

FAQS

Q. None of my bids or offers that were accepted by eMKT appear in eMKT why?

A. eMKT will only show data for locations that are contained within a portfolio. Therefore, even if your offers were accepted by the system, you will not be able to see those offers until you have placed the locations where the offers have occurred into a portfolio. You will first have to create a portfolio, and then add the locations that you are offering information into the portfolio. Instructions for creating a portfolio are shown below:

Instructions to add a location to a Portfolio

1. Log into eMKT <https://esuite.pjm.com/mui/index.htm>
2. Select the ADMIN button on the left side of the screen.
3. Select the USER SETUP tab and go to the CONFIGURE PARAMETERS page
4. Click on the button that says "**Submit Changes & Refresh Cache**"
5. Select the PORTFOLIO MANAGER tab
6. If you already have a portfolio assigned to your account, choose the appropriate name from the Portfolio drop down list. If you want to create a new portfolio, click the "Create New Portfolio" button and enter the name in the New Portfolio field.
7. **Click on the appropriate radio button that says "Demand" or "Generation" ... next to where it says "Select a type to filter location drop down list:"**
8. Click on the ADD button. Place the cursor in the open field. Start typing the location name, and the unit will appear in the drop down box.
9. Select the location, and click out of the field. The name should now be highlighted in red.
10. Click on the SUBMIT button to save to the portfolio.

Q. The Binding Limits report in eMkt for a specific date has two values for 1 of the hours where the Branch Name and the Contingency name are the same. Shouldn't there be just one value per hour ending?

A. The precision for the coefficient of control variables are different in both constraints (SPD-based on DC model and SFT-based on decoupled model), and when the difference in these coefficients are very small you'll see two constraints for the same hour. Only the constraint with the highest shadow price is taken into consideration while producing the prices.

Q. I am looking to see if you could provide me with some general definitions of these new tabs?

- DA Scheduling Reserve Award (in Market Results Tab)

- DA Scheduling Reserve Market (DA Scheduling Reserve Offers, DA Scheduling Reserve Bilaterals)

- Interface Pricing (Unit Hourly, Unit Offer & Area Forecast)

- **Opportunity Cost Calculator** (Opportunity Costs, Unit Parameters, Delivered Fuel, Unit Outages)

A. **Opportunity Cost Calculator** (Opportunity Costs, Unit Parameters, Delivered Fuel, Unit Outages)

<http://pjm.com/markets-and-operations/etools/emkt/opp-cost-calculator.aspx>

Generator Cost Development: Opportunity Cost Calculation for Energy & Environmentally Limited Units

This issue concerns stakeholder and board approved method for computing opportunity costs associated with an externally imposed energy or environmental run-hour restriction on a generation unit. This is still awaiting FERC approval and should not be used in offers.

- **DA Scheduling Reserve Award** (in Market Results Tab)

- **DA Scheduling Reserve Market** (DA Scheduling Reserve Offers, DA Scheduling Reserve Bilaterals)

The DA Scheduling Reserve Market is a supplementary reserve market that clears with the DA energy market. The details can be obtained from the link below and in Manual 11(2nd link).

<http://www.pjm.com/markets-and-operations/energy/day-ahead-sched.aspx#SliderItem1>

<http://www.pjm.com/~media/documents/manuals/m11.aspx>

- **Interface Pricing** (Unit Hourly, Unit Offer & Area Forecast)

<http://www.pjm.com/markets-and-operations/energy/~media/markets-ops/energy/lmp-model-info/20060929-interface-definition-methodology1.aspx>

Q. Under what situations is the Unit Hourly Updates, Schedule Details and Unit Details information used?

A. The hierarchy is as follows: Unit Hourly Updates overrides Schedule Details. Schedule Detail overrides Unit Detail.

Q. I was wondering how to go about updating the turndown ratio for my units even though they are past the open enrollment period?

A. You can update the PLS TDR through the eMKT daily exceptions screen. Exceptions are good for up to 7 days, any questions email parameterexceptions@pjm.com

Q. How is the turndown ratio defined, and how can I see what it is set to?

A. The Turndown Ratio is defined as the Eco Max/ Eco Min. It is just a calculation, it's not a separate field. You can see what the limit is by looking at the Parameter Limits/Participant Parameter Limits tab.

