PJM Command Line Interface

Table of Contents

Purpose

The purpose of this document is to give an overview of the command line interface (CLI), a Java based interface for transferring formatted files to and from PJM participant facing applications. Because the interface is "browser-less", it can be used by an end user or a custom automation program written by the participant.

PJM provides this specification to aid PJM customers in building an external interface to PJM applications. PJM will provide assistance to customers seeking to understand or clarify details in this specification. However, due to the customizable nature of this external interface and the varied environments in which PJM customers will implement it, PJM is unable to provide application support for these customer-built external interfaces.

System Requirements

All required software will either be provided by PJM or available for download from http://www.oracle.com or a vendor of your choice.

- A reliable internet connection
- Java Runtime Environment (JRE): Version 7 or higher is required. Java is available for Solaris, Linux, Mac OSX, AIX, and Windows. Java downloads are available at: http://www.oracle.com
- PJM provided Java CLI program: client side class to authenticate with PJM and transfer files. The current version is provided as a Java Archive (JAR) file.

Release History

Release	Date	Description
1.0.0	July 3, 2012	Initial Release including Gas Pipeline and InSchedule.
1.0.8	Sep 05, 2012	Allowed for digital signature handling of SAML and minor bug fixes and optimizations.
1.0.9	Oct 05, 2012	Write jsession temp file to java.io.tmpdir where the application will always have write permissions.
1.1.2	Oct 25, 2012	Added CSV to XML –c switch for backward compatibility. Fixed bug with XML parsing.
1.1.3	Dec 03, 2012	Fixed for use with SSL proxy server params https.proxyHost etc reported by Ventyx/JP Morgan.
1.1.4	Apr 01 2013	Fixed potential issue with more than one HTM L form on the login page.
1.1.5	Jul 09 2013	Added mime-type for application/xml if uploading XML file for ExSchedules
1.1.6	Feb 09 2014	Added instructions to ExSchedule action for withdrawing Day Ahead Bids.

1.1.7	March 31 2014	Added instructions for Power Meter
1.1.9	August 06, 2014	Updated ExSchedule and added Customer Outages
1.1.11	September 17, 2014	Updated for JCM Ramp Viewer, and added .NET example.
1.1.12	September 30, 2014	Updated to include Emergency Procedures.
1.1.13	January 18, 2015	Updated ExSchedule to add Interchange Available.
1.2.5	September 01, 2016	Added instructions for DR Hub
1.2.8	May 09, 2017	Updates for Power Meter 5 minute intervals
1.3.0	December 13, 2017	Single User Multiple Account (SUMA) Updates
1.3.3	December 12, 2018	Added OASIS information
1.3.4	January 4, 2019	Added Account Manager API information

Usage

The Command Line Interface can be used in one of two ways: either as a standalone program or through the API. The following section describes the two different usage options.

Standalone Application

The CLI can be run as a standalone application from the command line by passing arguments to the application.

Example Standalone Execution

```
C:\Personal\tools\pjm-cli>java -jar pjm-cli.jar
PJM Command Line Interface 1.3.3-SNAPSHOT
Copyright 2011-2018, PJM Interconnection LLC.  All rights reserved

Java: Java(TM) SE Runtime Environment 1.8.0_181-b25 Oracle Corporation
OS: Windows 7 6.1 x86

ERROR: MissingOptionException: Missing required options: u, p, s, a, d

usage: pjm-cli
 -a,--action <String>        Action to execute
 -c,--csvToXml               Convert CSV download to XML format
                             (optional)
 -d,--directory <Directory>  Directory location of destination output
```

```
                                  (required)
 -f,--file <File>             File location of upload file (required if
                              this is an upload operation)
 -h,--header <param=value>    Header parameters to attach (optional)
 -i,--idpUrl <URL>            PJM Identity Provider Single-Sign On URL
                              (optional for public URLs)
 -l,--logging <Level>         Log level of output TRACE, DEBUG, INFO,
                              WARN, ERROR (default INFO)
 -m,--mimeType <mimeType>     Specify mimeType, defaults to text/plain
 -o,--output <File>           Output result file or downloaded file
                              (optional - will use name sent from
                              application)
 -p,--password <String>       Password (either encrypted or clear text)
 -q,--query <param=value>     Query parameters to append to URL (optional
 -r,--realm <realm>           Realm for which HTTP Basic Auth credentials
                              apply (only valid for BASIC AuthType)
 -s,--serviceUrl <URL>        PJM Application URL
 -sso,--useSSO <Boolean>      Determines if credentials should be
                              authenticated with PJM SSO
 -t,--authType <type>         Authentication Type either BASIC or FORM
                              (default to FORM)
 -u,--username <String>       Username
 -x,--proxy <param=value>     Proxy parameters for internet proxy in form
                              of proxyPort=8000 for proxyHost, proxyPort,
                              proxyPortSsl, proxyUser, proxyPassword,
                              proxySSL (optional)
 -z,--timeout <arg>           Socket and connect timeout in milliseconds,
                              defaults to 60000 (optional)
```

As you can see the CLI is self-documenting and will alert you if you are missing any required parameters.

Parameter Details

The table below describes the parameters in detail.

