## Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Description</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>October 2005</td>
<td>1.0</td>
<td>Initial Draft. Updated from Phase1b Documentation.</td>
<td>Mark Lassiter</td>
</tr>
<tr>
<td>January 31, 2006</td>
<td>1.1</td>
<td>Revised based on feedback from LCG testers. Added TOC, Revision History and Cover page. Renamed the document to ADS API Specification.</td>
<td>Mark Lassiter</td>
</tr>
<tr>
<td>February 13, 2006</td>
<td>1.2</td>
<td>Removed PenaltyBidSegment field. Added bidDelay field. Added Change Log</td>
<td>Mark Lassiter</td>
</tr>
<tr>
<td>May 31, 2006</td>
<td>1.3</td>
<td>Add MSS LF methods and payloads. Added AS Award Type field to instruction payload.</td>
<td>Mark Lassiter</td>
</tr>
<tr>
<td>May 31, 2006</td>
<td>1.4</td>
<td>Added MSS data dictionary items. Updated TOC.</td>
<td>Mark Lassiter</td>
</tr>
<tr>
<td>October 12, 2006</td>
<td>1.5</td>
<td>Added XSD to Section 4.0 as per PMO standards request.</td>
<td>Karen Holly-Asper</td>
</tr>
<tr>
<td>October 13, 2006</td>
<td>1.6</td>
<td>Added Sample XML. Added Batch Type for OOS Instructions</td>
<td>Mark Lassiter</td>
</tr>
<tr>
<td>February 19, 2007</td>
<td>1.7</td>
<td>Modified spec for CR3B changes.</td>
<td>Mark Lassiter</td>
</tr>
<tr>
<td>February 21, 2007</td>
<td>1.8</td>
<td>Adjusted batch type definition and updated Market simulation reference. Added note regarding MSS LF.</td>
<td>Mark Lassiter</td>
</tr>
<tr>
<td>February 22, 2007</td>
<td>1.9</td>
<td>Removed Price field from Detail Segments</td>
<td>Mark Lassiter</td>
</tr>
<tr>
<td>October 5, 2007</td>
<td>2.1</td>
<td>Removed PassIndicator from XSD.</td>
<td>Troy Siegel</td>
</tr>
<tr>
<td>November 5, 2007</td>
<td>2.2</td>
<td>Changed data types for UID fields to string.</td>
<td>Mark Lassiter</td>
</tr>
<tr>
<td>August 22, 2008</td>
<td>3.0</td>
<td>Removed DA AS Award from XSD and data definition. Added new ADS API Response Web</td>
<td>Mark Lassiter</td>
</tr>
<tr>
<td>Date</td>
<td>Version</td>
<td>Service methods</td>
<td>Author</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------</td>
<td>----------------------------------------------------------------------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>September 17, 2008</td>
<td>3.1</td>
<td>Added DOP sequenceNumber and Trajectory Batch bindingFlag elements.</td>
<td>Mark Lassiter</td>
</tr>
<tr>
<td>December 7, 2009</td>
<td>4.0</td>
<td>Added MSG Configuration elements. Added ADS API Transition Support section. Updated UIDs to correctly reflect GUIDs instead of Integers.</td>
<td>Mark Lassiter</td>
</tr>
<tr>
<td>December 7, 2009</td>
<td>4.1</td>
<td>Added configuration id to Trajectory data.</td>
<td>Mark Lassiter</td>
</tr>
<tr>
<td>June 24, 2010</td>
<td>4.2</td>
<td>Add AGC and RMR flag</td>
<td>Ivan Loh</td>
</tr>
<tr>
<td>July 10, 2012</td>
<td>5.0</td>
<td>Add contingency dispatch elements.</td>
<td>Ivan Loh</td>
</tr>
<tr>
<td>August 28, 2012</td>
<td>5.1</td>
<td>Update CAISO header logo.</td>
<td>Ivan Loh</td>
</tr>
<tr>
<td>September 6, 2012</td>
<td>5.1.1</td>
<td>Update schema definitions</td>
<td>Ivan Loh</td>
</tr>
<tr>
<td>July 9, 2013</td>
<td>5.2.0</td>
<td>Add ADS Reason Codes</td>
<td>Ivan Loh</td>
</tr>
<tr>
<td>October 25, 2013</td>
<td>6.0.0</td>
<td>Add hourlyMwThreshold to instruction</td>
<td>Ivan Loh</td>
</tr>
<tr>
<td>March 17, 2014</td>
<td>7.0.0</td>
<td>EIM Release</td>
<td>Ivan Loh</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Add baseSchedule</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Add registeredIntertieFlag</td>
<td></td>
</tr>
</tbody>
</table>
# Table of Contents

1. OVERVIEW ............................................................................................................................................................ 5  
   1.1. REQUEST OVERVIEW ................................................................................................................................... 5  
   1.2. ADS API RESPONSE OVERVIEW ................................................................................................................. 6  
   1.3. ADS API RESPONSE SERVICE RESULT CODES ............................................................................................ 6  
   1.4. ERRORS AND DATA TYPES .......................................................................................................................... 6  

2. ADS API SERVICE OPERATIONS ................................................................................................................ 7  
   2.1. GETDISPATCHBATCHES SINCEUID .............................................................................................................. 7  
   2.2. GETBATCHSTATUS ...................................................................................................................................... 7  
   2.3. GETDISPATCHBATCH .................................................................................................................................. 7  
   2.4. GETBATCHHEADER ..................................................................................................................................... 8  
   2.5. ISNEWTRAJDATA ........................................................................................................................................ 8  
   2.6. GETTRAJECTORYDATA ................................................................................................................................ 8  
   2.7. VALIDATEDISPATCHBATCH ........................................................................................................................ 9  
   2.8. SUBMITMSSLFREQUEST............................................................................................................................. 9  

3. ADS API RESPONSE SERVICE OPERATIONS ........................................................................................ 10  
   3.1. ACCEPTINSTRUCTION ................................................................................................................................ 10  
   3.2. DECLINEINSTRUCTION ............................................................................................................................... 10  
   3.3. PARTIALACCEPTINSTRUCTION .................................................................................................................. 12  

4. RECOMMENDED USAGE ............................................................................................................................ 13  

5. XSD ................................................................................................................................................................... 15  
   5.1. SAMPLE XML ........................................................................................................................................... 19  

6. DATA DICTIONARY ..................................................................................................................................... 24  

7. ADS API TRANSITION SUPPORT .............................................................................................................. 31  

8. CHANGE LOG ................................................................................................................................................ 32
1. Overview

This document explains in detail the semantics of the function calls you can make using the ADS API web services. In this document the following are described:

- Syntax of requests.
- Format of the response data.

