



---

---

## **Application Notes**

### **For the PEMS XML Message Specification**

**April 22, 2008**

**Version 1.1**

---

---

**Trichord, Inc.**

12888 James Monroe Highway

Leesburg, VA 20176

Telephone: (703) 737-0162

Facsimile: (703) 777-9440

e-mail: [caf@trichord-inc.com](mailto:caf@trichord-inc.com)

## Table of Contents

<b>Introduction.....</b>	<b>1</b>
<b>Establishing a TCP/IP Socket connection .....</b>	<b>1</b>
<b>Login .....</b>	<b>1</b>
<b>Requesting Detector Inventory.....</b>	<b>3</b>
<b>Requesting Video Inventory .....</b>	<b>4</b>
<b>Requesting Detector Data .....</b>	<b>5</b>
<b>Requesting Changing Device Information .....</b>	<b>7</b>
<b>Canceling Subscriptions .....</b>	<b>8</b>

### Revision History

Filename	Version	Date	Author	Comment
Application Notes.doc	1.0	2/20/08	Craig A. Franklin	Initial Draft
Application Notes.doc	1.1	4/22/08	Craig A. Franklin Scott P. Butler	Updates from Implementation

## Introduction

This document is a companion document to the PEMS XML Message Specification and describes the details of how to connect to the PEMS server and exchange data over an XML data feed. This document is intended for an applications programmer and describes the message sequence and general parameters that must be sent. This document assumes that a separate server (called TIMS) wants to retrieve sensor data and video data from the PEMS server.

Details on data types and structures are presented in the PEMS XML Message Specification.

## Establishing a TCP/IP Socket connection

To connect to the PEMS Server, establish a TCP/IP socket connection to port 6000 at the IP address of the PEMS Server. Port 6000 is the default port for PEMS XML communication. The port can be changed by the PEMS System Administrator on the System Administration web page of the PEMS Server.

## Sending and Receiving Messages

All messages sent to and received from the PEMS Server are sent in ASCII (UTF-8) format. All messages must be valid XML. The XML definitions can be found in the Trichord PEMS Message Specification. All messages contain a 4 byte header that is the length of the message. The 4 byte header is calculated according to the algorithm below.

```
Header Calculation:
int nLength; // the number of characters in the message
byte byte1 = (nLength / 16777216) * 16;
byte byte2 = nLength % 256;
byte byte3 = (nLength % 65536) / 256;
byte byte4 = (nLength / 65536) % 256;
```

Message Example:

```
string xmlMessage = "<xml>test</xml>";
int nLength = 15;
byte1 = 0;
byte2 = 15;
byte3 = 0;
byte4 = 0;
```

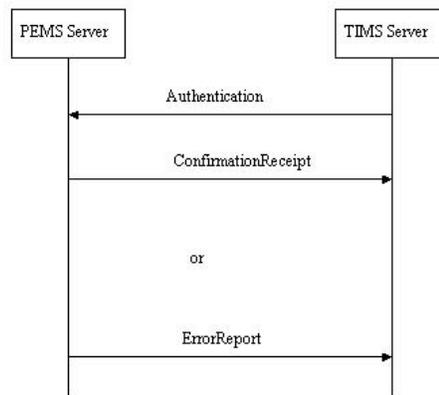
0	15	0	0	'<'	'x'	'm'	'l'	'>'	't'	'e'	's'	't'	'<'	'/'	'x'	'm'	'l'	'>'
---	----	---	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

---

## Login

The first message to send to the PEMS server is the Authentication message. This message confirms the authorization of the TIMS server for communicating with the PEMS server. The message is defined in the PEMS Message Specification document and is called LoginSet. The parameters of this message are a organization\_owning, organization\_requesting, User\_id and password. These parameters are set by the PEMS System Administrator.

Figure 1 presents a message sequence chart for authentication. TIMS sends the LoginSet message. PEMS responds with a ConfirmationReceipt message. If there is an error or problem with the LoginSet message, the PEMS server responds with an ErrorReport message.