Argument	Description	Required/Optional
-u,--username	PJM account username from CAM.	**Required**
-p,--password	PJM account password from CAM. Password may be encrypted to prevent having clear text in any files. If encrypted must be surrounded by ENC(..). See section "Password Encryption".	**Required**
-s,--serviceUrl	Service URL is the application URL you want to upload or download from. Example:	**Required**

	`https://gaspipe.pjm.com/gaspipe/`	
-a,--action	The action to execute which will be a partial URL appended onto the service URL. This will be either an upload or download action. Example: `/rest/public/download/csv/notifications`	**Required**
-q,--query	Query parameters to append to the URL. These are documented later and are specific for each application. They are in the form of <param=value>	Optional (depends on the application and which query params it may accept)
-d,--directory	Directory location of destination output files. These may be results from an upload or the contents of a download operation.	**Required**
-f,--file	File location of upload file	Optional (required if performing an upload operation)
-o,--output	Output result file or downloaded file. The default will be the file name sent from the application such as "companies_2012-06-29-152315.csv". This parameter overrides that name with whatever you pass in like "companies.txt" would always force the name to companies.txt.	Optional
-l,--logging	Log level of output TRACE, DEBUG, INFO,WARN, ERROR (default INFO)	Optional (defaults to INFO)
-m, --mimetype	Specify mimeType	Optional (defaults to text/plain)
-t,--authType	Authentication Type either BASIC or FORM (default to FORM)	Optional (default to FORM)
-r,--realm	Realm for which HTTP Basic Auth credentials apply (only valid for BASIC AuthType)	Optional (only applies to BASIC authentication)
-x,--proxy	Proxy parameters for internet proxy. If your company internet access requires a proxy enter it here. Some proxies only require a URL and port number while some are secured as well and require a username and password. Example: `-x proxyHost=myHost -x proxyPort=80 -x proxyPortSsl=443 -x proxyUser=myProxyUser -x proxyPassword=myProxyPass`	Optional (use only if your company uses a proxy)

| -z,--timeout | Sets the socket and connection timeouts. Defaults to 60 seconds however for some large files you may need to increase this timeout. | Optional (defaults to 60000 milliseconds) |
| -c, --csvtoxml | If downloading a CSV file and you would like it converted to an xml format for easier parsing. | Optional |

Password Encryption

Many participants choose to store their credentials in a file when calling the CLI.   PJM provides example setEnv.cmd file for storing all of the common parameters of the CLI so a participant can then create custom CMD files that extend it. However, as a good security practice it is never recommended to store a password in plain text in a file. The CLI comes with a password encryption tool and associated CMD file for encrypting your password.   You may then use this encrypted password in the password parameter of the CLI.

Example Password Encryption

Using the password.cmd command file included with the CLI distribution.

```
C:\Personal\tools\esuite-cli>password "pjm123"

PJM CLI Password Encryptor 1.0
Copyright 2011-2012, PJM Interconnection LLC.  All rights reserved

Java: Java(TM) SE Runtime Environment 1.6.0_30-b12 Sun Microsystems Inc.
OS: Windows Vista 6.0 x86

Encrypting password 'pjm123'...
Encrypted password: ENC(zoCSGqDTGqueZjXlI3a4Rg==)
```

In this example the password "pjm123" was encrypted to "ENC(zoCSGqDTGqueZjXlI3a4Rg==)".   The entire string including the ENC() must be passed to the password parameter for the CLI to realize the string is encrypted and needs to be decrypted.

Command Files (.cmd)

PJM includes DOS command files with the CLI release. These CMD files are executable in a Windows Command Prompt and allow for easier configuration of the file operations.   All re-usable parameters are stored in setEnv.cmd and then that CMD file is reference in all of the custom CMD files per application.   The command files are documented in the text of each file and all of the possible application calls will be included as examples.

API – Application Programming Interface

An application programming interface (API) is a specification intended to be used as an interface by software components to communicate with each other. If a participant would like to embed code directly in their Java application rather than calling the external CLI program, they can use the API in their application. This requires the developer to implement their own Java program and use the provided behaviors.   The API is designed to uses Streams for input and output. Downloads come in the form of a java.io.OutputStream allowing you to control the stream yourself using a FileOutputStream, ByteArrayOutputStream, or any OutputStream implementation.   Uploads are sent in the form of java.io.InputStream so you can use any stream you want and not be constrained to using a file on disk if you already have the data in memory.

The example can be found in the CLI distribution in the location /docs/api/ExampleApi.java.

The API parameters mirror the CLI parameters listed in the section "Parameter Details".

Java Download Example API Usage

```java
/**
 * Example call to download a file from PJM into a ByteArrayOutputStream in
 * memory if you did not want to write the results to disk.
 */
public static void downloadToMemory() {
   try {
      final PjmRemoteCommand command = new PjmRemoteCommand();

      // first set up URL and credentials
      command.setServiceUrl("https://insched.pjm.com/inschedule/");
      command.setUsername("pjmuser");
      command.setPassword("Pjm4567!");

      // the URL action you want to execute
      command.setAction("/rest/secure/download/csv/contracts");

      // any query parameters required by the download
      command.getQueryParams().put("start", "01-01-2012");
      command.getQueryParams().put("stop", "01-31-2012");

      // create an output stream to capture the output result
      final ByteArrayOutputStream baos = new ByteArrayOutputStream();
      command.setOutputStream(baos);

      // execute the command and inspect the results
      final PjmResult result = PjmClient.execute(command);

      if (result.getResultCode() == PjmResult.CLI_SUCCESS) {
         System.out.println("Download succeeded for file: " + result.getFileName());
      } else {
         System.out.println("Download failed for file: " + result.getFileName());
      }

      // now you can inspect or manipulate the Outputstream results
      final String output = new String(baos.toByteArray());
```

```
      System.out.println(output);
   } catch (PjmClientException ex) {
      ex.printStackTrace();
   }
}
```