Q. What do I need to do in eMKT to perform a black start test?

A. There isn't really anything to do in eMKT, other than to make sure your Startup Cost \$ is up-to-date. Settlements will credit you with a startup when you conduct the black start test. Other than that, you just need to coordinate the test time with the gen desk, and submit the forms when complete to eDART.

Q. Lately, my units have regularly been Price Capped in the DA market results screen due to failing the Three pivotal supplier test. Is there any way to get the information on which constraint are causing the units to fail the Three pivotal supplier test?

A. Simply hover your mouse over the units on the market results screen and the information will appear.

Q. I am trying to perform a demand bid at location 123456789 and I am receiving the following error back from the system: "You are not authorized to create a fixed-demand bid at node 123456789" what should I do?

A. Send an email to the markets_hotline@pjm.com and request to be granted access to the desired location. PJM will ensure that you are allowed to bid at that location and if so they will grant you permission to bid at the location and the error will go away.

Q. When I submit my XML using one of our internal applications – I'm receiving an error similar to the following: "(505) Http Version Not Supported" why?

A. There is a known .Net bug with handling simultaneous connections correctly. You can either force .Net to use Http 1.0 protocol which resolves the issue or download the .Net patch for this from the internet - <http://devproj20.blogspot.com/2008/01/force-c-web-service-proxy-to-use-http.html>. To enforce the code to use HTTP version 1.0 in the code, the line of code required for C# .Net would be:

```
((HttpRequest)request).ProtocolVersion = System.Net.HttpVersion.Version10;
```

A limitation of HTTP 1.1 is that it only supports 2 simultaneous connections to another server. This is usually OK, but if a customer has a multi-threaded application then this can quickly become a problem. The problem manifests itself with an error that looks like the server is down. In reality what you'll have is a bunch of threads all waiting for a chance to connect, and when they don't you tend to get timeout errors that make it look like the server didn't respond. To get around this you need to connect using HTTP 1.0, but by default Visual Studio creates a web reference using 1.1. So you need to do some tweaking. Basically we need to override the [GetWebRequest](#) method on our generated web reference and alter the properties on the [HttpRequest](#) so that we can assign the right version. When you add a Web Reference in Visual Studio you effectively add an auto generated class to your

project, which (if you look down the directory tree in Windows Explorer) is saved in a Reference.cs file. This file changes every time you tell Visual Studio to update the web reference. Because of this it isn't always possible to override that particular class. If you working with an API that can change and you want to be able to update it easily, then you'll have to use [partial classes](#) in order to override the method. Or you could sub class the proxy and override it in there.

Q. I submit my XML which appears to look valid, but I am receiving the following error:

<Text>Unable to read request: Unable to extract request: Parsing Error : Content is not allowed in prolog. Line : 1 Column : 1</Text>

A. Some editors introduce hidden non-printable characters into a file that are invalid when a file containing them is submitted to eMKT. Unfortunately, since the characters are invisible, the XML may appear fine. The following will work to resolve this issue:

Open the XML file in WordPad, save the file under a new name, and then re-submit the file.

Q. After submitting my XML, it seems like the response is either taking a very long time or is eventually timing out, why?

A. This can be due to a number of contributing factors. The first recommendation would be to ensure that your XML is as optimized as possible. This means not to repeat locations for a day or to repeat the same xml attributes within the file. A slow response time could also be due to your XML being very large. You should break a large XML file into smaller multiple XML submittals. The response time could also be impacted by the time that the submission occurs. The closer the submittal is performed to the Day Ahead noon deadline, the more likelihood that performance related issues are encountered due to high traffic. If you are submitting Virtual bids then there is some additional credit checking that eMKT performs, and this may contribute to longer response times. If you continue to receive inexplicable and exorbitantly long response times, feel free to contact the markets_hotline@pjm.com to investigate.

