**Interface Specifications**

**Remote Service Agent**

**With**

**Device Manager**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Author** | **Date** | **Version** | **Description** |
| Robert Boyer | September 6, 2012 | 0.1 | **Initial revision** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[1. Introduction 4](#_Toc334687887)

[1.1 Purpose and Scope 4](#_Toc334687888)

[1.2 Background System Description 4](#_Toc334687889)

[1.3 Background Model 5](#_Toc334687890)

[1.4 Remote Service Agent 5](#_Toc334687891)

[2. Publish / Subscribe Communications 5](#_Toc334687892)

[2.1 Notification Types 5](#_Toc334687894)

[2.2 Notification Actions 5](#_Toc334687895)

[3. GUIDs 6](#_Toc334687896)

[3.1 Mailbox Classification 6](#_Toc334687898)

[3.2 Device Classification 6](#_Toc334687899)

[3.3 Message/Object Type Classification 6](#_Toc334687900)

[4. Pertinent Types 6](#_Toc334687901)

[5. Device URIs 7](#_Toc334687902)

[6. Object IDs 7](#_Toc334687903)

[7. Base Message Structure 7](#_Toc334687904)

[6.1 Message Header 7](#_Toc334687908)

[6.2 Message Body 7](#_Toc334687909)

[8. Session GUIDs 8](#_Toc334687910)

[9. Transaction GUIDs 8](#_Toc334687911)

[10. Specific Commands 8](#_Toc334687912)

[11. Message Part Constructs 9](#_Toc334687913)

[11.1 Params 9](#_Toc334687920)

[11.2 Components 9](#_Toc334687921)

[11.3 Logs ?? 9](#_Toc334687922)

[12. Connection and Communications Flows 10](#_Toc334687923)

[12.1 Basic Connectivity and Session Management 10](#_Toc334687925)

[12.2 Connectivity and Extended Session Management 11](#_Toc334687926)

[12.3 Getting Headers and Associated Notifications 12](#_Toc334687927)

[12.4 Posting Notifications 13](#_Toc334687928)

[13. Sample Messages and Responses 14](#_Toc334687929)

[13.1 Session Management 14](#_Toc334687931)

[13.1.1 Create Session 14](#_Toc334687932)

[13.1.2 Open Session 15](#_Toc334687934)

[13.1.3 Close Session 16](#_Toc334687937)

[13.2 Authentication 17](#_Toc334687938)

[13.2.1 Login 17](#_Toc334687939)

[13.2.2 Logoff 18](#_Toc334687940)

[13.3 Device Management 19](#_Toc334687941)

[13.3.1 Create Device – NOT CURRENTLY IMPLEMENTED 19](#_Toc334687942)

[13.3.2 Delete Device – NOT CURRENTLY IMPLEMENTED 20](#_Toc334687943)

[13.3.3 Get Device List 21](#_Toc334687944)

[13.3.4 Stat Device 21](#_Toc334687945)

[13.4 Notification Management 22](#_Toc334687946)

[13.4.1 Get (Notification) Headers 22](#_Toc334687947)

[13.4.2 Get Notification 24](#_Toc334687948)

[13.4.3 Post Notification 26](#_Toc334687949)

[13.4.4 Delete Notification 28](#_Toc334687950)

[13.4.5 Update Notification 29](#_Toc334687951)

[13.4.6 Expunge (Notification) 31](#_Toc334687952)

# Introduction

The Covidien Gateway Remote Service capability is intended to enable “remote” interaction to medical devices performing such service oriented tasks as getting log files from devices or loading new software onto the device. This document describes one aspect of the system’s communications – between the device manager and the remote service agent.

## Purpose and Scope

The purpose of this document is to describe the messaging interface between the Device Manager (e.g., Vent Test System (VTS)) and the Remote Service Agent (RSA).

This document describes the “as built” communications interface. The interface between the RSA and Device Manager clients is managed by this communications contract.

## Background System Description

The complete system is composed of:

* **Hardware device** (e.g., Ventilator, Pulse Oximeter, etc.)
* **Device manager** (e.g., Vent Test System)
* **Remote Service Agent** (RSA)
* **Service Management Server**
* **Data feeds** (e.g., manufacturing information, service contracts, etc.)

The hardware device is, of course, the object to which we intend to interact with, from which to get device revisions, details, or logs or onto which to download software.

The device manager is intended (ultimately) to support multiple devices, which means communicating in a device specific way to gather desired information or to perform software downloads onto the device. As much as possible, devices will conform to the specifications herein.

The remote service agent is a service, serving as a proxy to the service manager server. It isolates the device manager from the potential connectivity issues which exist with the server itself. E.g., a typical laptop will only have a single network card, but the current ventilator is not truly network enabled, and must be connected directly – essentially forcing the laptop to be disconnected from all other networks, hence being disconnected from the server. Due to this situation, the RSA must become connected to the server (connected mode), get populated with information that will be needed, become disconnected from the server (disconnected mode), so that the device manager may then be connected to the device and desired operations ensue.

