

# **Marketplace Technical Conference**

## **Technical Design Considerations**

### **Load and Virtual Bidding Changes**

August 11, 2006

# Changes in Load Bid Data Processing

- **Why:** Imagine having to make 10 round trips to the gas station to get 10 gallons of gas in your car.
  
- **Legacy Implementation**
  - 100 bids submitted = 100 distinct requests to authenticate, authorize, evaluate and persist
  - High cost in limited computing resources due to redundant activity (network traffic, database sessions)
  - Offers incremental updates at a bid by bid level. If bid number 51 fails (validation, authorization, persistence) the previous 50 bids are already processed and persisted
  
- **New Implementation**
  - 100 bids submitted = 1 request to authenticate, authorize and evaluate. (*Approx 3 requests to persist*)
  - Fewer request/response cycles reduces load on network and database resources
  - Treats data submission as bulk data sets. If bid number 51 fails a business rule then the previous 50 bids are not persisted

# What Doesn't Change Due to Bulk Data Processing

- NO template changes
- NO data definition changes
- NO business rule changes
- Almost no GUI changes

# What Changes – Upload/Download

- **Scenario 1:** Upload 125 load bids. Data row 5 is missing required data. The rest of the file is reviewed for any other required data errors. 0 data rows persisted and one error messages for each row of data that has missing required data is displayed.
- **Scenario 2:** Upload 250 load bids. Data row 149 contains a bad PTID. Processing stops and the appropriate message is returned. 0 data rows persisted.
- **Scenario 3:** Upload 250 load bids. All data passes business rule validation. During the persistence process batches of data are sent to the database. After 2 batches there is a database error. Processing stops and the appropriate message is returned. X data rows are persisted (depends on the batch size) – this information is included in the error message.
- **Scenario 4:** One Upload in process, Second Upload on queue, Third upload submitted. Under this scenario the third upload will immediately be rejected. (Upload is identified by user and template combination).

# What Changes – GUI

- There are only two visible changes to the GUI
- Legacy

Virtual MWh Sum for all zones: \_\_\_\_\_ Virtual MWh Limit for all zones: \_\_\_\_\_

Time	Forecast MW	Fixed Bid MW	Price Cap #1		Price Cap #2		Price Cap #3		Interrupt Price Cap		Interrupt Fixed		Bid Status
			MW	\$/MW	MW	\$/MW	MW	\$/MW	MW	\$/MW	MW	\$/MW	
00:00	5.0	6.0	1.0		2.0		3.0						<a href="#">VALIDATION PASSED</a>
			1.00		2.00		3.00						

- New

Time	Forecast MW	Fixed Bid MW	Price Cap #1		Price Cap #2		Price Cap #3		Interrupt Price Cap		Interrupt Fixed		Bid Status
			MW	\$/MW	MW	\$/MW	MW	\$/MW	MW	\$/MW	MW	\$/MW	
00:00	5	6	1		2		3						<a href="#">VALIDATION PASSED</a>
			1.0		2.0		3.0						

## What Changes – Historical Data

- Currently historical load bid data (older than 10 days) is returned via the get load bids template.
- With the new implementation historical load bid data will only be accessible via DSS.
- What is the impact of this change to your systems?
- How much ramp up time is required to account for this change?

# What May Change

## → Request queuing

- Today a user can send multiple uploads and have them synchronously queue for up to 10 minutes waiting for previous uploads to complete.
- Is request queuing something you rely on or leverage?
- What is the impact to your systems if this feature was not available?

# Bidding Functions & Tiered Architecture

## → Legacy

- Logic for Physical, Virtual Load and Virtual Supply was intertwined and difficult to manage
- A business change was likely to result to a change to the UI
- Processing errors in the database are replicated to the user

## → New

- Each “bidding business vertical” is distinct. A change in the business rules for Virtual Load bidding can be implemented without impact to unrelated functions
- A business change will be absorbed by the business tier of the application, a data structure change will be managed at the data tier and UI changes are independent of the other tiers of the application
- Errors are managed and displayed with users in mind rather than developers



# Expected Value

- Reuse of business functions across business units
- Performance
- Flexibility
- Improved use of computing resources

# Web Services Interface

## → CSV Adapter Service

- Early consumer of the new services will be the CSV to XML adapter service. This allows us to fully support the existing CSV Upload/Download interface at the same time we offer an upgrade path

## → More Flexible

- Reuse of business logic
- Self defining rules that can be leveraged by the consumer via XML schema
- No need to provide checksums
- Improved security implementation via WS-Security

# Example of Get Load Bids XML Message

```
<?xml version="1.0" encoding="UTF-8"?>
<GetLoadBidsRequestMessage
  xmlns="http://services.nyiso.com/loadbidding/2006/LoadBiddingServiceSchema"
  xmlns:header="http://services.nyiso.com/2006/UploadHeaderSchema"
  xmlns:loadbid="http://services.nyiso.com/loadbidding/2006/LoadBidSchema"
  xmlns:loadbus="http://services.nyiso.com/loadbus/2006/LoadBusSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://services.nyiso.com/loadbidding/2006/LoadBiddingServiceSchema
  http://services.nyiso.com/loadbidding/2006/schemas/loadbidding/LoadBiddingServiceSchema.xsd">
  <GetLoadBidsRequest>
    <loadbid:BidDateTime>2006-08-17T00:00:00-05:00</loadbid:BidDateTime>
    <loadbus:LoadBusIdentifier>
      <loadbus:Ptid>54321</loadbus:Ptid>
    </loadbus:LoadBusIdentifier>
    <loadbid:BidStatusDescription>VALIDATION PASSED</loadbid:BidStatusDescription>
  </GetLoadBidsRequest>
</GetLoadBidsRequestMessage>
```