The following documents are referenced throughout this document:

<table>
<thead>
<tr>
<th>Document</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADSAPIWebService.wsdl</td>
<td>WSDL description of the ADS API Web Service.</td>
</tr>
<tr>
<td>ADSAPIResponseWebService.wsdl</td>
<td>WSDL description of the ADS API Response Web Service.</td>
</tr>
<tr>
<td>ads.caiso.com.xsd</td>
<td>XML Schema Definition describing the payload returned via the ADS API.</td>
</tr>
</tbody>
</table>

1.1. Request Overview

The following requests are available through the ADS API Service:

<table>
<thead>
<tr>
<th>Request</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getDispatchBatchesSinceUID</td>
<td>Retrieves a list of batch headers dispatched after a specified batch UID (unique identifier). Batches will not include instruction data.</td>
</tr>
<tr>
<td>getBatchStatus</td>
<td>Returns the batch status for a specified batch UID.</td>
</tr>
<tr>
<td>getBatchHeader</td>
<td>Returns the batch header for the specified batch UID.</td>
</tr>
<tr>
<td>getDispatchBatch</td>
<td>Retrieves the dispatch batch including the requestor’s set of instructions for the specified batch UID.</td>
</tr>
<tr>
<td>isNewTrajData</td>
<td>Returns true if new trajectory data has become available since a specified batch UID.</td>
</tr>
<tr>
<td>getTrajectoryData</td>
<td>Returns all new trajectory data received since the specified batch UID.</td>
</tr>
<tr>
<td>submitMSSLFRequest</td>
<td>Receives an MSS Load Following Request into the ADS system and returns validation results.</td>
</tr>
<tr>
<td>validateDispatchBatch</td>
<td>Informs the ADS system that the specified batch UID has been successfully received by the requestor.</td>
</tr>
</tbody>
</table>

The following requests are available through the ADS API Response Web Service:

<table>
<thead>
<tr>
<th>Request</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>acceptInstruction</td>
<td>Fully accepts the specified Instruction.</td>
</tr>
<tr>
<td>declineInstruction</td>
<td>Declines the specified Instruction.</td>
</tr>
<tr>
<td>partialAcceptInstruction</td>
<td>Partially accepts the specified Instruction.</td>
</tr>
</tbody>
</table>
1.2. **ADS API Response Overview**

Each ADS API request will return a single response that will contain the requested data or a fault condition. The following custom response types are described below:

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APIDispatchResponse</td>
<td>XML document containing a list of batches and associated information about each batch (e.g. batch UID, batch status, etc.).</td>
</tr>
<tr>
<td>DispatchBatch</td>
<td>XML Document containing a single ADS Dispatch Batch. May or may not contain instruction data depending on the call.</td>
</tr>
<tr>
<td>APITrajectoryResponse</td>
<td>XML document containing all trajectory data relevant to a particular client.</td>
</tr>
<tr>
<td>MSSLFResponse</td>
<td>XML document containing the validation results for each MSS LF Request submitted via <code>submitMSSLFRequest</code>.</td>
</tr>
</tbody>
</table>

1.3. **ADS API Response Service Result Codes**

The ADS API Response Web Service methods all return a single xsd:int value. The result codes are defined below:

<table>
<thead>
<tr>
<th>Response</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC_SUCCESS</td>
<td>0</td>
<td>The Instruction Response was received and accepted by the ADS system.</td>
</tr>
<tr>
<td>RC_INVALID_RESPONSE</td>
<td>1</td>
<td>The Instruction Response submitted is invalid. For example, the MW value may be outside the acceptable range for the instruction.</td>
</tr>
<tr>
<td>RC_INVALID_RESPONSE_PERIOD</td>
<td>2</td>
<td>The Instruction Response was received by ADS outside of the Market Participant response period.</td>
</tr>
<tr>
<td>RC_BATCH_NOT_FOUND</td>
<td>3</td>
<td>The Batch UID specified does not exist.</td>
</tr>
<tr>
<td>RC_INSTRUCTION_NOT_FOUND</td>
<td>4</td>
<td>The Instruction UID specified does not exist in the Batch specified.</td>
</tr>
<tr>
<td>RC_UNAUTHORIZED</td>
<td>5</td>
<td>You do not have sufficient privileges to respond to the resource associated with the submitted Instruction UID.</td>
</tr>
</tbody>
</table>

1.4. **Errors and Data Types**

All error conditions will be reported as SOAP `<fault>` elements. All other response types are standard xsd type definitions defined in the `ads.caiso.com.xsd` document.
2. ADS API Service Operations

2.1. getDispatchBatchesSinceUID

Returns all batches processed by the ADS system since the batch UID specified. Batches are returned in a list without instructions. The list will be empty if no batches are found. Batches should be processed in the order received. No assumptions should be made regarding the magnitude or ordering of the Batch UID. Always use the last batch UID processed for your next call or use -1 to retrieve the entire ADS API cache. At present, this would be all batches received by ADS in the last 24 hours.

The batch list will be returned as an XML document conforming to the `ads.caiso.com.xsd APIDispatchResponse` schema definition.

<table>
<thead>
<tr>
<th>I/O</th>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>batchUID</td>
<td>xsd:string</td>
<td>Last batch UID processed by your system.</td>
</tr>
<tr>
<td>Output</td>
<td>response</td>
<td>xsd:APIDispatchResponse</td>
<td>XML document containing a list of dispatches received since the batch UID specified.</td>
</tr>
</tbody>
</table>

2.2. getBatchStatus

This request returns the batch status associated with the provided batch UID.

<table>
<thead>
<tr>
<th>I/O</th>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>batchUID</td>
<td>xsd:string</td>
<td>Valid ADS batch UID.</td>
</tr>
<tr>
<td>Output</td>
<td>response</td>
<td>xsd:integer</td>
<td>See the section 4 for a list of valid Batch Status values.</td>
</tr>
</tbody>
</table>

2.3. getDispatchBatch

Returns the Dispatch Batch and associated instructions in Base64 encoded compressed XML for the batch UID specified. Only those instructions viewable by the requestor will be returned. This batch will be returned as an XML document conforming to the `ads.caiso.com.xsd DispatchBatch` schema definition. The data is returned as Base64 Encoded Compressed (GZIP) XML.

<table>
<thead>
<tr>
<th>I/O</th>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>batchUID</td>
<td>xsd:string</td>
<td>Valid ADS batch UID.</td>
</tr>
<tr>
<td>Output</td>
<td>response</td>
<td>xsd:DispatchBatch</td>
<td>XML document containing the requested batch and associated instructions viewable by the requestor.</td>
</tr>
</tbody>
</table>
### 2.4. getBatchHeader

Returns the batch header for the batch UID specified. This method can be used to check the status of batch as well as other batch attributes without the overhead of requesting the entire instruction set. The batch will be returned as an XML document conforming to the `ads.caiso.com.xsd DispatchBatch` schema definition. This data will not be compressed or encoded.

<table>
<thead>
<tr>
<th>I/O</th>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>batchUID</td>
<td>xsd:string</td>
<td>Valid ADS batch UID.</td>
</tr>
<tr>
<td>Output</td>
<td>response</td>
<td>xsd:DispatchBatch</td>
<td>XML document containing the requested batch. Instructions will not be returned.</td>
</tr>
</tbody>
</table>

### 2.5. isNewTrajData

Returns `true` if more recent trajectory data exists since the batch UID specified, otherwise returns `false`. Periodically execute to check for new trajectory data. When True is returned, execute the `getTrajectoryData` method to retrieve all new trajectory data in a single XML document.

