

# MANUAL



Power to Ontario.  
On Demand.

---

# IESO Developer's Toolkit (IDK)

Implementation Manual

---

[Issue 11.1](#)

This document contains a guide to accessing MIM data  
programmatically.

## Disclaimer

The posting of documents on this Web site is done for the convenience of *market participants* and other interested visitors to the *IESO* Web site. Please be advised that, while the *IESO* attempts to have all posted documents conform to the original, changes can result from the original, including changes resulting from the programs used to format the documents for posting on the Web site as well as from the programs used by the viewer to download and read the documents. The *IESO* makes no representation or warranty, express or implied, that the documents on this Web site are exact reproductions of the original documents listed. In addition, the documents and information posted on this Web site are subject to change. The *IESO* may revise, withdraw or make final these materials at any time at its sole discretion without further notice. It is solely your responsibility to ensure that you are using up-to-date documents and information.

This document may contain a summary of a particular *market rule*. Where provided, the summary has been used because of the length of the *market rule* itself. The reader should be aware, however, that where a *market rule* is applicable, the obligation that needs to be met is as stated in the “Market Rules”. To the extent of any discrepancy or inconsistency between the provisions of a particular *market rule* and the summary, the provision of the *market rule* shall govern.

<b>Document ID</b>	IMO_MAN_0023
<b>Document Name</b>	IESO Developer's Toolkit (IDK) Implementation Manual
<b>Issue</b>	<a href="#">Issue 11.1</a>
<b>Reason for Issue</b>	Revised for <a href="#">EDAC connectivity testing and future PKI decommissioning</a>
<b>Effective Date</b>	<a href="#">Tentative June 1, 2011</a>

## Document Change History

Issue	Reason for Issue	Date
1.0	First release for Baseline 8.0	September 25, 2002
2.0	Released for Baseline 9.1	June 4, 2003
3.0	Released for change of PKI Certification Authority	February 18, 2004
4.0	Released for baseline 12.0 due to upcoming changes to certificate mode of use.	September 1, 2004
5.0	Released for baseline 14.0 due to IMO to <i>IESO</i> name change and for Certification Authority IP address and domain name changes.	September 30, 2005
6.0	Change to MIM API URL to reflect name change	March 8, 2006
7.0	Release for baseline 17.0	March 7, 2007
8.0	Release for baseline 19.0	March 5, 2008
9.0	Release for baseline 20.0	September 10, 2008
10.0	Revised for Verizon Data Center Move date change from May to June 2009	June 5, 2009
11.0	Revised for Verizon CA Renewal May 2010	March 26, 2010
<a href="#">11.1</a>	<a href="#">Revised for EDAC connectivity testing and future PKI decommissioning</a>	<a href="#">Tentative June 1, 2011</a>

## Related Documents

Document ID	Document Title
IMP_GDE_0088	Market Manual 1: Market Entry, Maintenance & Exit, Part 1.3: Identity Management Operations Guide
IMO_MAN_0024	Market Manual 6: Participant Technical Reference Manual
IMO_GDE_0003	Market Participant Graphical User Interface User's Guide



# Table of Contents

<b>1. Introduction</b>	<b>5</b>
1.1 Purpose	5
1.2 Scope	5
1.2.1 Version 7.4 <del>2</del> Digital Certificates Use	5
1.3 Who Should Use This Document	5
1.4 Conventions	5
1.4.1 General Conventions	5
1.4.2 Specific Conventions	6
1.4.3 Notes, Cautions, Warnings, and Tips	7
<b>2. IDK Overview</b>	<b>8</b>
2.1 Introduction	8
2.1.1 MIM IDK Version 3.1 <del>and Higher</del> 3	9
2.1.2 MIM IDK System Requirements	10
2.1.3 API Design Concepts	11
2.1.4 API Overview	13
<b>3. IDK Usage</b>	<b>14</b>
3.1 Creating an MPIRequest Object	14
3.1.1 <u>Certificate Authentication: Automatic Key-Rollover</u>	14
3.1.2 <u>Certificate Authentication: No Automatic Key-Rollover</u>	16
<u>3.1.3 Username/Password Authentication</u>	17
3.2 Login/Logout of an MPIRequest Object	18
3.2.1 MIM IDK 3.1 <del>and Higher</del> 3	18
3.3 Uploading Bids	21
3.4 Downloading Bids	21
3.4.1 Resetting Bid Query Parameters	23
3.4.2 Setting Bilateral Bid Query Data	23
3.4.3 Setting Capacity Reserve Bid Query Data	24
3.4.4 Setting DAEFM Bid Query Data	24
3.4.5 Setting Operating Reserve Bid Query Data	25
3.4.6 Setting RTEM Bid Query Data	26
3.4.7 Setting Schedule Bid Query Data	27
3.4.8 <u>Setting Daily Generator Data (DGD) Query Data</u>	28
<u>3.4.9 Requesting Bids</u>	28
3.5 Downloading System Messages	29
3.6 Downloading Market Status Information	29
3.7 Miscellaneous Methods	29

---

<b>Appendix A:</b>	<b>IDK Bundle Descriptions .....</b>	<b>31</b>
--------------------	--------------------------------------	-----------

## Table of Changes

Reference (Section and Paragraph)	Description of Change
<a href="#">2.1.1</a>	<a href="#">Added content to update version of IDK and include support for UserID / Password login.</a>
<a href="#">3.1.2</a>	<a href="#">Added content to update version of IDK and remove content regarding CA replacement and IP address changes.</a>
<a href="#">2.1.4</a>	<a href="#">Added content for uploading and downloading Daily Generator Data.</a>
<a href="#">3.1</a>	<a href="#">Updated description of MPIRequest constructors and indicated availability of 2 test batch files included in the IDK bundle.</a>
<a href="#">3.1.1, 3.1.2</a>	<a href="#">Updated heading to include certificate authentication and removed old CA IP addresses.</a>
<a href="#">3.1.3</a>	<a href="#">Added section to deal with Username/password authentication method.</a>
<a href="#">3.2.1</a>	<a href="#">Added subsection headings and modified content to deal with Entrust (PKI) login and Username/password login as well as login/logout method Signatures</a>
<a href="#">3.3</a>	<a href="#">Updated content to include Daily Generator Data.</a>
<a href="#">3.4.8</a>	<a href="#">Added section to include method to deal with Daily Generator Data.</a>
<a href="#">Appendix A</a>	<a href="#">Added content to deal with availability of 2 test batch files included in the IDK bundle.</a>

|

# 1. Introduction

---

## 1.1 Purpose

The purpose of this document is to describe the programmatic application interface that allows access to the *IESO* MIM system (i.e. MIM IDK).