Java Upload Example API Usage

```java
/**
  * Uploads a file from disk to the PJM Servers.
  */
public static void upload() {
   try {
      final PjmRemoteCommand command = new PjmRemoteCommand();
      command.setServiceUrl("https://insched.pjm.com/inschedule/");
      command.setUsername("pjmuser");
      command.setPassword("Pjm4567!");

      // the URL action you want to execute
      command.setAction("/rest/secure/upload/file/");

      // a file upload requires an InputStream. It can be from a File like
      // this example, an in memory stream, or any other way of producing an
      // InputStream. The UploadFileName is purely a name for the server to
      // receive for logging purposes
      final File file = new File("my-upload.txt");
      command.setUploadFileName(file.getName());
      command.setInputStream(new FileInputStream(file));

      // create an output stream to capture the output result
      final ByteArrayOutputStream baos = new ByteArrayOutputStream();
      command.setOutputStream(baos);

      // execute the command and inspect the results
      final PjmResult result = PjmClient.execute(command);

      if (result.getResultCode() == PjmResult.CLI_SUCCESS) {
         System.out.println("Download succeeded for file: " + result.getFileName());
      } else {
         System.out.println("Download failed for file: " + result.getFileName());
      }

      // now you can inspect or manipulate the Outputstream results
      final String output = new String(baos.toByteArray());
      System.out.println(output);

   } catch (PjmClientException ex) {
      ex.printStackTrace();
   }
}
```

.NET Example

PJM does not provide a native .NET example however we can call our Java CLI from .NET very easily. Attached in the ZIP in /docs/.net you will find some sample source code for calling the CLI using .NET. A brief example is below:

```
//Create process
System.Diagnostics.Process pProcess = new System.Diagnostics.Process();

//strCommand is path and file name of command to run
pProcess.StartInfo.FileName = "java";

//Optional
pProcess.StartInfo.WorkingDirectory = "C:/Personal/tools/pjm-cli/";

//strCommandParameters are parameters to pass to program
pProcess.StartInfo.Arguments = "-Xms64m -Xmx256m -jar pjm-cli.jar -u "+txtUsername.Text+"
-p "+txtPassword.Text+" -d ./download/  -z 180000  -s
https://exschedule.pjm.com/exschedule/ -a /rest/secure/download/xml/schedules -q start=09-
18-2014 -q stop=09-19-2014";

// hide window
pProcess.StartInfo.CreateNoWindow = true;
pProcess.StartInfo.UseShellExecute = false;

//Set output of program to be written to process output stream
pProcess.StartInfo.RedirectStandardOutput = true;

//Start the process
pProcess.Start();

//Get program output
string strOutput = pProcess.StandardOutput.ReadToEnd();

//Wait for process to finish
pProcess.WaitForExit();
```
Once the process finishes you can parse the output for "SUCCESS" or "FAILURE" reported from the CLI and anything else printed out by the CLI.

Single User Multiple Account (SUMA)

A certain set of applications will be enabled with the ability to use the Single User Multiple Account functionality. This functionality allows users to use one username and password across multiple accounts in applications. By default when connecting to an application the user's primary account will be used to retrieve or submit data. The default behavior can be overridden by providing an additional header in the request to set the account to be used. The user must have access to this account as well as the appropriate application access to use this functionality. The header which is used to override this behavior is "Suma-Account" and the value is the account's short name. Below are examples of overriding the default behavior to query data for account short name ABCD.

Command File Example (.cmd)

The setenv.cmd file has been updated to include this header in the file. If you wish to override the account set the header to the accounts short name. For example:

set ACCOUNT=-h Suma-Account=ABCD

Java API Example

When using the API provided by the CLI you can add the Suma-Account header to the command. For example:

```
command.getHeaderParams().put("Suma-Account","ABCD");
```

.NET Example

As mentioned above PJM does not provide a native .NET example, however, the header can be added to the example shown above. For example:

```
pProcess.StartInfo.Arguments = "-Xms64m -Xmx256m -jar pjm-cli.jar -u "+txtUsername.Text+"
-p "+txtPassword.Text+" -d ./download/  -z 180000  -s
https://exschedule.pjm.com/exschedule/ -a /rest/secure/download/xml/schedules -q start=09-
18-2014 -q stop=09-19-2014" –h Suma-Account=ABCD;
```

Supported Applications

As PJM refreshes each application in its portfolio they will be designed to use this Command Line Interface.   The goal of this CLI is to have "convention over configuration". This means that each application will handle file uploads and downloads using the same convention allowing this CLI to support current and future applications without needing to be released every time a new application is released.

Each application will be listed below as well as what Actions that application exposes to the CLI.