**Figure 1: Message Sequence Chart for Authentication**

Message Parameters:

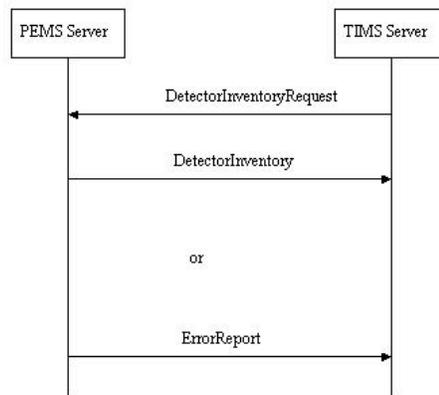
LoginSet (organization\_owning, organization\_requesting, User\_id, Password)  
ConfirmationReceipt(organization\_owning, organization\_requesting, errorcode, errortext)

Note: If for some reason the TCP/IP connection is broken, the TIMS server must establish a new connection and send a new LoginSet message.

## Requesting Detector Inventory

Once the connection is established and confirmed, the TIMS server should request an inventory of traffic sensor.

To request a traffic sensor inventory, the TIMS server should send a DetectorInventoryRequest message. The PEMS server shall respond with either a Detector Inventory message or an ErrorReport. Figure 2 illustrates the message sequence chart for the DetectorInventoryRequest message.



**Figure 2: Message Sequence Chart for DetectorInventoryRequest**

Message Parameters:

DetectorInventoryRequest( organization\_information, organization\_requesting,  
device\_type, device\_information\_type)

DetectorInventory(DetectorInventoryHeader, DetectgorInventoryDetails)

Note: In the first message, device\_type should equal (1) one to indicate that you want sensor information. A device\_type of (2) two indicates that you want video information.

The DetectorInventoryHeader is a structure with the following parameters:

Organization\_information, device\_id, device\_location

The DetectorInventoryDetails is an array with the following information:

DetectorInventoryHeader, detector\_type, detector\_lanes

Please note that the DetectorInventoryHeader is repeated within the DetectorInventoryDetails. Also note that detector\_lanes is an array of lane\_id.

Another way to display the parameters for DetectorInventory is as follows:

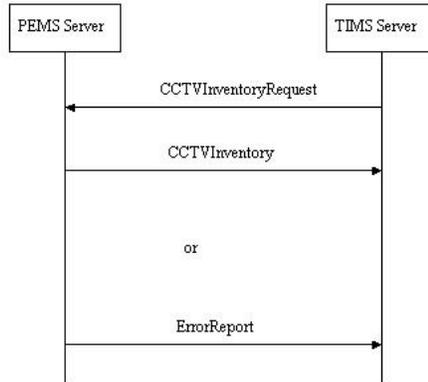
```
DetectorInventoryHeader
{
    organization_information,
    device_id,
    device_location
}
```

```
DetectorInventoryDetails [1 – n ]
{
    DeviceInventoryHeader,
    Detector_type,
    Detector_lanes [ 1 – n ]
    {
        lane_id
    }
}
```

## Requesting Video Inventory

Once the connection is established and confirmed, the TIMS server should request an inventory of video cameras currently deployed.

To request a video camera inventory, the TIMS server should send a CCTVInventoryRequest message. The PEMS server shall respond with either a CCTVInventory message or an ErrorReport. Figure 3 illustrates the message sequence chart for the CCTVInventoryRequest message.



**Figure 3: Message Sequence Chart for CCTVInventoryRequest**

Message Parameters:

CCTVInventoryRequest( organization\_information, organization\_requesting,  
device\_type, device\_information\_type)

CCTVInventory(CCTVInventoryDetails)

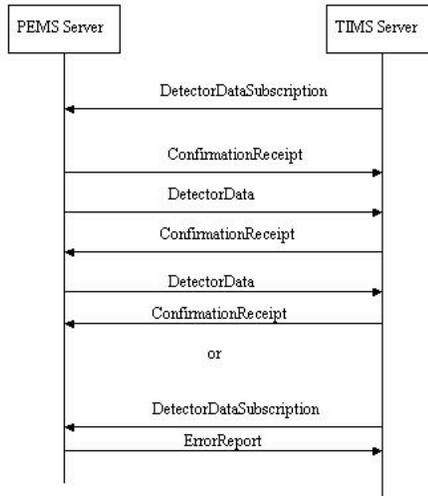
Note: In the first message, device\_type should equal (2) two to indicate that you want video information. A device\_type of (1) one indicates that you want sensor information.

CCTVInventoryDetails is an array of CCTVInventory which is a structure consisting of device\_inventory\_header, and cctv\_image. cctv\_image is an IP address for that specific camera. TIMS accesses the video image directly from the camera. Video images are not relayed through the PEMS server. Sensor data is relayed through the PEMS server.

