

**Electric Industry Data Exchange**

**Protocol Specification**

Version 4.7

Publication Date: May 24, 2017

Classification: Public

Contents

[External EIDE Interface for Peak Data 3](#_Toc483312893)

[Publication Note 3](#_Toc483312894)

[Terminology 3](#_Toc483312895)

[Submission Frequency 3](#_Toc483312896)

[Forecast Interval 4](#_Toc483312897)

[Data Submission and Error Detection 4](#_Toc483312898)

[Web Service 4](#_Toc483312899)

[SFTP Service 4](#_Toc483312900)

[Obtaining Service Credentials 4](#_Toc483312901)

[Peak Web Services Network Information 5](#_Toc483312902)

[Submitting Changed Data Only 5](#_Toc483312903)

[EIDE XML SOAP Message Overview 5](#_Toc483312904)

[**MessageInfo** 6](#_Toc483312905)

[**Schedules** 7](#_Toc483312906)

[**ScheduleDescription** 7](#_Toc483312907)

[**Quantities** 7](#_Toc483312908)

[**Quantity** 7](#_Toc483312909)

[Example: Load Forecast with an Hourly Forecast Interval 7](#_Toc483312910)

[Schedule Description 7](#_Toc483312911)

[Quantities 8](#_Toc483312912)

[Quantity 8](#_Toc483312913)

[Example: Unit Commitment with an Hourly Forecast Interval 8](#_Toc483312914)

[Schedule Description 9](#_Toc483312915)

[Quantities 9](#_Toc483312916)

[Quantity 9](#_Toc483312917)

[Jointly Owned Units 10](#_Toc483312918)

[Example C# code showing how to submit data via .NET 10](#_Toc483312919)

[.config file binding information 11](#_Toc483312920)

[EIDE web service PutSchedule WSDL, SOAP and class definitions 11](#_Toc483312921)

[Account Codes 12](#_Toc483312922)

[Version History 13](#_Toc483312923)

# External EIDE Interface for Peak Data

In order to perform Reliability Coordination (RC) functions for WECC members, Peak Reliability needs to receive various forecasts as specified in the Peak RC Data Request. Peak may also receive data from WECC members on behalf of other entities such as the Northwest Power Pool. Those non-RC entities are responsible for specifying and requesting the data they wish Peak to receive from their participants.

Peak relies on the Electric Industry Data Exchange (EIDE) standards as set forth by the WECC Data Exchange working group to receive this data. This document details the EIDE web services protocol to be used for data submission to Peak and provides some examples to help entities send data to Peak via EIDE.

This document is intended as a technical specification and reference document only. It has been written to accommodate submission of a variety of data sets, some of which your company may not be obligated to provide. Please consult the appropriate data request documents to determine what data you are required to submit to Peak. This document specifies “how” to submit data via the EIDE protocol, not “what” data” you may be required to submit.

Peak provides an Internet facing web service that utilizes PKI compliant certificates for authentication over standard https so members can automate the submission of data. The certificate can be any NAESB PKI compliant certificate or their equivalents (Verisign class 3). Once a certificate is obtained Peak will need the OID to uniquely identify you as the submitter. A secure ftp (SFTP) site will also be made available for submission for those that cannot submit the data via a web service. The internet web service will consist of only one of the EIDE methods called “PutSchedule”. The synchronous PutSchedule method schema fits all the required needs of the types of data Peak currently plans to receive.

# Publication Note

This new version of the EIDE Protocol Specification supersedes all previously published versions. This version of the specification maintains backwards compatibility with the December 2010 version. Systems that are compliant with the 2010 version of the specification should continue to operate unmodified. However, you are encouraged to adopt the new versions of the account codes specified in the Account Code section of this document.

# Terminology

## Submission Frequency

The time frequency establishing how often a particular data submission must be made is the submission frequency. Example: For the RC the minimum submission frequency for Unit Commitment data is once every day.