Q. I am receiving the following error after I attempt to submit my XML:

<Text>Invalid request object: It must be a Submit Request.</Text>

A. You are either submitting the XML to the wrong tab within eMKT – the Query Upload tab, rather than using the Submit Upload tab.... OR you failed to wrap the XML in valid SOAP. All XML must look as follows:

```
<?xml version="1.0"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
<env:Body>
  Your Specific XML
```

```
</env:Body>
</env:Envelope>
```

Q. How do I delete my bids or offers?

A. There are two methods by which you can remove your offers. You can do it via manually logging into eMKT, and going to the appropriate screen, and use the delete button(s) to remove the offer. Alternatively, you can submit XML to remove the offers. Below is a snippet of documentation as extracted from the eMKT External Specification guide in terms of how to use XML to properly remove bids. This snippet is specific to fixed and price sensitive demand bids but the idea for the other offer types is similar:

To delete all the bids submitted at a given pricing node location, the <DemandBid> element is submitted as an empty set as shown below.

```
<DemandBid location="zzz" day="yyyy-mm-dd"/>
```

A delete may only be specified prior to market close at 12 Noon on the day before the operating day. Bid data may not be deleted once it is used in the day-ahead market clearing.

To delete a bid at a given hourly interval, you specify the hourly interval as an empty set as shown in the example below.

```
<DemandBid location="zzz" day="yyyy-mm-dd">
  <DemandBidHourly hour="hh"/>
</DemandBid>
```

The above example will delete the hourly interval specified.

To delete the price-sensitive portion of a demand bid, the PriceSensitiveDemand element is specified including each of the bid segments submitted as an empty elements as shown in the example below. This will delete all specified bid segments for the given hour and location. Also, the fixed demand is not affected by this operation.

```
<DemandBid location="zzz" day="yyyy-mm-dd">
  <DemandBidHourly hour="hh">
    <PriceSensitiveDemand>
      <BidSegment id="1"/>
      <BidSegment id="2"/>
      <BidSegment id="3"/>
      --- repeat for all defined bid segments ---
    </PriceSensitiveDemand>
  </DemandBidHourly>
</DemandBid>
```

Individual bid segments can be deleted by submitting empty BidSegment elements as shown below for only those segments to be deleted. Other bid segments not included are left in place.

```
<DemandBid location="zzz" day="yyyy-mm-dd">
  <DemandBidHourly hour="hh">
```

```
<PriceSensitiveDemand>
  <BidSegment id="3"/>
</PriceSensitiveDemand>
</DemandBidHourly>
</DemandBid>
```

Data may be replaced merely by resubmitting any portion of data to be modified. To replace all segments of the price sensitive demand with new segments (possibly with new segment id) the previous segments should be deleted. Segments that are not deleted are replaced by new data if the same segment id is used. If a segment is not deleted and it is not replaced by a message then it is left intact.

Q. Is there some way to view whether an eMKT submission has been sent or received successfully?

A. There are several ways to verify a submittal – one method is via XML, another method is via the eMKT screens themselves, and lastly via the XML transaction log (if submitted via XML). Below is a description of how you would verify using each of the methods:

Via XML:

First off if you sent the submission via XML then you should have received a transaction ID in response if it was successful or an error otherwise. If you didn't receive anything, and still want to confirm the submission was made, then you can still query that location or that day as shown. However, please know that this is not to be confused with querying for market results (this does not mean that your bid was assigned in dayahead).

Query for all demand bids for a specific market day:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
<QueryRequest xmlns="http://emkt.pjm.com/emkt/xml">
<QueryDemandBid day="2011-07-10">
<All/>
</QueryDemandBid>
</QueryRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Via eMKT Screens

You would first have to ensure that you have a portfolio defined that contains the location or locations of interest within eMKT. Once they are in a portfolio then you should be able to select the "Demand" or "Generator" button on the left hand side of the screen and then use the appropriate tabs across the top of the page. You can then select the portfolio that contains the location(s), and then the actual location from the drop down, and lastly select the day that the submittal was for.

