**Operational Parameters API**

*Overview*

*Design Details*

**Process Code**

The process code for Operational Parameters Inbound API is **APIOPM**. This code will be used as the process name to identify the process in the File Listener and HTTP Listener.

**Trigger Events**

There are three ways in which the interface can be initiated to load the XML file:

* 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 an offer file to the file listener directory. nMarket will be equipped with a listener to look for the completed XML file and automatically load it.

|  |  |
| --- | --- |
| **Note** | Look in the MWFileListener.properties file to determine the location of the **File Listener** directory. |

* A programmatic load can be executed by submitting a file to the Application Server via http listener.

**General Conditions**

This API provides the ability for nMarket Midwest to receive Default Operational Parameters and Combined Cycle Status information from a participant’s external system. The participant is responsible for sending the schedule data according to the requirements specified in this API.

**Process**

The nMarket API for Operational Parameters provides the ability for nMarket to receive Combined Cycle Status and Default Operational Parameter information for units from a participant’s external system. The participant is responsible for sending data according to the requirements specified in this API.

The records are stored in nMarket in the MC\_OPPARM\_COMBINED\_CYCLE and MC\_OPPARM\_DFLT tables.

**General** **Conditions**

1. The ACTION field with values of INSERT, UPDATE, or DELETE has to be provided to determine which data manipulation should occur for that Parameter in nMarket. The DML is based on the unique key for the Offer Data. The unique key includes the following fields:

* Offer:
  + Participant Code
  + Transaction Point
  + Effective Date
  + Source
  + Clearing Code (Default Op Parms Only)

1. The Combined Cycle and Default Operational Parameters parts of the API are maintained independent of each other.
2. If new data is entered into the external system, nMarket will receive an action of INSERT or UPDATE and the new data will be inserted into nMarket with a status of NEW.

* If the data does not exist for the Participant, Transaction Point, Effective Date, Source, [Clearing Code] then the record will be inserted.
* If data exists for the Participant, Transaction Point, Effective Date, Source, [Clearing Code] update non key fields based on the unique keys.

1. If a record is deleted in the external system, nMarket will receive an action of DELETE.
   * If the record exists in nMarket, the record will be physically deleted from the nMarket Database.
   * If the record sent by the external system is not found in nMarket, the following error message will be generated and no action will be performed for that record.

“Record cannot be deleted because the record does not exist for <XP\_ID> for the period <effective date>.”

* Any existing record in nMarket will remain active unless a Delete is specifically issued by the external system.

1. The API can update any internal records in the database.
2. Multiple actions should not be included in the same file for records having the same key values. For instance, it is not permissible to delete and then re-insert records for the same key values in the same API file. If such an operation is required, it should be performed in two separate API files.
3. To update an existing record all parts must be sent by the API, there are no partial updates. If a tag is missing or null the corresponding value in the Database will be removed.
4. Any tags included in the API file not explicitly defined by this document should be ignored.

**Versioning**

Generation Offers are not versioned in nMarket.

**API**

* The internal status of Combined Cycle and Default records are maintained separately using the following logic:
* The first time a record is inserted it will be assigned a status of NEW.
* If that record is updated before being submitted, the status of the record will remain NEW.
* If the record is updated after being submitted, the status of the record will change to UPDATED
* If the record being updated has the status of UPDATED, ERROR or UNKNOWN then the record will be updated and the status will change to UPDATED
* If the updated record is deleted, the record will by physically deleted from the database.
* If the SUBMITTED record is deleted then
  + If the DELSUB Configuration Parameter is set to No then this should result in an error.
  + If the DELSUB Configuration Parameter is set to Yes then the record will be physically deleted from the database