Gas Pipeline

Type	Action	Parameters
Download	`/rest/public/download/csv/notifications` Downloads a CSV file of all current critical notifications.	None

JCM Ramp Viewer

Type	Action	Parameters
Download	`/rest/public/download/xml/jcmramp/` Downloads an XML file of available ramp as either composite view, PJM	view=COMPOSITE,MISO,PJM

	only view, or MISO only view.	location=PJM,MISO timezone=UTC, EST5EPT

InSchedule

Type	Action	Parameters
Upload	`/rest/secure/upload/file/` Uploads a flat file type of either internal schedules or reconciliations.	File to upload
Download	`/rest/secure/download/csv/contracts` Downloads a CSV file of contracts. Can choose a date range or pending only if you only want pending contracts	pending=true/false start=01-01-2012 stop=01-31-2012
Download	`/rest/secure/download/csv/schedules` Downloads a CSV file of internal schedules. Can choose a date range or pending only if you only want pending schedules	pending=true/false start=01-01-2012 stop=01-31-2012
Download	`/rest/secure/download/csv/companies` Downloads a CSV file of Company static data. This is a complete list of active companies in the system. (includeHeader will put the header at top of file making it invalid CSV)	includeHeader=true/false
Download	`/rest/secure/download/csv/reconciliations` Downloads a CSV file of Reconciliation data for a date range. (includeHeader will put the header at top of file making it invalid CSV)	includeHeader=true/false start=01-01-2012 stop=01-31-2012
Download	`/rest/public/download/csv/edclossfactor` Downloads a CSV file of EDC Loss Factor data for a date range. (includeHeader will put the header at top of file making it invalid CSV)	includeHeader=true/false start=01-01-2012 stop=01-31-2012

Power Contract Bulletin Board

Type	Action	Parameters
Download	`/rest/public/download/csv/postings` Downloads a CSV file of all current bulletin board postings.	None

ExSchedule

Type	Action	Parameters
Download	`/rest/secure/download/xml/ramp/` Downloads an XML file of the current ramp availability for now plus the next 72 hours.	None
Download	`/rest/secure/download/xml/interchange/` Downloads an XML file of the current interchange availability for now plus the next 72 hours.	None
Download	`/rest/secure/download/xml/schedules` Downloads a XML file of all current schedules including Tags, Ramp Reservations and Day Ahead Bids for a given date range. All parameters are optional and will be given defaults. **start (optional)** = the Start date of the date range you are specifying. If no start date is defined it will default to Today's date. **stop (optional)** = the Stop date of the date range you are specifying. **status (optional)** = Status is a comma separated list of only the ramp reservation statuses to return such as "Withdrawn,Expired".	start=01-01-2012 stop=01-31-2012 status=Approved
Download	`/rest/secure/download/xml/tagreservations` Downloads an XML file of the Tag Reservations for the date range. This report only contains implemented tags / approved ramp reservations. **start (optional)** = the Start date of the date range you are specifying. If no start date is defined it will default to Today's date.	start=01-01-2012 stop=01-31-2012

	stop (optional) = the Stop date of the date range you are specifying.	
Download	`/rest/secure/download/xml/bids` Downloads an XML file of the Day Ahead Bids for the date range. **start (optional)** = the Start date of the date range you are specifying. If no start date is defined it will default to Today's date. **stop (optional)** = the Stop date of the date range you are specifying.	start=01-01-2012 stop=01-31-2012
Download	`/rest/secure/download/xml/bid` Downloads an XML file of the Day Ahead Bid by name. **bid (required)** = the name of the bid.	bid=T2000220
Upload	`/rest/secure/upload/xml/bid/` Uploads an XML file for a Day Ahead Bid.	File to upload
Upload	`/rest/secure/withdraw/bid/` Withdraw profiles from a Day Ahead Bid by Bid Name **-q** = Indicates query mode and to use the bid parameter to locate the bid for update. **bid** = the name of the bid to withdraw. **profile** = (optional) the status of the particular profile to withdraw from the bid. If no profile specified the entire bid is withdrawn.	-q bid=T2000220 -q profile=PENDING_OASIS
Upload	`/rest/secure/upload/xml/reservation/` Uploads an XML file for a Ramp Reservation.	File to upload
Download	`/rest/secure/download/xml/reservations` Downloads an XML file of the Ramp Reservations for the date range. **start (optional)** = the Start date of the date range you are specifying. If no start date is defined it will default to Today's date. **stop (optional)** = the Stop date of the date range you are specifying.	start=01-01-2012 stop=01-31-2012

Download	`/rest/secure/download/xml/reservation` Downloads an XML file of the Ramp Reservation by name. **reservation (required)** = the name of the reservation.	reservation=XYZ2000124
Download	`/rest/secure/download/xml/creditusage` Downloads an XML file of the Credit Usage on all tags for the date range. **start (optional)** = the Start date of the date range you are specifying. If no start date is defined it will default to Today's date. **stop (optional)** = the Stop date of the date range you are specifying.	start=01-01-2012 stop=01-31-2012
Download	`/rest/secure/download/xml/ctsbids` Downloads an XML file of the CTS Bids for the date range. **start (optional)** = the Start date of the date range you are specifying. If no start date is defined it will default to Today's date. **stop (optional)** = the Stop date of the date range you are specifying.	start=01-01-2012 stop=01-31-2012
Download	`/rest/secure/download/xml/ctsbid` Downloads an XML file of the CTS Bid by name. **bid (required)** = the name of the bid.	bid=CTS2000220
Upload	`/rest/secure/upload/xml/ctsbid/` Uploads an XML file for a CTS Bid.	File to upload
Upload	`/rest/secure/withdraw/ctsbid/` Withdraw profiles from a CTS Bid by Bid Name **-q** = Indicates query mode and to use the bid parameter to locate the bid for update. **bid** = the name of the bid to withdraw. **profile** = (optional) the status of the particular profile to withdraw from the bid. If no profile specified the entire bid is withdrawn.	-q bid=CTS2000220 -q profile=REQUESTED

OASIS

Type	Action	Parameters
Download/Upload	`/rest/secure/`***templatename*** Replace ***templatename*** with valid NAESB or PJM offered template name.	Refer to NAESB WEQ Business Practice Standards and PJM OASIS API User Guide for more information about the accepted template requests.