Via the Transaction Logs

"Admin" button -> "XML" -> "Transaction Log" tab -> *Select Date*

Q. My client wants me to grab out DA LMP data and pointed me to this part of your website: <http://www.pjm.com/markets-and-operations/energy/day-ahead/lmpda.aspx>

Is this data exposed via a web service as well or is the RSS feed the only way to get access to this data programmatically other than building the http header out of the date and the format you use (which is a possibility)? Is there a query-able aspect to it as well where I can just grab the total LMP for each node at each hour? I'm hoping to get the real-time information as well.

A. Query for Market Prices

Purpose

This query returns the day-ahead market clearing prices for the requested locations. This is a public report. The report is an hourly report for the requested operating day.

The market clearing prices are available daily at 4 pm on the day before the operating day.

The query results are distinguished by how the type specifier (Public, Demand, Generation, LoadResponse) and the <All/> location specifier are used together. The query logic is described in the following table.

| Type | Description |
|--------------|--|
| Public | If <All/> is specified then the pricing nodes returned in the response include all publicly available pnodes that have a price. All Public queries are independent of participant identity or resource ownership. |
| Demand | If <All/> is specified then the pricing nodes returned in the response include all pnodes ever specified by the participant for a demand or virtual increment/decrement bid. |
| Generation | If <All/> is specified then then all generators locations owned by the participant are included in the response. |
| LoadResponse | If <All/> is specified then all load response resources that are owned by the participant are included in the response. |

Message Format

The Market Prices Query request is shown below:

```
<QueryRequest>
  <QueryMarketPrices type="xxx" day="yyyy-mm-dd">
    <All/>
    <LocationName>xxx</LocationName>
    <PortfolioName>xxx</PortfolioName>
  </QueryMarketPrices>
</QueryRequest>
```

The following table describes each of the elements and attributes and how they are used:

| Element or Attribute | Data Type | Description |
|----------------------|--|---|
| <QueryRequest> | Complex Type | The root element containing all query request elements in this message. Query request elements can be intermixed within the message. Occurs just once and specified in the SOAP Body element. |
| <QueryMarketPrices> | Complex Type | Specifies query for day-ahead market clearing prices for the given operating day and locations. |
| type | (Public, Generation, Demand, LoadResponse) | Required field specifying the type of market response data to query. This field determines the interpretation of the locations specified by <All/>, <LocationName>, or <PortfolioName>. |
| day | YYYY-MM-DD | Required field specifying the operating day. |
| <All/> | N/A | Optional element specifying that all <i>resources</i> of the type specified (see type description above in section 7.6.1). |

| Element or Attribute | Data Type | Description |
|----------------------|------------------|--|
| <LocationName> | Character String | Optional element specifying identifier of the resource of the type specified above. If not specified, then the <PortfolioName> element must be specified or the <All/> element must be specified. You must choose between <All/>, <LocationName>, or <PortfolioName> but you cannot specify more than one of these qualifiers in the same query request. |
| <PortfolioName> | Character String | Optional element specifying the name of a portfolio of locations. See comments above for the <LocationName> element. |

Q. I would like to automate the process of submitting our offers into eMKT, how can this be done?

A. There are many different means, and methods of performing browserless communications with the eMKT system. The following are the most common methods used: "Web Services", "VBA", ".Net", and "Java". PJM is only obligated to support the supplied browserless "Java" application. However, further below is a brief description on how VBA and .Net can be used with simple example code.

This information should provide all that you need to get started. Your main focus should be to PJM's eMKT External Specification Guide. Additionally, included below is a link containing information about PJM's Java API, and browserless app:

Other links and info:

Link to the external specification guide DOC: <http://www.pjm.com/markets-and-operations/etools/emkt.aspx>

Java Browserless app location:
<http://www.pjm.com/pub/etools/edart/xmldocs/xmldoc.html>

Select the top link for the "browserless.zip v5.1" and download. All of the documentation and code required to use the Java supplied browserless application is obtained from this 1 link.

First here is some important URL info:

Please note that although the url for the sandbox is different than production (assuming you may want to test in the sandbox first), the

application should use the same emkt namespace(the xmlns part of the XML) for the query. In other words the emktrn.pjm.com designation should only be specified in the POST and HOST commands and not changed in the xmlns string within the XML itself. Hopefully, the source code shown below will make it apparent as far as what substitutions you would need to make in order to get this to work for production.