**Generation Offer API Elements/Data Mapping**

Below is the high-level interface specification. It is a listing of all the data elements that can be specified by the user in an XML file, and how those

elements relate back to the nMarket data model.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Req | Valid Values | Format | nMarket Database Mapping | Description/Comments |
| ACTION | Y | INSERT UPDATE DELETE | String | N/A |  |
| PTCPT\_CD | Y | Valid Code defined in Participants | String | MC\_OPPARM\_COMBINED\_CYCLE MC\_OPPARM\_DFLT |  |
| TX\_PT | Y | Valid Name or ISO Ref defined in Transactions points | String | MC\_OPPARM\_COMBINED\_CYCLE MC\_OPPARM\_DFLT | Verify whether only owned assets or not. |
| EFFECTIVE\_DATE | Y |  | String  (yyyy-mm-dd) | MC\_OPPARM\_COMBINED\_CYCLE MC\_OPPARM\_DFLT | Start Date of the effective period of the operational parameters. |
| SOURCE | Y | INTERNAL | String | MC\_OPPARM\_COMBINED\_CYCLE MC\_OPPARM\_DFLT | Source Code: A code that indicates the source of the record.  NEED TO FOLLOW UP WITH JIM WITH THIS DO NOT SEE A FK SET UP |
| ROW\_ID |  |  | String | N/A |  |
| COMBINED CYCLE |  |  |  | N/A | Group header tag Note: This section should only be included for TX\_PTs of type Combined Cycle Resource (CCR) |
| STATUS |  | AGGREGATE INDIVIDUAL | String | MC\_OPPARM\_COMBINED\_CYCLE.ISO\_STATUS\_ID | Status of Combined Cycle Unit Valid values stored in MC\_OPPARM\_COMBINED\_CYCLE\_TYPE |
| EXT\_ID |  |  | String | MC\_OPPARM\_COMBINED\_CYCLE | Combined Cycle Status Op Parm External ID: An identifier assigned by the initiator of the operation is echoed back in the response. |
| COMMENTS |  |  | String | MC\_OPPARM\_COMBINED\_CYCLE | Combined Cycle Status User Comments: Used to store message generated by the User(s). |
| DEFAULTS |  |  |  | N/A | Group header tag |
| CLRG\_CD | Y | DA RT BOTH | String | MC\_OPPARM\_DFLT |  |
| MIN\_EMERGENCY\_LIMIT |  |  | Number | MC\_OPPARM\_DFLT | Min Emergency Limit: The minimum MW level at which a Resource may operate under Emergency system conditions. |
| MIN\_ECONOMIC\_LIMIT |  |  | Number | MC\_OPPARM\_DFLT | Min Economic Limit: The minimum MW level at which a Resource may operate under Economic system conditions. |
| MIN\_REGULATION\_LIMIT |  |  | Number | MC\_OPPARM\_DFLT | Min Regulation Limit: The minimum MW level at which a Regulation Qualified Resource may operate while providing Regulation Deployment. |
| MAX\_REGULATION\_LIMIT |  |  | Number | MC\_OPPARM\_DFLT | Max Regulation Limit: The maximum MW level at which a Regulation Qualified Resource may operate while providing Regulation Deployment. |
| MAX\_ECONOMIC\_LIMIT |  |  | Number | MC\_OPPARM\_DFLT | Max Economic Limit: The maximum MW level at which a Resource may operate under Economic system conditions. |
