***Overview***

This API specification was designed to explain the process for transferring price information into nMarket from a participant’s external systems. A market participant can define the types of data sources for which they will be transferring prices into nMarket. One version of prices for each data source will be stored in nMarket for a transaction point and interval.

Supported Markets

* SPP
* MISO

***Process***

The nMarket API for Prices provides the ability for nMarket to receive data from a participant’s external system. The participant is responsible for sending the data according to the requirements specified in this API. Each nMarket product will store the price data in market defined tables. One version of prices can be stored for each data source, transaction point, per date and interval. Data Sources can be defined in nMarket Codes.

***General Conditions***

1. The combination of the following fields will be used to designate uniqueness on the price data file and must be included as part of the insert/update/delete key:

* Market
* Start Time
* End Time
* Transaction Point
* Price Type
* Price Source

2. The ACTION field with values of INSERT, UPDATE, or DELETE must be provided to determine what data manipulation (DML) should occur for the price data in nMarket. The DML is based on the unique key for the price data, which includes the fields listed above.

* If a new price data is entered into the external system, nMarket should receive a status of INSERT or UPDATE and the new price will be inserted into nMarket.
  + If the price records do not exist, the records will be inserted.
  + If the price records already exist in nMarket, the existing price records will be deleted and the new price records will be inserted.
* If price data is updated in the external system, nMarket should receive an action of UPDATE or INSERT. To update any non-key fields, such as price amount, an update needs to be sent with the updated value and the all other key values equal to the values in the original record.
* If price data is deleted in the external system, nMarket should receive an action of DELETE.
  + If the price records exist in nMarket, the price records will be physically deleted from the nMarket Database.
  + If the records sent by the external system are not found in nMarket, the following error message will be generated and no action will be performed for those price records that were not found. “Record cannot be deleted because the record does not exist for <TX\_PT> for the period <Start Date and Interval> <End Date and Interval>.”
* Any existing prices in nMarket will remain in nMarket unless a Delete is specifically issued by the external system.

3. Any tags included in the API file not explicitly defined by this document will be ignored.

***Process Code***

The process code for the nMarket Cross-Market Price API is XPRICE. This code will be used as the process name to identify the process in the File Listener and HTTP Listener, and Web Services.

***Trigger Events***

There are four ways to initiate the load of the Price XML file into nMarket.

* A manual load can be executed by any authorized user logged into nMarket via the load from file process.
* An automated load can be executed by the source system by writing a file to the file listener directory. nMarket will be equipped with a listener to look for the completed XML file and automatically load it into nMarket.
* A programmatic load can be executed by submitting a file to the Application Server via http.
* A programmatic load can be executed by submitting a file to the Application Server via Web Services.

***Rollback/Commits***

The load process will have two options when loading a file and encountering an error. In X\_IMPORTEXPORTCONFIG table, if RAISE\_ERROR\_NUM is set to 0, records that failed to load will not abort the load. If RAISE\_ERROR\_NUM is set to 1, records that failed to load will result in the load being rolled back.

***Versioning***

Price Information is not versioned in nMarket. If price data is sent through the API to be loaded into nMarket and previous prices exist in nMarket for the same price type, transaction point, price source, date and time the existing price data in nMarket will be replaced with the new prices.