Note that the Trajectory Batch UID is tracked separately form the Dispatch Batch UID. You should also track these values separately in your code.

<table>
<thead>
<tr>
<th>I/O</th>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>batchUID</td>
<td>xsd:string</td>
<td>Valid ADS trajectory batch UID.</td>
</tr>
<tr>
<td>Output</td>
<td>response</td>
<td>xsd:boolean</td>
<td><code>True</code> if new trajectory exists, otherwise <code>false</code>.</td>
</tr>
</tbody>
</table>

### 2.6. getTrajectoryData

Retrieves trajectory data received by the system since the trajectory batch UID specified. This batch list will be returned as an XML document conforming to the `ads.caiso.com.xsd APITrajectoryResponse` schema definition. The data is returned as Base64 Encoded Compressed (GZIP) XML.

<table>
<thead>
<tr>
<th>I/O</th>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>batchUID</td>
<td>xsd:string</td>
<td>Valid ADS trajectory batch UID.</td>
</tr>
<tr>
<td>Output</td>
<td>response</td>
<td>xsd:APITrajectoryResponse</td>
<td>XML document containing a list of trajectory batches received since the batch UID specified. Each batch will contain the DOP and Compliance data associated with the batch.</td>
</tr>
</tbody>
</table>
2.7. validateDispatchBatch
Informs the ADS system that the specified batch UID has been successfully received by the requestor. Only those instructions for which the requestor has primary access to the resource will be validated. If the instruction has already been validated, the instruction is skipped. This method should ONLY be passed ADS Dispatch Batch UIDs. Do not pass trajectory UIDs.

<table>
<thead>
<tr>
<th>I/O</th>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>batchUID</td>
<td>xsd:string</td>
<td>Valid ADS batch UID.</td>
</tr>
</tbody>
</table>

2.8. submitMSSLFRequest
Use to submit an MSS Load Following Request into the ADS system. This method performs a simple validation on each submitted instruction and forwards valid instructions to IFM. This method will return an MSS LF Response with the newly assigned CAISO batch identifier and a list of the MSS LF Instruction Response items. Each item also includes the newly assigned CAISO Instruction identifier and a flag indicating whether or not the instruction was valid.

The submitted request is an XML document conforming to the ads.caiso.com.xsd MSSLFRequest schema definition. The response is an XML document conforming to the ads.caiso.com.xsd MSSLFResponse schema definition.

**NOTE:** This functionality is **ONLY available to MSS Load Following Market Participants.**

<table>
<thead>
<tr>
<th>I/O</th>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>request</td>
<td>xsd:MSSLFRequest</td>
<td>XML document containing a list of MSS LF Instruction requests.</td>
</tr>
<tr>
<td>Output</td>
<td>response</td>
<td>xsd:MSSLFResponse</td>
<td>XML document containing a list of MSS LF Instructions submitted along with an assigned CAISO identifier and a flag indicating whether or not the instruction passed validation.</td>
</tr>
</tbody>
</table>
3. ADS API Response Service Operations

3.1. acceptInstruction
Accepts the specified ADS Instruction for the instructed DOT value. You must have a Primary or Secondary role for the resource associated with the instruction. The batch must not be expired and must be accepting responses. For hourly pre-dispatch instructions, this means the batch status must be 1.

A result code as defined in section 1.3 will be returned.

<table>
<thead>
<tr>
<th>I/O</th>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>batchUID</td>
<td>xsd:string</td>
<td>Batch UID of DispatchBatch associated with Instruction.</td>
</tr>
<tr>
<td>Input</td>
<td>instructionUID</td>
<td>xsd:string</td>
<td>Instruction UID of instructions to be accepted.</td>
</tr>
<tr>
<td>Output</td>
<td>response</td>
<td>xsd:int</td>
<td>Result code as defined in section 1.3.</td>
</tr>
</tbody>
</table>

3.2. declineInstruction
Declines the specified ADS Instruction for the entire instructed DOT value, returning the instruction to its Hour Ahead schedule or 0 if not scheduled. You must have a Primary or Secondary role for the resource associated with the instruction. The batch must not be expired and must be accepting responses. For hourly pre-dispatch instructions, this means the batch status must be 1.

A result code as defined in section 1.3 will be returned.

<table>
<thead>
<tr>
<th>I/O</th>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>batchUID</td>
<td>xsd:string</td>
<td>Batch UID of DispatchBatch associated with Instruction.</td>
</tr>
<tr>
<td>Input</td>
<td>instructionUID</td>
<td>xsd:string</td>
<td>Instruction UID of instructions to be accepted.</td>
</tr>
<tr>
<td>Input</td>
<td>reasonCode</td>
<td>xsd:int</td>
<td>A valid ADS reason code for declining the instruction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10 - Line down</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11 - Economic Considerations</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12 - Bad Bid Submitted</td>
</tr>
<tr>
<td>Output</td>
<td>response</td>
<td>xsd:int</td>
<td>Result code as defined in section 1.3.</td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
<td>---------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td></td>
<td>Unit Derate</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td></td>
<td>No Available Transmission</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td></td>
<td>Timed Out: Minimum Accepted</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td></td>
<td>Timed Out: Forced Decline</td>
</tr>
</tbody>
</table>
3.3. partialAcceptInstruction

Partially accepts the specified ADS Instruction for the specified MW amount. The specified MW amount must be between the Instruction DOT value and the resources Hour Ahead schedule.

You must have a Primary or Secondary role for the resource associated with the instruction. The batch must not be expired and must be accepting responses. For hourly pre-dispatch instructions, this means the batch status must be 1.

A result code as defined in section 1.3 will be returned.

<table>
<thead>
<tr>
<th>I/O</th>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>batchUID</td>
<td>xsd:string</td>
<td>Batch UID of DispatchBatch associated with Instruction.</td>
</tr>
<tr>
<td>Input</td>
<td>instructionUID</td>
<td>xsd:string</td>
<td>Instruction UID of instructions to be accepted.</td>
</tr>
<tr>
<td>Input</td>
<td>acceptDOT</td>
<td>xsd:double</td>
<td>The accepted DOT amount which must be between the instructed DOT MW and the Hour Ahead schedule.</td>
</tr>
<tr>
<td>Input</td>
<td>reasonCode</td>
<td>xsd:int</td>
<td>A valid ADS reason code for partially accepting the instruction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10 - Line down</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11 - Economic Considerations</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12 - Bad Bid Submitted</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>13 - Unit Derate</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>14 - No Available Transmission</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15 - Timed Out: Minimum Accepted</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>16 - Timed Out: Forced Decline</td>
</tr>
<tr>
<td>Output</td>
<td>response</td>
<td>xsd:int</td>
<td>Result code as defined in section 1.3.</td>
</tr>
</tbody>
</table>
4. Recommended Usage

The ADS API was designed for real time usage and not as a historical query tool. For the latter, please use the ADS Query Tool.