## Forecast Interval

The duration of time for which a specific forecast value is provided in a given data submission; it is bounded by a start time and an end time, sometimes referred to as ‘granularity’. EIDE supports forecast intervals as short as 1 minute in length.

Note: The required submission frequency and forecast interval for each data set is specified in the applicable data request document(s) as noted in the preceding section.

# Data Submission and Error Detection

It is your responsibility to monitor for submission attempt failure and take corrective action in the event of a failure. If you believe the problem is with the Peak web service please contact the Peak help desk for assistance.

Peak’s systems will continue to use the last successfully received data until such time as new data is received and successfully processed.

## Web Service

When a successful submission is made via the web service your client application will receive a “processedOK” reply code response, or an error code indicating the problem. If you attempt submission and receive no response or a “timeout” the problem may be on our end of the communication session. In this case please contact the Peak help desk to report the problem.

## SFTP Service

If you are unable to utilize the web service to submit data, an alternative method is to upload data files to Peak’s SFTP site. An Excel spreadsheet and instructions are available that allow you to enter data into the spreadsheet and produce the required XML data payload files which you must then upload to the SFTP site. The data file transfer to Peak's SFTP site is something you must do yourself using SFTP client software. Consult your IT department for assistance with this.

This spreadsheet is available for download from the PeakRC.org website or email helpdesk@peakrc.com to request a copy.

When submitting data to the SFTP site, you will place the file into the “BASent” folder. The file will be processed and placed in the “BAGood” folder if it was a properly formatted with good data. If a file was incorrectly formatted or there was a problem with the data it will be placed into the “BABad” folder. You should routinely check the “BABad” folder and if you find it contains data files, contact a Peak support engineer to help you correct the problem.

# Obtaining Service Credentials

If you do not already have credentials for the service you intend to use, call or email the Peak Helpdesk at helpdesk@peakrc.com for assistance.

# Peak Web Services Network Information

Technical information regarding Peak's web and SFTP services can be found in Peak's document: Peak Web Services Network Information. To obtain a copy of this document submit an email request to: helpdesk@peakrc.com

# Submitting Changed Data Only

While not required by this specification, in a given data submission, EIDE supports data payloads of only the data that has changed since the last submission. From a processing and bandwidth utilization perspective this method is less costly to both parties and is preferred.

Example - Hourly submission of Load Forecast with 15 minute intervals:

Last hour you submitted the following Load Forecast Profile for current day hour ending 6:

|  |  |  |
| --- | --- | --- |
| StartTime | EndTime | Value |
| 05.00.00 | 05.15.00 | 100 |
| 05.15.00 | 05.30.00 | 100 |
| 05.30.00 | 05.45.00 | 100 |
| 05.45.00 | 05.60.00 | 100 |

[NOTE: The date/time convention used is UTC “hour ending” with ordinal hours 00-23. So hour ending 1 notation is 00 and hour ending 24 notation is 23.]

At the next hourly submission, only the values for the first and third quarters of hour ending 6 have changed. In this case the following submission is allowable in the payload:

|  |  |  |
| --- | --- | --- |
| StartTime | EndTime | Value |
| 05.00.00 | 05.15.00 | 150 |
| 05.30.00 | 05.45.00 | 150 |

EIDE will continue to use the previously submitted values for the second and fourth quarters of hour ending 6 until such time as new data for these specific intervals is received.

# EIDE XML SOAP Message Overview

EIDE implements an XML SOAP message that gets sent to a Web Service. A brief description of the XML SOAP message follows. The same XML schema is used for all data submissions. Only dates, times, Account Codes (which identify the type of data), and the quantity of values being submitted change with each SOAP message.

This is the “PutSchedule” schema that is used to submit all the requested EIDE data to Peak. Each type of data set has different requirements for Submission Frequency and Forecast Interval. Reference the appropriate data request document(s) for those specific requirements.