***Price API Field Descriptions/Data Mapping***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **API Data Element** | **XML Tag** | **Req?** | **Domain / (Default)** | **Format** | **Mapping** | **Notes** |
| Market Code | MKT\_CD | Yes | SPPIM or SPP,  MISO | String |  | The market code associated w/ the scheduled job. The value must be set to the appropriate *market* to invoke the correct scheduler job in nMarket.  The market code in combination with PRICE\_TYP will indicate which tables the data needs to be loaded into. |
| Data Record Action | ACTION | Yes | INSERT, UPDATE, DELETE | String (6) | N/A | This is a required filed that tells nMarket if this record is a new record to be inserted, an update to a previously inserted record or a request to delete a previously inserted or updated record |
| Transaction Point | TX\_PT | Yes |  | String | XP\_ID | XP Name or ISOREF |
| Price Type | PRICE\_TYP | Yes | Price Type Definition table below | String | PRICE\_TYPE\_CD | The price type identifier.    The market code in combination with PRICE\_TYP will indicate which tables the data needs to be loaded into. |
| Price Source | PRICE\_SRC | Yes | Price Source Definition table below  Created in Codes | String | PRICE\_SOURCE\_CD | Identify which system the price data came from.    NOTE: API does not load for Price Source where Category = ISO. |
| Start Time | START\_TIME | Yes |  | String  YYYY-MMDDThh:  mm:sszz:ZZ | START\_DT\_GMT | The start date and time of the price record (the start hour or  beginning hour).  See Date/Time and Daylight Savings Time section for more  details. |
| End Time | END\_TIME | Yes |  | String  YYYY-MMDDThh:  mm:sszz:ZZ | END\_DT\_GMT | The end time of the price record (the end hour or ending  hour).  See Date/Time and Daylight Savings Time section for more  details. |
| Price Amount | PRICE\_AMT | Yes |  | Number (10v2) | PRICE\_AMT | Specifies the price in $ per MW.    NOTE: Allow negative values |
| Energy Component | ENERGY\_AMT | No |  | Number (10v2) | ENERGY\_AMT | Specifies the Energy Component in $ per MW.  NOTE: Value is only valid where price type is a LMP. |
| Congestion Component | CONGEST\_AMT | No |  | Number (10v2) | CONGEST\_AMT | Specifies the Congestion Component in $ per MW.  NOTE: Value is only valid where price type is a LMP. |
| Loss Component | LOSS\_AMT | No |  | Number (10v2) | LOSS\_AMT | Specifies the Loss Component in $ per MW.  NOTE: Value is only valid where price type is a LMP. |
| External ID | EXT\_ID | No |  | String | EXT\_PRICE\_ID | Optional field that is used to store a source system’s id that describes this record. |
| Row ID | ROW\_ID | No |  | String | N/A | Optional field that is used to identify this record in error messages. |