## Requesting Detector Data

Detector data comes to the PEMS server every minute. In order to receive real-time detector data, the TIMS server should establish a subscription for the detector data. TIMS sends a DetectorDataSubscription message to the PEMS server. TIMS receives a ConfirmationReceipt message followed by Detector Data messages at one minute intervals.

Figure 4 illustrates the message sequence chart for the detector data subscription.



**Figure 4: Message Sequence Chart for DetectorDataSubscription**

Note: There are no parameters associated with the DetectorDataSubscription message. The PEMS Server knows which socket requested the data and responds over that socket with the ConfirmationReceipt message. Each DetectorData message is acknowledged by TMS.

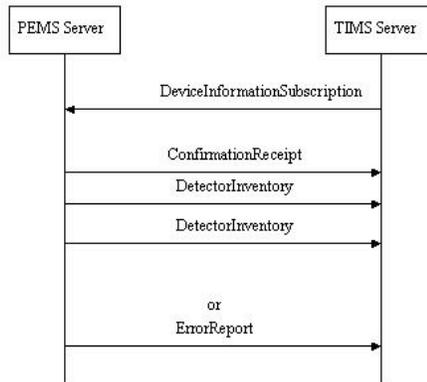
The DetectorData message has a parameter of DetectorDataDetails which is an array of the following structure:

```
DetectorDataDetails [ 1 - n ]
{
    organization_information,
    station_identifier,
    detector_id,
    vehicle_count,
    vehicle_occupancy,
    vehicle_speed
}
```

## Requesting Changing Device Information

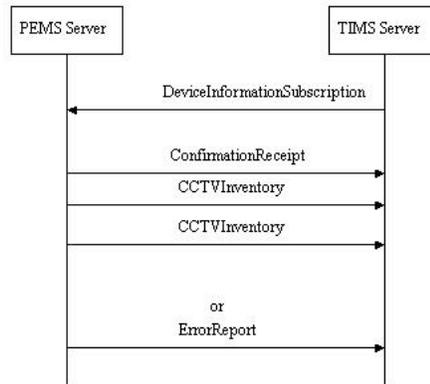
The PEMS server collects data from mobile monitoring platforms in the field. Over time, some platforms are taken down and other platforms are added. In order to know what devices have been added or deleted from the field deployments, the TIMS server should establish a Device Information subscription to the PEMS server. When a new monitoring platform is added, the PEMS server shall inform TIMS. There are two types of devices deployed in the field – traffic sensors and video cameras. TIMS should establish a device information subscription for each type of device being deployed.

Figure 5 presents the message sequence chart for the Device Information Subscription where the device\_type equals one (1) – thus requesting updated detector inventories.



**Figure 5: Message Sequence Chart for Device Information Subscription with device\_type = 1 for Traffic Sensor Inventory**

Figure 6 presents the message sequence chart for the Device Information Subscription where the device\_type equals (2) two – thus requesting updated CCTV inventories.

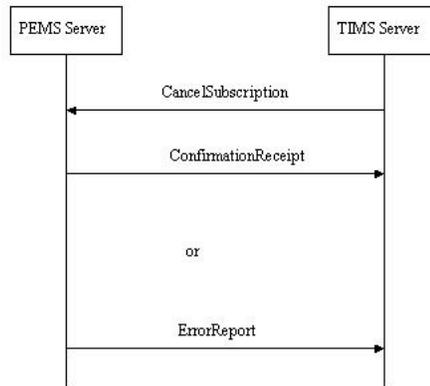


**Figure 6: Message Sequence Chart for Device Information Subscription with device\_type = 2 for CCTV Inventory**

The DeviceInformationSubscription message has the following parameters: Organization\_information, Organization\_requesting, Device\_type, Device\_information\_type.

## Canceling Subscriptions

To cancel all subscriptions, the TIMS server sends the CancelSubscription message to the PEMS server. This message cancels ALL subscriptions between the PEMS server and the TIMS server. Figure 7 presents the message sequence chart for the CancelSubscription message.



**Figure 7: Message Sequence Chart for CancelSubscription**

The CancelSubscription message has three parameters – organization\_information, center\_id, and center\_name.

Note: To “logout” the TMS server should cancel all subscriptions and disconnect the TCP/IP socket. There is no explicit logout message.