## 1.2 Scope

This document discusses the programmatic API method whereby *market participants* can gain access to the *IESO* MIM system. The browser method using Internet Explorer is discussed in the *Market Participant Graphical User Interface User's Guide*.

Both methods use digital certificate authentication credentials.

### 1.2.1 Version 7.2 Digital Certificates Use

The upgraded MIM IDK uses digital certificates issued from the version 7.2 Entrust Authority system operated by Verizon to authenticate to the *IESO* MIM.

The version 7.2 production certificate may be used with both the Production and Sandbox MIM systems where applicable (i.e. when and where the *market participant* has been issued the same user account ID for both Production and Sandbox MIM systems by the *IESO*).

## 1.3 Who Should Use This Document

The document is intended for the use of *market participants* who require programmatic access to the MIM.

## 1.4 Conventions

### 1.4.1 General Conventions

The standard conventions followed in this document are as follows:

- The word 'shall' denotes a mandatory requirement;
- Terms and acronyms used in this document that are *italicized* have the meanings ascribed thereto in Chapter 11 of the *Market Rules*;

- Double quotation marks are used to indicate titles of legislation, publications, forms and other documents.

## 1.4.2 Specific Conventions

The following text conventions are used in this document to distinguish between different type of text. Refer to this section when clarification is needed to determine the proper convention to use.

Item	Convention	Example
a.m., p.m.	Small caps	A.M., P.M.
Book, document, section and chapter titles	Initial capitals, italics	Refer to the <i>Visual Basic Custom Control Reference</i>
Buttons on screens	Uppercase, bold, italics	<b>SAVE</b> button <b>QUERY</b> button
Class Name	Lowercase, Helvetica 9 pt., italics	LDatabase
Command lines and options	Lowercase, Helvetica 9 pt., bold	copy command /a option
Command Parameter	Lowercase, Helvetica 9 pt., bold, italics	(blue)
Database table names	All capitals, regular	Cust_org
File name	Lowercase, italics	docid.txt
GUI Field names	Initial capitals, italic	<i>Primary Provider</i> field
Icon names	Initial capitals, italic	Click the <i>Microsoft Excel</i> icon.
Key names, key combinations, and key sequences.	Small caps enclosed in brackets.	<ctrl>, <tab>, <ctrl+alt+del>, <shift+f7>
Menu commands	Initial capitals, separated by >, bold.	Select <b>File &gt; Save</b> . Select <b>Format &gt; Font &gt; Arial</b> .
Menu names	Bold, initial capitals	<b>Insert</b> menu <b>File</b> menu <b>Format</b> menu
Product Reference	Lowercase, bold, italics	eaplus
Program application names	Lowercase, italics	condmgr.exe
Program code (Use when lines number 2 or less)	Lowercase, Helvetica 9 pt., regular	\$ set def clc\$cin
Program code 2 (Use when there are more than two lines of code)	Lowercase, Helvetica 9 pt., regular	\$ set def clc\$cin

<b>Item</b>	<b>Convention</b>	<b>Example</b>
Tech Keyword	Lowercase, italics	garbage
User input	Lowercase, italic	Type <i>password</i>

### 1.4.3 Notes, Cautions, Warnings, and Tips

Notes, Cautions, Warnings, and Tips are used throughout this document to call attention to information of special importance. The different circumstance where each of these is used is described below.

**Note:** A Note is used to indicate neutral or positive information that emphasizes or supplements important points of the main text, to supply information that may apply only in special cases.

**Caution:** A Caution is used when system performance may be affected and of potential damage to the equipment, data, or software.

**Warning** : A Warning is used when system performance may be affected and of potential damage to the equipment, data, or software.

**Tip:** A Tip helps apply the techniques and procedures described in the text to their specific needs. A tip suggests alternative methods that may not be obvious and helps the user understand the benefits and capabilities of the application.

– End of Section –

## 2. IDK Overview

---

### 2.1 Introduction

The MIM programmatic API represents the Internet-based client gateway to functionality provided by the *IESO* Energy Bidding System. Communications with the MIM system is conducted using the industry standard HTTPS protocol. Within HTTPS, the SSL version 3.0 protocol in conjunction with *IESO* issued digital certificates is used to establish mutual trust and security between the participant and the MIM system. The MIM programmatic API is used to usher participant requests/responses to and from the MIM in a simple, secure and robust manner without the use of a browser.

The *IESO* Development toolkit (MIM IDK) which provides the programmatic API emulates the functionality provided by browsers with an emphasis on text-based data responses.

Participants are thus allowed programmatic access to MIM functionality via participant-built applications. Consequently, MIM operations may be scheduled based on *market participant* system's timing and or data constraints. Browsers, on the other hand, are GUI based entities that interpret tag languages, such as HTML, and are typically driven via the mouse/keyboard. This document discusses the IDK.

Requests are routed from the participants to the Web server. These requests are then routed to the *IESO* application server (IAS); participant profile Directory Server (DS) or external *IESO* subsystem (the latter will not be discussed in this document) based on their types. During login, participant specific identification information contained within the presented digital certificate is validated against information within the Certification Authority Manager and directory server systems, and the *IESO* MIM Directory Server (DS). If a valid authentication and authorization match is made, login is granted. Part of login is determining what privileges are granted to the participant user identity presented via the digital certificate. For example, access to the Physical Market may be granted or denied based on the referenced authorization profile. This information is fetched from the MIM DS.

There may be circumstances where communications to the designated Certification Authority (CA) directory server (via the domain name/IP parameters as specified in section 3.1) used to authenticate the presented digital certificate and corresponding profile may not be possible by any *market participant*. This may occur due to planned or unplanned outages of the Certification Authority systems or outages of Internet communication infrastructure systems elsewhere that the *market participant* or *IESO* have no control over, that affects all users. Under these conditions the normal, certificate on-line login where revocation checking against the current Certificate Revocation List located on the specified CA directory server will not be possible and login to the MIM system with the MIM API will fail with a 'directory not available' or related type error. To counteract these situations, the *IESO* shall, in addition to planned CA outage situations, continually monitor the availability of the relevant CA directory servers and centrally enable off-line mode certificate use within the MIM API for *market participants* when the need arises. This shall enable general continuity of market systems access via the MIM API (and MPI). At no time shall the *market participant* need to or be able to configure this on the local client or modify any of the client CA IP address configurations as they are now used.