***Price Type Definition***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Market** | **Price Type** | **Interval** | **Description** | **Source Database Table** | **Target Database Table** |
| SPP | DALMP | Hourly | Day Ahead Location Marginal Price | SPP\_PRICE\_DA\_TYPE | SPP\_PRICE\_DA |
| SPP | DAREGUP | Hourly | Day Ahead Regulation Up Market Clearing Price | SPP\_PRICE\_OTHER\_TYPE | SPP\_PRICE\_OTHER |
| SPP | DAREGDN | Hourly | Day Ahead Regulation Down Market Clearing Price | SPP\_PRICE\_OTHER\_TYPE | SPP\_PRICE\_OTHER |
| SPP | DASPIN | Hourly | Day Ahead Spinning Reserve Market Clearing Price | SPP\_PRICE\_OTHER\_TYPE | SPP\_PRICE\_OTHER |
| SPP | DASUPP | Hourly | Day Ahead Supplemental Reserve market Clearing Price | SPP\_PRICE\_OTHER\_TYPE | SPP\_PRICE\_OTHER |
| SPP | RTLMP | 5-Minute | Real-Time Locational Marginal Price | SPP\_PRICE\_RT\_TYPE | SPP\_PRICE\_RT |
| SPP | RTREGUP | 5-Minute | Real-Time Regulation Up Market Clearing Price | SPP\_PRICE\_OTHER\_TYPE | SPP\_PRICE\_OTHER |
| SPP | RTREGDN | 5-Minute | Real-Time Regulation Down Market Clearing Price | SPP\_PRICE\_OTHER\_TYPE | SPP\_PRICE\_OTHER |
| SPP | RTSPIN | 5-Minute | Real-Time Spinning Reserve Market Clearing Price | SPP\_PRICE\_OTHER\_TYPE | SPP\_PRICE\_OTHER |
| SPP | RTSUPP | 5-Minute | Real-Time Supplemental Reserve Market Clearing Price | SPP\_PRICE\_OTHER\_TYPE | SPP\_PRICE\_OTHER |
| MC | DALMP | Hourly | Day Ahead LMP | MC\_PRICE\_DA\_TYPE | MC\_PRICE\_DA |
| MC | DAREGMCP | Hourly | Day Ahead Regulation Market Clearing Price | MC\_PRICE\_DA\_TYPE | MC\_PRICE\_DA |
| MC | DASPINMCP | Hourly | Day Ahead Spinning Reserve Market Clearing Price | MC\_PRICE\_DA\_TYPE | MC\_PRICE\_DA |
| MC | DASUPPMCP | Hourly | Day Ahead Supplemental Reserve Market Clearing Price | MC\_PRICE\_DA\_TYPE | MC\_PRICE\_DA |
| MC | DAGENREGMCP | Hourly | Day Ahead Gen-based Regulation Market Clearing Price | MC\_PRICE\_DA\_TYPE | MC\_PRICE\_DA |
| MC | DAGENSPINMCP | Hourly | Day Ahead Gen-based Spinning Reserve Market Clearing Price | MC\_PRICE\_DA\_TYPE | MC\_PRICE\_DA |
| MC | DAGENSUPPMCP | Hourly | Day Ahead Gen-based Supplemental Reserve Market Clearing Price | MC\_PRICE\_DA\_TYPE | MC\_PRICE\_DA |
| MC | DADEMREGMCP | Hourly | Day Ahead Demand-based Regulation Market Clearing Price | MC\_PRICE\_DA\_TYPE | MC\_PRICE\_DA |
| MC | DADEMSPINMCP | Hourly | Day Ahead Demand-based Spinning Reserve Market Clearing Price | MC\_PRICE\_DA\_TYPE | MC\_PRICE\_DA |
| MC | DADEMSUPPMCP | Hourly | Day Ahead Demand-based Supplemental Reserve Market Clearing Price | MC\_PRICE\_DA\_TYPE | MC\_PRICE\_DA |
| MC | ROUND1ONPEAKSHADOWPRICE | Monthly | Shadow Price Round 1 On Peak | MC\_PRICE\_FTR\_TYPE | MC\_PRICE\_FTR |
| MC | ROUND1OFFPEAKSHADOWPRICE | Monthly | Shadow Price Round 1 Off Peak | MC\_PRICE\_FTR\_TYPE | MC\_PRICE\_FTR |
| MC | ROUND2ONPEAKSHADOWPRICE | Monthly | Shadow Price Round 2 On Peak | MC\_PRICE\_FTR\_TYPE | MC\_PRICE\_FTR |
| MC | ROUND2OFFPEAKSHADOWPRICE | Monthly | Shadow Price Round 2 Off Peak | MC\_PRICE\_FTR\_TYPE | MC\_PRICE\_FTR |
| MC | ROUND3ONPEAKSHADOWPRICE | Monthly | Shadow Price Round 3 On Peak | MC\_PRICE\_FTR\_TYPE | MC\_PRICE\_FTR |
| MC | ROUND3OFFPEAKSHADOWPRICE | Monthly | Shadow Price Round 3 Off Peak | MC\_PRICE\_FTR\_TYPE | MC\_PRICE\_FTR |
| MC | ROUND4ONPEAKSHADOWPRICE | Monthly | Shadow Price Round 4 On Peak | MC\_PRICE\_FTR\_TYPE | MC\_PRICE\_FTR |
| MC | ROUND4OFFPEAKSHADOWPRICE | Monthly | Shadow Price Round 4 Off Peak | MC\_PRICE\_FTR\_TYPE | MC\_PRICE\_FTR |
| MC | MONTHLYONPEAKSHADOWPRICE | Monthly | Shadow Price Monthly On Peak | MC\_PRICE\_FTR\_TYPE | MC\_PRICE\_FTR |
| MC | MONTHLYOFFPEAKSHADOWPRICE | Monthly | Shadow Price Monthly Off Peak | MC\_PRICE\_FTR\_TYPE | MC\_PRICE\_FTR |
| MC | FLOWGATESHADOWPRICE | Hourly | Flowgate Shadow Price | MC\_PRICE\_OTHER\_TYPE | MC\_PRICE\_OTHER |
| MC | 5MINRTLMP | 5-Minute | 5-Minute Real-Time Locational Marginal Price | MC\_PRICE\_RT\_TYPE | MC\_PRICE\_RT |
| MC | 5MINRTREGMCP | 5-Minute | 5-Minute Real-Time Regulation Market Clearing Price | MC\_PRICE\_RT\_TYPE | MC\_PRICE\_RT |
| MC | 5MINRTREGMILMCP | 5-Minute | 5-Minute Real-Time Regulation Mileage Market Clearing Price | MC\_PRICE\_RT\_TYPE | MC\_PRICE\_RT |
| MC | 5MINRTSPINMCP | 5-Minute | 5-Minute Real-Time Spinning Reserve Market Clearing Price | MC\_PRICE\_RT\_TYPE | MC\_PRICE\_RT |
| MC | 5MINRTSUPPMCP | 5-Minute | 5-Minute Real-Time Supplemental Reserve Market Clearing Price | MC\_PRICE\_RT\_TYPE | MC\_PRICE\_RT |
| MC | RTGENREGMCP | 5-Minute | Real-Time Gen-based Regulation Market Clearing Price | MC\_PRICE\_RT\_TYPE | MC\_PRICE\_RT |
| MC | RTGENSPINMCP | 5-Minute | Real-Time Gen-based Spinning Reserve Market Clearing Price | MC\_PRICE\_RT\_TYPE | MC\_PRICE\_RT |
| MC | RTGENSUPPMCP | 5-Minute | Real-Time Gen-based Supplemental Reserve Market Clearing Price | MC\_PRICE\_RT\_TYPE | MC\_PRICE\_RT |
| MC | RTDEMREGMCP | 5-Minute | Real-Time Demand-based Regulation Market Clearing Price | MC\_PRICE\_RT\_TYPE | MC\_PRICE\_RT |
| MC | RTDEMSPINMCP | 5-Minute | Real-Time Demand-based Spinning Reserve Market Clearing Price | MC\_PRICE\_RT\_TYPE | MC\_PRICE\_RT |
| MC | RTDEMSUPPMCP | 5-Minute | Real-Time Demand-based Supplemental Reserve Market Clearing Price | MC\_PRICE\_RT\_TYPE | MC\_PRICE\_RT |
| MC | RTLMP | Hourly | Real-Time Integrated Locational Marginal Price | MC\_PRICE\_RT\_TYPE | MC\_PRICE\_RT |
| MC | RTREGMCP | Hourly | Real-Time Integrated Regulation Market Clearing Price | MC\_PRICE\_RT\_TYPE | MC\_PRICE\_RT |
| MC | RTSPINMCP | Hourly | Real-Time Integrated Spinning Reserve Market Clearing Price | MC\_PRICE\_RT\_TYPE | MC\_PRICE\_RT |
| MC | RTSUPPMCP | Hourly | Real-Time Integrated Supplemental Reserve Market Clearing Price | MC\_PRICE\_RT\_TYPE | MC\_PRICE\_RT |

