPObligation>

<ns1:status>ACCEPTED</ns1:status>

<ns1:source>R\_KING\_NE\_1</ns1:source>

<ns1:sink>VALERO\_L\_A</ns1:sink>

<ns1:bidId>109</ns1:bidId>

<ns1:CapacitySchedule>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T22:00:00-05:00</ns1:ending>

<ns1:value1>54</ns1:value1>

<ns1:multiHourBlock>true</ns1:multiHourBlock>

</ns1:TmPoint>

</ns1:CapacitySchedule>

<ns1:MaximumPrice>

<ns1:startTime>2009-06-25T01:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-25T22:00:00-05:00</ns1:endTime>

<ns1:price>18</ns1:price>

</ns1:MaximumPrice>

</ns1:PTPObligation>

<ns1:PTPObligation>

<ns1:status>ACCEPTED</ns1:status>

<ns1:source>WHCCS\_E\_1</ns1:source>

<ns1:sink>SCURRYSWCH</ns1:sink>

<ns1:bidId>110</ns1:bidId>

<ns1:CapacitySchedule>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:00:00-05:00</ns1:ending>

<ns1:value1>55</ns1:value1>

<ns1:multiHourBlock>true</ns1:multiHourBlock>

</ns1:TmPoint>

</ns1:CapacitySchedule>

<ns1:MaximumPrice>

<ns1:startTime>2009-06-25T01:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-25T23:00:00-05:00</ns1:endTime>

<ns1:price>21</ns1:price>

</ns1:MaximumPrice>

</ns1:PTPObligation>

<ns1:PTPObligation>

<ns1:status>ACCEPTED</ns1:status>

<ns1:source>BTE\_BTE\_G3</ns1:source>

<ns1:sink>CTL\_ST\_101</ns1:sink>

<ns1:bidId>111</ns1:bidId>

<ns1:CapacitySchedule>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:00:00-05:00</ns1:time>

<ns1:ending>2009-06-26T00:00:00-05:00</ns1:ending>

<ns1:value1>36</ns1:value1>

<ns1:multiHourBlock>false</ns1:multiHourBlock>

</ns1:TmPoint>

</ns1:CapacitySchedule>

<ns1:MaximumPrice>

<ns1:startTime>2009-06-25T02:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-26T00:00:00-05:00</ns1:endTime>

<ns1:price>8</ns1:price>

</ns1:MaximumPrice>

</ns1:PTPObligation>

<ns1:PTPObligation>

<ns1:status>ACCEPTED</ns1:status>

<ns1:source>R\_MCSES\_3</ns1:source>

<ns1:sink>CALLAHAN\_E\_1</ns1:sink>

<ns1:bidId>112</ns1:bidId>

<ns1:CapacitySchedule>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:00:00-05:00</ns1:time>

<ns1:ending>2009-06-26T00:00:00-05:00</ns1:ending>

<ns1:value1>57</ns1:value1>

<ns1:multiHourBlock>false</ns1:multiHourBlock>

</ns1:TmPoint>

</ns1:CapacitySchedule>

<ns1:MaximumPrice>

<ns1:startTime>2009-06-25T02:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-26T00:00:00-05:00</ns1:endTime>

<ns1:price>24</ns1:price>

</ns1:MaximumPrice>

</ns1:PTPObligation>

<ns1:PTPObligation>

<ns1:status>ACCEPTED</ns1:status>

<ns1:source>LCSES\_E\_1</ns1:source>

<ns1:sink>LEON\_CRK\_E\_1</ns1:sink>

<ns1:bidId>113</ns1:bidId>

<ns1:CapacitySchedule>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:00:00-05:00</ns1:ending>

<ns1:value1>51</ns1:value1>

<ns1:multiHourBlock>true</ns1:multiHourBlock>

</ns1:TmPoint>

</ns1:CapacitySchedule>

<ns1:MaximumPrice>

<ns1:startTime>2009-06-25T01:00:00-05:00</ns1:startTime>

<ns1:endTime>2009-06-25T23:00:00-05:00</ns1:endTime>

<ns1:price>23</ns1:price>

</ns1:MaximumPrice>

</ns1:PTPObligation>

</ns1:BidSet>

</ns0:Payload>

</ns0:RequestMessage>

SOAP Request Example for SelfSchedule

<ns0:RequestMessage xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/ews/message">

<ns0:Header>

<ns0:Verb>create</ns0:Verb>

<ns0:Noun>BidSet</ns0:Noun>

<ns0:ReplayDetection>

<ns0:Nonce>abebfa51ad6ed7c621b4442767c96b5b</ns0:Nonce>

<ns0:Created>2009-07-20T16:23:20.86-05:00</ns0:Created>

</ns0:ReplayDetection>

<ns0:Revision>1.0</ns0:Revision>

<ns0:Source>CPS</ns0:Source>

<ns0:UserID>USER1</ns0:UserID>

<ns0:Comment>Sample message template</ns0:Comment>

</ns0:Header>

<ns0:Request/>

<ns0:Payload>

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:tradingDate>2009-06-25</ns1:tradingDate>