Below is an example EIDE XML SOAP Message for a Load Forecast submission:

**<?xml version="1.0" encoding="utf-8"?>**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

 <soap:Body>

 <PutSchedule xmlns="http://www.nwpp.org/eide">

 <MessageInfo>

 <SysGenID>123</SysGenID>

 <TimeStamp>2014-07-13T13:00:00</TimeStamp>

 <Sender>NERCBA</Sender>

 <Receiver>Peak</Receiver>

 <EntityCode>NERCBA</EntityCode>

 </MessageInfo>

 <Schedules>

 <Schedule>

 <ScheduleDescription>

 <StartTime>2014-07-13T00:00:00</StartTime>

 <EndTime>2014-07-20T00:00:00</EndTime>

 <AccountCode>LoadForecast</AccountCode>

 </ScheduleDescription>

 <Quantities>

 <Quantity>

 <Value>100</Value>

 <StartTime>2014-07-13T00:00:00</StartTime>

 <EndTime>2014-07-13T01:00:00</EndTime>

 </Quantity>

 //Additional Quantities would be added in here………..

 </Quantities>

 </Schedule>

 </Schedules>

 </PutSchedule>

 </soap:Body>

 </soap:Envelope>

The message is comprised of the following sections:

**MessageInfo** **–** The MessageInfo section describes who is sending the data and who is to receive it. It contains the following tags:

**Sender**: This tag must contain the code assigned by Peak to the sender of the data. This is the entity actually transmitting the data to Peak regardless of which entity the data is for. If you are sending data for your own entity then the value for **Sender** and **EnityCode** will be the same. If you are sending data for some other entity then the value for **Sender** and **EnityCode** will be different.

**EntityCode**: This tag must contain the code assigned by Peak to the entity whom the data is for. If you are sending data for your own entity then the value for **Sender** and **EnityCode** will be the same. If you are sending data for some other entity then the value for **Sender** and **EnityCode** will be different.

If you are unsure what **Sender** or **EnityCode** codes to use, submit an email request to: helpdesk@peakrc.com

**TimeStamp**: Current date and time (UTC) that this data message was created.

**SysGenID**: This can be any number you may wish to use for tracking or audit purposes.

**Schedules** – Is an array of individual schedule submissions. Multiple schedules can be sent in a single message.

**ScheduleDescription** – ScheduleDescription describes what each individual schedule is, and the time it covers. It consists of the schedule start and end times and applicable Account Code. The AccountCode field is what tells Peak what type of values you are submitting. There can be only one account code in the ScheduleDescription section of a Schedule.

**Start DateTime**: The start date and time of the earliest Quantity record contained in the set of quantity record(s) for this particular Schedule.

**End DateTime**: The End date and time of the latest Quantity record contained in the set of quantity record(s) for this particular Schedule.

**AccountCode**: This describes what type of values you are submitting for this Schedule. The “Schedule Description” section of a data message may only have one account code.

See the Account Code section of this document for valid AccountCode definitions.

**Quantities** – This is an array of the individual quantities being submitted.

**Quantity** – This section is where the actual value is placed. It contains the number value and the start and end time for that number. You will most likely have multiple quantities being submitted per schedule.

# Example: Load Forecast with an Hourly Forecast Interval

The BA should utilize the PutSchedule method to submit this data to Peak, or deposit the XML Soap message to the SFTP site.

Schedule Description – Identifies the start time, stop time, and Account Code (type of submission) for the schedule. The start and stop time hour should be 00 in UTC time. Account Code will be “LoadForecast” that indicates to Peak this is a LF submission.

Quantities – Is an array of the load forecast values. Data will be read as Hour ending so the first quantity should have a start time hour of 00 and could have an end time hour of 01 indicating it is hour ending 1’s value. It is encouraged to use compression if the same value spans multiple hours. To indicate that hour ending 1, 2, and 3 will all be the same value, you would indicate a start time hour of 00 and an end time hour of 03 for that quantity. You do not need to submit each hour individually, but can if your pre-existing code was written that way.