Energy bids may be uploaded via the MIM IDK for validation and potential insertion by the bid processing software within the IAS. Likewise, inserted bids may be retrieved for subsequent viewing, editing and/or resubmission.

Participants may retrieve their messages from the MIM system message database. These messages have both a priority value and target audience. The priority values are normal, urgent and emergency. An individual message may be targeted towards an individual participant, all participants of a given organization, a market group or everyone. Web based *Market Participants* have their messages automatically polled for via an applet running within the browser. MIM IDK participants ask for this data programmatically via a MIM IDK provided method.

The *Market Status* display data consists of bid submission table information. This table contains the status of all markets for all hours and the submission windows, which make up those markets. For example, bids may not be submitted via the MIM unless the corresponding *Market Status* table entry allows it. Web based *Market Participants* have Market Status information polled for automatically. MIM IDK participants are provided a method with which to fetch this data, which must be used by the *market participant* application in order to do so.

As of release 20.0 the MIM system will no longer be used to return reports to Market Participants. Report categories affected are as follows: :

- Financial Market
- Physical Market
- AMP (all *market participants*)
- Settlement
- Invoice

Note: All reports within the categories listed above are -available through the *IESO* Reports Site at: <https://reports.ieso.ca/private/>.

The MIM IDK is entirely Java based, built using Sun's J2SE JDK along with PKI code using the Entrust Java toolkit. The MIM IDK wraps lower level *IESO* /ABB provided Java class libraries to maximize ease of use. These same wrappers are utilized within the Navigator and IE browser environments. They provide the following functionality:

- Template Upload
- Template Download
- System Message Download
- Market Status Download

### 2.1.1 MIM IDK Version 3.23

The Entrust Java Toolkit version 7.2 libraries have been bundled within with the MIM IDK version 3.23 to provide updated PKI functionality. This enables the use of updated and improved Entrust PKI

methods required for use with the Entrust Authority 7.2 version system operated by the Cybertrust Certification Authority (now owned by Verizon). [The MIM IDK version 3.3 also supports username/password authentication with SSLv3.](#)

## 2.1.2 MIM IDK System Requirements

The system requirements are as follows:

### MIM IDK Version 3.23 Requirements

Version 3.23 of the MIM IDK has been tested to be compatible with the J2SE JDK version as indicated on the IESO Supported Client Platform web page at : [http://www.ieso.ca/imoweb/ti/ti\\_Supported-Client-Platform.asp](http://www.ieso.ca/imoweb/ti/ti_Supported-Client-Platform.asp). Client side programs using the supported version of the MIM IDK will need to be updated to have access to a JVM capable of processing Java byte-code generated from this JDK.

The required JDK and JRE are currently available at: <http://java.sun.com/products/archive/>

Note that the JDK is not mandatory on client machine running the IDK; while the JRE is. The JDK install contains the JRE by default but the JRE can be installed by itself. The JDK and JRE can be downloaded from Sun's Java products website at no cost (financially) although a click-through license agreement is required to be agreed to.

An EPF (Entrust Profile) file containing digital certificates created with the *IESO* CLS software and *IESO* issued activation codes is required for login to the *IESO* secure web server with the MIM IDK.

Version 3.23 of the MIM IDK enables the following certificate usage functionality:

- A certificate created or recovered with the *IESO* CLS version 1.8 from the Entrust Authority 7.2 system at Cybertrust may be used.
- A certificate created or recovered with Entrust Authority Administration tool provided by the Cybertrust Certification Authority from the Entrust Authority 7.2 system at Cybertrust may be used.

Use of the *IESO* CLS application and the Entrust Authority Administration tool are both documented in the *IESO Identity Management Operations Guide*.

### Certificate Account Requirement

The certificate accounts used with the MIM IDK are Application Subscriber role certificates typically identifying *the Market Participant* and computer system the IDK is operating on. The EPF file contains all the PKI information necessary to authenticate the user/machine account to the MIM system, establish secure Internet communications and generate digital signatures of transactions as required by the Web and application servers.

### Other System Requirements

The IDK has the following other requirements:

- Sufficient memory for the JVM and associated programs. For example, a PC with a minimum of 256 MB system memory.
- A UNIX or PC platform. For example, Intel P4 1.0 GHz based PC running Windows NT, Service Pack 6 or later. However current dual-core CPU, multi gigahertz and multi-gigabyte memory platforms running the latest recommended version of Windows, UNIX or Linux would be more appropriate. The *IESO* has not tested the MIM IDK with multi CPU platforms.
- Sufficient disk space to hold the IDK, approximately 1.5-MB.

### IDK Version 3.23 Communication Parameter Requirements

1. The domain name (i.e. IP address) of the 7.2 Certificate Authority (CA) directory server located at the CA operations center. The *IESO*'s CA updates the directory servers on a scheduled and as-needed basis.
2. The port number of the 7.2 CA directory server at the *IESO*'s CA for LDAP communications.
3. The domain name (i.e. IP address) of the version 7.2 CA Manager (optional but required for automatic certificate updates).
4. The port number of the version 7.2 CA Manager for encrypted PKIX-CMP communications (optional but required for automatic certificate updates).

