

IRC Browserless Guide

- [Overview](#)
- [IRC download types](#)
- [Upload and Revise](#)
- [Schemas](#)
- [Commands](#)

Overview

The new IRC consists of several downloads and one upload. To use the command line API for the new IRC (and in general for any other eDART function), one has to call filetransfer.jar with a number of arguments. For example, to do a selfcheck IRC download:

```
java -jar filetransfer.jar app=edart sandbox=true username=<username>
password=<password> action=download type=ircselfcheck
```

Or to do an IRC upload:

```
java -jar filetransfer.jar app=edart sandbox=true username=<username>
password=<password> action=upload source=ircrevise.xml
```

These can be run either from a Windows batch file (.bat), or from a shell script (.sh), or you can even call the API directly from Java. We can provide a sample Java project to show how the API can be used from Java. But for now in this document we'll focus on command line.

The current design does not expose the actual server URLs to users. You control which environment you want to use through the "sandbox" flag. So while you don't have to deal with actual URLs, here is how filetransfer.jar will translate the requests, just for information:

sandbox	URL
false (or not present)	https://edart.pjm.com (production)
true	https://edarttrain.pjm.com (training)

The "app" argument should always be "edart". Action can be "download" or "upload". For downloads you need to further specify the type of download using "type" argument. For uploads there is no need to specify "type", because the application can determine what kind of upload is needed by looking directly at the file contents.

IRC download types

These are the download types available for IRC:

type	Can be run when there is an open IRC in progress	Can be run when there is no open IRC
ircselfcheck	no	yes
ircfull	yes	yes (can request a particular past IRC ID)
ircpooltotals	yes	yes
ircrevise	yes	no

The selfcheck download is a view of reserve numbers calculated by the PJM Energy Management System (EMS) every 5 minutes. So this download is actually independent of the reserve numbers submitted by member companies, and it can be run at any time as long as there is no open IRC in eDART.

The "full" download, as the name implies, contains the complete information about an IRC. It includes pool totals numbers, and a list of all generators with values for all their attributes. Each generator section includes both the PJM number and the company submitted numbers, so that

discrepancies can be easily identified.

When you execute a full download you may specify an IRC ID or not. If no IRC ID is specified, the download will return the current IRC in progress (if there is one) or the most recent posted IRC if there is none in progress.

The pooltotals section, although it's included in the full download, is also available as a separate download mainly for Transmission or Generic users. It does not accept an IRC ID, just like the full download it will return the current IRC in progress if there is one, or the most recent posted IRC if none is currently open.

Upload and Revise

To upload reserve numbers when an IRC is open, the steps are:

1. Execute a revise download. This will give you an xml file, say ircrevise.xml, containing all the units that can be modified (some units are read-only and will not appear here)
2. Open the xml file in your favorite editor (preferably an editor with build-in xml validation) and make the desired changes
3. Execute an upload with this xml file

The whole reason why there exists a revise download is to make uploads easy. Instead of asking the user to create from scratch an xml file that conforms to our upload schema, we provide through the revise download a template that can simply be edited then submitted back to PJM. The revise download already contains only the units that can be modified with only the fields that can be modified.

To execute the revise download:

```
java -jar filetransfer.jar app=edart username=<username> password=<password>  
action=download type=ircrevise
```

The output will be available in "result.xml" in the current directory - you can rename this file to your preference.

Say we get back 3 units:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ircReviseDownload xsi:schemaLocation="http://www.pjm.com/external/schemas/irc/v2
irc.xsd" xmlns:ns2="http://www.pjm.com/external/schemas/irc/v2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <reportRequestId>2967</reportRequestId>
  <reportPostedTimestamp xsi:nil="true"/>
  <requestTimestamp>2015-06-25T09:22:09.000-04:00</requestTimestamp>
  <currentTimestamp>2015-06-25T15:10:48.379-04:00</currentTimestamp>
  <comments></comments>
  <acknowledgeAll>false</acknowledgeAll>
  <Generator id="1464">
    <acknowledged>false</acknowledged>
    <beyondSecondary>0</beyondSecondary>
    <quickStart>0</quickStart>
    <secondary>0</secondary>
    <synchronizedReserves>0</synchronizedReserves>
  </Generator>
  <Generator id="1963">
    <acknowledged>false</acknowledged>
    <beyondSecondary>0</beyondSecondary>
    <quickStart>0</quickStart>
    <secondary>0</secondary>
    <synchronizedReserves>0</synchronizedReserves>
  </Generator>
  <Generator id="1973">
    <acknowledged>false</acknowledged>
    <beyondSecondary>0</beyondSecondary>
    <quickStart>0</quickStart>
    <secondary>0</secondary>
    <synchronizedReserves>0</synchronizedReserves>
  </Generator>
</ns2:ircReviseDownload>

```

Each unit has an acknowledged flag and the 4 numbers that can be modified. To modify something in a unit, it is not enough to set the numbers, you also need to set the acknowledged flag to true. In other words a change must be acknowledged. The global acknowledge flag (acknowledgeAll) works in this way: if it's false, it's ignored. If it is set to true however it overrides all the unit individual ack flags. The global ack flag applies to the units present in the file, or if there are no units in the upload file to ALL company's units. For example you can submit just the global ack flag, with no units in the file:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ircReviseDownload xsi:schemaLocation="http://www.pjm.com/external/schemas/irc/v2
irc.xsd" xmlns:ns2="http://www.pjm.com/external/schemas/irc/v2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <reportRequestId>2967</reportRequestId>
  <reportPostedTimestamp xsi:nil="true"/>
  <requestTimestamp>2015-06-25T09:22:09.000-04:00</requestTimestamp>
  <currentTimestamp>2015-06-25T15:10:48.379-04:00</currentTimestamp>
  <comments>Acknowledging all units</comments>
  <acknowledgeAll>true</acknowledgeAll>
</ns2:ircReviseDownload>