Quantity – All quantities should cover the entire time of the schedule’s start and end times. For example; if you submit one schedule that covers a 7 day period you should have 168 (7 days X 24 Hours) individual quantities starting at hour 00 and ending at hour 00 if not using compression. If using compression, you must ensure the entire time is covered in the quantities submitted.

Below is an example submission for Load Forecast to Peak:

 <?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

 <soap:Body>

 <PutSchedule xmlns="http://www.nwpp.org/eide">

 <MessageInfo>

 <SysGenID>1</SysGenID>

 <TimeStamp>2014-07-13T19:02:23Z</TimeStamp>

 <Sender>NERCBA</Sender>

 <Receiver>Peak</Receiver>

 <EntityCode>NERCBA</EntityCode>

 </MessageInfo>

 <Schedules>

 <Schedule>

 <ScheduleDescription>

 <StartTime>2014-07-13T00:00:00</StartTime>

 <EndTime>2014-07-20T00:00:00</EndTime>

 <AccountCode>LoadForecast</AccountCode>

 </ScheduleDescription>

 <Quantities>

 <Quantity>

 <Value>800</Value>

 <StartTime>2014-07-13T00:00:00</StartTime>

 <EndTime>2014-07-13T01:00:00</EndTime>

 </Quantity>

 //You would add the remaining Quantity hours here.

 </Quantities>

 </Schedule>

 </Schedules>

 </PutSchedule>

 </soap:Body>

 </soap:Envelope>

# Example: Unit Commitment with an Hourly Forecast Interval

The BA should utilize the PutSchedule method to submit this data to Peak, or deposit the XML Soap message to the SFTP site

Schedule Description – Identifies the start time, stop time, and Account Code (type of submission) for the schedule. The start and stop time hour should be 00 in UTC time. Account Codes for UC will be one of the following values:

Substation.UnitName (with 1 of the following .suffixes appended)

BaseMW

WMax

WMin

OnAGC

AGCParticipationFactor

UCFlag

Consult the AccountCode section of this document for valid definitions and guidelines.

If this schedule was for the BaseMW of generation Unit 1 in the George substation the AccountCode string for this schedule would be: George.Unit1.BaseMW

Quantities – Is an array of the UC values. Data will be read as Hour ending so the first quantity could have a start time hour of 05 and could have an end time hour of 06 indicating it is hour ending 6’s value. It is encouraged to use compression if the same value spans multiple hours. To indicate that hour ending 5, 6, and 7 will all be the same value, you would indicate a start time hour of 04 and an end time hour of 07 for that quantity. You do not need to submit each hour individually, but can if your pre-existing code was written that way.

Quantity – All quantities should cover the time the unit is committed and may not match the schedule’s start and end times. The *schedule* start and stop time indicate the duration of the schedule you are submitting. The *Quantities* indicate only at what times during the schedule the unit is committed to run.

Below is an example submission for the BaseMW value of a Unit Commitment to Peak for substation George Unit1.

 <?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

 <soap:Body>

 <PutSchedule xmlns="http://www.nwpp.org/eide">

 <MessageInfo>

 <SysGenID>1</SysGenID>

 <TimeStamp>2014-07-13T19:36:44Z</TimeStamp>

 <Sender>NERCBA</Sender>

 <Receiver>Peak</Receiver>

 <EntityCode>NERCBA</EntityCode>

 </MessageInfo>

 <Schedules>

 <Schedule>

 <ScheduleDescription>

 <StartTime>2014-07-13T00:00:00</StartTime>

 <EndTime>2014-07-16T00:00:00</EndTime>

 <AccountCode>George.Unit1.BaseMW</AccountCode>

 </ScheduleDescription>

 <Quantities>

 //here is the first quantity showing the unit is starting at 16:00 on July 13th.

 <Quantity>

 <Value>200</Value>

 <StartTime>2014-07-13T16:00:00</StartTime>

 <EndTime>2014-07-13T22:00:00</EndTime>

 </Quantity>

 //You would add the remaining committed times for BaseMW for this unit here.