~~**Note:** The IESO PKI service vendor, Verizon is implementing a replacement CA system in early March 2010 to handle the transition from the existing CA system on which the CA issuing certificate is expiring May 13, 2010. IP address parameters for the CA Manager and Directory Server are changing for the replacement CA system as documented here and in the *Participant Technical Reference manual*. The current IP addresses can continue to be used until a new certificate has been created from the replacement CA for use with the MIM IDK. Once the new certificate has been created, IP address parameters must be updated to the new ones by the market participant.~~

### 2.1.3 API Design Concepts

The most important functionality that is provided by the IDK is the ability to upload and download bid data. The IDK transfers this data in template format (Refer to the specific bid template documentation for a description of the required formats). A template format consists of ASCII lines of data in CSV format. For example, a template formatted RTEM bid might look as follows:

**PM, RTEM, 20011231, PARTICIPANT\_ID, USER\_ID, , NORMAL;**

\*

**GENERATOR, RESOURCE\_ID, , , 50.0, SUBMIT, NORMAL;**

**1-24, NERCTAGID, {(20.00,0.0),(20.00,6.4),(30.00,9.5)},  
{(100.0,1.0,2.0),(120.0,2.0,1.0),(140.0,2.0,3.0)}, N, N, N;**

\*\*

Template data is transferred to and from the MIM with this format style. This is convenient for a number of reasons:

- The CSV format makes the generation and importation of these templates quite straightforward with, for example, spreadsheet applications.
- More than one bid may be uploaded or downloaded in one transaction. The MIM GUI supports both HTML and template format. The HTML format is not useful to the MIM IDK because only one bid may be transferred at a time and HTML tags surround the data.
- To establish communication with the MIM, one must create an IDK object and login to the MIM with it. All communications with the MIM require this logged-in object. It will oversee the normalization of MIM IDK method invocations into messages sent to and from the MIM system. PKI operations are performed as required along the way during transaction processing.
- Secure encrypted operations are provided via the Entrust PKI technologies combined with *IESO* server certificates for SSL protocol communications. The Entrust Java Toolkit based PKI software code is bundled within the MIM IDK. The MIM IDK invokes Entrust Java Toolkit software code when secure operations are necessary. . To increase security, private encryption and signing keys used within the PKI model are periodically scheduled for update at the rollover point of the keys.
- The certificate/key updates are implemented based on a schedule set by the *IESO's* CA policy. Typically the update schedule is once per year for encryption keys in the EPF file and once per year as well for the signing keys based on the date of initial creation or last update. This may be subject to change based on the CA policy. The triggering point for update is approximately 100 days before expiry (i.e. 70% of certificate lifetime). When done, this is called key-rollover. When the update is completed this means that the “old” keys become invalid and new ones are put in their place. This does limit the amount of time a “hacker” has to try to discover keys. If not updated the keys will expire and recovery of the certificates and keys will be required.
- In order to enable this rollover or key update functionality automatically, the 7.2 CA Manager must be specified during login time as shown in section 3.1.1. The Entrust Java Toolkit software code within the MIM IDK supports automatic key-rollover along with the *IESO's* CA systems. However, if it is desired to manually manage certificate updates via the use of the *IESO* CLS software (see *Market Manual 1: Market Entry, Maintenance & Exit Part 1.3: Identity Management Operations Guide*) then use the constructor shown in section 3.1.2. However this is not recommended as certificate key update is then dependent upon a market participant custodian remembering when to manually update the EPF file. If the automatic update constructor is used the Certificate Lifecycle System (CLS) is not required for update of certificates when they are used to login to API on a regular basis. The choice is of course up to the *Market Participant*. Contact *IESO* Market Services for assistance if required.

## 2.1.4 API Overview

The above requirements dictate the following IDK method suite:

- Create MIM IDK instance.
- Login MIM IDK instance.
- Upload bids- [and/or Daily Generator Data \(DGD\)](#).
- Download bids- [and/or Daily Generator Data \(DGD\)](#).
- Download Reports. (-This capability although existing in the API will no longer be usable as of release 20.0
- Download System Messages.
- Download Market Status.
- Logout MIM IDK instance.

The *MPIRequest* class within the IDK embodies the above functionality. All operations are performed using instances of it. The IDK Javadoc for all versions goes into detail about the appropriate usage of this class.

– End of Section –

## 3. IDK Usage

---

### 3.1 Creating an MPIRequest Object

~~Two~~ There are three *MPIRequest* constructors ~~exist. They~~; the first two are used for PKI certificate-based authentication, while the third is required for username/password based authentication. See Appendix A regarding the two batch files included in the IDK bundle that provide user interactive demonstration of the login methods and IDK functionality.

The two constructors for Entrust certificate-based authentication contain one major difference between them: one uses the CA directory server alone while the other uses the CA directory server and the CA Manager. If automatic key-rollover is desired, then the CA Manager must be specified. The CA directory server is mandatory and the *Market Participant* EPF certificate data is validated against the appropriate and current CRL (certificate revocation list) on it during login (see the login methods below). ~~The two forms of constructors follows:~~

The list of *MPIRequest* constructors follows:

#### 3.1.1 Certificate Authentication: Automatic Key-Rollover

- Use the following constructor when automatic key-rollover (key update management) is required or desired for digital certificates/ keys used with the IDK.

```
public MPIRequest (java.lang.String managerHost,
                  int managerPort,
                  java.lang.String dirHost,
                  int dirPort,
                  java.lang.String epf,
                  char password,
                  java.lang.String host,
                  int port,
                  int execMode) throws java.lang.Exception
```

**Parameters:** Note that the values must be matched appropriately for the issuing CA and for the Sandbox or Production MIM environments. Availability of the CA environments is subject to change during any planned or unplanned CA outages. These situations will be communicated appropriately by the *IESO*:

**MIM IDK 3.23**

- managerHost - CA Manager domain name in `String` format. Specify one of:
  - ~~"ccica1.idm.cybertrust.com" or IP Address of "195.217.198.65" (prior to using a new certificate from the replacement CA)~~
  - or;
  - ~~"ccica2.idm.cybertrust.com" or IP Address of "195.217.198.67" (upon using a new certificate from the replacement CA)~~
  - or;
  - "64.18.21.140" (under CA long term failover circumstances, to be advised by IESO) - Verizon (formerly Cybertrust) version 7.2 Production System CA Manager Domain for both *IESO* Production and Sandbox MIM - "when using 7.2 Production Certificates"

**Note:** It is not recommended to use the domain name instead of the IP address for the managerHost argument. This is due to technical restrictions when the IESO must go to PKI offline mode during an outage for the PKI systems at the Certification Authority.