Power Meter

Type	Action	Parameters
Download	`/rest/secure/download/xml/dailysubmission` Downloads a XML file of daily meter submissions for a given date range and meter account type. All parameters are optional and will be given defaults. **start (optional)** = the start date of the date range you are specifying. If no start date is defined it will default to the last business day. **stop (optional)** = the stop date of the date range you are specifying. If no stop date is defined it will default to start date + one day. **type (optional)** = the type of meter account which can be all, gen, or tie. **id (optional)** = comma separated list of meter account ids.	start=01-01-2012 stop=01-31-2012 type=tie
Download	`/rest/secure/download/csv/dailysubmission` Downloads a CSV file of daily meter submissions for a given date range and meter account type. All parameters are optional and will be given defaults. **start (optional)** = the start date of the date range you are specifying. If no start date is defined it will default to the last business day. **stop (optional)** = the stop date of the date range you are specifying. If no stop date is defined it will default to start date + one day.	start=01-01-2012 stop=01-31-2012 type=tie

	type (optional) = the type of meter account which can be all, gen, or tie. **id (optional)** = comma separated list of meter account ids.	
Download	`/rest/secure/download/xml/fiveminutesubmission` Downloads a XML file of five minute meter submissions for a given date range and meter account type. All parameters are optional and will be given defaults. **start (optional)** = the start date of the date range you are specifying. If no start date is defined it will default to the last business day. **stop (optional)** = the stop date of the date range you are specifying. If no stop date is defined it will default to start date + one day. **type (optional)** = the type of meter account which can be all, gen, or tie. **id (optional)** = comma separated list of meter account ids.	start=01-01-2012 stop=01-31-2012 type=tie
Download	`/rest/secure/download/csv/fiveminutesubmission` Downloads a CSV file of five minute meter submissions for a given date range and meter account type. All parameters are optional and will be given defaults. **start (optional)** = the start date of the date range you are specifying. If no start date is defined it will default to the last business day. **stop (optional)** = the stop date of the date range you are specifying. If no stop date is defined it will default to start date + one day. **type (optional)** = the type of meter account which can be all, gen, or tie. **id (optional)** = comma separated list of meter account ids.	start=01-01-2012 stop=01-31-2012 type=tie
Download	`/rest/secure/download/xml/dailyallocation` Downloads a XML file of Daily Meter Allocations for a given date range and meter account type. All parameters are optional and will be given defaults. **start (optional)** = the start date of the date range you are specifying. If no start date is defined it will default to the last business day. **stop (optional)** = the stop date of the date range you are specifying. If no	start=01-01-2012 stop=01-31-2012 type=tie

	stop date is defined it will default to start date + one day. **type (optional)** = the type of meter account which can be all, gen, or tie. **id (optional)** = comma separated list of meter account ids.	
Download	`/rest/secure/download/csv/dailyallocation` Downloads a CSV file of Daily Meter Allocations for a given date range and meter account type. All parameters are optional and will be given defaults. **start (optional)** = the start date of the date range you are specifying. If no start date is defined it will default to the last business day. **stop (optional)** = the stop date of the date range you are specifying. If no stop date is defined it will default to start date + one day. **type (optional)** = the type of meter account which can be all, gen, or tie. **id (optional)** = comma separated list of meter account ids.	start=01-01-2012 stop=01-31-2012 type=tie
Download	`/rest/secure/download/xml/monthlycorrection` Downloads a XML file of Monthly Corrections for a given date range and meter account type. All parameters are optional and will be given defaults. **start (optional)** = the start date of the date range you are specifying. If no start date is defined it will default to the last business day. **stop (optional)** = the stop date of the date range you are specifying. If no stop date is defined it will default to start date + one day. **type (optional)** = the type of meter account which can be all, gen, or tie. **id (optional)** = comma separated list of meter account ids.	start=01-01-2012 stop=01-31-2012 type=tie
Download	`/rest/secure/download/csv/monthlycorrection` Downloads a CSV file of Monthly Corrections for a given date range and meter account type. All parameters are optional and will be given defaults. **start (optional)** = the start date of the date range you are specifying. If no start date is defined it will default to the last business day. **stop (optional)** = the stop date of the date range you are specifying. If no	start=01-01-2012 stop=01-31-2012 type=tie

