

# External Interface Requirements

Data Submissions

WECC Version 2.0



01 Dec 10

## Contents

External EIDE Interface for WECC RC Data .....	4
MessageInfo.....	5
Schedules .....	5
Quantities.....	5
Quantity .....	6
Path Limits Submission .....	6
MessageInfo.....	6
Schedules .....	6
Schedule Description .....	6
Quantities.....	6
Quantity .....	7
Hourly Load Forecast Submission .....	8
MessageInfo.....	8
Schedules .....	8
Schedule Description .....	8
Quantities.....	8
Quantity .....	8
Unit Commitment Submission .....	9
MessageInfo.....	10
Schedules .....	10
Schedule Description .....	10
Quantities.....	10
Quantity .....	10
Scheduled Interchange Submission .....	11

MessageInfo.....	11
Schedules .....	11
Schedule Description .....	12
Quantities.....	12
Quantity .....	12
Reserves Submission.....	13
MessageInfo.....	13
Schedules .....	13
Schedule Description .....	13
Quantities.....	13
Quantity .....	13
Example C# code showing how to submit data via .NET .....	14

## External EIDE Interface for WECC RC Data

WECC needs to receive Load Forecast, Operating Reserves, Unit Commitment, and Scheduled Interchange data thru an external interface to fulfill various study activities. WECC is going to utilize the EIDE standards as set forth by the Data Exchange Working group to accomplish this. This document will detail what WECC needs to receive along with examples to help the BA fulfill these requirements.

WECC will be providing an Internet facing web service that will utilize PKI compliant certificates for authentication over standard https so BA's 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 WECC will need the OID to uniquely identify you as the submitter. A secure ftp 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 "putSchedule". The synchronous putSchedule method schema fits all the required needs of the types of data WECC currently plans to receive.

EIDE is basically an XML Text SOAP message that gets sent to a Web Service. This same XML Text SOAP message could be deposited to a secure ftp site that is also available. A brief description of the XML SOAP message follows. The same schema will be used for all submissions with the only things changing being the date and times, account codes (which identify the type of data), and the quantity of values being submitted.

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>2008-07-13T13:00:00</TimeStamp>
        <Sender>NERCBA</Sender>
        <Receiver>WECC</Receiver>
        <EntityCode>NERCBA</EntityCode>
      </MessageInfo>
      <Schedules>
        <Schedule>
          <ScheduleDescription>
```

```

    <StartTime>2008-07-13T00:00:00</StartTime>
    <EndTime>2008-07-20T00:00:00</EndTime>
    <AccountCode>LoadForecast</AccountCode>
</ScheduleDescription>
<Quantities>
  <Quantity>
    <Value>100</Value>
    <StartTime>2008-07-13T00:00:00</StartTime>
    <EndTime>2008-07-13T01:00:00</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 sections describes who is sending the data and who is to receive it. It contains an EntityCode field which allows for a different company to submit for other companies on their behalf. All senders, receivers, and entities should be represented by the NERC ID number registered to NERC for that company. It also contains a SysGenID which is a number created by the creator of the message that uniquely identifies this message. The final field of interest is the TimeStamp field which indicates the time the message was sent.

**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 the AccountCode. The AccountCode is what tells WECC what kind of values you are submitting. The AccountCodes will be:

1. LoadForecast
2. ScheduledInterchange
3. Spinning.Reserves
4. Operating.Reserves
5. MostSevereContingency.Reserves
6. SubstationName.UnitName.BaseMW
7. SubstationName.UnitName.WMax
8. SubstationName.UnitName.WMin
9. SubstationName.UnitName.OnAGC
10. PathName.LimitMW

These are the only possible account codes and types of information we are expecting.

**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.

That is the base form that is used to submit all the requested data to WECC. Each type of data has different numbers of days of data we are asking for. Each AccountCode data type submission and time periods are described in the sections below.

An Excel spreadsheet has been included that allows you to enter data via a spreadsheet and produced the desired output message. Please see the Spreadsheets readme file for instructions on use.

## Path Limits Submission

WECC needs to receive Path Limits from each Path Operator. The Path Data WECC needs to receive is:

TIME = The time of the limit in UTC hour ending.

LimitMW = Path Limit value for the hour in MW for the Path.

The BA should utilize the putSchedule method to submit this data to WECC, or deposit the XML Soap message to the secure ftp site. This data should be submitted as soon as a future path limits are available or it changes. You can submit this data as many times as you wish during the day. The most recent data submitted is what will be used. Please send a rolling 3 day forecast for each submission, starting on the current day of submission at hour 00.