Production URL: esuite.pjm.com

Production Web Service URL: emkt.pjm.com

Sandbox URL: esuitetrain.pjm.com

Sandbox Web Service URL : emktrn.pjm.com

Another means by which browserless communications can be achieved with the eMKT system is via VBA code as described below:

VBA information:

- #1 Open the Microsoft Excel application and select the develop VB icon in Excel
- #2 Drop the simple code located at the bottom of this email, and make it get called when the workbook is open or from a button click.
- #3 Select the References menu option in Excel and check the 'Microsoft HTTP Services 5.1' or later version
- #4 You can change the code so that the XML string which is currently hardcoded in the routine gets passed to the function or is obtained from a cell or cells from within the worksheet or from a flat file.

Here is the VB routine to perform XML uploads/downloads. Please note that right now this is set up to go against the sandbox but you can easily substitute per my comments above to go against production in the future.:

Note: You will need to substitute the appropriate XML and user name and password for your needs. Also, if you have to go through a proxy server then there is a line of code below that is currently commented out but you MAY have to uncomment it and put the info in that's particular to your company. Chances are you shouldn't need this line - so leave it commented out:

VBA Code Sample:

```
Public Sub xmlPost()  
    Dim oWinhttp As WinHttpRequest  
    Dim sXML As String  
    Set oWinhttp = New WinHttpRequest
```

Dim result As Variant

'Hardcoded XML to obtain Market Results. You could obtain this info or part of the info like the date
'from a spreadsheet cell or a flat file. The double quotes are to get the string itself to be able to contain
'a quote.

```
sXML = "<?xml version=""1.0""?>" & _  
"<Envelope xmlns=""http://schemas.xmlsoap.org/soap/envelope/"">" & _  
"<Header/>" & _  
"<Body>" & _  
"<QueryRequest xmlns="" http://emkt.pjm.com/emkt/xml"">" & _  
"<QueryDemandBid day=""2005-11-16"">" & _  
"<All/>" & _  
"</QueryDemandBid>" & _  
"</QueryRequest>" & _  
"</Body>" & _  
"</Envelope>"
```

With oWinhttp

'Only Uncomment this next line if you need to go through a Firewall

```
'Call .SetProxy(2, "myproxy.mycompany.com:80", "*.YOURCO.com")
```

```
Call .SetTimeouts(300000, 300000, 300000, 300000)
```

```
Call .Open("POST", "https://emkt.pjm.com/emkt/xml/query")
```

```
Call .SetCredentials("your user name", "your password", 0)
```

```
Call .SetRequestHeader("POST", "/emkt/xml/query HTTP/1.1")
```

```
Call .SetRequestHeader("HOST", "emktrn.pjm.com")
```

```
Call .SetRequestHeader("Content-Type", "text/xml")
```

```
Call .SetRequestHeader("Content-Length", Len(sXML))
```

```
Call .SetRequestHeader("SOAPAction", "/emkt/xml/query")
```

```
Call .Send(sXML)
```

'Wait for the response asynchronously.

```
Call .WaitForResponse
```

'These next 2 lines display the response text in a popup window.

'You could also store to a variable or to a cell or to a flat file

```
result = .ResponseText
```

```
MsgBox result
```

```
Call .Abort
```