	stop date is defined it will default to start date + one day. **type (optional)** = the type of meter account which can be all, gen, or tie. **id (optional)** = comma separated list of meter account ids.	
Download	`/rest/secure/download/xml/meteraccount` Downloads a XML file of individual meter account for a given date and meter account id. Meter account Id is required. **date (optional)** = the date of the meter account. If no start date is defined it will default to the last business day. **id (required)** = the id of the meter account.	Date=01-01-2012 id=12345
Download	`/rest/secure/download/csv/meteraccount` Downloads a CSV file of individual meter account for a given date and meter account id. Meter account Id is required. **date (optional)** = the date of the meter account. If no start date is defined it will default to the last business day. **id (required)** = the id of the meter account.	Date=01-01-2012 id=12345
Download	`/rest/secure/download/xml/meteraccounts` Downloads a XML file of Monthly Corrections for a given date range and meter account type. All parameters are optional and will be given defaults. **date (optional)** = the date of the meter account. If no start date is defined it will default to the last business day. **type (optional)** = the type of meter account which can be all, gen, or tie.	date=01-01-2012 type=tie
Download	`/rest/secure/download/csv/meteraccounts` Downloads a CSV file of Monthly Corrections for a given date range and meter account type. All parameters are optional and will be given defaults. **date (optional)** = the date of the meter account. If no start date is defined it will default to the last business day. **type (optional)** = the type of meter account which can be all, gen, or tie.	date=01-01-2012 type=tie

Download	`/rest/secure/download/xml/loadsubmission` Downloads a XML file of Load Submissions for a given date range. All parameters are optional and will be given defaults. **start (optional)** = the start date of the date range you are specifying. If no start date is defined it will default to the last business day. **stop (optional)** = the stop date of the date range you are specifying. If no stop date is defined it will default to start date + one day.	
Download	`/rest/secure/download/csv/loadsubmission` Downloads a CSV file of Load Submissions for a given date range. All parameters are optional and will be given defaults. **start (optional)** = the start date of the date range you are specifying. If no start date is defined it will default to the last business day. **stop (optional)** = the stop date of the date range you are specifying. If no stop date is defined it will default to start date + one day.	
Upload	`/rest/secure/upload/xml/dailysubmission/` Uploads an XML file for Daily/Five Minute Meter Submission.	File to upload
Upload	`/rest/secure/upload/csv/fiveminutesubmission/` Uploads a CSV file for Five Minute Meter Submission.	File to upload
Upload	`/rest/secure/upload/xml/loadsubmission/` Uploads an XML file for Load Submission.	File to upload
Upload	`/rest/secure/upload/xml/monthlycorrection/` Uploads an XML file for Monthly Correction.	File to upload

Emergency Procedures

Type	Action	Parameters
Download	`/rest/public/posting` Downloads postings active in the current and previous 2 days.	none

| Download | `/rest/public/posting`

Downloads postings active for a given date range. All parameters are optional and will be given defaults.

start (optional) = the Start date of the date range you are specifying. If no start date is defined it will default to Today's date minus 2 days.

stop (optional) = the Stop date of the date range you are specifying. If no stop date is defined it will default to Tomorrow at midnight.

priority (optional) =If specified, will only retrieve postings matching the specified priority type. See the Message Definitions page for a list of available priorities.

drill (optional) =If specified, will apply the appropriate filter for PJM drills. Options include:

INCLUDE – include PJM drills in download (default)
EXCLUDE – exclude PJM drills in download.
ONLY – only include PJM drills in download.

activeOnly (optional) = true or false indicating whether or not the download should only include currently active postings. Defaults to false.

paiOnly(optional)=true or false indicating whether or not the download should only include PAI postings. Defaults to false. | start=01-01-2012

stop=01-31-2012

priority=Informational

drill=exclude

activeOnly=true

paiOnly=true |

DR Hub

Events

Type	Action	Parameters
Download	`/rest/secure/download/event/loadmanagement/acknowledgeable` Downloads unacknowledged events for Load Management Events. Note: The Request will download Unacknowledged Events with an End Date equal to null or less than one hour in the past.	none

Download	`/rest/secure/download/event/realtimedispatch/acknowledgeable` Downloads unacknowledged events for Real Time Events Note: The Request will download Unacknowledged Events with an End Date equal to null or less than one hour in the past.	none
Download	`/rest/secure/download/event/synchronizedreserve/acknowledgeable` Downloads unacknowledged events for Synchrnonized Reserve Events Note: The Request will download Unacknowledged Events with an End Date equal to null or less than one hour in the past.	none
Download	`/rest/secure/download/event/testloadmanagementevent/acknowledgeable` Downloads unacknowledged events for Test Load Management Events Note: The Request will download Unacknowledged Events with an End Date equal to null or less than one hour in the past.	none
Download	`/rest/secure/download/event/zonalemergencyenergy/acknowledgeable` Downloads unacknowledged events for Zonal Emergency Energy Events Note: The Request will download Unacknowledged Events with an End Date equal to null or less than one hour in the past.	none
Download	`/rest/secure/download/event/loadmanagement/123` Downloads Load Management Event by ID. To download other event types replace loadmanagement with one of the following: dayaheadenergy, loadmanagementsummary, loadmanagementtestsummary, realtimedispatch, realtimedispatchsummary, resourceoutage, synchronizedreserve, synchronizedreservesummary, zonalemergencyenergy, or zonalemergencyenergysummary	none
Download	`/rest/secure/download/event/loadmanagement` Downloads Load Management Event by date range. To download other event types replace loadmanagement with one of the following: dayaheadenergy, loadmanagementsummary, realtimedispatch,	startDate=01-01-2016 stopDate=01-31-2016

| | realtimedispatchsummary, resourceoutage, synchronizedreserve, synchronizedreservesummary, testloadmanagementevent, testloadmanagementsummary, zonalemergencyenergy, or zonalemergencyenergysummary

startDate (optional) = the Start date of the date range you are specifying.
stopDate (optional) = the Stop date of the date range you are specifying. | |
|---|---|---|
| Download | `/rest/secure/download/event/testloadmanagementsummary/123`