***Price Source Definition***

|  |  |  |  |
| --- | --- | --- | --- |
| **Market** | **Source** | **Category** | **Source Database Table** |
| SPP | ISO | ISO | SPP\_PRICE\_SOURCE\_CD\_TYPE |
| SPP | INTERNAL | INTRNL | SPP\_PRICE\_SOURCE\_CD\_TYPE |
| MC | ISOEXA | ISO | MC\_PRICE\_SOURCE\_CD\_TYPE |
| MC | ISOEXP | ISO | MC\_PRICE\_SOURCE\_CD\_TYPE |
| MC | INTERNAL | INTRNL | MC\_PRICE\_SOURCE\_CD\_TYPE |
| MC | ISO | ISO | MC\_PRICE\_SOURCE\_CD\_TYPE |

NOTE: User can add additional Internal Source Codes via the Codes window.

***Error Handling***

All errors generated by the API will be identified with the following parameters from the erroneous record (if populated): row\_id=<row\_id>, ext\_id=<ext\_id>, mkt\_cd=<mkt\_cd>, tx\_pt =<tx\_pt>, price\_typ=<price\_typ>, start\_time=<start\_time>, end\_time=<end\_time>, price\_src=<price\_src>

***Errors***

If the following conditions exist, the record for that interval should not be loaded and the following error messages should be generated:

* Another Price Load occurring
  + Process Already Locked. Unable to continue the process XPRICE since the process is currently locked by another transaction.
* Action is invalid
  + Invalid Data passed from the file. Action <action> is invalid.
* Action is missing
  + Insufficient number of parameters is being passed. Action is a required field.
* Delete action is invalid
  + Invalid Data passed from the file. The action code passed is DELETE but no matching data exists.
* Market is invalid
  + Invalid Data passed from the file. Market <mkt\_cd> is invalid.
* Market is missing
  + Insufficient number of parameters is being passed. Market is a required field.
* Transaction point is missing
  + Insufficient number of parameters is being passed. Transaction Point is a required field.
* Transaction point is invalid
  + Invalid Data passed from the file. Transaction Point <tx\_pt> is invalid.
* Price Type is missing
  + Insufficient number of parameters is being passed. Price Type is a required field.
* Price Type is invalid
  + Invalid Data passed from the file. Price Type is invalid.
* Price Type and Transaction Point combination is invalid
  + Invalid Data passed from the file. Price Type and Transaction Point is invalid.
* Start Time is missing
  + Insufficient number of parameters is being passed. Start Time is a required field.
* Start Time is invalid
  + Invalid Data passed from the file. Start Time is invalid.
* End Time is missing
  + Insufficient number of parameters is being passed. End Time is required.
* End Time is invalid
  + Invalid Data passed from the file. End Time is invalid.
* Invalid Time Combination (i.e. End Time starts before Start Time)
  + Invalid Data passed from the file. Start Date <start\_dt> and Start Time <start\_tm> is more than/equal to the End Date <end\_dt> and End Time <end\_tm>.
* Price Source is missing
  + Insufficient number of parameters is being passed. Price Source is required.
* Price Source is invalid
  + Invalid Data passed from the file. Price Source is invalid.
* Price Amount is missing
  + Insufficient number of parameters is being passed. Invalid Price Amount for Start Date "<start\_dt>" Start Time "<start\_time>" and End Date "<end\_dt>" End Time "<end\_time>".
* Price Amount is invalid (incorrect number of digits or combination with decimal point)
  + Invalid Data passed from the file. Invalid Price Amount for Start Date "<start\_dt>" Start Time "<start\_time>" and End Date "<end\_dt>" End Time "<end\_time>". Price Amount = <price\_amt>
* Non-Numeric Price Amount (incorrect number of digits or combination with decimal point)
  + Invalid Data passed from the file. Invalid Price Amount for Start Date "<start\_dt>" Start Time "<start\_time>" and End Date "<end\_dt>" End Time "<end\_time>". Price Amount = <price\_amt>
* Energy Amount is not null where Price Type is not LMP
  + Invalid Data passed from the file. Energy Amount is invalid.
* Congestion Amount is not null where Price Type is not LMP
  + Invalid Data passed from the file. Congestion Amount is invalid.
* Loss Amount is not null where Price Type is not LMP
  + Invalid Data passed from the file. Loss Amount is invalid.

***Data File Specification (DTD)***

<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT CROSS\_MARKET (MKT\_CD, DOCUMENT)>

<!ELEMENT MKT\_CD (#PCDATA)>

<!ELEMENT DOCUMENT (LOCATION+)>

<!ELEMENT LOCATION (ACTION, TX\_PT, PRICE\_TYP, PRICE\_SRC, PRICES)>

<!ELEMENT TX\_PT (#PCDATA)>

<!ELEMENT PRICE\_TYP (#PCDATA)>

<!ELEMENT PRICE\_SRC (#PCDATA)>

<!ELEMENT ACTION (#PCDATA)>

<!ELEMENT PRICES (PRICE+)>

<!ELEMENT PRICE (START\_TIME, END\_TIME, PRICE\_AMT, ENERGY\_AMT?, CONGEST\_AMT?,

LOSS\_AMT?, EXT\_ID?, ROW\_ID?)>

<!ELEMENT START\_TIME (#PCDATA)>

<!ELEMENT END\_TIME (#PCDATA)>

<!ELEMENT PRICE\_AMT (#PCDATA)>

<!ELEMENT ENERGY\_AMT (#PCDATA)>

<!ELEMENT CONGEST\_AMT (#PCDATA)>

<!ELEMENT LOSS\_AMT (#PCDATA)>

<!ELEMENT EXT\_ID (#PCDATA)>

<!ELEMENT ROW\_ID (#PCDATA)>

***Sample Files***

Below are examples of the Cross-Market Price file per market:

***SPP***

<CROSS\_MARKET>

<MKT\_CD>SPP</MKT\_CD>

<DOCUMENT>

<LOCATION>

<ACTION>INSERT</ACTION>

<TX\_PT>TransactionPoint</TX\_PT>

<PRICE\_TYP>DALMP</PRICE\_TYP>

<PRICE\_SRC>INTERNAL</PRICE\_SRC>

<PRICES>

<PRICE>

<START\_TIME>2013-01-01T00:00:00-06:00</START\_TIME>

<END\_TIME>2013-01-02T00:00:00-06:00</END\_TIME>

<PRICE\_AMT>6.00</PRICE\_AMT>

<ENERGY\_AMT></ENERGY\_AMT>

<CONGEST\_AMT></CONGEST\_AMT>

<LOSS\_AMT></LOSS\_AMT>

<EXT\_ID>00001</EXT\_ID>

<ROW\_ID>1</ROW\_ID>

</PRICE>

<PRICE>

<START\_TIME>2013-01-02T00:00:00-06:00</START\_TIME>

<END\_TIME>2013-01-03T00:00:00-06:00</END\_TIME>

<PRICE\_AMT>7.00</PRICE\_AMT>

<ENERGY\_AMT></ENERGY\_AMT>

<CONGEST\_AMT></CONGEST\_AMT>

<LOSS\_AMT></LOSS\_AMT>

<EXT\_ID>00002</EXT\_ID>

<ROW\_ID>2</ROW\_ID>

</PRICE>

</PRICES>

</LOCATION>

<LOCATION>

<ACTION>INSERT</ACTION>

<TX\_PT>TransactionPoint2</TX\_PT>

<PRICE\_TYP>DALMP</PRICE\_TYP>

<PRICE\_SRC>INTERNAL</PRICE\_SRC>

<PRICES>

<PRICE>

<START\_TIME>2013-01-01T00:00:00-06:00</START\_TIME>

<END\_TIME>2013-01-02T00:00:00-06:00</END\_TIME>

<PRICE\_AMT>6.00</PRICE\_AMT>

<ENERGY\_AMT></ENERGY\_AMT>

<CONGEST\_AMT></CONGEST\_AMT>

<LOSS\_AMT></LOSS\_AMT>

<EXT\_ID>00001</EXT\_ID>

<ROW\_ID>1</ROW\_ID>

</PRICE>

<PRICE>

<START\_TIME>2013-01-02T00:00:00-06:00</START\_TIME>

<END\_TIME>2013-01-03T00:00:00-06:00</END\_TIME>

<PRICE\_AMT>7.00</PRICE\_AMT>

<ENERGY\_AMT></ENERGY\_AMT>

<CONGEST\_AMT></CONGEST\_AMT>

<LOSS\_AMT></LOSS\_AMT>

<EXT\_ID>00002</EXT\_ID>

<ROW\_ID>2</ROW\_ID>

</PRICE>

</PRICES>

</LOCATION>

</DOCUMENT>

</CROSS\_MARKET>

***File*** ***Naming Convention***

The XML file generated as part of this API should be named in the following manner:

mkt\_XPRICE\_ YYYYMMDD[HH][MM].xml

Example: = SPP\_XPRICE\_200107100845.xml

***Outstanding Questions***

-Expanding days/hours

***Testing Approach***

|  |  |
| --- | --- |
| * + **API LOADS** | **(A)utomated / (M)anual** |
| Load from File | M |
| FileListener | On hold |
| Inclusive of all optional & required fields | A |
| INSERT, UPDATE, DELETE (all day & partial) | A |
| Only Required fields | A |
| Unique Key Validation (verifying duplications don’t exist) (ex: per Mkt/Transaction Point/Price type/Source) | M |
| Validate end-dated XP validation | M |
| Validate ISO & Internal API duplicate data with different Sources can exist w/o wiping each other out | M |
| Validate Start Date is always less than the End Date | M |
| Validate each valid Participant type for all valid XP types | M |
| Error handling- Duplicate data validation for data integrity | M |
| Error handling - Formatting | M |
| Validate the Expanding of multiple hours / days | A |
| Test on Fall DST day - MISO should only have 24 hours worth of records, other markets 25 hours worth of records | A |
| Test on Spring DST day - MISO should only have 24 hours worth of records, other markets 23 hours worth of records | A |