<ns1:SelfSchedule>

<ns1:status>ACCEPTED</ns1:status>

<ns1:source>STP\_E\_1</ns1:source>

<ns1:sink>BRAUNIG</ns1:sink>

<ns1:EnergySchedule>

<ns1:TmPoint>

<ns1:time>2009-06-25T00:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T00:15:00-05:00</ns1:ending>

<ns1:value1>24.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T00:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T00:30:00-05:00</ns1:ending>

<ns1:value1>24.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T00:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T00:45:00-05:00</ns1:ending>

<ns1:value1>24.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T00:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T01:00:00-05:00</ns1:ending>

<ns1:value1>24.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T01:15:00-05:00</ns1:ending>

<ns1:value1>23.8</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T01:30:00-05:00</ns1:ending>

<ns1:value1>23.8</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T01:45:00-05:00</ns1:ending>

<ns1:value1>23.8</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T01:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T02:00:00-05:00</ns1:ending>

<ns1:value1>23.8</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T02:15:00-05:00</ns1:ending>

<ns1:value1>25.5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T02:30:00-05:00</ns1:ending>

<ns1:value1>25.5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T02:45:00-05:00</ns1:ending>

<ns1:value1>25.5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T02:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T03:00:00-05:00</ns1:ending>

<ns1:value1>25.5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T03:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T03:15:00-05:00</ns1:ending>

<ns1:value1>24.2</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T03:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T03:30:00-05:00</ns1:ending>

<ns1:value1>24.2</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T03:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T03:45:00-05:00</ns1:ending>

<ns1:value1>24.2</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T03:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T04:00:00-05:00</ns1:ending>

<ns1:value1>24.2</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T04:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T04:15:00-05:00</ns1:ending>

<ns1:value1>22</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T04:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T04:30:00-05:00</ns1:ending>

<ns1:value1>22</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T04:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T04:45:00-05:00</ns1:ending>

<ns1:value1>22</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T04:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T05:00:00-05:00</ns1:ending>

<ns1:value1>22</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T05:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T05:15:00-05:00</ns1:ending>

<ns1:value1>22</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T05:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T05:30:00-05:00</ns1:ending>

<ns1:value1>22</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T05:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T05:45:00-05:00</ns1:ending>

<ns1:value1>22</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T05:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T06:00:00-05:00</ns1:ending>

<ns1:value1>22</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T06:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T06:15:00-05:00</ns1:ending>

<ns1:value1>23.4</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T06:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T06:30:00-05:00</ns1:ending>

<ns1:value1>23.4</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T06:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T06:45:00-05:00</ns1:ending>

<ns1:value1>23.4</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T06:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T07:00:00-05:00</ns1:ending>

<ns1:value1>23.4</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T07:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T07:15:00-05:00</ns1:ending>

<ns1:value1>23.4</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T07:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T07:30:00-05:00</ns1:ending>

<ns1:value1>23.4</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T07:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T07:45:00-05:00</ns1:ending>

<ns1:value1>23.4</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T07:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T08:00:00-05:00</ns1:ending>

<ns1:value1>23.4</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T08:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T08:15:00-05:00</ns1:ending>

<ns1:value1>25</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T08:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T08:30:00-05:00</ns1:ending>

<ns1:value1>25</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T08:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T08:45:00-05:00</ns1:ending>

<ns1:value1>25</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T08:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T09:00:00-05:00</ns1:ending>

<ns1:value1>25</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T09:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T09:15:00-05:00</ns1:ending>

<ns1:value1>24.2</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T09:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T09:30:00-05:00</ns1:ending>

<ns1:value1>24.2</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T09:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T09:45:00-05:00</ns1:ending>

<ns1:value1>24.2</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T09:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T10:00:00-05:00</ns1:ending>

<ns1:value1>24.2</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T10:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T10:15:00-05:00</ns1:ending>

<ns1:value1>22.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T10:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T10:30:00-05:00</ns1:ending>

<ns1:value1>22.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T10:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T10:45:00-05:00</ns1:ending>

<ns1:value1>22.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T10:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T11:00:00-05:00</ns1:ending>

<ns1:value1>22.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T11:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T11:15:00-05:00</ns1:ending>

<ns1:value1>22.3</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T11:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T11:30:00-05:00</ns1:ending>

<ns1:value1>22.3</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T11:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T11:45:00-05:00</ns1:ending>

<ns1:value1>22.3</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T11:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T12:00:00-05:00</ns1:ending>

<ns1:value1>22.3</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T12:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T12:15:00-05:00</ns1:ending>

<ns1:value1>21.6</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T12:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T12:30:00-05:00</ns1:ending>

<ns1:value1>21.6</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T12:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T12:45:00-05:00</ns1:ending>

<ns1:value1>21.6</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T12:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T13:00:00-05:00</ns1:ending>

<ns1:value1>21.6</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T13:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T13:15:00-05:00</ns1:ending>

<ns1:value1>23.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T13:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T13:30:00-05:00</ns1:ending>

<ns1:value1>23.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T13:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T13:45:00-05:00</ns1:ending>

<ns1:value1>23.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T13:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T14:00:00-05:00</ns1:ending>

<ns1:value1>23.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T14:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T14:15:00-05:00</ns1:ending>

<ns1:value1>23.4</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T14:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T14:30:00-05:00</ns1:ending>

<ns1:value1>23.4</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T14:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T14:45:00-05:00</ns1:ending>