Downloads Test Load Management Summary by ID. | none |
| Download | `/rest/secure/download/event/testloadmanagementsummary`

Downloads Test Load Management Summary by date parameters. | startDate=MM-DD-YYYY stopDate=MM-DD-YYYY |
| Upload | `/rest/secure/upload/event/testloadmanagementsummary/create/`

Upload and create a Test Load Management Summary. | File to upload |
| Upload | `/rest/secure/upload/event/testloadmanagementsummary/12345`

Upload and update a Test Load Management Summary. | File to upload |
| Upload | `/rest/secure/upload/event/loadmanagement/acknowledge`

Upload and acknowledge a list of Load Management Events. | File to upload |
| Upload | `/rest/secure/upload/event/realtimedispatch/acknowledge`

Upload and acknowledge a list of Real Time Dispatch Events. | File to upload |
| Upload | `/rest/secure/upload/event/synchronizedreserve/acknowledge`

Upload and acknowledge a list of Synchronized Reserve Events. | File to upload |
| Upload | `/rest/secure/upload/event/testloadmanagementevent/acknowledge`

Upload and acknowledge a list of Test Load Management Events. | File to upload |
| Upload | `/rest/secure/upload/event/zonalemergencyenergy/acknowledge`

Upload and acknowledge a list of Zonal Emergency Energy Events. | File to upload |

Settlements

Type	Action	Parameters

Download	`/rest/secure/download/settlement/123` Downloads the Settlement by ID.	none
Download	`/rest/secure/download/settlement/` Downloads the Settlement by date range. **start (optional)** = the Start date of the date range you are specifying. **stop (optional)** = the Stop date of the date range you are specifying.	start=01-01-2016 stop=01-31-2016
Upload	`/rest/secure/settlement/123/deny` Deny the Settlement by ID. **comment (optional)** = A comment for denying a Settlement.	comment=reason for denying a settlement.
Upload	`/rest/secure/settlement/123/confirm` Confirm the Settlement by ID. **comment (optional)** = A comment for confirming a Settlement.	comment=reason for confirming a settlement.
Upload	`/rest/secure/settlement/123/dispute` Dispute the Settlement by ID. **comment (required)** = A comment for disputing a Settlement.	comment=reason for disputing a settlement.
Upload	`/rest/secure/settlement/123/submit` Submit the Settlement by ID.	none
Upload	`/rest/secure/upload/settlement/modify` Modify the settlement by uploading an XML file.	File to upload

Meter Data

Type	Action	Parameters

Download	`/rest/secure/download/meterdata/daily`
	
	`/rest/secure/download/meterdata/interval`
		registrationId=ID of the Registration for which to download

	Downloads the Daily or Interval Meter Data based on the parameters. Note: either of locationId or registrationId is required; optionally both may be provided	Meter Data (required if locationId is not provided) locationId=ID of the Location for which to download Meter Data (required if registrationId is not provided) startDate=01-01-2016 stopDate=01-31-2016
Download	`/rest/secure/download/meterdata/expected-reduction` Downloads the Expected Reduction Data based on the parameters.	zones=A comma-separated list of zone names for which to download Expected Reduction data startDate=01-01-2016 stopDate=01-31-2016

Upload	`/rest/secure/upload/meterdata/daily`
	
	`/rest/secure/upload/meterdata/interval`
	
	Upload the given Meter Data XML file.
		File to upload

Upload	`/rest/secure/upload/meterdata/expected-reduction` Upload the given Expected Reduction XML file.	File to upload

CBL

Type	Action	Parameters
Download	`/rest/secure/download/cblcalculation/cbltest/1234` Downloads the CBL Test for the Registration ID provided in the request. Note: This download returns a zip file with all the CBL test results	Registration ID
Download	`/rest/secure/download/cblcalculation/settlement/1234` Downloads the CBL Calculation for the Settlement ID provided in the request.	Settlement ID
Upload	`/rest/secure/upload/cblcalculation/cbltest/execute` Upload the given CBL Test parameters in the XML file and executes the CBL test.	File to upload
Upload	`/rest/secure/upload/cblcalculation/settelement/execute` Upload the given CBL Calculation parameters in the XML file and executes the CBL Calculation.	File to upload

Locations

Type	Action	Parameters
Download	`/rest/secure/download/location/1234` Downloads the Location for the Location ID provided in the request.	Location ID
Download	`/rest/secure/download/location` Downloads the Locations for the IDs or name provided in the request.	ids=A comma-separated list of Location IDs name=A string representing part, or all, of a Location's name. If multiple Locations match the name value, they will all be returned zones=A comma-separated list of Zone

		names
		edcAccountNumber=A string representing part, or all, of an EDC Account Number. If multiple Locations match the EDC Account Number, they will all be returned lastModifiedDate=Returns Locations modified after the given date
Download	`/rest/secure/download/location/1234/document` Downloads a ZIP archive that includes all the documents associated with the given Location.	Location ID
Upload	`/rest/secure/upload/location/create` Validates and creates the given Location in the XML file.	File to upload
Upload	`/rest/secure/upload/location/1234` Validates and updates the given Location in the XML file.	File to upload
Upload	`/rest/secure/upload/location/1234/document` Validates and uploads the given document associated with the Location. Note: Description is required. A file cannot be larger than 10 MB. Supported file types include CSV, XLS, XLSX, PPT, DOC, DOCX, PDF, and ZIP.	Location ID Description – string description of the file File to upload