The following is pseudo code demonstrating the expected usage of the ADS API in a real time client application. This logic should keep your client application current with all ADS Dispatch and Trajectory Batch data. ADS API methods are highlighted in red.

**Pseudo Code**

```plaintext
Load LastDispatchUID  // or default to -1 the first time
Load LastTrajectoryUID // or default to -1 the first time

Start Loop
  // Check for new batches
  Batches = getDispatchBatchesSinceUID( LastDispatchUID )
  // Iterate batches returned (may be zero if no new)
  For Each Batch in Batches
    // Retrieve Instructions
    BatchData = getDispatchBatch( Batch.BatchUID )
    // Decode and decompress
    DecodeAndDecompress( BatchData )
    // Optional Step: Validate receipt
    validateDispatchBatch( Batch.BatchUID )
    // Process Batch Data (your logic)
    Process( BatchData )
    // Update the last batch uid processed
    Set LastDispatchUID = Batch.BatchUID
  End for each
  // Check for new trajectory batches
  If isNewTrajData( LastTrajectoryUID ) then
    // Get New Trajectory data
    TrajBatches = getTrajectoryData( LastTrajectoryUID )
    // Decode and decompress
    DecodeAndDecompress( TrajBatches )
    // Iterate Trajectory Batches
    For Each TrajBatch in TrajBatches
      // Process Batch Data (your logic)
      ProcessTrajectory( TrajBatch )
      // Update the last batch uid processed
      Set LastTrajectoryUID = TrajBatch.BatchUID
    End for each
  End if
  // Sleep
  Sleep 10 Seconds
End Loop
```
5. XSD

Below is the ADS XML schema definition.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema
targetNamespace="http://ads.caiso.com/api/schema/v6"
xmlns:tns="http://ads.caiso.com/api/schema/v6"
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://ads.caiso.com"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <element name="DispatchBatch">
    <complexType>
      <sequence>
        <element name="marketID" type="xsd:string"/>
        <element name="batchStatus" type="xsd:int"/>
        <element name="batchReceived" type="xsd:dateTime"/>
        <element minOccurs="0" name="batchSent" type="xsd:dateTime"/>
        <element minOccurs="0" name="batchExpires" type="xsd:dateTime"/>
        <element name="batchType" type="xsd:int"/>
        <element name="startTime" type="xsd:dateTime"/>
        <element name="dispatchMode" type="xsd:int"/>
        <element name="bindingFlag" type="xsd:string"/>
        <element name="revisionNo" type="xsd:int"/>
        <element minOccurs="0" name="contingencyType" type="xsd:string"/>
        <element minOccurs="0" name="pathExclusion" type="xsd:string"/>
        <element minOccurs="0" name="instructions">
          <complexType>
            <sequence>
              <element maxOccurs="unbounded" minOccurs="0" ref="tns:instruction"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
      <attribute name="batchUID" type="xsd:string" use="required"/>
    </complexType>
  </element>
  <element name="instruction">
    <complexType>
      <sequence>
        <element name="batchUID" type="xsd:string"/>
        <element name="resourceId" type="xsd:string"/>
        <element minOccurs="0" name="startTime" type="xsd:dateTime"/>
        <element minOccurs="0" name="endTime" type="xsd:dateTime"/>
        <element minOccurs="0" name="dot" type="xsd:double"/>
        <element minOccurs="0" name="oosEnergyCode" type="xsd:string"/>
        <element minOccurs="0" name="asType" type="xsd:string"/>
        <element name="instructionType" type="xsd:int"/>
      </sequence>
    </complexType>
  </element>
</schema>
```
<element minOccurs="0" name="preGoto" type="xsd:double"/>
<element minOccurs="0" name="bidDelay" type="xsd:int"/>
<element minOccurs="0" name="minAccept" type="xsd:double"/>
<element minOccurs="0" name="acceptDot" type="xsd:string"/>
<element minOccurs="0" name="acceptStatus" type="xsd:string"/>
<element minOccurs="0" name="responder" type="xsd:string"/>
<element minOccurs="0" name="reasonCode" type="xsd:int"/>
<element minOccurs="0" name="oprAcceptDot" type="xsd:double"/>
<element minOccurs="0" name="oprAcceptStatus" type="xsd:string"/>
<element minOccurs="0" name="oprResponder" type="xsd:string"/>
<element minOccurs="0" name="oprReasonCode" type="xsd:int"/>
<element minOccurs="0" name="validated" type="xsd:dateTime"/>
<element minOccurs="0" name="validatedBy" type="xsd:string"/>
<element minOccurs="0" name="apiValidated" type="xsd:dateTime"/>
<element minOccurs="0" name="apiValidatedBy" type="xsd:string"/>
<element name="revisionNumber" type="xsd:int"/>
<element name="statusCode" type="xsd:int"/>
<element minOccurs="0" name="clearedMW" type="xsd:double"/>
<element minOccurs="0" name="awardMW" type="xsd:double"/>
<element minOccurs="0" name="selfSchedMW" type="xsd:double"/>
<element minOccurs="0" name="hourlyMw" type="xsd:double"/>
<element minOccurs="0" name="configurationId" type="xsd:string"/>
<element minOccurs="0" name="transitionFromConfigId" type="xsd:string"/>
<element minOccurs="0" name="transitionToConfigId" type="xsd:string"/>
<element minOccurs="0" name="agcFlag" type="xsd:string"/>
<element minOccurs="0" name="rmrFlag" type="xsd:string"/>
<element minOccurs="0" name="hourlyMwThreshold" type="xsd:double"/>
<element minOccurs="0" name="baseSchedule" type="xsd:double"/>
<element minOccurs="0" name="registeredIntertieFlag" type="xsd:string"/>

<complexType name="instructionDetail">
<sequence>
  <element minOccurs="0" name="serviceType" type="xsd:string"/>
  <element minOccurs="0" name="mw" type="xsd:double"/>
</sequence>
<attribute name="segNo" type="xsd:int" use="required"/>
</complexType>

<element name="instructionUID" type="xsd:string" use="required"/>
</element>
<element name="trajectoryBatch">
    <complexType>
        <sequence>
            <element name="batchReceived" type="xsd:dateTime"/>
            <element name="bindingFlag" type="xsd:string"/>
            <element minOccurs="0" name="batchSent" type="xsd:dateTime"/>
            <element minOccurs="0" name="dopList">
                <complexType>
                    <sequence>
                        <element maxOccurs="unbounded" minOccurs="0" ref="tns:trajectoryDop"/>
                    </sequence>
                </complexType>
            </element>
            <element minOccurs="0" name="complianceList">
                <complexType>
                    <sequence>
                        <element maxOccurs="unbounded" minOccurs="0" ref="tns:trajectoryCompliance"/>
                    </sequence>
                </complexType>
            </element>
            <attribute name="batchUID" type="xsd:string" use="required"/>
        </sequence>
    </complexType>
</element>

<element name="trajectoryDop">
    <complexType>
        <sequence>
            <element name="resourceId" type="xsd:string"/>
            <element name="dop" type="xsd:double"/>
            <element name="targetTime" type="xsd:dateTime"/>
            <element name="sequenceNumber" type="xsd:int"/>
            <element minOccurs="0" name="configurationId" type="xsd:string"/>
        </sequence>
        <attribute name="dopUID" type="xsd:string" use="required"/>
    </complexType>
</element>

<element name="trajectoryCompliance">
    <complexType>
        <sequence>
            <element name="resourceId" type="xsd:string"/>
            <element name="startTime" type="xsd:dateTime"/>
            <element name="mwh" type="xsd:double"/>
            <element name="complFlag" type="xsd:string"/>
            <element minOccurs="0" name="configurationId" type="xsd:string"/>
        </sequence>
        <attribute name="complianceUID" type="xsd:string" use="required"/>
    </complexType>
</element>

<element name="APIDispatchResponse"/>
<complexType>
  <sequence>
    <element name="dispatchBatchList">
      <complexType>
        <sequence>
          <element maxOccurs="unbounded" minOccurs="0" ref="tns:DispatchBatch"/>
        </sequence>
      </complexType>
    </element>
  </sequence>
</complexType>

<element name="APITrajectoryResponse">
  <complexType>
    <sequence>
      <element name="trajectoryBatchList">
        <complexType>
          <sequence>
            <element maxOccurs="unbounded" minOccurs="0" ref="tns:trajectoryBatch"/>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>
</element>

<element name="MSSLFRequest">
  <complexType>
    <sequence>
      <element name="scMSSBatchId" type="xsd:string"/>
      <element minOccurs="0" name="mssLFInstructionRequests">
        <complexType>
          <sequence>
            <element maxOccurs="unbounded" minOccurs="0" ref="tns:MSSLFInstructionRequest"/>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>
</element>

<element name="MSSLFInstructionRequest">
  <complexType>
    <sequence>
      <element name="scMSSBatchId" type="xsd:string"/>
      <element name="scMSSLFInstructionId" type="xsd:string"/>
      <element name="resourceId" type="xsd:string"/>
      <element name="startTime" type="xsd:dateTime"/>
      <element name="endTime" type="xsd:dateTime"/>
      <element name="loadFollowingMW" type="xsd:double"/>
    </sequence>
  </complexType>
</element>
5.1. Sample XML