<ns1:value1>23.4</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T14:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T15:00:00-05:00</ns1:ending>

<ns1:value1>23.4</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T15:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T15:15:00-05:00</ns1:ending>

<ns1:value1>22.9</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T15:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T15:30:00-05:00</ns1:ending>

<ns1:value1>22.9</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T15:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T15:45:00-05:00</ns1:ending>

<ns1:value1>22.9</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T15:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T16:00:00-05:00</ns1:ending>

<ns1:value1>22.9</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T16:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T16:15:00-05:00</ns1:ending>

<ns1:value1>24.3</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T16:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T16:30:00-05:00</ns1:ending>

<ns1:value1>24.3</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T16:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T16:45:00-05:00</ns1:ending>

<ns1:value1>24.3</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T16:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T17:00:00-05:00</ns1:ending>

<ns1:value1>24.3</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T17:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T17:15:00-05:00</ns1:ending>

<ns1:value1>22.6</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T17:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T17:30:00-05:00</ns1:ending>

<ns1:value1>22.6</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T17:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T17:45:00-05:00</ns1:ending>

<ns1:value1>22.6</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T17:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T18:00:00-05:00</ns1:ending>

<ns1:value1>22.6</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T18:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T18:15:00-05:00</ns1:ending>

<ns1:value1>22.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T18:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T18:30:00-05:00</ns1:ending>

<ns1:value1>22.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T18:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T18:45:00-05:00</ns1:ending>

<ns1:value1>22.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T18:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T19:00:00-05:00</ns1:ending>

<ns1:value1>22.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T19:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T19:15:00-05:00</ns1:ending>

<ns1:value1>22.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T19:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T19:30:00-05:00</ns1:ending>

<ns1:value1>22.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T19:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T19:45:00-05:00</ns1:ending>

<ns1:value1>22.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T19:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T20:00:00-05:00</ns1:ending>

<ns1:value1>22.1</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T20:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T20:15:00-05:00</ns1:ending>

<ns1:value1>22.5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T20:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T20:30:00-05:00</ns1:ending>

<ns1:value1>22.5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T20:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T20:45:00-05:00</ns1:ending>

<ns1:value1>22.5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T20:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T21:00:00-05:00</ns1:ending>

<ns1:value1>22.5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T21:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T21:15:00-05:00</ns1:ending>

<ns1:value1>24.6</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T21:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T21:30:00-05:00</ns1:ending>

<ns1:value1>24.6</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T21:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T21:45:00-05:00</ns1:ending>

<ns1:value1>24.6</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T21:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T22:00:00-05:00</ns1:ending>

<ns1:value1>24.6</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T22:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T22:15:00-05:00</ns1:ending>

<ns1:value1>22.9</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T22:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T22:30:00-05:00</ns1:ending>

<ns1:value1>22.9</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T22:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T22:45:00-05:00</ns1:ending>

<ns1:value1>22.9</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T22:45:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:00:00-05:00</ns1:ending>

<ns1:value1>22.9</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T23:00:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:15:00-05:00</ns1:ending>

<ns1:value1>23.5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T23:15:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:30:00-05:00</ns1:ending>

<ns1:value1>23.5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T23:30:00-05:00</ns1:time>

<ns1:ending>2009-06-25T23:45:00-05:00</ns1:ending>

<ns1:value1>23.5</ns1:value1>

</ns1:TmPoint>

<ns1:TmPoint>

<ns1:time>2009-06-25T23:45:00-05:00</ns1:time>

<ns1:ending>2009-06-26T00:00:00-05:00</ns1:ending>

<ns1:value1>23.5</ns1:value1>

</ns1:TmPoint>

</ns1:EnergySchedule>

</ns1:SelfSchedule>

</ns1:BidSet>

</ns0:Payload>

</ns0:RequestMessage>

SOAP Reply Example

<?xml version="1.0" encoding="UTF-8"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Body>

<ns0:ResponseMessage xmlns:SOAP-ENV = "http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns0 = "http://www.ercot.com/schema/2007-05/nodal/ews/msg" xmlns:wsu = "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

<ns0:Header>

<ns0:Verb>reply</ns0:Verb>

<ns0:Noun>BidSet</ns0:Noun>

<ns0:ReplayDetection>

<ns0:Nonce>2:36 PM</ns0:Nonce>

<ns0:Created>2007-08-08T11:54:32.105-05:00</ns0:Created>

</ns0:ReplayDetection>

<ns0:Revision>1.0</ns0:Revision>

<ns0:Source>ERCOT</ns0:Source>

</ns0:Header>

<ns0:Reply>

<ns0:ReplyCode>OK</ns0:ReplyCode>

<ns0:Timestamp>2007-08-08T11:54:52.214-05:00</ns0:Timestamp>

</ns0:Reply>

<ns0:Payload>

<ns1:BidSet xmlns:ns1 = "http://www.ercot.com/schema/2007-05/nodal/ews">

<ns1:tradingDate>2007-05-10</ns1:tradingDate>

<ns1:ThreePartOffer>

<ns1:mRID>111111112.20070517.TPO.SOBAY\_7\_SY4</ns1:mRID>

<ns1:externalId>12345</ns1:externalId>

<ns1:status>SUBMITTED</ns1:status>

</ns1:ThreePartOffer>

</ns1:BidSet>

</ns0:Payload>

</ns0:ResponseMessage>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

SOAP Notification Example

<?xml version="1.0" encoding="UTF-8"?><SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Body>