- managerPort - CA Manager port number in 'int' format~~...~~
  - 829 - all CA instances and versions.
- dirHost - CA directory server domain name in `String` format. For the corresponding CA Manager system above, specify one of:
  - ~~"ccipdir.idm.cybertrust.com" or IP Address of "195.217.199.14" (prior to using a new certificate from the replacement CA)~~
  - or;
  - ~~"ccipdir2.idm.cybertrust.com" or IP Address of "195.217.199.26" (upon using a new certificate from the replacement CA)~~
  - or;
  - ~~"64.18.29.142" (under CA long term failover circumstances) - Verizon (formerly - Cybertrust) version 7.2 Production System Directory Server Domain for both *IESO* Production and Sandbox MIM- "when using 7.2 Production Certificates".~~

**Note:** It is not recommended to use the domain name instead of the IP address for the dirHost argument. This is due to technical restrictions when the IESO must go to PKI offline mode during an outage for the PKI systems at the Certification Authority."

- dirPort - CA directory server port number in 'int' format~~...~~
  - 389 - all CA instances and versions.
- epf - EPF file specification.
- password - EPF password - char array.
- host - MIM domain name. Specify one of:

- "mos.ieso.ca" for *IESO* Production MIM.
- "moswebh.ieso.ca" for *IESO* Sandbox MIM.
- port – MIM server communications port. in 'int' format..
  - i.e. 443 all MIM instances.
- execMode - Value specifying what environment the code is running in. Since the IDK always runs as an application, use the MPIRequest.APPLICATION literal. Specify one of the field constants labeled Execution mode type constant

**Throws:**

java.lang.Exception - Returned for any URL or Entrust object creation failures.

**3.1.2 Certificate Authentication: No Automatic Key-Rollover**

Use the following constructor when automatic key-rollover is not required or desired.

```
public MPIRequest( java.lang.String dirHost,
                  int dirPort,
                  java.lang.String epf,
                  char [ password,
                  java.lang.String host,
                  int port,
                  int execMode) throws java.lang.Exception
```

**Parameters:** Note that the values must be matched appropriately for the issuing CA and for the Sandbox or Production MIM environments. Availability of the CA environments is subject to change during any planned or unplanned CA outages. These situations will be communicated appropriately by the *IESO*:

**MIM IDK 3.23**

- dirHost - CA directory server domain name. Specify one of:
  - ~~"ccipdir.idm.cybertrust.com" or IP Address of "195.217.199.14" (prior to using a new certificate from the replacement CA)~~
  - ~~or;~~
  - ~~"ccipdir2.idm.cybertrust.com" or IP Address of "195.217.199.26" (upon using a new certificate from the replacement CA)~~
  - ~~or;~~
  - "64.18.29.142" (under CA long term failover circumstances) - Verizon (formerly – Cybertrust) Production version 7.2 System Directory Server Domain for both *IESO* Production and Sandbox MIM - “when using 7.2 Production Certificates”

**Note:** It is not recommended to use the domain name instead of the IP address for the dirHost argument. This is due to technical restrictions when the IESO must go to PKI offline mode during an outage for the PKI systems at the Certification Authority.

- dirPort - CA directory server port number.
  - 389 - all CA instances and versions.
- epf - EPF file specification.
- password - EPF password char array.
- host - MIM server domain name. Specify one of:
  - "mos.ieso.ca" for *IESO* Production
  - "moswebh.ieso.ca" for *IESO* Sandbox
- port - MIM server port.
  - i.e. 443 all MIM instances.
- execMode - Value specifying what environment the code is running in. Since the IDK always runs as an application, use the MPIRequest.APPLICATION literal. Specify one of the field constants labeled Execution mode type constant

**Throws:**

java.lang.Exception - Returned for any URL or Entrust object creation failures.

### **3.1.3 Username/Password Authentication**

- Use the following constructor when username/password authentication is used with the IDK.

```
public MPIRequest( java.lang.String host,
                  java.lang.String protocol,
                  int port,
                  java.lang.String username,
                  char[] password,
                  int execMode) throws java.io.IOException
```

**Parameters:** Note that the account values must match those issued for the Sandbox or Production MIM environments. These will be communicated appropriately by the IESO:

#### **MIM IDK 3.3**

- host - MIM domain name. Specify one of:

- "mos.ieso.ca" for IESO Production MIM.
- "moswebh.ieso.ca" for IESO Sandbox MIM.
- protocol – Communications protocol in 'String' format.
  - i.e. https all MIM instances.
- port – MIM server communications port in 'int' format.
  - i.e. 443 all MIM instances.
- username - username in 'String' format, i.e. UserID@marketParticipant.
  - i.e. this is the issued user account UserID followed by the @ symbol followed by the market participant shortname with no spaces.
- password - password - char array.
- execMode - Value specifying what environment the code is running in. Since the IDK always runs as an application, use the MPIRequest.APPLICATION literal. Specify one of the field constants labeled Execution mode type constant

**Throws:**

java.io.IOException - Returned for any URL object creation failures.

## 3.2 Login/Logout of an MPIRequest Object

### 3.2.1 MIM IDK 3.23

Successful login into the MIM system is mandatory to access MIM data. It is required before any other MIM communications may occur.-

#### **Entrust Login**

Login may be either in CA on-line or off-line mode as controlled by the IESO centrally. Several steps are taken during the normal on-line login process:

- The EPF use is validated with the provided password (locally on the workstation or server that the API is running on).
- The specified and valid, version 7.2 CA directory server (i.e. IESO PKI directory server) is checked for a valid certificate entry as specified in the EPF. Note, at this time, if a valid CA Manager is specified, key-rollover may occur, if triggered and necessary.
- The *Market Participant* certificate is checked to see that it has not been revoked (against the current and appropriate Certificate Revocation List (CRL) on the directory server – updated every 4 hours or as required by the CA).
- An Entrust instance is created with which to perform secure operations with.

- SSL security context is created to connect to the MIM with using encrypted communications.