| MAX\_EMERGENCY\_LIMIT |  |  | Number | MC\_OPPARM\_DFLT | Max Emergency Limit: The maximum MW level at which a Resource may operate under Emergency system conditions. |
| OFFLINE\_RESOURCE\_LIMIT |  |  | Number | MC\_OPPARM\_DFLT | Offline Resource Limit: The maximum MW level at which a Resource may operate when offering to provide Offline Supplemental Reserves. |
| NO\_LOAD\_COST |  |  | Number | MC\_OPPARM\_DFLT | No Load Cost: No Load Offer. |
| COLD\_STARTUP\_COST |  |  | Number | MC\_OPPARM\_DFLT | Cold Startup Cost: Cold Start-up Offer. |
| INTER\_STARTUP\_COST |  |  | Number | MC\_OPPARM\_DFLT | Inter Startup Cost: Intermediate Start-up Offer. |
| HOT\_STARTUP\_COST |  |  | Number | MC\_OPPARM\_DFLT | Hot Startup Cost: Hot Start-up Offer. |
| RAMP\_RATE |  |  | Number | MC\_OPPARM\_DFLT | Ramp Rate: Single Ramp Rate value in MW/minute. |
| RAMP\_RATE\_UP |  |  | Number | MC\_OPPARM\_DFLT | Ramp Rate Up: Up Ramp Rate used in RT Market. |
| RAMP\_RATE\_DOWN |  |  | Number | MC\_OPPARM\_DFLT | Ramp Rate Down: Down Ramp Rate used in RT Market. |
| RAMP\_RATE\_BIDIRECTIONAL |  |  | Number | MC\_OPPARM\_DFLT | Ramp Rate Bidirectional: Bi-Directional or Oscillating Ramp Rate. |
| COMMITMENT\_STATUS |  | ECONOMIC EMERGENCY  MUSTRUN  OUTAGE  NOTPARTICIPATING  AVAILABLE  UNAVAILABLE | String | MC\_OPPARM\_DFLT | Commitment Status Code: Commitment Status for Energy Offer.  Valid values stored in MC\_OP\_COMMITMENT\_STS\_TYPE. |
| EO\_DISPATCH\_STATUS |  | ECONOMIC SELFSCHEDULE | String | MC\_OPPARM\_DFLT | EO Dispatch Status Code: Dispatch Status for Energy Offer. Valid value stored in MC\_OP\_DISPATCH\_STS\_EO\_TYPE. |
| REG\_DISPATCH\_STATUS |  | ECONOMIC SELFSCHEDULE NOTQUALIFIED NOTPARTICIPATING | String | MC\_OPPARM\_DFLT | REG Dispatch Status Code: Regulation Offer Dispatch Status. Valid value stored in MC\_OP\_DISPATCH\_STS\_REG\_TYPE. |
| SPIN\_DISPATCH\_STATUS |  | ECONOMIC SELFSCHEDULE EMERGENCY NOTQUALIFIED NOTPARTICIPATING | String | MC\_OPPARM\_DFLT | Spin Dispatch Status Code: Spinning Reserve Offer Dispatch Status. Valid value stored in MC\_OP\_DISPATCH\_STS\_SPIN\_TYPE. |
| SUPP\_ON\_DISPATCH\_STATUS |  | ECONOMIC SELFSCHEDULE EMERGENCY NOTQUALIFIED NOTPARTICIPATING | String | MC\_OPPARM\_DFLT | Supp On Dispatch Status Code: Online Supplemental Offer Dispatch Status. Valid value stored in MC\_OP\_DISPATCH\_STS\_SUPON\_TYPE. |
| SUPP\_OFF\_DISPATCH\_STATUS |  | ECONOMIC SELFSCHEDULE EMERGENCY NOTQUALIFIED NOTPARTICIPATING | String | MC\_OPPARM\_DFLT | Supp Off Dispatch Status Code: Offline Supplemental Offer Dispatch Status. Valid value stored in MC\_OP\_DISPATCH\_STS\_SUPOFF\_TYPE. |
| ECO\_LOW\_LIMIT\_MW |  |  | Number | MC\_OPPARM\_DFLT | Economic Low Limit MW: Temperature based Economic Low Limit MW. |
