***Load RT Zonal Prices***

***Query Response (Team Aricent)***

The query request should be formatted as outlined below and the request should be submitted to the MISO MOS System for processing:

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<env:Header/>

<env:Body>

<QueryResponse xmlns="http://markets.midwestiso.org/dart/xml">

<RealTimeZonalPrices type="xxx" interval="YYYY-MM-ddTHH:MI:SS">

<ReserveZonePrices zone="xx">

<GenRegMCP>xx</GenRegMCP>

<GenSpinMCP>xx</GenSpinMCP>

<GenSuppMCP>xx</GenSuppMCP>

<DemandRegMCP>xx</DemandRegMCP>

<DemandSpinMCP>xx</DemandSpinMCP>

<DemandSuppMCP>xx</DemandSuppMCP>

</ReserveZonePrices>

</RealTimeZonalPrices>

</QueryResponse>

</env:Body>

</env:Envelope>

***Description***

|  |  |  |  |
| --- | --- | --- | --- |
| **Field**  **MC\_PRICE\_RT** | Description | Source | Lookup/Notes |
| XP\_ID | Specifies the pricing node location name. | zone= “xxxx” | X\_XP |
| START\_DT | Specifies the market operating day date | interval="YYYY-MM-ddTHH:MI:SS" - 5 minutes | Stored as the start of the interval |
| DT\_INTRVL\_VAL |  | +00 00:05:00.000000 |  |
| SOURCE\_CD | Type specifies the ExPost or ExAnte prices being queried for depending on selection in IM | Load from Market screen dropdown | Type = “ExAnte” or "UDS' then 'ISOEXA'  Or  Type = “ExPost” or "LMP" then 'ISOEXP' | MC\_PRICE\_SOURCE\_CODE\_TYPE |
| PRICE\_TYPE\_CD | An identifier that uniquely identifies the type of price.    Unique price types are driven by the tags in the file. If all tags exist and are populated then 6 physical records will be created in the table each with a different price type. | <GenRegMCP> = RTGENREGMCP  <GenSpinMCP> = RTGENSPINMCP  <GenSuppMCP> = RTGENSUPPMCP  <DemandRegMCP> = RTDEMREGMCP  <DemandSpinMCP> = RTDEMSPINMCP  <DemandSuppMCP> = RTDEMSUPPMCP | MC\_PRICE\_RT\_TYPE |
| PRICE\_AMT | price in $ per MW  One physical record for each populated tag. | <GenRegMCP>  <GenSpinMCP>  <GenSuppMCP>  <DemandRegMCP>  <DemandSpinMCP>  <DemandSuppMCP> |  |
| USER\_ID | User id who loaded/updated/created the record | The user id who loaded the file. |  |
| REVISED\_DT | Date and time that the record was loaded/updated/created | The system date when the file was loaded. |  |

Processing

If price for a given start datetime/location/price type/source DOES NOT exist, INSERT the new record.

If price for a given start datetime/location/price type/source ALREADY exists, UPDATE the existing record.

Examples

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<env:Header/>

<env:Body>

<QueryResponse xmlns="http://markets.midwestiso.org/dart/xml">

<RealTimeZonalPrices type="LMP" interval="2009-01-01T01:05:00">

<ReserveZonePrices zone="1">

<GenRegMCP>10.00</GenRegMCP>

<GenSpinMCP>10.00</GenSpinMCP>

<GenSuppMCP>10.00</GenSuppMCP>

<DemandRegMCP>10.00</DemandRegMCP>

<DemandSpinMCP>10.00</DemandSpinMCP>

<DemandSuppMCP>10.00</DemandSuppMCP>

</ReserveZonePrices>

<ReserveZonePrices zone="2">

<GenRegMCP>10.00</GenRegMCP>

<GenSpinMCP>10.00</GenSpinMCP>

<GenSuppMCP>10.00</GenSuppMCP>

<DemandRegMCP>10.00</DemandRegMCP>

<DemandSpinMCP>10.00</DemandSpinMCP>

<DemandSuppMCP>10.00</DemandSuppMCP>

</ReserveZonePrices>

</RealTimeZonalPrices>

<RealTimeZonalPrices type="LMP" interval="2009-01-01T01:10:00">

<ReserveZonePrices zone="1">

<GenRegMCP>20.00</GenRegMCP>

<GenSpinMCP>20.00</GenSpinMCP>

<GenSuppMCP>20.00</GenSuppMCP>

<DemandRegMCP>20.00</DemandRegMCP>

<DemandSpinMCP>20.00</DemandSpinMCP>

<DemandSuppMCP>20.00</DemandSuppMCP>

</ReserveZonePrices>

<ReserveZonePrices zone="2">

<GenRegMCP>20.00</GenRegMCP>

<GenSpinMCP>20.00</GenSpinMCP>

<GenSuppMCP>20.00</GenSuppMCP>

<DemandRegMCP>20.00</DemandRegMCP>

<DemandSpinMCP>20.00</DemandSpinMCP>

<DemandSuppMCP>20.00</DemandSuppMCP>

</ReserveZonePrices>

</RealTimeZonalPrices>

</QueryResponse>

</env:Body>

</env:Envelope>

***Testing Approach***

|  |  |
| --- | --- |
| * + - **ISO LOADS** | **(A)utomated / (M)anual** |
| Connectivity - Download From Market | **On hold** |
| FileListener | **M** |
| Job Scheduler | **On hold** |
| Load From File | **M** |
| Verify price data appears correctly on View Prices screen. | **On hold** |
| Load from File: Verify the correct Load from File process is recognized. | **On hold** |
| Verify data is stored in the tables correctly. | **M** |
| 3 consecutive days' worth of data (Validate all 24 hours remain for all 3 days) validate 1st and the last hrs. are intact | **A** |
| Unique key validations for data integrity | **A** |
| Date- one day | **A** |
| File with All the Valid Types that are supported (price types, Source, Com\_CD etc) | **A** |
| XP\_Validate end-dated XP validation | **A** |
| Validate user is able to Delete ISO records on the UI - on hold | **On hold** |
| Validate the ~~Auditpath~~ and Load Summary window has correct information | **M** |
| Error handling- Duplicate data validation for data integrity | **A** |
| Error handling- Business Keys (required fields and unique keys) | **A** |
| Error handling - Formatting - look below | **A** |

***Error Handling***

* + Invalid Transaction Point. Skip record.
    - *No transaction point exists in the nMarket Database for the location name = <zone>.*
  + Invalid Date
    - *Invalid date passed from the file <interval> is not a valid value.*
  + Invalid Value
    - *Invalid price passed from the file <GenRegMCP>/<GenSpinMCP>/<GenSuppMCP>/<DemandRegMCP>/<DemandSpinMCP>/<DemandSuppMCP> is not a valid value.*