As a part of MIM API login, if the on-line CA directory server access fails for any reason in the initial login attempt by the client (e.g. the specified and valid version 7.2 CA directory not available etc.), the following will be performed on the client:

- A preliminary off-line CA login attempt is performed next for the specified and valid version 7.2 CA system. This ensures correct EPF syntax, password validity and date validity of the enclosed certificates. Any failures for these issues at this level result in login failure and no further progress until the user corrects the problem with those issues. For example expired certificates cannot be used to login in either on-line or off-line mode. If off-line login is successful this fetches credentials used to create a valid SSL session context for communication with the MIM Web server.
- If the preliminary off-line CA login is successful for the specified and valid, version 7.2 CA system, a MIM Web server configuration file is then securely fetched within the SSL session. The file controlled by the *IESO* shall specify the CA off/on-line access status for the specified and valid, CA system. Note that this is the same file used by the *IESO* for specifying what certificate mode will be used for the GUI MPI. The *IESO* will modify this file as required, based on planned version 7.2 CA system outages and its monitoring of the CA directory servers availability to permit CA off-line certificate use and connection to the market systems by *market participants*.
- If the CA directory server IP address and port values specified by the *market participant* client application match those listed in the file downloaded from the *IESO* Sandbox or Production web server and; the relevant CA directory server status in the file is configured off-line by the *IESO*, the client login will be successful in off-line mode. Thus the *IESO* will centrally be able to set mode of certificate use without administration effort at the *market participants*.
- If the CA directory server IP address and port values specified by the *market participant* client application is not listed or rather does not match those in the file downloaded from the *IESO* Sandbox or Production web server or do indeed match those in the file, but the CA mode is still specified to be on-line because the *IESO* has not changed the mode yet or login is not permitted for other reasons, the client login will fail. *The market participant* cannot input an invalid CA domain name or IP address or port to spoof off-line mode.
- MIM API users are not allowed to specify on-line or off-line login mode, for the users it is always on-line by default. Only the *IESO* shall be able to control this mode for them. The *IESO* monitors and coordinates with the Certification Authority regarding the availability of the directory servers for both planned and unplanned directory server outages.

~~**Note:**The IESO will be updating the IP Addresses and port values in its server configuration file for the Verizon CA update in early March 2010 for transitioning to new certificates in advance of the existing CA May 13, 2010 certificate expiry. This will enable the IESO to accept certificates from either the existing or replacement CA systems from early to mid March up to May 13<sup>th</sup>, and enable Market Participants to update their API client IP address parameters for the CA Manager and Directory Servers.~~

### **Username/Password Login**

The following steps are taken during the normal login process:

- The username is validated with the provided password (against the IESO MIM directory server).

- SSL security context is created to connect to the MIM using encrypted communications.

### Login/Logout Method Signatures

The login method signatures are as follows:

```
public boolean login()
```

Login the MIM user.-

For Entrust login, this method contacts the specified and valid, Entrust CA directory server for verification of the certificates within the selected EPF file and verifies the `password` parameter specified in the corresponding constructor. If successful, access to PKI operations on behalf of the logged-in user is now possible. When the CA Manager is specified, digital certificate keys may be rolled-over if required through PKI management protocols with the specified and valid CA Manager system.-

For username/password login, this method records authentication information for subsequent Web server I/O.

#### **Returns:**

true if login is successful, and false if it is not.

```
public boolean login(boolean online)
```

Login the MIM user.-

For Entrust login, this method contacts the specified and valid Entrust CA directory server for verification of the certificates within the selected EPF file and associated `password` parameter specified in the corresponding constructor. If successful, access to PKI operations on behalf of the logged-in user is now possible. When -the CA Manager is specified, -digital certificate keys may be rolled-over if required through PKI management protocols with the specified and valid, CA Manager system.-

For username/password login, this method records authentication information for subsequent Web server I/O.

#### **Parameters:**

`online` - Determines login mode - true for online mode, false for offline mode. (Added for code consistency between MIM API and applet). The IDK as of version 2.4 permits offline login mode as controlled by the *IESO*. This method has not changed for version 3.1 of the IDK.

**Note:** There is no offline mode applicable for username/password login.

#### **Returns:**

true if login is successful, and false if it is not.

When no more I/O is required with the MIM, a logout call is made. It releases and masks the memory associated with secure operations. It is important to invoke this when MIM access is no longer necessary. The logout method signature follows:

```
public void logout()
```

~~Logout MIM/Entrust user. Invoke this method to release resources consumed by logged-in Entrust user and disable PKI operations~~

### 3.3 Uploading Bids

Bids are uploaded by specifying a file, which contains the bids to upload, and a file to hold the MIM response generated while processing those bids. Any number of bids and bid types may be specified in the file. Note that the same method is used to upload Daily Generator Data (DGD).

The upload-bid-file method signature follows:

```
public boolean sendUploadFileRequest(java.lang.String inputFile,  
                                     java.lang.String outputFile)
```

#### Parameters:

- inputFile - File containing bids to upload.
- outputFile - File containing results from MIM after processing inputFile.

#### Returns:

A Boolean value representing upload status, true for success, false for failure. A response of true means server I/O was successful, the outputFile must be inspected to determine whether bids were actually accepted into the MIM database or not.

### 3.4 Downloading Bids

Bids are downloaded in template format. Target bid data is controlled by query methods accessible from the *MPIRequest* instance. Methods exist which determine the type of bid data that is queried. Note that the same method is used to download Daily Generator Data (DGD).

Adhere to the following multi-step process to download bids:

1. Reset query information.
2. Set query data for bid type.
3. Set query data for next bid type, etc.
4. Request bids.
5. Process requested bids.

## 6. Reset query structure...

Resetting the query information erases any previously set bid query parameters.

Each bid type has its own query data setting method. Wildcards are possible using the *MPIRequest* literal *IMO\_DEFAULT*. Use this value when no criteria are necessary on a given field. For example, in the bilateral query method below, if one does not care what the participant generation bid value is, set the first parameter to *IMO\_DEFAULT*.

This does not apply to *Standing Flags*, *Day Types* or *Date* fields. They must be specified and be unique.

The following standing flags are specified as literals within the *MPIRequest* class and must be used when specifying standing flag query data. They specify whether the bids being queried for are standing or not, respectively.

- IMO\_STANDING
- IMO\_NO\_STANDING

The values are listed in the *MPIRequest* public final array name *StandingFlags*. It is referenced when validating standing flag values.

The following valid day types are specified as literals within the *MPIRequest* class and must be used when specifying day type query data. They specify which day of the week (or all days) standing schedule bid data that is being queried:

- IMO\_MON
- IMO\_TUE
- IMO\_WED
- IMO\_THU
- IMO\_FRI
- IMO\_SAT
- IMO\_SUN
- IMO\_ALL

The values are listed in the *MPIRequest* public final array named *DayTypes*. It is referenced when validating day type values.

MIM date values are required by various methods including query methods. The format of such strings must adhere to the following formats:

yyyymmdd or yyyymmddhh

Where yyyy is the integer year, mm is the one-relative integer month, dd is the one-relative integer day of the month and hh is the one-relative integer hour of the day.