<ns0:Notify xmlns:ns0="http://www.ercot.com/schema/2007-06/nodal/notification">

<ns0:NotificationMessage>

<ns0:Message>

<ns0:ResponseMessage xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/ews/msg">

<ns0:Header>

<ns0:Verb>changed</ns0:Verb>

<ns0:Noun>BidSet</ns0:Noun>

<ns0:ReplayDetection>

<ns0:Nonce>f148e6952a1370b5019b512f71145bd9</ns0:Nonce>

<ns0:Created>2007-08-08T11:55:38.08-05:00</ns0:Created>

</ns0:ReplayDetection>

<ns0:Revision>1.02</ns0:Revision>

<ns0:Source>ERCOT</ns0:Source>

</ns0:Header>

<ns0:Reply>

<ns0:ReplyCode>OK</ns0:ReplyCode>

<ns0:Timestamp>2007-08-08T11:55:38.08-05:00</ns0:Timestamp>

</ns0:Reply>

<ns0:Payload>

<ns1:BidSet xmlns:ns1="http://www.ercot.com/schema/2007-05/nodal/ews">

<ns1:tradingDate>2007-05-10</ns1:tradingDate>

<ns1:ThreePartOffer>

<ns1:mRID>111111112.20070517.TPO.SOBAY\_7\_SY4</ns1:mRID>

<ns1:externalId>12342</ns1:externalId>

<ns1:status>ACCEPTED</ns1:status>

</ns1:ThreePartOffer>

</ns1:BidSet>

</ns0:Payload>

</ns0:ResponseMessage>

</ns0:Message>

</ns0:NotificationMessage>

</ns0:Notify>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

# Appendix F: XSL Examples

The purpose of this appendix is to provide XSL examples related to the processing and transformation of the XML described in this specification. One example of XSL usage would be to convert UTC time into separate date and hour values based on the local time zone. For example, given the following XML:

<startTime>2007-12-17T06:00:00Z</startTime>

An XSL template can be used to generate tags that are desired by a consumer of the message such as the following where the hour is an hour ending in local time:

<startTime>2007-12-17T06:00:00Z</startTime>

<startDate>2007-12-17</startDate>

<startHour>1</startHour>

The following XSL example is implemented using custom functions within TIBCO BusinessWorks.

<?xml version='1.0'?>

<xsl:stylesheet version="1.0"

xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

xmlns:msg="http://www.uisol.com"

xmlns:tib="http://www.tibco.com/bw/xslt/custom-functions"

exclude-result-prefixes="msg tib">

<xsl:template match="/">

<xsl:apply-templates/>

</xsl:template>

<xsl:template match="\*">

<xsl:choose>

<xsl:when test="name() = 'msg:startTime'">

<xsl:copy>

<xsl:apply-templates/>

</xsl:copy>

<xsl:variable name="vStartTime">

<xsl:value-of select="tib:translate-timezone(text(),

tib:get-timezone-from-dateTime(current-dateTime()))"/>

</xsl:variable>

<msg:startDate>

<xsl:value-of select="substring($vStartTime,1,10)"/>

</msg:startDate>

<msg:startTime>

<xsl:choose>

<xsl:when test="substring($vStartTime,12,2) = '00'">24</xsl:when>

<xsl:otherwise><xsl:value-of select="format-number(substring($vStartTime,12,2),'#0')"/></xsl:otherwise>

</xsl:choose>

</msg:startTime>

</xsl:when>

<xsl:when test="name() = 'msg:endTime'">

<xsl:copy>

<xsl:apply-templates/>

</xsl:copy>

<xsl:variable name="vEndTime">

<xsl:value-of select="tib:translate-timezone(text(),

tib:get-timezone-from-dateTime(current-dateTime()))"/>

</xsl:variable>

<msg:endDate>

<xsl:value-of select="substring($vEndTime,1,10)"/>

</msg:endDate>

<msg:endTime>

<xsl:choose>

<xsl:when test="substring($vEndTime,12,2) = '00'">24</xsl:when>

<xsl:otherwise>

<xsl:value-of select="format-number(substring($vEndTime,12,2),'#0')"/>

</xsl:otherwise>

</xsl:choose>

</msg:endTime>

</xsl:when>

<xsl:otherwise>

<xsl:copy>

<xsl:apply-templates/>

</xsl:copy>

</xsl:otherwise>

</xsl:choose>

</xsl:template>

</xsl:stylesheet>

The following XSL example is implemented using Xalan:

<?xml version='1.0'?>

<xsl:stylesheet version="1.0"

xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

xmlns:msg="http://www.uisol.com"

xmlns:lxslt="http://xml.apache.org/xslt"

xmlns:mk="ext1"

extension-element-prefixes="mk"

exclude-result-prefixes="msg lxslt">

<lxslt:component prefix="mk" functions="ConvertDate">

<lxslt:script lang="javascript">

function ConvertDate(inYear,inMonth,inDay,inTime,inType){

var strDate = inMonth + "/" + inDay + "/" + inYear + " " + inTime + " UTC";

var GMTDate = new Date(strDate);

strDate = inType == "Date" ? GMTDate.getFullYear() + "-" + GMTDate.getMonth() + "-" + GMTDate.getDate() : GMTDate.getHours();

return strDate.toString();

