RRC Browserless Guide

Overview

The new RRC consists of several downloads and one upload. To use the command line API for the new RRC (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 RRC download in eDART Sandbox:

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

Or to do an RRC upload in eDART Sandbox:

```
java -jar filetransfer.jar app=edart sandbox=true username=<username>
password=<password> action=upload source=rrcrevise.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 actual server URLs are not exposed 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.

RRC download types

These are the download types available for RRC:

type	Can be run when there is an open RRC in progress	Can be run when there is no open RRC
rrcselfcheck	yes *(will return last update before RRC was opened)*	yes
rrcfull	yes	yes
rrcequipmentlist	yes	yes
rrcrevise	yes	no

The "selfcheck" download is a view of derived reserves, updated every 5 minutes, with respect to static and dynamic reactive resources as determined by the PJM Energy Management System [EMS] based upon equipment availability, status, and limitations. This download is actually independent of the reserve numbers submitted by member companies, and it can be run at any time.

The "full" download, as the name implies, contains the complete information about an RRC. It includes the RRC comments and a list of all equipment in the RRC with values for all their attributes. Sections for Capacitors, Reactors and SVCs/Statcoms (referred to as SVCs in XML tags) include both the PJM number and the company submitted numbers, so that discrepancies can be easily identified. The section for units all have the PJM values as this is read only for the TOs.
When you execute a full download, the download will return the current RRC in progress (if there is one) or the most recent posted RRC if there is none in progress.

The "equipment list" download gives details of equipment included in the RRC. Equipment TA IDs which are used in the other RRC downloads and uploads can be translated to Station, Voltage and Equipment Name using this download. Users can review the list and notify PJM if any equipment should be excluded from the RRC.

Upload and Revise

An example RRC Revise file is available at http://www.pjm.com/pub/etools/edart/xmldocs/rrc_revise.xml.

Users can edit the file to suit their purpose.

<to_responses> is the Company Response field and can be used to send relevant information to PJM when responding to the RRC.

Acknowledge all PJM derived values by setting the following to true:

```
<acknowledgeAllCapacitors>
<acknowledgeAllReactors>
<acknowledgeAllSVC>
```

Updates to Rated or Active, Min and Max MVAR values must be submitted by TA ID. For example:

```
<Capacitor ta_id="1234">
    <ratedMVAR>30</ratedMVAR>
</Capacitor>
```

You can submit just the global acknowledgement flags, with no equipment in the file:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rrc:rrcRevise xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:rrc="http://www.pjm.com/external/schemas/rrc/v2"
xsi:schemaLocation="http://www.pjm.com/external/schemas/rrc/v2 rrc.xsd">
    <reportRequestId>2967</reportRequestId>
    <to_responses>
     <to_response  to_name="ABC">Acknowledging all equipment</to_response>
    </to_responses>
    <acknowledgeAllCapacitors>true</acknowledgeAllCapacitors>
    <acknowledgeAllReactors>true</acknowledgeAllReactors>
    <acknowledgeAllSVC>true</acknowledgeAllSVC>
</rrc:rrcRevise>
```

This will acknowledge all the PJM derived reserve values.

If you submit however:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rrc:rrcRevise xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:rrc="http://www.pjm.com/external/schemas/rrc/v2"
xsi:schemaLocation="http://www.pjm.com/external/schemas/rrc/v2 rrc.xsd">
    <reportRequestId>2967</reportRequestId>
    <to_responses>
     <to_response  to_name="ABC">Acknowledging all equipment</to_response>

    </to_responses>
    <acknowledgeAllCapacitors>false</acknowledgeAllCapacitors>
    <acknowledgeAllReactors>true</acknowledgeAllReactors>
    <acknowledgeAllSVC>true</acknowledgeAllSVC>
    <Capacitor ta_id="1234">
 <ratedMVAR>30</ratedMVAR>
    </Capacitor>
</rrc:rrcRevise>
```

This will acknowledge the PJM derived reserve values for all Reactors and SVCs/Statcom and at the same time, update the Rated or Active MVAR value for Capacitor ID 1234 to 30.

Note that since <acknowledgeAllCapacitors> is false, only the Capacitor updated will be acknowledged. Set it to true to acknowledge all Capacitors and update the value for ID 1234.

The reserve numbers can be positive, negative or decimal values.

The response from the upload is in xml format.

For each equipment submitted, it will the status will be "OK" if the numbers were uploaded successfully, or it will give an error message if something was wrong and the MVAR value could not be updated. For example:

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<rrc:rrcReviseResponse
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:rrc="http://www.pjm.com/external/schemas/rrc/v2"
xsi:schemaLocation="http://www.pjm.com/external/schemas/rrc/v2 rrc.xsd">
    <status ta_id="1234">
        <message>OK</message>
    </status>
    <error ta_id="1975">
        <message>Error message here</message>
    </error>
</rrc:rrcReviseResponse>
```

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/pub/etools/edart/xmldocs/rrc.xsd

Commands

To execute a self check download use type=rrcselfcheck:

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

To execute a full download use type=rrcfull:

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

To execute an equipment list download:

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

To execute an upload:

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