```xml
<complexType>
  <element name="MSSLFRResponse">
    <complexType>
      <sequence>
        <element name="caisoMSSBatchId" type="xsd:string"/>
        <element name="scMSSBatchId" type="xsd:string"/>
        <element minOccurs="0" name="mssLFInstructionResponses">
          <complexType>
            <sequence>
              <element maxOccurs="unbounded" minOccurs="0" ref="tns:MSSLFInstructionResponse"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </complexType>
  </element>
</complexType>

<complexType>
  <element name="MSSLFInstructionResponse">
    <complexType>
      <sequence>
        <element name="caisoMSSBatchId" type="xsd:string"/>
        <element name="caisoMSSLFInstructionId" type="xsd:string"/>
        <element name="scMSSBatchId" type="xsd:string"/>
        <element name="scMSSLFInstructionId" type="xsd:string"/>
        <element name="validated" type="xsd:boolean"/>
      </sequence>
    </complexType>
  </element>
</complexType>
</schema>

APIDispatchResponse
<xml version="1.0" encoding="utf-8">...
</xml>
```

Draft Page 19 of 33
<DispatchBatch xmlns="http://ads.caiso.com/api/schema/v6" batchUID="DISP-59F16A91-46CA-402B-FFDC-0A090014DA8C">
    <marketID>5MinDOT</marketID>
    <batchStatus>3</batchStatus>
    <batchReceived>2006-10-13T14:55:36Z</batchReceived>
    <batchSent>2006-10-13T14:55:37Z</batchSent>
    <batchExpires>2006-10-13T14:09:45Z</batchExpires>
    <batchType>0</batchType>
    <startTime>2006-10-13T14:10:00Z</startTime>
    <dispatchMode>0</dispatchMode>
    <bindingFlag>Y</bindingFlag>
    <revisionNo>4</revisionNo>
    <instructions>
        <instruction instructionUID="DISP-59F16A91-46CA-402B-FFDC-0A090014DA8C-I0">
            <batchUID>DISP-59F16A91-46CA-402B-FFDC-0A090014DA8C</batchUID>
            <resourceId>TEST_RESOURCE_1</resourceId>
            <startTime>2006-10-13T14:10:00Z</startTime>
            <dot>12.0</dot>
            <instructionType>0</instructionType>
            <preGoto>0.0</preGoto>
            <bidDelay>0</bidDelay>
            <acceptDot>12.0</acceptDot>
            <apiValidated>2006-10-13T14:55:50Z</apiValidated>
            <apiValidatedBy>TEST_John Smith</apiValidatedBy>
            <revisionNumber>2</revisionNumber>
            <statusCode>3</statusCode>
            <hourlyMw>3.0</hourlyMw>
            <configurationId>TEST_RESOURCE_1_1x1</configurationId>
            <agcFlag>Y</agcFlag>
            <rmrFlag>N</rmrFlag>
            <detail/>
        </instruction>
    </instructions>
</DispatchBatch>
<serviceType>SCHD</serviceType>
<mw>3.0</mw>
</instructionDetail>
<instructionDetail segNo="2">
<serviceType>SUPP</serviceType>
<mw>0.0</mw>
</instructionDetail>
<instructionDetail segNo="3">
<serviceType>RMPS</serviceType>
<mw>3.0</mw>
</instructionDetail>
<instructionDetail segNo="4">
<serviceType>SPIN</serviceType>
<mw>0.0</mw>
</instructionDetail>
<instructionDetail segNo="5">
<serviceType>NSPN</serviceType>
<mw>3.0</mw>
</instructionDetail>
<instructionDetail segNo="6">
<serviceType>HASE</serviceType>
<mw>0.0</mw>
</instructionDetail>
<instructionDetail segNo="7">
<serviceType>MSS</serviceType>
<mw>3.0</mw>
</instructionDetail>
<instructionDetail segNo="8">
<serviceType>TBD</serviceType>
<mw>0.0</mw>
</instructionDetail>
</detail>
</instruction>
<instruction instructionUID="DISP-59F16A91-46CA-402B-FFDC-0A090014DA8C-I1">
<batchUID>DISP-59F16A91-46CA-402B-FFDC-0A090014DA8C</batchUID>
<resourceId>TEST_RESOURCE_2</resourceId>
<startTime>2006-10-13T14:10:00Z</startTime>
<dot>11.0</dot>
<instructionType>0</instructionType>
<preGoto>0.0</preGoto>
<bidDelay>0</bidDelay>
<acceptDot>11.0</acceptDot>
(apiValidated)="2006-10-13T14:55:50Z"</apiValidated>
<apiValidatedBy>TEST_John Smith</apiValidatedBy>
<revisionNumber>2</revisionNumber>
<statusCode>3</statusCode>
<hourlyMw>0.0</hourlyMw>
</instruction>
<instruction instructionUID="DISP-59F16A91-46CA-402B-FFDC-0A090014DA8C-I2">
<batchUID>DISP-59F16A91-46CA-402B-FFDC-0A090014DA8C</batchUID>
<resourceId>TEST_RESOURCE_2</resourceId>
<startTime>2006-10-13T14:10:00Z</startTime>
<dot>11.0</dot>
<instructionType>0</instructionType>
<preGoto>0.0</preGoto>
<bidDelay>0</bidDelay>
<acceptDot>11.0</acceptDot>
(apiValidated)="2006-10-13T14:55:50Z"</apiValidated>
<apiValidatedBy>TEST_John Smith</apiValidatedBy>
<revisionNumber>2</revisionNumber>
<statusCode>3</statusCode>
<hourlyMw>0.0</hourlyMw>
</instruction>
<trajectoryCompliance complianceUID="TRAJ-59F16A91-46CA-402B-FFDC-0A090014DA8C-C1">
  <resourceId> TEST_RESOURCE_2 </resourceId>
  <startTime>2006-10-13T09:55:00.0000000-04:00</startTime>
  <mwh>0</mwh>
  <complFlag>N</complFlag>
</trajectoryCompliance>
</complianceList>
</trajectoryBatch>
<trajectoryBatch batchUID="TRAJ-59F16A91-46CA-402B-FFDC-0A090014BB4BE">
  <batchReceived>2006-10-13T11:08:36.0000000-04:00</batchReceived>
  <batchSent>2006-10-13T11:08:36.0000000-04:00</batchSent>
  <dopList>
    <trajectoryDop dopUID="TRAJ-59F16A91-46CA-402B-FFDC-0A090014BB4BE-D0">
      <resourceId> TEST_RESOURCE_1 </resourceId>
      <dop>16</dop>
      <targetTime>2006-10-13T11:08:00.0000000-04:00</targetTime>
      <sequenceNumber>1</sequenceNumber>
      <configurationId>TESTRESOURCE_1_1x1</configurationId>
    </trajectoryDop>
    <trajectoryDop dopUID="TRAJ-59F16A91-46CA-402B-FFDC-0A090014BB4BE-D1">
      <resourceId> TEST_RESOURCE_2 </resourceId>
      <dop>14</dop>
      <targetTime>2006-10-13T11:07:00.0000000-04:00</targetTime>
      <sequenceNumber>1</sequenceNumber>