Any combination of query setting methods may be utilized. For example, if only RTEM bids are required, use the RTEM query parameter setting method and then request bids. If both capacity

reserve and bilateral data are required, use the capacity reserve and bilateral data query parameter-setting methods and then request bids.

After the desired query data is set, the request is now ready to be sent to the MIM server for processing. To be clear, clearing and setting bid query data only alters the contents of client local data structures. Set query parameters must be sent to the MIM for processing.

Note that in the query methods described below, if standing query data is not being queried for (standing flag is *IMO\_NO\_STANDING*), the day type and expiry date parameters are ignored.

### 3.4.1 Resetting Bid Query Parameters

The query reset method signature follows:

```
public void resetBidQueryParams()
```

Invoke this whenever bid query parameters need to be cleared.

### 3.4.2 Setting Bilateral Bid Query Data

To set bilateral bid query information, use the following method:

```
public boolean  
setQueryBilateralParams(java.lang.String participantGen,  
                          java.lang.String participantLoad,  
                          java.lang.String resourceGen,  
                          java.lang.String standingFlag,  
                          java.lang.String dayType,  
                          java.lang.String expiryDate)
```

#### Parameters:

- participantGen - Generation Participant name.
- participantLoad - Load Participant name.
- resourceGen - Resource Generation.
- standingFlag - Standing Flag literal.
- dayType - Day type literal.
- expiryDate - Date value.

#### Returns:

Boolean value, true when arguments are valid, false when they are not.

### 3.4.3 Setting Capacity Reserve Bid Query Data

To set capacity reserve query information, use the following method:

```
public boolean
setQueryCapacityReserveParams(java.lang.String[] bidTypes,
                               java.lang.String resourceId,
                               java.lang.String tiePoint,
                               java.lang.String standingFlag,
                               java.lang.String dayType,
                               java.lang.String expiryDate)
```

#### Parameters:

- bidTypes - Bid types array. The following is the list of valid bid types for capacity reserve bids.
  - IMO\_CAP\_GEN
  - IMO\_CAP\_INJECTION

For convenience, they are listed in the *MPIRequest* array *CapacityReserveBidTypes*.

**Note:** To select all bid types, pass an array of size one with the value set to *IMO\_DEFAULT*.

- resource - Resource identifier.
- tiePoint - Tiepoint.
- standingFlag - Standing Flag literal.
- dayType - Day type literal.
- expiryDate - Date value.

#### Returns:

Boolean value, true when arguments are valid, false when they are not.

### 3.4.4 Setting DAEFM Bid Query Data

To set DAEFM query information, use the following method:

```
public boolean setQueryDAEFMPParams(java.lang.String bidType,
                                     java.lang.String standingFlag,
                                     java.lang.String dayType,
                                     java.lang.String expiryDate)
```

#### Parameters:

- `bidType` - Bid type literal. Specify one of the following *MPIRequest* literals:
  - `IMO_BID_TYPE`
  - `IMO_OFFER_TYPE`
  - `IMO_BID_OFFER_TYPE`

They are listed in the *BidTypes* array for convenience.

- `standingFlag` - Standing flag literal.
- `dayType` - Day type literal.
- `expiryDate` - Date value

**Returns:**

Boolean value, true when arguments are valid, false when they are not.

### 3.4.5 Setting Operating Reserve Bid Query Data

To set Operating Reserve query information, use the following method:

```
public boolean
setQueryOperatingReserveParams(java.lang.String[] bidTypes,
                               java.lang.String[] classTypes,

                               java.lang.String resourceId,

                               java.lang.String tiePoint,
                               java.lang.String standingFlag,

                               java.lang.String dayType,

                               java.lang.String expiryDate)
```

**Parameters:**

- `bidTypes` - Bid types array. The following is the list of valid bid types for operating reserve bids.
  - `IMO_OPER_DISP`
  - `IMO_OPER_GEN`
  - `IMO_OPER_INJECTION`
  - `IMO_OPER_OFFTAKE`

For convenience, they are listed in the *MPIRequest* array *OperatingReserveBidTypes*.

**Note:** To select all bid types, pass an array of size one with the value set to *IMO\_DEFAULT*.

- classTypes - Class types array. The following is the list of valid class types for operating reserve bids.
  - IMO\_OPER\_SPIN\_10MIN
  - IMO\_OPER\_NSPIIN\_10MIN
  - IMO\_OPER\_RESV\_30MIN

For convenience, they are listed in the *MPIRequest* array *OperatingReserveClassTypes*.

**Note:** To select all class types, pass an array of size one with the value set to *IMO\_DEFAULT*.

- resource - Resource identifier.
- tiePoint - Tiepoint value.
- standingFlag - Standing Flag literal.
- dayType - Day type literal.
- expiryDate - Date value.

**Returns:**

Boolean value, true when arguments are valid, false when they are not.

### 3.4.6 Setting RTEM Bid Query Data

To set RTEM query information, use the following method:

```
public boolean setQueryRTEParams(      java.lang.String[] bidTypes,
                                       java.lang.String resourceId,
                                       java.lang.String tiePoint,
                                       java.lang.String standingFlag,
                                       java.lang.String dayType,
                                       java.lang.String expiryDate)
```

**Parameters:**

- bidTypes - Bid types array. The following is the list of valid bid types for RTEM bids.
  - IMO\_RTEM\_LOAD
  - IMO\_RTEM\_GEN
  - IMO\_RTEM\_OFFTAKE
  - IMO\_RTEM\_INJECTION

For convenience, they are listed in the *MPIRequest* array *RTEMBidTypes*.

**Note:** To select all bid types, pass an array of size one with the value set to *IMO\_DEFAULT*.

- resource - Resource identifier.
- tiePoint - Tiepoint value.
- standingFlag - Standing Flag literal.
- dayType - Day type literal.
- expiryDate - Date value.

**Returns:**

Boolean value, true when arguments are valid, false when they are not.

### 3.4.7 Setting Schedule Bid Query Data

To set Schedule query information, use the following method:

```
public boolean setQueryScheduleParams(java.lang.String[] bidTypes,
                                     java.lang.String resourceId,
                                     java.lang.String standingFlag,
                                     java.lang.String dayType,
                                     java.lang.String expiryDate)
```

**Parameters:**

- bidTypes - Bid types array. The following is the list of valid bid types for Schedule bids.
  - IMO\_SCHED\_SELF\_STR
  - IMO\_SCHED\_INTGEN\_STR
  - IMO\_SCHED\_NONDLOAD\_STR

For convenience, they are listed in the *MPIRequest* array *ScheduleBidTypes*.