| ECO\_LOW\_LIMIT\_TEMP |  |  | Number | MC\_OPPARM\_DFLT | Economic Low Limit Temp: Temperature based Economic Low Limit Temp. |
| ECO\_MID\_LIMIT\_MW |  |  | Number | MC\_OPPARM\_DFLT | Economic Mid Limit MW: Temperature based Economic Mid Limit MW. |
| ECO\_MID\_LIMIT\_TEMP |  |  | Number | MC\_OPPARM\_DFLT | Economic Mid Limit Temp: Temperature based Economic Mid Limit Temp. |
| ECO\_HIGH\_LIMIT\_MW |  |  | Number | MC\_OPPARM\_DFLT | Economic High Limit MW: Temperature based Economic High Limit MW. |
| ECO\_HIGH\_LIMIT\_TEMP |  |  | Number | MC\_OPPARM\_DFLT | Economic High Limit Temp: Temperature based Economic High Limit Temp. |
| EMGCY\_LOW\_LIMIT\_MW |  |  | Number | MC\_OPPARM\_DFLT | Emergency Low Limit MW: Temperature based Emergency Low Limit MW. |
| EMGCY\_LOW\_LIMIT\_TEMP |  |  | Number | MC\_OPPARM\_DFLT | Emergency Low Limit Temp: Temperature based Emergency Low Limit Temp. |
| EMGCY\_MID\_LIMIT\_MW |  |  | Number | MC\_OPPARM\_DFLT | Emergency Mid Limit MW: Temperature based Emergency Mid Limit MW. |
| EMGCY\_MID\_LIMIT\_TEMP |  |  | Number | MC\_OPPARM\_DFLT | Emergency Mid Limit Temp: Temperature based Emergency Mid Limit Temp. |
| EMGCY\_HIGH\_LIMIT\_MW |  |  | Number | MC\_OPPARM\_DFLT | Emergency High Limit MW: Temperature based Emergency High Limit MW. |
| EMGCY\_HIGH\_LIMIT\_TEMP |  |  | Number | MC\_OPPARM\_DFLT | Emergency High Limit Temp: Temperature based Emergency High Limit Temp. |
| REG\_LOW\_LIMIT\_MW |  |  | Number | MC\_OPPARM\_DFLT | Emergency Low Limit MW: Temperature based Emergency Low Limit MW. |
| REG\_LOW\_LIMIT\_TEMP |  |  | Number | MC\_OPPARM\_DFLT | Emergency Low Limit Temp: Temperature based Emergency Low Limit Temp. |
| REG\_MID\_LIMIT\_MW |  |  | Number | MC\_OPPARM\_DFLT | Emergency Mid Limit MW: Temperature based Emergency Mid Limit MW. |
| REG\_MID\_LIMIT\_TEMP |  |  | Number | MC\_OPPARM\_DFLT | Emergency Mid Limit Temp: Temperature based Emergency Mid Limit Temp. |
| REG\_HIGH\_LIMIT\_MW |  |  | Number | MC\_OPPARM\_DFLT | Emergency High Limit MW: Temperature based Emergency High Limit MW. |
| REG\_HIGH\_LIMIT\_TEMP |  |  | Number | MC\_OPPARM\_DFLT | Emergency High Limit Temp: Temperature based Emergency High Limit Temp. |
| MAX\_CR\_DEPLOY\_MW |  |  | Number | MC\_OPPARM\_DFLT | Max Regulation Down Deployment. |
| MAX\_REG\_UP\_DEPLOY\_MW |  |  | Number | MC\_OPPARM\_DFLT |  |
| MAX\_REG\_DOWN\_DEPLOY\_MW |  |  | Number | MC\_OPPARM\_DFLT |  |
| EXT\_ID |  |  | String | MC\_OPPARM\_DFLT | Default Op Parm External ID: An identifier assigned by the initiator of the operation is echoed back in the response. |
| COMMENTS |  |  | String | MC\_OPPARM\_DFLT | Default Op Parms User Comments: Used to store message generated by the User(s). |