The putSchedule method of EIDE is composed of the following parts:

**MessageInfo** – Which uniquely identifies the submission of the data and identifies the sender and receivers. The NERC ID's will be used to identify the sender and receiver. SysGenID is a unique number that identifies the message as identified by the sender. You may use any integer number for tracking purposes. TimeStamp is the time you created the message.

**Schedules** – Is an array of Path Limit forecast submissions. You may submit one schedule that has a start and end time covering the entire 3 days, or you can submit one schedule for each day of submission.

**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. AccountCode will be "PathNumber.LimitMW" that indicates to WECC this is a Path Limit submission (Path Numbers should match the WECC Path Catalog Numbers). TagID will hold the annotated description of why the Path is restricted (this is a requirement for restricted submissions). A separate schedule should be submitted for each restriction time period.

**Quantities** – Is an array of the Path Limit 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. If you submit one schedule that covers the entire 3 day period you should have 72 (3 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 Path Limit Submission to WECC:

```
<?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>2008-07-13T19:02:23Z</TimeStamp>
        <Sender>NERCBA</Sender>
        <Receiver>WECC</Receiver>
        <EntityCode>NERCBA</EntityCode>
      </MessageInfo>
      <Schedules>
        <Schedule>
          <ScheduleDescription>
            <StartTime>2008-07-13T00:00:00</StartTime>
            <EndTime>2008-07-20T00:00:00</EndTime>
            <AccountCode>40.LimitMW</AccountCode>
            <TagID>If limits are restricted then a schedule should be submitted just covering the restriction period and
annotating the reason for restriction here. There should be a schedules to cover the entire 3 day submission
period.</TagID>
          </ScheduleDescription>
          <Quantities>
            <Quantity>
              <Value>800</Value>
              <StartTime>2008-07-13T00:00:00</StartTime>
              <EndTime>2008-07-13T01:00:00</EndTime>
            </Quantity>
            //You would add the remaining Quantity hours here.
          </Quantities>
        </Schedule>
      </Schedules>
    </PutSchedule>
  </soap:Body>
</soap:Envelope>
```

When a successful submission is made via the web service you will receive a "processedOK" reply code as indicated in the EIDE standard for a response, or an error code indicating the problem. When submitting the data to the secure ftp site, you will place the file into the "BASent" folder. The message will be processed and placed in the "BAGood" folder if it was a properly formatted with good data. Files will be placed in the "BABad" folder indicating the file was incorrectly formatted or there was a problem with the data.

## Hourly Load Forecast Submission

WECC needs to receive Load Forecast Data from each BA. The Load Forecast Data WECC needs to receive is:

TIME = The time of the forecast in UTC hour ending.

LOAD = Load value for the hour in MW for the BA.

The BA should utilize the putSchedule method to submit this data to WECC, or deposit the XML Soap message to the secure ftp site. This data should be submitted as soon as a Load Forecast is available or it changes. You can submit this data as many times as you wish during the day. The most recent data submitted is what will be used. Please send a rolling 7 day forecast for each submission, starting on the current day of submission at hour 00.

The putSchedule method of EIDE is composed of the following parts:

**MessageInfo** – Which uniquely identifies the submission of the data and identifies the sender and receivers. The NERC ID's will be used to identify the sender and receiver. SysGenID is a unique number that identifies the message as identified by the sender. You may use any integer number for tracking purposes. TimeStamp is the time you created the message.

**Schedules** – Is an array of forecast submissions. You may submit one schedule that has a start and end time covering the entire 7 days, or you can submit one schedule for each day of submission.

**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. AccountCode will be "LoadForecast" that indicates to WECC 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. If you submit one schedule that covers the entire 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 WECC:

```
<?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>2008-07-13T19:02:23Z</TimeStamp>
      <Sender>NERCBA</Sender>
      <Receiver>WECC</Receiver>
      <EntityCode>NERCBA</EntityCode>
    </MessageInfo>
    <Schedules>
      <Schedule>
        <ScheduleDescription>
          <StartTime>2008-07-13T00:00:00</StartTime>
          <EndTime>2008-07-20T00:00:00</EndTime>
          <AccountCode>LoadForecast</AccountCode>
        </ScheduleDescription>
        <Quantities>
          <Quantity>
            <Value>800</Value>
            <StartTime>2008-07-13T00:00:00</StartTime>
            <EndTime>2008-07-13T01:00:00</EndTime>
          </Quantity>
          //You would add the remaining Quantity hours here.
        </Quantities>
      </Schedule>
    </Schedules>
  </PutSchedule>
</soap:Body>
</soap:Envelope>

```

When a successful submission is made via the web service you will receive a “processedOK” reply code as indicated in the EIDE standard for a response, or an error code indicating the problem. When submitting the data to the secure ftp site, you will place the file into the “BASent” folder. The message will be processed and placed in the “BAGood” folder if it was a properly formatted with good data. Files will be placed in the “BABad” folder indicating the file was incorrectly formatted or there was a problem with the data.