}

</lxslt:script>

</lxslt:component>

<xsl:template match="/">

<xsl:apply-templates/>

</xsl:template>

<xsl:template match="\*">

<xsl:choose>

<xsl:when test="(name() = 'msg:startTime' or name() = 'msg:endTime')”>

<xsl:copy>

<xsl:apply-templates/>

</xsl:copy>

<xsl:variable name="vYear"><xsl:value-of select="substring(text(),1,4)"/></xsl:variable>

<xsl:variable name="vMonth"><xsl:value-of select="substring(text(),6,2)"/></xsl:variable>

<xsl:variable name="vDay"><xsl:value-of select="substring(text(),9,2)"/></xsl:variable>

<xsl:variable name="vTime"><xsl:value-of select="substring(text(),12,8)"/></xsl:variable>

<xsl:choose>

<xsl:when test="name() = 'msg:startTime'">

<msg:startDate>

<xsl:value-of select="mk:ConvertDate(string($vYear),string($vMonth),string($vDay),string($vTime),'Date')"/>

</msg:startDate>

<msg:startTime>

<xsl:value-of select="mk:ConvertDate(string($vYear),string($vMonth),string($vDay),string($vTime),'Time')"/>

</msg:startTime>

</xsl:when>

<xsl:when test="name() = 'msg:endTime'">

<msg:endDate>

<xsl:value-of select="mk:ConvertDate(string($vYear),string($vMonth),string($vDay),string($vTime),'Date')"/>

</msg:endDate>

<msg:endTime>

<xsl:value-of select="mk:ConvertDate(string($vYear),string($vMonth),string($vDay),string($vTime),'Time')"/>

</msg:endTime>

</xsl:when>

</xsl:choose>

</xsl:when>

<xsl:otherwise>

<xsl:copy>

<xsl:apply-templates/>

</xsl:copy>

</xsl:otherwise>

</xsl:choose>

</xsl:template>

</xsl:stylesheet>

# Appendix G: Payload Compression Example

The purpose of this section is to provide an example of the code required to compress and encode a payload, where the payload is then passed as the contents of the Payload/Compressed element. The following is a Java example that leverages commonly used classes for compression and base64 encoding.

package BusinessService.TS.WebServiceClient.CompressionClient;

import java.util.\*;

import java.io.\*;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.InputStream;

import java.io.IOException;

import java.util.zip.\*;

import org.apache.soap.encoding.soapenc.Base64;

public class CompressionClientCompressandEncode{

protected byte[] input = null;

protected String base64GzipInput = "";

public byte[] getinput() {

return input;

}

public void setinput(byte[] val) {

input = val;

}

public String getbase64GzipInput() {

return base64GzipInput;

}

public void setbase64GzipInput(String val) {

base64GzipInput = val;

}

public CompressionClientCompressandEncode() {

}

public void invoke() throws Exception {

In : byte[] input

Out : String base64GzipInput

ByteArrayOutputStream bas = new ByteArrayOutputStream();

GZIPOutputStream bis = new GZIPOutputStream(bas);

bis.write(input);

bis.close();

base64GzipInput = Base64.encode(bas.toByteArray());

}

}

# Appendix H: DST XML Examples

The purpose of this section is to provide an high level DST(daylight time) XML examples for different types operations.

* BidSet or other submission examples for DST long day:

<BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews">

<tradingDate>2010-11-07</tradingDate>

<ASOffer>

<startTime>2010-11-07T00:00:00.000-05:00</startTime>

<endTime>2010-11-08T00:00:00.000-06:00</endTime>

<externalId>ERCOTTESt</externalId>

<expirationTime>2010-11-07T23:59:00.000-06:00</expirationTime>

<resource>XXXX\_UNIT1</resource>

<asType>REGUP-RRS-ONNS</asType>

<ASPriceCurve>

<startTime>2010-11-07T00:00:00.000-05:00</startTime>

<endTime>2010-11-07T01:00:00.000-05:00</endTime>

<OnLineReserves>

<xvalue>10</xvalue>

<ONNS>2</ONNS>

<block>VARIABLE</block>

</OnLineReserves>

</ASPriceCurve>

<ASPriceCurve>

<!—DST normal hour -->

<startTime>2010-11-07T01:00:00.000-05:00</startTime>

<endTime>2010-11-07T01:00:00.000-06:00</endTime>

<OnLineReserves>

<xvalue>11</xvalue>

<ONNS>1</ONNS>

<block>VARIABLE</block>

</OnLineReserves>

</ASPriceCurve>

<ASPriceCurve>

<!—DST extra hour -->

<startTime>2010-11-07T01:00:00.000-06:00</startTime>

<endTime>2010-11-07T02:00:00.000-06:00</endTime>

<OnLineReserves>

<xvalue>11</xvalue>

<ONNS>1</ONNS>

<block>VARIABLE</block>

</OnLineReserves>

</ASPriceCurve>

<ASPriceCurve>

<startTime>2010-11-07T02:00:00.000-06:00</startTime>

<endTime>2010-11-07T03:00:00.000-06:00</endTime>

<OnLineReserves>

<xvalue>11</xvalue>

<ONNS>1</ONNS>

<block>VARIABLE</block>

</OnLineReserves>

</ASPriceCurve>

<ASPriceCurve>

<startTime>2010-11-07T03:00:00.000-06:00</startTime>

<endTime>2010-11-07T04:00:00.000-06:00</endTime>

<OnLineReserves>

<xvalue>11</xvalue>

<ONNS>1</ONNS>

<block>VARIABLE</block>

</OnLineReserves>

</ASPriceCurve>

<ASPriceCurve>

<startTime>2010-11-07T04:00:00.000-06:00</startTime>

<endTime>2010-11-07T05:00:00.000-06:00</endTime>

<OnLineReserves>

<xvalue>11</xvalue>

<ONNS>1</ONNS>

<block>VARIABLE</block>

</OnLineReserves>

</ASPriceCurve>

</ASOffer>

------

------

-------

* BidSet or other submission examples for DST short day:

<BidSet xmlns="http://www.ercot.com/schema/2007-06/nodal/ews">

<tradingDate>2011-03-13</tradingDate>

<ASOffer>

<startTime>2011-03-13T00:00:00.000-06:00</startTime>

<endTime>2011-03-14T00:00:00.000-05:00</endTime>

<externalId>ERCOTTESt</externalId>

<expirationTime>2011-03-13T23:59:00.000-05:00</expirationTime>

<resource>XXXX\_UNIT1</resource>

<asType>REGUP-RRS-ONNS</asType>

<ASPriceCurve>

<startTime>2011-03-13T00:00:00.000-06:00</startTime>

<endTime>2011-03-13T01:00:00.000-06:00</endTime>

<OnLineReserves>

<xvalue>11</xvalue>

<ONNS>1</ONNS>

<block>VARIABLE</block>

</OnLineReserves>

</ASPriceCurve>

<ASPriceCurve>

<!—DST short hour -->

<startTime>2011-03-13T01:00:00.000-06:00</startTime>

<endTime>2011-03-13T03:00:00.000-05:00</endTime>

<OnLineReserves>

<xvalue>11</xvalue>

<ONNS>1</ONNS>

<block>VARIABLE</block>

</OnLineReserves>

</ASPriceCurve>

<ASPriceCurve>

<startTime>2011-03-13T03:00:00.000-05:00</startTime>

<endTime>2011-03-13T04:00:00.000-05:00</endTime>

<OnLineReserves>

<xvalue>11</xvalue>

<ONNS>1</ONNS>

<block>VARIABLE</block>

</OnLineReserves>

</ASPriceCurve>

<ASPriceCurve>

<startTime>2011-03-13T04:00:00.000-05:00</startTime>

<endTime>2011-03-13T05:00:00.000-05:00</endTime>

<OnLineReserves>

<xvalue>11</xvalue>

<ONNS>1</ONNS>

<block>VARIABLE</block>

</OnLineReserves>

</ASPriceCurve>

<ASPriceCurve>

<startTime>2011-03-13T05:00:00.000-05:00</startTime>

<endTime>2011-03-13T06:00:00.000-05:00</endTime>

<OnLineReserves>

<xvalue>11</xvalue>

<ONNS>1</ONNS>

<block>VARIABLE</block>

</OnLineReserves>

</ASPriceCurve>

</ASOffer>

------

------

-------

* Refer to Section 2.3.4 for dateTime value conversions into mRID hours for continuous hour queries or cancels. This is applicable for BidSet related operations.
* XML examples for Report payloads(Awards etc) that contain dateTime values in ISO format.

DST Long Day:

<?xml version="1.0" encoding="UTF-8"?>

<AwardSet>

<tradingDate>2010-11-07</tradingDate>

<marketType>DAM</marketType>

<AwardedCRR>

<qse>TestQSE</qse>

<startTime>2010-11-07T00:00:00-05:00</startTime>

<endTime>2010-11-07T01:00:00-05:00</endTime>

<tradingDate>2010-11-07</tradingDate>

<marketType>DAM</marketType>

<awardedMW>3</awardedMW>

<price>0</price>

<source>Source1</source>

<sink>Sink1</sink>

<crrId>12345</crrId>

<offerId>1234</offerId>

</AwardedCRR>

<AwardedCRR>

<qse>TestQSE</qse>

<startTime>2010-11-07T01:00:00-05:00</startTime>

<endTime>2010-11-07T01:00:00-06:00</endTime>

<tradingDate>2010-11-07</tradingDate>

<marketType>DAM</marketType>

<awardedMW>3</awardedMW>

<price>0</price>

<source>Source1</source>

<sink>Sink1</sink>

<crrId>12345</crrId>

<offerId>1234</offerId>

</AwardedCRR>

<AwardedCRR>

<qse>TestQSE</qse>

<startTime>2010-11-07T01:00:00-06:00</startTime>

<endTime>2010-11-07T02:00:00-06:00</endTime>

<tradingDate>2010-11-07</tradingDate>

<marketType>DAM</marketType>

<awardedMW>3</awardedMW>

<price>0</price>

<source>Source1</source>

<sink>Sink1</sink>

<crrId>12345</crrId>

<offerId>1234</offerId>

</AwardedCRR>

<AwardedCRR>

<qse>TestQSE</qse>

<startTime>2010-11-07T02:00:00-06:00</startTime>

<endTime>2010-11-07T03:00:00-06:00</endTime>

<tradingDate>2010-11-07</tradingDate>

<marketType>DAM</marketType>

<awardedMW>3</awardedMW>

<price>0</price>

<source>Source1</source>

<sink>Sink1</sink>

<crrId>12345</crrId>

<offerId>1234</offerId>

</AwardedCRR>

<AwardedCRR>

<qse>TestQSE</qse>

<startTime>2010-11-07T03:00:00-06:00</startTime>

<endTime>2010-11-07T04:00:00-06:00</endTime>

<tradingDate>2010-11-07</tradingDate>

<marketType>DAM</marketType>

<awardedMW>3</awardedMW>

<price>0</price>

<source>Source1</source>

<sink>Sink1</sink>

<crrId>12345</crrId>

<offerId>1234</offerId>

</AwardedCRR>

</AwardSet>

DST Short Day

<?xml version="1.0" encoding="UTF-8"?>

<ns1:ASObligations xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ercot.com/schema/2007-06/nodal/ews ErcotInformation.xsd">

<ns1:ASObligation>

<ns1:startTime>2011-03-13T00:00:00-06:00</ns1:startTime>

<ns1:endTime>2011-03-13T01:00:00-06:00</ns1:endTime>

<ns1:TmPoint>

<ns1:time>2011-03-13T00:00:00-06:00</ns1:time>

<ns1:ending>2011-03-13T01:00:00-06:00</ns1:ending><ns1:value1>133.8</ns1:value1>

</ns1:TmPoint>

<ns1:asType>Reg-Down</ns1:asType>

<ns1:qse>QSE</ns1:qse>

</ns1:ASObligation>

<ns1:ASObligation>

<ns1:startTime>2011-03-13T01:00:00-06:00</ns1:startTime>

<ns1:endTime>2011-03-13T03:00:00-05:00</ns1:endTime>

<ns1:TmPoint>

<ns1:time>2011-03-13T01:00:00-06:00</ns1:time>

<ns1:ending>2011-03-13T03:00:00-05:00</ns1:ending>

<ns1:value1>164.7</ns1:value1>

</ns1:TmPoint>

<ns1:asType>Reg-Down</ns1:asType>

<ns1:qse>QSE</ns1:qse>

</ns1:ASObligation>

<ns1:ASObligation>

<ns1:startTime>2011-03-13T03:00:00-05:00</ns1:startTime>

<ns1:endTime>2011-03-13T04:00:00-05:00</ns1:endTime>

<ns1:TmPoint>

<ns1:time>2011-03-13T03:00:00-05:00</ns1:time>

<ns1:ending>2011-03-13T01400:00-05:00</ns1:ending>

<ns1:value1>473.6</ns1:value1>

</ns1:TmPoint>

<ns1:asType>Reg-Down</ns1:asType>

<ns1:qse>QSE</ns1:qse>

</ns1:ASObligation>

</ns1:ASObligations>

* XML examples for Report payloads(StartupShutdownInstructions etc) that contain deliveryHout/hourEnding rinstead of timestmps in payload.

<?xml version="1.0" encoding="UTF-8"?>

<ns1:StartupShutdownInstructions xmlns:ns0="http://www.ercot.com/schema/2007-05/nodal/eip/il" xmlns:ns1="http://www.ercot.com/schema/2007-06/nodal/ews">