**Note:** To select all bid types, pass an array of size one with the value set to *IMO\_DEFAULT*.

- resource - Resource identifier.
- standingFlag - Standing Flag literal.
- dayType - Day type literal.
- expiryDate - Date value.

**Returns:**

Boolean value, true when arguments are valid, false when they are not.

### **3.4.8 Setting Daily Generator Data (DGD) Query Data**

To set DGD query information, use the following method:

```
public boolean setQueryDGDPParams(java.lang.String resourceId)
```

#### **Parameters:**

- resourceId - Resource identifier 'String'.

#### **Returns:**

Boolean value, true when arguments are valid, false when they are not.

### **3.4.83.4.9 Requesting Bids**

Use this method to send the selected bid query data to the MIM server for processing.

```
public boolean sendBidQueryRequest(java.lang.String mode,
                                   java.lang.String date,
                                   java.lang.String[] hours,
                                   java.lang.String outputFile)
```

Request bid data to be downloaded using bid query presets.

#### **Parameters:**

- **mode** - Mode literal. Bids may be downloaded in full (complete data format) or summary (header only) format. The following is the list of valid modes.
  - IMO\_FULL
  - IMO\_SUMMARY

For convenience, they are listed in the *MPIRequest* array *ModeTypes*.

- **date** - Date value.
- **hours** - Hours enable/disable array. Each entry in this array describes whether data for the corresponding hour entry is being requested or not. Note that the array must have a size of 24. The following is the list of valid hour specification values:
  - IMO\_HOUR\_SET
  - IMO\_HOUR\_CLEAR
- **outputFile** - Server bid processing response file String.

#### **Returns:**

A Boolean value representing upload status, true for success, false for failure. A response of true means server I/O was successful; the `outputFile` must be inspected to determine whether bid data was actually returned.

### 3.5 Downloading System Messages

System Messages targeted to the logged-in participant may be fetched with the following method:

```
public boolean  
sendDownloadMessagesRequest(java.lang.String outputFile)
```

**Parameters:**

- `outputFile` - Local file specification to store system messages information in.

**Returns:**

A Boolean value representing download status, true for success, false for failure. A response of true means server I/O was successful and the `outputFile` file has been populated with MIM server data.

### 3.6 Downloading Market Status Information

Physical and Financial Market Status information may be fetched with following method:

```
public boolean sendDownloadMarketStatusRequest(java.lang.String outputFile)
```

**Parameters:**

- `outputFile` - Local file specification to store market status information in.

**Returns:**

A Boolean value representing download status, true for success, false for failure. A response of true means server I/O was successful and the `outputFile` file has been populated with MIM server data.

### 3.7 Miscellaneous Methods

The following is a list of miscellaneous methods contained in the *MPIRequest* class. Since they do not fall into any one category, they are included here.

For debugging purposes, the following method is provided. It writes information describing the various stages of CA/MIM message creation and transfer to the standard output device. The debug method signature follows:

```
public void setDebug(boolean flag)
```

**Parameters:**

- **flag** - Set true to turn debugging on and false to turn it off.

Use the following method to fetch the name of the currently logged in participant.

```
public java.lang.String getUsername()
```

**Returns:**

The logged in participant user login name. If no value is set, null will be returned.

– End of Section –

## Appendix A: IDK Bundle Descriptions

This appendix describes how to use the IDK software bundle. For this example, assume the environment variable `JDK_HOME` points to the JDK installation and `IDK_HOME` points to the IDK software. The IDK bundle is contained within a self extracting archive file named to include the version number (for example, `sidk_v3.1.exe` or `-sidk_v3.2.exe`). The extraction creates a top-level directory called Java.

In the top-level directory are the `README.txt` file and any files related to the digital signing of the IDK zip file and its contents. It also contains an overview of the IDK, its usage plus revision history of the IDK. [Included are a couple of batch files for testing and demonstration of the IDK with user interactive prompts for login and functionality.](#)

- [\*IESOIDK33PKITest.bat\* – for user interactive PKI certificate login and use of the IDK](#)
- [\*IESOIDK33UIDTest.bat\* – for user interactive UserID/Password login and use of the IDK](#)

The Java directory contains the following directories:

- classes
- doc
- source

The classes directory contains two Java library files:

- *capsapi\_classes.zip*
- *MPIApplication.jar*

The *capsapi\_classes.zip* contains Netscape provided privilege manager software. It is used to ensure participant authorization has been given before secure operations occur. This is used within MIM Applet software. It is included here since the same set of core class libraries is utilized in both the MIM GUI and IDK.

The *MPIApplication.jar* contains the following software:

- *MPIRequest.class* - Class library which oversees communication with the MIM.
- *FormObject.class* - Class library that oversees the building of HTTP Post objects.
- *MultiPartFormData.class* - Class library that oversees the normalization of client data into RFC 1867 formatted data.
- *PostOutputStream.class* - Class library that oversees the creation of *FormObjects* objects into HTTP FORM POST objects.
- *EntrustRequest.class* - Class library which oversees PKI operations.
- Entrust security classes.

The doc directory contains the following:

- A version of this document.

- IDK API description in Javadoc format.

The source directory contains the following IDK test application file:

- *MPIApplicationTest.java*

This file may be perused, compiled and executed to gain further understanding through example of the IDK and its usage. It is strongly suggested that *market participants* not familiar with the MIM IDK, utilize the test application, bundled with the MIM IDK zip file, in conjunction with the Sandbox MIM system in order to gain a working understanding of its functionality and methods.

Note that the *capsapi\_classes.zip* and *MPIApplication.jar* files must be set in the CLASSPATH environment variable for the IDK to run properly.

For example, to compile and execute the test program (*MPIApplicationTest.java*) within a Windows environment perform the following:

```
set
CLASSPATH=%CLASSPATH%;%IDK_HOME%\java\classes\capsapi_classes.zip;%IDK_HOME%\java\classes\MPIApplication.jar
javac MPIApplicationTest.java
java MPIApplicationTest
```

– End of Document –