## Unit Commitment Submission

WECC needs to receive Unit Commitment Data from each BA. The Unit Commitment data WECC needs to receive is:

Start Time =The start time of the Unit in UTC.

Stop Time =The stop time of the Unit in UTC.

Unit Name=Unit substation and name

Base MW = Unit’s base MW

On AGC = Indicator of AGC status (1=On AGC, and a 2=Off AGC)

WMAX = The MW max of the unit.

WMIN = The MW min of the unit

The BA should utilize the putSchedule method to submit this data to WECC, or deposit the XML Soap message to the secure ftp site. This data should be submitted as soon as a UC is available or it changes. Submit Unit Commitment for as many days out as you have. At a minimum please submit for the current day and next 2 days.

The putSchedule method of EIDE is composed of the following parts:

**MessageInfo** – Which uniquely identifies the submission of the data and identifies the sender and receivers. The NERC ID's will be used to identify the sender and receiver. SysGenID is a unique number that identifies the message as created by the sender. TimeStamp is the time the message was created.

**Schedules** – Is an array of UC schedules. You will be submitting multiple schedule information for each Unit's values required by WECC during its committed time periods.

**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. AccountCodes for UC will be one of the following values SubstationName.UnitName.BaseMW, SubstationName.UnitName.OnAGC, SubstationName.UnitName.WMax, or SubstationName.UnitName.WMin where SubstationName is your unit's substation and UnitName is your unit's name. An example for a BaseMW at the George substation for Unit 1 would have an account code of 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 complete days you are submitting for. The *Quantities* indicate only at what times during the schedule days the unit is committed to run.

Below is an example submission for the BaseMW value of a Unit Commitment to WECC 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>2008-07-13T19:36:44Z</TimeStamp>
        <Sender>NERCBA</Sender>
        <Receiver>WECC</Receiver>
```

```

    <EntityCode>NERCBA</EntityCode>
  </MessageInfo>
  <Schedules>
    <Schedule>
      <ScheduleDescription>
        <StartTime>2008-07-13T00:00:00</StartTime>
        <EndTime>2008-07-16T00:00:00</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>2008-07-13T16:00:00</StartTime>
          <EndTime>2008-07-13T22:00:00</EndTime>
        </Quantity>
        //You would add the remaining committed times for BaseMW for this unit here.
      </Quantities>
    </Schedule>
  </Schedules>
</PutSchedule>
</soap:Body>
</soap:Envelope>

```

You will be submitting 4 schedules for each unit's values. When a successful submission is made via the web service you will receive a "processedOK" reply code as indicated in the EIDE standard for a response, or an error code indicating the problem. When submitting the data to the secure ftp site, you will place the file into the "BASent" folder. The message will be processed and placed in the "BAGood" folder if it was a properly formatted with good data. Files will be placed in the "BABad" folder indicating the file was incorrectly formatted or there was a problem with the data.

## Scheduled Interchange Submission

WECC needs to receive Scheduled Interchange Data from each BA. The Scheduled Interchange data WECC needs to receive is:

TIME =The time of the Scheduled Interchange in UTC.

MW\_SCHEDULE = Interchange value for the specified hour.

The BA should utilize the putSchedule method to submit this data to WECC or can be posted to the secure ftp site. This data should be submitted as soon as a Scheduled Interchange is available or it changes. Please send a rolling 3 day submission (current day and 2 additional days).

The putSchedule method of EIDE is composed of the following parts:

**MessageInfo** – Which uniquely identifies the submission of the data and identifies the sender and receivers. The NERC ID's will be used to identify the sender and receiver. SysGenID is a unique number that identifies the message as identified by the sender. TimeStamp is the time the message was created.

**Schedules** - - Is an array of Interchange Schedules. You may submit one schedule that has a start and end time covering the entire period, or you can submit one schedule for each day of submission.

***Schedule Description*** – Identifies the start time and stop time for that Interchange Schedule, as well as identifies the BA Interchange NERC ID. This name is typically the same NERC ID as the sender. The start and stop time should be on Hour 00. AccountCode should be “ScheduledInterchange” to indicate to WECC that this is a Scheduled Interchange submission.

***Quantities*** – Should contain individual quantity entries to identify the Interchange for each hour of the submission period.

***Quantity*** – Should contain the start and stop time of a submission time period along with the Interchange value for that period’s hours. There should be quantity submissions to cover the entire time of the schedule.

Below is an example submission for Scheduled Interchange to WECC:

```
<?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>2008-07-13T17:37:49</TimeStamp>
        <Sender>NERCBA</Sender>
        <Receiver>WECC</Receiver>
        <EntityCode>NERCBA</EntityCode>
      </MessageInfo>
      <Schedules>
        <Schedule>
          <ScheduleDescription>
            <StartTime>2008-07-13T00:00:00</StartTime>
            <EndTime>2008-07-16T00:00:00</EndTime>
            <AccountCode>ScheduledInterchange</AccountCode>
          </ScheduleDescription>
          <Quantities>
            <Quantity>
              <Value>800</Value>
              <StartTime>2008-07-13T00:00:00</StartTime>
              <EndTime>2008-07-13T01:00:00</EndTime>
            </Quantity>
            //You would add remaining quantities here
          </Quantities>
        </Schedule>
      </Schedules>
    </PutSchedule>
  </soap:Body>
</soap:Envelope>
```

When a successful submission is made via the web service you will receive a “processedOK” reply code as indicated in the EIDE standard for a response, or an error code indicating the problem. When submitting the data to the secure ftp site, you will place the file into the “BASent” folder. The message will be processed and placed in the “BAGood” folder if it was a properly formatted with good data. Files will be placed in the “BABad” folder indicating the file was incorrectly formatted or there was a problem with the data.

## Reserves Submission

WECC needs to Reserve information from each BA. The Reserve data WECC needs to receive is:

Spinning Reserves= An hour ending value indicating the BA's Spinning Reserves.

Operating Reserves = An hour ending value indicating the BA's Operating Reserves.

Most Severe Contingency =An hour ending value for the single largest contingency in MW's for that BA's area.

The BA should utilize the putSchedule method to submit this data to WECC, or deposit the XML Soap message to the secure ftp site. This data should be submitted as soon as a Reserve requirement is defined. Submit Reserve schedules for as many days out as you have. At a minimum please submit for the current day and next 2 days.

The putSchedule method of EIDE is composed of the following parts:

**MessageInfo** – Which uniquely identifies the submission of the data and identifies the sender and receivers. The NERC ID's will be used to identify the sender and receiver. SysGenID is a unique number that identifies the message as created by the sender. TimeStamp is the time the message was created.

**Schedules**- Is an array of Reserve schedules. You will be submitting multiple schedule information for each Reserve component required by WECC.

**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. AccountCodes for Reserves will be one of the following values Spinning.Reserves,Operating.Reserves, or MostSevereContingency.Reserves.

**Quantities** – Is an array of the Reserve values. Data will be read as Hour ending so the first quantity would have a start time hour of 00 and would have an end time hour of 01 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 entire time of the schedule's start and end times.

Below is an example submission for Spinning.Reserves:

```
<?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>2008-07-13T19:36:44Z</TimeStamp>
    <Sender>NERCBA</Sender>
    <Receiver>WECC</Receiver>
    <EntityCode>NERCBA</EntityCode>
  </MessageInfo>
  <Schedules>
    <Schedule>
      <ScheduleDescription>
        <StartTime>2008-07-13T00:00:00</StartTime>
        <EndTime>2008-07-16T00:00:00</EndTime>
        <AccountCode>Spinning.Reserves</AccountCode>
      </ScheduleDescription>
      <Quantities>
        //here is the first quantity showing the unit is starting at 16:00 on July 13th.
        <Quantity>
          <Value>200</Value>
          <StartTime>2008-07-13T00:00:00</StartTime>
          <EndTime>2008-07-13T01:00:00</EndTime>
        </Quantity>
        //You would add the remaining hours for spinning reserve here.
      </Quantities>
    </Schedule>
  </Schedules>
</PutSchedule>
</soap:Body>
</soap:Envelope>

```

You will be submitting schedules for all three account codes for reserves. When a successful submission is made via the web service you will receive a “processedOK” reply code as indicated in the EIDE standard for a response, or an error code indicating the problem. When submitting the data to the secure ftp site, you will place the file into the “BASent” folder. The message will be processed and placed in the “BAGood” folder if it was a properly formatted with good data. Files will be placed in the “BABad” folder indicating the file was incorrectly formatted or there was a problem with the data.

## 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);
}
}
}

```

That's it. This is showing a Load Forecast submission, all other submissions are similar you are just changing the AccountCode, time periods and values.