    </trajectoryDop>
  </dopList>
</trajectoryBatch>
</trajectoryBatchList>
</APITrajectoryResponse>

MSSLF Request

<?xml version="1.0" encoding="utf-16"?>
  <scMSSBatchId>TestBatch1</scMSSBatchId>
  <mssLFInstructionRequests>
    <MSSLFInstructionRequest>
      <scMSSBatchId>TestBatch1</scMSSBatchId>
      <mssLFInstructionId>TestInstruction1</mssLFInstructionId>
      <resourceId>test</resourceId>
      <startTime>2006-09-25T01:00:00.0000000-04:00</startTime>
      <endTime>2006-09-25T02:00:00.0000000-04:00</endTime>
      <loadFollowingMW>100</loadFollowingMW>
    </MSSLFInstructionRequest>
  </mssLFInstructionRequests>
</MSSLFRequest>

MSSLF Response

<?xml version="1.0" encoding="UTF-8"?>
<xml version="1.0" encoding="UTF-8" xmlns="http://ads.caiso.com">
  <caisoMSSBatchId>8</caisoMSSBatchId>
</MSSLFResponse>
6. Data Dictionary

This section contains a data dictionary of those attributes and elements found in the ADS XSD document.

<table>
<thead>
<tr>
<th>Element/Attribute</th>
<th>XML Schema Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DispatchBatch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>batchUID</td>
<td>xsd:string</td>
<td>Unique identifier for ADS Dispatch Batch.</td>
</tr>
<tr>
<td>marketID</td>
<td>xsd:string</td>
<td>Market Run Identifier</td>
</tr>
<tr>
<td>batchStatus</td>
<td>xsd:int</td>
<td>The batch status. One of the following integer values: 0 = New, 1 = Active, 2 = Time Out (hourly only), 3 = Closed, 4 = Emergency Cancelled, 5 = Operator Response Period (hourly only), 6 = Deferred.</td>
</tr>
<tr>
<td>batchReceived</td>
<td>xsd:dateTime</td>
<td>Time batch was received into the ADS system.</td>
</tr>
<tr>
<td>batchSent</td>
<td>xsd:dateTime</td>
<td>Time batch was set to Active (batchStatus = 1).</td>
</tr>
<tr>
<td>batchExpires</td>
<td>xsd:dateTime</td>
<td>Time the batch expires/times out or otherwise transitions the batch status from its current status to the next. This differs depending on the batch type.</td>
</tr>
<tr>
<td>batchType</td>
<td>xsd:int</td>
<td>The batch type. 0 = 5 minute dispatchable, 1 = Hourly Pre-Dispatch</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>startTime</td>
<td>xsd:dateTime</td>
<td>The start time associated with the batch. This time should correspond to the start time of the interval for which the batch is targeted.</td>
</tr>
<tr>
<td>dispatchMode</td>
<td>xsd:int</td>
<td>The source system’s dispatch mode: 0 = Interval, 1 = Manual, 2 = Contingency</td>
</tr>
<tr>
<td>bindingFlag</td>
<td>xsd:string</td>
<td>Whether or not the batch is a binding batch. For the first release of ADS MRTU, this will always be Y (binding).</td>
</tr>
<tr>
<td>revisionNo</td>
<td>xsd:int</td>
<td>A revision number used to track incremental changes to the batch. Each time an instruction within the batch changes or the batch status changes, this value is incremented.</td>
</tr>
<tr>
<td>contingencyType</td>
<td>xsd:string</td>
<td>Valid contingency type values are: RTCD = Real-Time Contingency Dispatch, RTDD = Real-Time Disturbance Dispatch</td>
</tr>
<tr>
<td>pathExclusion</td>
<td>xsd:string</td>
<td>Valid path exclusion values are: NTE = Northern Ties Excluded, STE = Southern Ties Excluded, NSTE = Both Northern &amp; Southern Ties Excluded</td>
</tr>
<tr>
<td>instructionUID</td>
<td>xsd:string</td>
<td>ADS Unique identifier for each instruction. Unique across all batches.</td>
</tr>
<tr>
<td>resourceId</td>
<td>xsd:string</td>
<td>Resource ID from ISO Master File. Either the MF registered resource ID or the dynamic Intertie.</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>configurationId</td>
<td>xsd:string</td>
<td>MSG Configuration ID if applicable.</td>
</tr>
<tr>
<td>startTime</td>
<td>xsd:dateTime</td>
<td>Start time for instruction. Aka target time for DOT instructions.</td>
</tr>
<tr>
<td>endTime</td>
<td>xsd:dateTime</td>
<td>End time for instruction. Does not apply to all instruction types.</td>
</tr>
<tr>
<td>Dot</td>
<td>xsd:double</td>
<td>DOT for 5 minute, hourly and OOS instructions.</td>
</tr>
<tr>
<td>awardMW</td>
<td>xsd:double</td>
<td>AS MW Award for Ancillary Service Award instructions.</td>
</tr>
<tr>
<td>clearedMW</td>
<td>xsd:double</td>
<td>AS Cleared MW for AS Award instructions. Cleared MW is the total MW amount; equal to awardMW + daMW + selfSchedMW.</td>
</tr>
<tr>
<td>selfSchedMW</td>
<td>xsd:double</td>
<td>AS Self Scheduled MW for AS Award instructions.</td>
</tr>
<tr>
<td>asType</td>
<td>xsd:string</td>
<td>AS Type for Ancillary Service Award instructions: EN = Energy RU = Regulation Up RD = Regulation Down SR = Spinning Reserve NR = Non-Spinning Reserve RC = Residual Unit Commitment SR = Spinning Reserve LFU = Load Following Up LFD = Load Following Down</td>
</tr>
<tr>
<td>instructionType</td>
<td>xsd:int</td>
<td>Instruction Type. One of the following values: 0 = DOT (5 minute/Hourly/OOS) 1 = Min Constraint (OOS) 2 = Max Constraint (OOS) 3 = Fixed Constraint (OOS) 4 = Start up (Commitment) 5 = Shut down (Commitment) 6 = Capacity Award (AS Award) 7 = MSG Transition Instruction</td>
</tr>
<tr>
<td>transitionFromConfigId</td>
<td>xsd:string</td>
<td>Defines the “from” configuration id for an MSG transition</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>instruction</td>
<td></td>
<td>instruction (instructionType=7). Note that the startTime and endTime elements will define the transition period.</td>
</tr>
<tr>
<td>transitionToConfigId</td>