```

This will acknowledge all the company's units (in the background the app will retrieve all the modifiable units and acknowledge each one). If you submit however:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ircReviseDownload xsi:schemaLocation="http://www.pjm.com/external/schemas/irc/v2
irc.xsd" xmlns:ns2="http://www.pjm.com/external/schemas/irc/v2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <reportRequestId>2967</reportRequestId>
  <reportPostedTimestamp xsi:nil="true"/>
  <requestTimestamp>2015-06-25T09:22:09.000-04:00</requestTimestamp>
  <currentTimestamp>2015-06-25T15:10:48.379-04:00</currentTimestamp>
  <comments>upload unit 1464</comments>
  <acknowledgeAll>true</acknowledgeAll>
  <Generator id="1464">
    <acknowledged>false</acknowledged>
    <beyondSecondary>1</beyondSecondary>
    <quickStart>2</quickStart>
    <secondary>3</secondary>
    <synchronizedReserves>4</synchronizedReserves>
  </Generator>
</ns2:ircReviseDownload>

```

this will only acknowledge unit 1464.

The reserve numbers have to be positive integers, so nothing negative or fractional is accepted.

The response from the upload is in xml format and is self-explanatory. For each unit, it will say "OK" if the numbers were uploaded successfully, or it will give the error message if something was wrong and the unit could not be updated. For example:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ircUploadResponse xsi:schemaLocation="http://www.pjm.com/external/schemas/irc/v2
irc.xsd" xmlns:ns2="http://www.pjm.com/external/schemas/irc/v2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Generator id="1973">
    <status>GEN_ID: 1973 - IGNORED. IRC Exception: Bad quickStart value '23.43',
it must be a positive integer</status>
  </Generator>
  <Generator id="1975">
    <status>GEN_ID: 1975 - IGNORED. IRC Exception: Bad beyondSecondary value
'-44', it must be a positive integer</status>
  </Generator>
  <Generator id="1963">
    <status>GEN_ID: 1963 - OK</status>
  </Generator>
  <Generator id="53102801">
    <status>GEN_ID: 53102801 - Error: Any Change must be Acknowledged</status>
  </Generator>
</ircUploadResponse>

```

Schemas

All xml schemas, for downloads and for the upload, including the upload response, are in the same xsd file, available here: http://www.pjm.com/pu/b/etools/edart/xmldocs/irc_new.xsd

Commands

To execute a self check download use type=ircselfcheck:

```
java -jar filetransfer.jar app=edart username=<username> password=<password>  
action=download type=ircselfcheck
```

To execute a full download without specifying a particular IRC number use type=ircfull:

```
java -jar filetransfer.jar app=edart username=<username> password=<password>  
action=download type=ircfull
```

To execute a full download for a specific posted IRC, use type=ircfull ircId=<ircRequestId>:

```
java -jar filetransfer.jar app=edart username=<username> password=<password>  
action=download type=ircfull ircId=1234
```

To execute a pool totals download, use type=ircpooltotals:

```
java -jar filetransfer.jar app=edart username=<username> password=<password>  
action=download type=ircpooltotals
```

To execute a revise download:

```
java -jar filetransfer.jar app=edart username=<username> password=<password>  
action=download type=ircrevise
```

To execute an upload:

```
java -jar filetransfer.jar app=edart username=<username> password=<password>  
action=upload source=ircrevise.xml
```