*Interface Business Logic*

**Common**

1. Validate the data format. Check for syntactical errors in the XML file as well as incorrect data types. Write error messages to error logs where specified validation criteria are not met. If any required fields are missing, log the appropriate error and skip processing for that record.
2. If a single API file contains more than one participant, then the API data will load properly, but any errors will only be made to the first participant in the API file. In order for nMarket to process the errors by participant, a separate API file must be created for each participant.
3. Records that do not meet validation criteria will not be loaded into nMarket. Instead, the appropriate error message should be triggered.
4. For default parameters, if the action of ‘UPDATE’ is sent – the entire record will have to be resent.
5. For combined cycle status, if the action of ‘UPDATE’ is sent – the entire record will have to be sent.

**Assumptions**

All records will be sent with INSERT/UPDATE/DELETE action.

**Error Handling**

**Error**

* Invalid or missing participant code:

*Participant code could not be obtained or missing. Participant code does not exist in the database with the code <PTCPT\_CD>.*

* Invalid or missing Effective Date:

*Invalid or missing data passed from file. <Effective\_Date> is an invalid value.*

* Invalid or missing transaction point:

*No transaction point exists in the nMarket database for the node name = <TX\_PT> on operating day = <operating\_day>.*

* Invalid or missing action code:

*Invalid or missing data passed from file. <ACTION\_CD> is not a valid action code.*

* Invalid or missing Source:

*Invalid or missing data passed from file. <source> is not a valid source.*

* Invalid or missing clearing code:

*Invalid or missing data passed from file. <CLRG\_CD> is not a valid clearing code.*

* Invalid delete request under circumstance. This is applicable only if the DELSUB config parm is set to N. If set to Y, then delete is allowed.

*Offer cannot be deleted because the record has been submitted for transaction point: <XP\_ID> for the period <operating hour>.*

* Invalid Combined Cycle Status:

*Invalid data passed from file. <STATUS> is not a valid combined cycle status.*

* Invalid Commitment Status:

*Invalid passed from file. <COMMITMENT\_STATUS> is not a valid commitment status.*

* Invalid Energy Offer Dispatch Status:

*Invalid passed from file. <EO\_DISPATCH\_STATUS> is not a valid dispatch status.*

* Invalid Regulation Dispatch Status:

*Invalid passed from file. <REG\_DISPATCH\_STATUS> is not a valid dispatch status.*

* Invalid Spin Dispatch Status:

*Invalid passed from file. <SPIN\_DISPATCH\_STATUS> is not a valid dispatch status.*

* Invalid Online Supplemental Dispatch Status:

*Invalid passed from file. <SUPP\_ON\_DISPATCH\_STATUS> is not a valid dispatch status.*

* Invalid Offline Supplemental Dispatch Status:

*Invalid passed from file. <SUPP\_OFF\_DISPATCH\_STATUS> is not a valid dispatch status.*

**Data File Specifications**

Below is the Data Type Definition for the XML file interface.