<td>xsd:string</td>
<td>Defines the “to” configuration id for an MSG transition instruction (instructionType=7). Note that the startTime and endTime elements will define the transition period.</td>
</tr>
<tr>
<td>agcFlag</td>
<td>xsd:string</td>
<td>The AGC flag will communicate when a resource was on AGC (Automatic Generation Control) at the start of the real-time market run. Y/N.</td>
</tr>
<tr>
<td>rmrFlag</td>
<td>xsd:string</td>
<td>The RMR flag will communicate when a resource is being incremented above the day-ahead schedule in real time, due to a RMR (Reliability Must Run) contract. Y/N.</td>
</tr>
<tr>
<td>hourlyMwThreshold</td>
<td>xsd:double</td>
<td>Accept/decline MW threshold</td>
</tr>
<tr>
<td>baseSchedule</td>
<td>xsd:double</td>
<td>EIM participating resource hourly base schedule</td>
</tr>
<tr>
<td>registeredIntertieFlag</td>
<td>xsd:string</td>
<td>Y: Physical resource has a MF registered resource Id.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>N: Physical resource does not have a MF registered resource Id.</td>
</tr>
<tr>
<td>preGoto</td>
<td>xsd:double</td>
<td>Previous DOT if available.</td>
</tr>
<tr>
<td>minAccept</td>
<td>xsd:double</td>
<td>Minimum Accept DOT for Intertie instructions. May be null if not applicable.</td>
</tr>
<tr>
<td>acceptDot</td>
<td>xsd:double</td>
<td>Accepted DOT for Intertie instructions.</td>
</tr>
<tr>
<td>acceptStatus</td>
<td>xsd:string</td>
<td>Accept Status for Intertie instructions.</td>
</tr>
<tr>
<td>responder</td>
<td>xsd:string</td>
<td>Responder for Intertie instructions.</td>
</tr>
<tr>
<td>reasonCode</td>
<td>xsd:int</td>
<td>Reason Code for declined or partially accepted Intertie instructions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid values:</td>
</tr>
<tr>
<td>oprAcceptDot</td>
<td>xsd:double</td>
<td>Operator Accept DOT for intertie instructions.</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>oprAcceptStatus</td>
<td>xsd:string</td>
<td>Operator Accept Status for intertie instructions.</td>
</tr>
<tr>
<td>oprResponder</td>
<td>xsd:string</td>
<td>Operator responder for intertie instructions.</td>
</tr>
<tr>
<td>oprReasonCode</td>
<td>xsd:int</td>
<td>Operator Reason Code for declined or partially accepted intertie instructions.</td>
</tr>
<tr>
<td>validated</td>
<td>xsd:dateTime</td>
<td>Time instruction was validated by ADS Client. If null, the instruction was not received by an ADS Client with Primary access to the resource.</td>
</tr>
<tr>
<td>validatedBy</td>
<td>xsd:string</td>
<td>The user common name responsible for the instruction validation.</td>
</tr>
<tr>
<td>apiValidated</td>
<td>xsd:dateTime</td>
<td>Time instruction was validated by ADS API Client. If null, the instruction was not received by an ADS API Client with Primary access to the resource OR the ADS API client has not been coded to call validateDispatchBatch. See the pseudo code for an example.</td>
</tr>
<tr>
<td>apiValidatedBy</td>
<td>xsd:string</td>
<td>The client certificate common name responsible for the instruction validation.</td>
</tr>
<tr>
<td>revisionNumber</td>
<td>xsd:int</td>
<td>Each time an instruction is updated, its revision number is incremented.</td>
</tr>
<tr>
<td>statusCode</td>
<td>xsd:int</td>
<td>The status of the instruction. Should always correspond to the instruction’s batch status. However may differ during during</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>bidDelay</td>
<td>int</td>
<td>Emergency cancel situations.</td>
</tr>
<tr>
<td>rmrTestRequestor</td>
<td>string</td>
<td>Minutes to sync</td>
</tr>
<tr>
<td>InstructionDetail</td>
<td></td>
<td></td>
</tr>
<tr>
<td>segNo</td>
<td>int</td>
<td>The unique number used to order the instruction detail elements.</td>
</tr>
<tr>
<td>serviceType</td>
<td>string</td>
<td>The service type associated with the instruction detail element.</td>
</tr>
<tr>
<td>mw</td>
<td>double</td>
<td>The MW amount.</td>
</tr>
<tr>
<td>TrajectoryBatch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>batchUID</td>
<td>string</td>
<td>Unique identifier for ADS Dispatch Batch.</td>
</tr>
<tr>
<td>batchReceived</td>
<td>date/time</td>
<td>Time batch was received into the ADS system.</td>
</tr>
<tr>
<td>bindingFlag</td>
<td>string</td>
<td>Y/N indicating whether or not the DOPs in this batch are binding.</td>
</tr>
<tr>
<td>batchSent</td>
<td>date/time</td>
<td>Time batch was set to Active (batchStatus = 1).</td>
</tr>
<tr>
<td>TrajectoryDOP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dopUID</td>
<td>string</td>
<td>Unique identifier for the ADS DOP record.</td>
</tr>
<tr>
<td>resourceId</td>
<td>string</td>
<td>Resource ID from the ISO Master File.</td>
</tr>
<tr>
<td>configurationId</td>
<td>string</td>
<td>MSG Configuration ID if applicable.</td>
</tr>
<tr>
<td>Dop</td>
<td>double</td>
<td>Dispatch Operating Point (DOP)</td>
</tr>
<tr>
<td>targetTime</td>
<td>date/time</td>
<td>Target Time for DOP.</td>
</tr>
<tr>
<td>sequenceNumber</td>
<td>int</td>
<td>A sequence value to indicate correct ordering of DOP points when there are two points within the same batch with the same targetTime. To properly order points, use the following ordering: DOP Target Time, Trajectory Batch batchReceived time, DOP sequenceNumber.</td>
</tr>
<tr>
<td>TrajectoryCompliance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>complianceUID</td>
<td>string</td>
<td>Unique identifier for the ADS Compliance record.</td>
</tr>
<tr>
<td>resourceId</td>