<ns1:StartupShutdown>

<ns1:rucType>DRUC</ns1:rucType>

<ns1:hourEnding>01</ns1:hourEnding>

<ns1:deliveryDate>2010-11-07</ns1:deliveryDate>

<ns1:resource>Res\_UNIT1</ns1:resource>

<ns1:clearedCommit>true</ns1:clearedCommit>

<ns1:clearedDecommit>false</ns1:clearedDecommit>

<ns1:decommitReasonCode/>

<ns1:reportDate>2010-11-06T15:47:55-05:00</ns1:reportDate>

</ns1:StartupShutdown>

<ns1:StartupShutdown>

<ns1:rucType>DRUC</ns1:rucType>

<ns1:hourEnding>2</ns1:hourEnding>

<ns1:deliveryDate>2010-11-07</ns1:deliveryDate>

<ns1:resource>Res\_UNIT1</ns1:resource>

<ns1:clearedCommit>true</ns1:clearedCommit>

<ns1:clearedDecommit>false</ns1:clearedDecommit>

<ns1:decommitReasonCode/>

<ns1:reportDate>2010-11-06T15:47:55-05:00</ns1:reportDate>

</ns1:StartupShutdown>

<!-- DST Extra hour -- >

<ns1:StartupShutdown>

<ns1:rucType>DRUC</ns1:rucType>

<ns1:hourEnding>2\*</ns1:hourEnding>

<ns1:deliveryDate>2010-11-07</ns1:deliveryDate>

<ns1:resource>Res\_UNIT1</ns1:resource>

<ns1:clearedCommit>true</ns1:clearedCommit>

<ns1:clearedDecommit>false</ns1:clearedDecommit>

<ns1:decommitReasonCode/>

<ns1:reportDate>2010-11-06T15:47:55-05:00</ns1:reportDate>

</ns1:StartupShutdown>

<ns1:StartupShutdown>

<ns1:rucType>DRUC</ns1:rucType>

<ns1:hourEnding>3</ns1:hourEnding>

<ns1:deliveryDate>2010-11-07</ns1:deliveryDate>

<ns1:resource>Res\_UNIT1</ns1:resource>

<ns1:clearedCommit>true</ns1:clearedCommit>

<ns1:clearedDecommit>false</ns1:clearedDecommit>

<ns1:decommitReasonCode/>

<ns1:reportDate>2010-11-06T15:47:55-05:00</ns1:reportDate>

</ns1:StartupShutdown>

<ns1:StartupShutdown>

<ns1:rucType>DRUC</ns1:rucType>

<ns1:hourEnding>04</ns1:hourEnding>

<ns1:deliveryDate>2010-11-07</ns1:deliveryDate>

<ns1:resource>Res\_UNIT1</ns1:resource>

<ns1:clearedCommit>true</ns1:clearedCommit>

<ns1:clearedDecommit>false</ns1:clearedDecommit>

<ns1:decommitReasonCode/>

<ns1:reportDate>2010-11-06T15:47:55-05:00</ns1:reportDate>

</ns1:StartupShutdown>

<ns1:StartupShutdown>

<ns1:rucType>DRUC</ns1:rucType>

<ns1:hourEnding>05</ns1:hourEnding>

<ns1:deliveryDate>2010-11-07</ns1:deliveryDate>

<ns1:resource>Res\_UNIT1</ns1:resource>

<ns1:clearedCommit>true</ns1:clearedCommit>

<ns1:clearedDecommit>false</ns1:clearedDecommit>

<ns1:decommitReasonCode/>

<ns1:reportDate>2010-11-06T15:47:55-05:00</ns1:reportDate>

</ns1:StartupShutdown>…

....

..

</StartupShutdownInstructions>

* XML notice examples for notices that contain deliveryHour or hourEnding in the message.

<ns1:ResponseMessage xmlns:ns1="<http://www.ercot.com/schema/2007-06/nodal/ews/message>">

                    <ns1:Header>

                        <ns1:Verb>created</ns1:Verb>

                        <ns1:Noun>Alert</ns1:Noun>

                        <ns1:ReplayDetection>

                            <ns1:Nonce>1c311e4c96d6e983a36ac3aac1b3116a</ns1:Nonce>

                            <ns1:Created>2010-11-07T00:02:48.024-05:00</ns1:Created>

                        </ns1:ReplayDetection>

                        <ns1:Revision>1.19O</ns1:Revision>

                        <ns1:Source>ERCOT</ns1:Source>

                        <ns1:UserID>USER1</ns1:UserID>

                        <ns1:MessageID>SCED-TWOHR-FAIL-QSE-2010-11-07T00:02:47.747-05:00</ns1:MessageID>

                    </ns1:Header>

                    <ns1:Reply>

                        <ns1:ReplyCode>OK</ns1:ReplyCode>

                        <ns1:Timestamp>2010-11-07T00:02:48.024-05:00</ns1:Timestamp>

                    </ns1:Reply>

                    <ns1:Payload>

                        <ns2:Event xmlns:ns2="<http://www.ercot.com/schema/2007-06/nodal/ews>">

                            <ns2:qse>QSE</ns2:qse>

                            <ns2:ID>SCED-TWOHR-FAIL</ns2:ID>

                            <ns2:type>Submission Warning</ns2:type>

                            <ns2:priority>Medium</ns2:priority>

                            <ns2:source>ERCOT</ns2:source>

                            <ns2:issued>2010-11-07T00:02:47.747-05:00</ns2:issued>

                            <ns2:summary>SCED-TWOHR-FAIL Two Hour Notification: Check FAILED due to missing Output Schedules/Energy Offer Curves for Resource: LPCCS\_CC2\_4, Hour: 02\* and Date: 11/07/2010</ns2:summary>

                        </ns2:Event>

                        <ns1:format>XML</ns1:format>

                    </ns1:Payload>

                </ns1:ResponseMessage>

Note: 02\* indicating DST extra hour

# 

1. Note that the bidID element isn’t defined with consistent letter case across all submission and award types. To verify the letter case, please refer to the corresponding XSD. [↑](#footnote-ref-1)
2. Note that the bidID element isn’t defined with consistent letter case across all submission and award types. To verify the letter case, please refer to the corresponding XSD. [↑](#footnote-ref-2)
3. Note that the bidID element isn’t defined with consistent letter case across all submission and award types. To verify the letter case, please refer to the corresponding XSD. [↑](#footnote-ref-3)
4. Note that the bidID element isn’t defined with consistent letter case across all submission and award types. To verify the letter case, please refer to the corresponding XSD. [↑](#footnote-ref-4)
5. Note that the bidID element isn’t defined with consistent letter case across all submission and award types. To verify the letter case, please refer to the corresponding XSD. [↑](#footnote-ref-5)
6. Note that the bidID element isn’t defined with consistent letter case across all submission and award types. To verify the letter case, please refer to the corresponding XSD. [↑](#footnote-ref-6)