End With

End Sub

.NET Code Example

```
Imports System.Net
Imports System.Web
Imports System.Configuration
Imports System.IO
Imports System.Text
Imports System.Threading
Public Class RequestState
    ' This class stores the state of the request
    Private Shared BUFFER_SIZE As Integer = 1024
    Public _RequestData As StringBuilder
    Public _BufferRead() As Byte
    Public _Request As WebRequest
    Public _Response As WebResponse
    Public _ResponseStream As Stream
    Public Sub New()
        _BufferRead = New Byte(BUFFER_SIZE) {}
        _requestData = New StringBuilder("")
        _Request = Nothing
        _ResponseStream = Nothing
    End Sub ' New
End Class ' RequestState

Public Class RetrieveData
    Public Shared _allDone As New ManualResetEvent(False)
    Private _DataDate As String
    Private Shared BUFFER_SIZE As Integer = 1024
    Protected Shared log As log4net.ILog

    Shared Sub New()
        'log4net.Config.DOMConfigurator.Configure()
        log = log4net.LogManager.GetLogger("RetrieveData")
    End Sub
    Public Sub BeginDownload()
        Try
            ' Create a new request to the mentioned URL.
            Dim myWebRequest As WebRequest =
WebRequest.Create(ConfigurationSettings.AppSettings("PJM-URL"))

            Dim CProxy As New
WebProxy(ConfigurationSettings.AppSettings("Proxy-URL"))
            CProxy.Credentials = New NetworkCredential(ConfigurationSettings.AppSettings("Proxy-UID"), _
                ConfigurationSettings.AppSettings("Proxy-PWD"))
            myWebRequest.Proxy = CProxy

            ' Create an instance of the RequestState and assign
'myWebRequest' to it's request field.
            Dim RequestStateObject As New RequestState
            RequestStateObject._Request = myWebRequest
            myWebRequest.ContentType = "text/xml"
```

```

' Set the 'Method' property to 'POST' to post data to a Uri.
RequestStateObject._Request.Method = "POST"
RequestStateObject._Request.ContentType = "text/xml"
RequestStateObject._Request.Headers().Set("POST",
"/emkt/xml/query HTTP/1.1")
RequestStateObject._Request.Headers().Set("SOAPAction",
"/emkt/xml/query")
Dim PJMCredentials As NetworkCredential = New NetworkCredential
PJMCredentials.UserName =
ConfigurationSettings.AppSettings("PJM-UID")
PJMCredentials.Password =
ConfigurationSettings.AppSettings("PJM-PWD")
myWebRequest.Credentials = PJMCredentials

' Start the Asynchronous 'BeginGetRequestStream' method
call.
Dim r As IAsyncResult = CType(myWebRequest.BeginGetRequestStream(AddressOf ReadCallback,
RequestStateObject), IAsyncResult)

' Assign the response object of 'WebRequest' to a 'WebResponse' variable.
Dim WebResponseObject As WebResponse =
myWebRequest.GetResponse()
Dim StreamResponseObject As Stream =
WebResponseObject.GetResponseStream()
Dim StreamReaderObject As New
StreamReader(StreamResponseObject)
Dim readBuff(256) As [Char]
Dim count As Integer = streamRead.Read(readBuff, 0, 256)

While count > 0
Dim outputData As New [String](readBuff, 0, count)
Console.WriteLine(outputData)
count = StreamReaderObject.Read(readBuff, 0, 256)
End While

' Close the Stream Object.
StreamResponseObject.Close()
StreamReaderObject.Close()
_allDone.WaitOne()

' Release the HttpWebResponse Resource.
WebResponseObject.Close()
Catch e As WebException
log.Error("Web Exception" & e.Message)
Catch e As Exception
log.Error("General Exception " & e.Message)
End Try
End Sub ' Main

Private Shared Sub ReadCallback(ByVal asynchronousResult As
IAsyncResult)
Try

```

```

    Dim sXML As String =
ConfigurationSettings.AppSettings("DemandXML")
    ' State of request is set to asynchronous.
    Dim myRequestState As RequestState = CType(asynchronousResult.AsyncState, RequestState)
    Dim myWebRequest2 As WebRequest = myRequestState._Request

    ' End of the Asynchronous request.
    Dim streamResponse As Stream =
myWebRequest2.EndGetRequestStream(asynchronousResult)

    Dim d As Byte() = System.Text.Encoding.ASCII.GetBytes(sXML)

    streamResponse.Write(d, 0, d.Length)
    streamResponse.Close()
    _allDone.Set()

Catch e As WebException
    log.Error("Web Exception" & e.Message)
Catch e As Exception
    log.Error("General Exception " & e.Message)
End Try

End Sub ' ReadCallback

Public Property DataDate() As String
    Get
        Return _DataDate
    End Get
    Set(ByVal Value As String)
        _DataDate = Value
    End Set
End Property

End Class ~~~~~

```