 </Quantities>

 </Schedule>

 </Schedules>

 </PutSchedule>

 </soap:Body>

 </soap:Envelope>

## Jointly Owned Units

EIDE has the capability to accommodate different modeling constructs for this. For example; if a unit or plant total is supplied to EIDE but that unit/plant has been modeled in the WSM as a separate facility for each share owner, the appropriate portion of the total will be allocated to each share owner’s facility in the model. The inverse of this operation is also supported.

# Example C# code showing how to submit data via .NET

using System;

using System.Collections.Generic;

using System.Text;

using EIDEConsumer.localhost; //This is the reference to the webservice

using System.Security.Cryptography.X509Certificates;

namespace EIDEConsumer

{

 class EIDEConsumer

 {

 static void Main(string[] args)

 {

 SendLF();

 }

 public static void SendLF()

 {

 MessageInfoType info = null;

 ScheduleTypeSchedule[] schedules = null;

 // Instantiate an EIDE MessageInfo object

 info = new MessageInfoType();

 info.SysGenID = 1;

 info.TimeStamp = DateTime.Now;

 info.Sender = "NERCBA";

 info.Receiver = "004";

 info.EntityCode = "NERCBA";

 // Instantiate an EIDE PutSchedule Object.

 schedules = new ScheduleTypeSchedule[1]; //Here we are only creating 1 schedule

 schedules[0] = new ScheduleTypeSchedule();

 schedules[0].ScheduleDescription = new ScheduleDescriptionType();

 schedules[0].ScheduleDescription.StartTime = DateTime.Parse(DateTime.Now.Year + "-" + DateTime.Now.Month + "-" + DateTime.Now.Day + " 00:00:00");

 schedules[0].ScheduleDescription.EndTime = DateTime.Parse(DateTime.Now.Year + "-" + DateTime.Now.Month + "-" + DateTime.Now.Day + " 00:00:00").AddDays(7);

 schedules[0].ScheduleDescription.AccountCode = "LoadForecast";

 schedules[0].Quantities = new QuantityType[168]; //Creating 168 hours of values.

 DateTime dtstart = DateTime.Parse(DateTime.Now.Year + "-" + DateTime.Now.Month + "-" + DateTime.Now.Day + " 00:00:00");

 DateTime dtend = DateTime.Parse(DateTime.Now.Year + "-" + DateTime.Now.Month + "-" + DateTime.Now.Day + " 01:00:00");

 for (int i = 0; i < 168; i++)//Loop thru and add a value for each hour

 {

 schedules[0].Quantities[i] = new QuantityType();

 schedules[0].Quantities[i].StartTime = dtstart;

 schedules[0].Quantities[i].EndTime = dtend;

 schedules[0].Quantities[i].Value = 800;//Your value would go here

 dtstart = dtstart.AddHours(1); //Move to next hour

 dtend = dtend.AddHours(1); //Move to next hour

 }

 // Set up for the web service call.

 EIDEService service = new EIDEService();

 // Set up client certificate for communications with EIDE server.

 X509Certificate certificate = X509Certificate.CreateFromCertFile(@"c:\eide\mts.cer");

 service.ClientCertificates.Add(certificate); //Add in certificate

 // Call the EIDE PutPID Method

 PutScheduleResult ret;

 ret = service.PutSchedule(info, schedules); //Send and get a response

 // Display the response

 Console.WriteLine(ret.ReplyBlock.Reply.ReplyText);

 }

 }

}

# .config file binding information

Once a reference to the EIDE service has been added to the .NET project, edit your applications .config file (typically: app.config or web.config) and ensure the binding is configured to use the certificate as the client credential as shown below:

 <basicHttpBinding>

                <binding name="EIDEServiceSoap">

                  <security mode="Transport">

                    <transport clientCredentialType="Certificate" />

                  </security>

                </binding>

                <binding name="EIDEServiceSoap1" />

            </basicHttpBinding>

# EIDE web service PutSchedule WSDL, SOAP and class definitions

This information can be obtained by using a tool like MS Visual Studio and wsdl.exe to interrogate the EIDE test web service. The URL for that service can be found in the document: Peak Web Services Network Information. To obtain a copy of this document submit an email request to: helpdesk@peakrc.com

# Account Codes

|  |  |
| --- | --- |
| **Supported EIDE Account Codes** | **Description\*** |
| ScheduledInterchange | BA Net Area Interchange forecast |
| Operating.Reserves | BA Area Operating Reserve forecast |
| Spinning.Reserves | BA Area Spinning Reserve forecast |
| Contingency.Reserves | BA Area Contingency Reserve forecast |
| ContingencySpin.Reserves | Total Spinning Contingency Reserve forecast |
| ContingencyReq.Reserves | Total Required Contingency Reserve forecast |
| MostSevereContingency.Reserves | BA Area MSSC Reserves forecast |
| LoadForecast | BA Area Load forecast |
| [SubstationName].[UnitName].BaseMW OR RP.[AreaName].]ResourceName].[SubstationName].[Description].BaseMW+ | Unit Operational Base MW forecast |
| [SubstationName].[UnitName].UCFlag OR RP.[AreaName].]ResourceName].[SubstationName].[Description].UCFlag+ | Unit Dispatch MW forecast |
| [SubstationName].[UnitName].WMax OR RP.[AreaName].]ResourceName].[SubstationName].[Description].WMax+ | Unit Operational Maximum MW forecast |
| [SubstationName].[UnitName].WMin OR RP.[AreaName].]ResourceName].[SubstationName].[Description].WMin+ | Unit Operational Minimum MW forecast |
| [SubstationName].[UnitName].OnAGC OR RP.[AreaName].]ResourceName].[SubstationName].[Description]. OnAGC+ | Unit AGC Status forecast |
| [SubstationName].[UnitName].AGCPF OR RP.[AreaName].]ResourceName].[SubstationName].[Description]. AGCPF+ | Unit AGC Participation Factor forecast |
| [PathName].LimitMW OR [PathName].[WSMPathName].[TOP].[FlowDirection].LimitMW+ | Path Limit SOL forecast |
| SOL.[TOP].[InterfaceName].[WSMInterfaceName].[FlowDirection].[Description]+ | Interface Limit SOL forecast |

\*Refer to the Peak RC Data Request or the applicable Data Exchange document for complete definitions and submission requirements.

+Work with Peak Reliability staff by contacting data.request@peakrc.com to develop the names for marked account codes.

# Version History

| **Rev.** | **When** | **Who** | **What** |
| --- | --- | --- | --- |
| 3.1 | 07/31/2014 | Peak | Draft Release |
| 3.2 | 08/21/2014 | Peak | Draft for review. Note: Previous versions of this document were titled: External EIDE Interface for Peak RC Data. Changed the title somewhat but did not re-set the version number for continuity. Added new Account Codes. |
| 4.0 | 09/08/2014 | Peak | Draft for review. Changed major version number to 4 since this is a significant format change from previous versions.Normalized repetitive content down to a single representative example.Removed the account code listing since the canonical list of account codes will be maintained in other document(s) from this point forward.  |
| 4.1 | 09/22/2014 | Peak | New release for publication to WECC members.  |
| 4.2 | 09/22/2014 | Peak | Withdrawn |
| 4.3 | 09/25/2014 | Peak | Correction to data service addresses. |
| 4.4 | 11/10/2014 | Peak | Correction to .x509 certificate info.  |
| 4.5 | 12/22/2015 | D. Aronson | Updates for new account codes |
| 4.6 | 02/21/2017 | N. Timinskas | Updated “Supported EIDE Account Codes” table |
| 4.7 | 05/24/2017 | D. Aronson | Added binding info for client .config file |