**Data File Example**

<OPERATIONAL\_PARAMETERS>

<OPERATIONAL\_PARAMETER>

<ACTION>INSERT</ACTION>

<PTCPT\_CD>TEST</PTCPT\_CD>

<TX\_PT>TEST\_GEN1</TX\_PT>

<EFFECTIVE\_DATE>2013-10-01</EFFECTIVE\_DATE>

<SOURCE> </SOURCE>

<ROW\_ID/>

<COMBINED\_CYCLE>

<STATUS>INDIVIDUAL</STATUS>

<EXT\_ID>test</EXT\_ID>

<COMMENTS>test comment</COMMENTS>

</COMBINED\_CYCLE>

<DEFAULTS>

<CLRG\_CD>DA</CLRG\_CD>

<MIN\_EMERGENCY\_LIMIT>100.1</MIN\_EMERGENCY\_LIMIT>

<MIN\_ECONOMIC\_LIMIT>100.1</MIN\_ECONOMIC\_LIMIT>

<MIN\_REGULATION\_LIMIT>100.1</MIN\_REGULATION\_LIMIT>

<MAX\_REGULATION\_LIMIT>100.1</MAX\_REGULATION\_LIMIT>

<MAX\_ECONOMIC\_LIMIT>100.1</MAX\_ECONOMIC\_LIMIT>

<MAX\_EMERGENCY\_LIMIT>100.1</MAX\_EMERGENCY\_LIMIT>

<OFFLINE\_RESOURCE\_LIMIT>100.1</OFFLINE\_RESOURCE\_LIMIT>

<NO\_LOAD\_COST>100.11</NO\_LOAD\_COST>

<COLD\_STARTUP\_COST>100.11</COLD\_STARTUP\_COST>

<INTER\_STARTUP\_COST>100.11</INTER\_STARTUP\_COST>

<HOT\_STARTUP\_COST>100.11</HOT\_STARTUP\_COST>

<RAMP\_RATE>100.1</RAMP\_RATE>

<RAMP\_RATE\_UP>100.1</RAMP\_RATE\_UP>

<RAMP\_RATE\_DOWN>100.1</RAMP\_RATE\_DOWN>

<RAMP\_RATE\_BIDIRECTIONAL>100.1</RAMP\_RATE\_BIDIRECTIONAL>

<COMMITMENT\_STATUS>ECONOMIC</COMMITMENT\_STATUS>

<EO\_DISPATCH\_STATUS>ECONOMIC</EO\_DISPATCH\_STATUS>

<REG\_DISPATCH\_STATUS>ECONOMIC</REG\_DISPATCH\_STATUS>

<SPIN\_DISPATCH\_STATUS>ECONOMIC</SPIN\_DISPATCH\_STATUS>

<SUPP\_ON\_DISPATCH\_STATUS>ECONOMIC</SUPP\_ON\_DISPATCH\_STATUS>

<SUPP\_OFF\_DISPATCH\_STATUS>ECONOMIC</SUPP\_OFF\_DISPATCH\_STATUS>

<ECO\_LOW\_LIMIT\_MW>100.1</ECO\_LOW\_LIMIT\_MW>

<ECO\_LOW\_LIMIT\_TEMP>100.1</ECO\_LOW\_LIMIT\_TEMP>

<ECO\_MID\_LIMIT\_MW>100.1</ECO\_MID\_LIMIT\_MW>

<ECO\_MID\_LIMIT\_TEMP>100.1</ECO\_MID\_LIMIT\_TEMP>

<ECO\_HIGH\_LIMIT\_MW>100.1</ECO\_HIGH\_LIMIT\_MW>

<ECO\_HIGH\_LIMIT\_TEMP>100.1</ECO\_HIGH\_LIMIT\_TEMP>

<EMGCY\_LOW\_LIMIT\_MW>100.1</EMGCY\_LOW\_LIMIT\_MW>

<EMGCY\_LOW\_LIMIT\_TEMP>100.1</EMGCY\_LOW\_LIMIT\_TEMP>

<EMGCY\_MID\_LIMIT\_MW>100.1</EMGCY\_MID\_LIMIT\_MW>

<EMGCY\_MID\_LIMIT\_TEMP>100.1</EMGCY\_MID\_LIMIT\_TEMP>

<EMGCY\_HIGH\_LIMIT\_MW>100.1</EMGCY\_HIGH\_LIMIT\_MW>

<EMGCY\_HIGH\_LIMIT\_TEMP>100.1</EMGCY\_HIGH\_LIMIT\_TEMP>

<REG\_LOW\_LIMIT\_MW>100.1</REG\_LOW\_LIMIT\_MW>

<REG\_LOW\_LIMIT\_TEMP>100.1</REG\_LOW\_LIMIT\_TEMP>

<REG\_MID\_LIMIT\_MW>100.1</REG\_MID\_LIMIT\_MW>

<REG\_MID\_LIMIT\_TEMP>100.1</REG\_MID\_LIMIT\_TEMP>

<REG\_HIGH\_LIMIT\_MW>100.1</REG\_HIGH\_LIMIT\_MW>

<REG\_HIGH\_LIMIT\_TEMP>100.1</REG\_HIGH\_LIMIT\_TEMP>

<MAX\_CR\_DEPLOY\_MW>100.1</MAX\_CR\_DEPLOY\_MW>

<MAX\_REG\_UP\_DEPLOY\_MW>100.1</MAX\_REG\_UP\_DEPLOY\_MW>

<MAX\_REG\_DOWN\_DEPLOY\_MW>100.1</MAX\_REG\_DOWN\_DEPLOY\_MW>

<EXT\_ID>test default</EXT\_ID>

<COMMENTS>test default comments</COMMENTS>

</DEFAULTS>

</OPERATIONAL\_PARAMETER>

</OPERATIONAL\_PARAMETERS>

**File Naming Convention**

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

OP\_PARM\*\_YYYYMMDD.xml

Example: = OP\_PARM\_20020208.xml

[[A1]](file:///M:\nMarket-Global\MISO\Designs\FirstGen_VSS_Designs\MW\API's\Inbound%20API%20-%20Energy%20Offers.doc#_msoanchor_1)Update