<td>string</td>
<td>Resource ID from the ISO Master File.</td>
</tr>
</tbody>
</table>
### ADS API Specification

<table>
<thead>
<tr>
<th>configurationId</th>
<th>xsd:string</th>
<th>MSG Configuration ID if applicable.</th>
</tr>
</thead>
<tbody>
<tr>
<td>startTime</td>
<td>xsd:dateTime</td>
<td>5 minute interval start time associated with this compliance record.</td>
</tr>
<tr>
<td>MWh</td>
<td>xsd:double</td>
<td>Expected energy in MWh’s for the 5 minute interval.</td>
</tr>
<tr>
<td>complFlag</td>
<td>xsd:string</td>
<td>Flag indicating compliance with the expected energy. Y/N.</td>
</tr>
</tbody>
</table>

#### MSSLFRequest

<table>
<thead>
<tr>
<th>scMSSBatchId</th>
<th>xsd:String</th>
<th>Unique identifier assigned by the SC for this batch of MSS Requests.</th>
</tr>
</thead>
<tbody>
<tr>
<td>mssLFInstructionRequests</td>
<td>0..n</td>
<td>xsd:MMSSLFInstructionRequest</td>
</tr>
</tbody>
</table>

#### MSSLFInstructionRequest

<table>
<thead>
<tr>
<th>scMSSBatchId</th>
<th>xsd:String</th>
<th>Unique identifier assigned by the SC for this batch of MSS requests.</th>
</tr>
</thead>
<tbody>
<tr>
<td>scMSSLFInstructionId</td>
<td>xsd:String</td>
<td>Unique identifier assigned by the SC for this Load Following Instruction request.</td>
</tr>
<tr>
<td>resourceId</td>
<td>xsd:String</td>
<td>Resource Id. ADS will verify that SC has sufficient privileges to submit on behalf of resource.</td>
</tr>
<tr>
<td>startTime</td>
<td>xsd:dateTime</td>
<td>Start time for LF MW.</td>
</tr>
<tr>
<td>endTime</td>
<td>xsd:dateTime</td>
<td>End time for LF MW.</td>
</tr>
<tr>
<td>loadFollowingMW</td>
<td>xsd:double</td>
<td>Load Following MW Positive for follow-up and negative for follow-down.</td>
</tr>
</tbody>
</table>

#### MSSLFResponse

<table>
<thead>
<tr>
<th>caisoMSSBatchId</th>
<th>xsd:String</th>
<th>Unique identifier assigned by the CAISO for this batch of MSS requests. Used internally by the ISO to track these requests.</th>
</tr>
</thead>
<tbody>
<tr>
<td>scMSSBatchId</td>
<td>xsd:String</td>
<td>Unique identifier assigned by the SC for this batch of MSS requests passed back to SC so they can use either the newly assigned ISO id or their own.</td>
</tr>
</tbody>
</table>

#### MSSLFInstructionResponse

| caisoMSSBatchId | xsd:String | Unique identifier assigned by the CAISO for this batch of MSS requests. Used internally by the ISO to track these requests. |

---

Draft Page 30 of 33
scMSSBatchId | xsd:String | Unique identifier assigned by the SC for this batch of MSS requests passed back to the SC for their internal coding.

caisoMSSLFInstructionId | xsd:String | Unique identifier assigned by CAISO for this Load Following Instruction request. Used internally by the ISO to track this request.

cMSSLFInstructionId | xsd:String | Unique identifier assigned by the SC for this Load Following Instruction request. Passed back to the SC for their internal coding.

validated | xsd: boolean | Flag specifying whether or not the submitted request was valid.

7. ADS API Transition Support

As of version 4.0 of the ADS API, we will be maintaining two versions of the ADS API to allow for an easy transition from the existing API to the latest release.

Version 5.0 has been deprecated and is no longer supported.

The supported ADS APIs are available at the following URIs:

8. Change Log

2009-12-7
- Added MSG Instruction data
- Updated example XML with GUIDs instead of integer UID values
- WSDL and XSD namespace changes

2008-09-17
- Added bindingFlag to TrajectoryBatch.
- Added sequenceNumber to TrajectoryDOP.

2008-08-22
- Removed DA MW.
- Added ADS API Response Service.

2006-02-13
- Removed getBatchListByDate method. Method is expensive. For historical queries, use the Query Tool. See the pseudo code for preferred real time API usage.
- Changed all references to BatchSeq to BatchUID. BatchSeq implies the numbers are always sequential. This may not be the case.
- GetInstructionSet was renamed to GetDispatchBatch.
- The return payloads have been changed for MRTU. New XSD attached.
- ValidateDispatchBatch has been added. This method may be used to programmatically confirm the receipt of a batch via the ADS API.

2006-05-31
- Added submitMSSLFRequest.
- Added AS Award Type. For now, the first two enumerations are reserved.

2006-10-12
- Added XSD to document in Section 4.
- Added Section 4.1 for Sample XML
- New Section 4, which moved Data Dictionary and Change Log to Sections 5 and 6.

2006-10-30
- Added Sample XML
- Added Batch Type for OOS Instructions

2007-02-19
- Removed AS Award Type
- Added new AS Breakdown fields:
  - awardMW (renamed asMW to awardMW)
  - clearedMW
  - selfProvidedMW
• dayAheadMW
  ▪ Added QLFC to AS Type list.
  ▪ Updated XSD and Sample XML.

2007-02-22
• Removed Price from Instruction Detail.

2007-04-04
• Updated ads.caiso.com.xsd to the new version.
• Added new possible values to DispatchBatch.batchStatus.
• Changed DispatchBatch.bindingFlag from unsignedInt to string.
• Changed AS breakdown fields:
  ▪ from Instruction.selfProvidedMW to Instruction.selfSchedMW (to reflect name in ResourceAwards.xsd v5.1)
  ▪ from Instruction.dayAheadMW to instruction.daMW (to reflect name in ADS database)
• Changed Instruction.asType to string and filled in all possible values.

2007-10-05
• Removed passIndicator from ADS xsd to make consistent with published ads.caiso.com-release-4.1.7.xsd

2007-11-05
• Changed UID data types from xsd:int to xsd:string.