Registrations

Type	Action	Parameters
Download	`/rest/secure/download/registration/1234` Downloads the Registration for the Registration ID provided in the request.	Registration ID
Download	`/rest/secure/download/registration`	status=A comma-separated list of Registration Statuses

	Downloads the Registrations for the parameters provided in the request.	startDate=Start Date to query against the Registration Effective Date stopDate=Stop Date to query against the Registration Effective Date name=A string representing part, or all, of a Registration's name. If multiple Registrations match the name value, they will all be returned lastModifiedDate=Returns Registrations modified after the given date
Download	`/rest/secure/download/registration/1234/document` Downloads a ZIP archive that includes all the documents associated with the given Registration.	Registration ID registrationDocumentType – section of the registration document belongs to; valid values are SYNCH_RESERVE, REGULATION_PLAN, ONE_LINE_DIAGRAM, DA_SCHEDULING_RESERVE, or INCREMENTAL_COST. Description – string description of the file
Upload	`/rest/secure/upload/registration/create` Validates and creates the given Registration in the XML file.	File to upload
Upload	`/rest/secure/registration/123/submit` Submit the Registration by ID.	override=Override duplicate registration warning
Upload	`/rest/secure/registration/123/withdraw` Withdraw the Registration by ID.	none
Upload	`/rest/secure/registration/123/dispute`	comment=required

	Dispute the Registration by ID.	
Upload	`/rest/secure/upload/registration/calculatewpl` Calculate winter peak load for the Registration IDs in the XML file	File to upload
Upload	`/rest/secure/upload/registration/complete` Marks the given Registration IDs as complete in the XML file.	File to upload
Upload	`/rest/secure/upload/registration/1234/terminate` Terminates the Registration by ID.	terminateDate=required for registrations that allow for it
Upload	`/rest/secure/upload/registration/1234/extend` Extends the Registration by ID.	terminateDate=required
Upload	`/rest/secure/upload/registration/1234/document` Validates and uploads the given document associated with the Registration in the specified section. Note: RegistrationDocumentType and Description are required. A file cannot be larger than 10 MB. Supported file types include CSV, XLS, XLSX, PPT, DOC, DOCX, PDF, and ZIP.	Registration ID registrationDocumentType – section of the registration document belongs to; valid values are SYNCH_RESERVE, REGULATION_PLAN, ONE_LINE_DIAGRAM, DA_SCHEDULING_RESERVE, or INCREMENTAL_COST. Description – string description of the file File to upload
Upload	`/rest/secure/upload/registration/1234 -file` Modifies an existing registration by ID.	Registration ID File to upload

Task

Type	Action	Parameters
Download	`/rest/secure/download/task/` Downloads all task for user.	none
Download	`/rest/secure/download/task/registration/edc/`	none

	Downloads all EDC Registration tasks for user.	
Download	`/rest/secure/download/task/registration/lse/` Downloads all LSE Registration tasks for user.	none
Upload	`/rest/secure/upload/task/registration/edc/` Acts on the EDC Registration tasks in the upload file.	File to upload
Upload	`/rest/secure/upload/task/registration/lse/` Acts on the LSE Registration tasks in the upload file.	File to upload

Compliance

Type	Action	Parameters
Download	`/rest/secure/download/compliance/` Downloads Compliance records based on the supplied parameters.	registrationId=The ID of the Registration on the Compliance records startDate=The start of the range to select (based on Event Date) stopDate=The end of the range to select (based on Event Date) status=The status to select
Download	`/rest/secure/download/compliance/validate/` Validates and returns the Compliance records for the supplied IDs.	Ids=A comma-separated list of Compliance IDs to validate

Dispatch Group

Type	Action	Parameters
Download	/rest/secure/download/dispatchgroup/1234 Downloads the Dispatch Group for the ID provided in the request.	The ID of the dispatch group.
Download	/rest/secure/download/dispatchgroup	name=A string representing part, or all, of a Dispatch Group's name. If multiple Dispatch Group match the name value, they will all be returned effectiveDate =The specific day to search

Account Manager

Type	Action	Parameters
Download	/rest/secure/download/csv/userSearch Downloads a CSV file of user information based on the supplied search criteria.	Username=The username to search (* wildcard available) activeOnly=T/F flag indicating whether to search active or terminated users - defaults to T accountId=The ID of the user's employer accountName=The Name of the user's

		employer (* wildcard available)
		accountShortName=The Short Name of the user's employer (* wildcard available)
		firstName=The first name of the user (* wildcard available)
		lastName=The last name of the user (* wildcard available)
		systemId=T/F flag indicating whether to search systems IDs or not - defaults to show all
		emailAddress=The email address of the user (* wildcard available)
		phone=The phone number of the user (* wildcard available)