The service manager server is populated with information and business rules to execute so that a device manager can be provided with the proper information to service each device to which it is connected, based on the logged-in user’s rights.

The data feeds are I.T. and Manufacturing feeds to the server providing knowledge about initial device configurations by device type and serial number, service contracts, software release information, etc. These provide the basis for activity at the device management level, but are not significant to the remainder of this document.

## Background Model

Communications are asynchronous, with messages being sent from one system to another without expectation of an immediate response. This is, in part, to account for the expectation that the RSA, may, from time to time, act only as a forwarding agent, passing (re-cast) messages on to the backend service manager server.

As such, a publish / subscribe (pub/sub) model has been chosen as the base communications style.

For the message store and forward, the iMAP communication mailbox approach is used by the RSA.

Communications are XML based, with all the rights and restrictions thereof. Where appropriate, binary data will likely be base64 encoded (and appropriately identified as such).

## Remote Service Agent

An objective for the Remote Service Agent was for it to be “invisible”, such that the Device Manager would act as if it was connected to the server itself, but without having to know that. To an extent, that objective is met. However, due to the very nature of the connectivity, the Device Manager does sometimes need to interact with the RSA knowing that it does exist.

# Publish / Subscribe Communications

The original intent of the communications system was to follow a basic publish / subscribe model where messages would contain, in their header content, a high-level description of what the message is intended to accomplish. Two key pieces of information are provided: **notification type** and **notification action**.



## Notification Types

There are three notification types:

* Event – a “one-time” occurrence being reported
* Command – a command / demand / instruction to a partnering component
* Response – the response to a prior command

## Notification Actions

There are six notification actions:

* Create
* Delete
* Modify
* Replace
* Get
* Set

These actions have their normal meanings.

# XML Messages

The basic message structure is composed of the standard XML header tag, followed by a message tag and the “**schema\_version**” attribute which it encodes.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e">

--- content ---

</message>

The message may be composed of XML elements of **head**, **body** or neither, which vary with need and content. Further detail is provided in the subsections below.



## Request Message

Most specific commands from the device manager to the RSA are identified as **requests**. The general structure is an xml element request followed by a transaction guid and, once known, a session guid. The following shows an example.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e">

<request type='createsession' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c' />

</message>

## Response Message

Requests made are expected to receive a **response**, usually indicating that the request was **ok** or **bad** and often receiving associated parameters / results. The general structure is an xml element response followed by a transaction guid to indicate to which request the response is associated. The following shows an example (the response to the createsession request above).

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<response type='param' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c' >

<params>

**<param name='session'>bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3</param>**

</params>

</response>

</message>

## Message Header

When used, this might read

<header>

<session>bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3</session>

<example>sample content</example>

</header>

## Message Body

When used, this might read

<body>

<is\_accessible>true</is\_accessible>

<last\_update>08/17/2012 12:00:00</last\_update>

<components>

-- component list here --

</components>

</body>

## Overview of Key Message Parts

When used, this xxx

### Message Parts

When used, this xxx

# GUIDs

To provide a level of flexibility and obscurity, GUIDs (global unique identifiers) are used to provide key tags used when identifying message types, destinations, etc. The guids defined below are considered “well known” (as they are documented here). Currently these are used for transactions and session ids as well as grouped to “define” specific object instances, such as mailbox classifications, device classifications, and message type classifications.



## Session GUID

To manage connections and authentication, a session guid will be created (as a result of a createsession message) which will always be passed with the messages thereafter. Once the session has been created or opened, the message element will contain the “**session\_guid**” attribute, as follows:

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3' >

--- content ---

</message>

Creation of the session guid is the responsibility of the RSA.

## Transaction GUID

Because the system follows the pub / sub asynchronous communication model, there is no inherent mechanism by which to associate a reply with the originating message. To solve this, a transaction guid, “**xaction\_guid**” attribute will be supplied with each request or response message (element). This should be, but is not required to be, unique. It might appear as follows:

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e">

<request type='createsession' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c' />

</message>

This allows the originator the ability to recognize which response goes with which request.

## Mailbox Classifications

These guids are used to represent different mailboxes (making it more difficult for someone snooping on the line to understand what is actually happening).

|  |  |
| --- | --- |
| Description (debug string) | GUID |
| DEVICE\_INBOX | 34caa93b-6ed3-4a93-9be7-4e31eb4b22d5 |
| DEVICE\_OUTBOX | 3bd775ad-7250-42ff-adf4-188e0b691bff |
| SYSTEM\_INBOX | b20af193-6353-4cf5-95cc-e79b84d88907 |
| SYSTEM\_OUTBOX | 5b5fd529-4244-46be-88cd-1a1304c179df |

The system outbox is included for completeness, but is not expected to be used.

## Device Classifications

Each device is associated with a guid.

|  |  |
| --- | --- |
| Description (debug string) | GUID |
| PB980\_VENTILATOR | 7a85f0c9-531e-4754-ad68-04c77ed63657 |
| PB840\_VENTILATOR | a7a65225-ef2d-48e9-89c8-6975dd7dc054 |

Other devices to be added as implemented.

## Message/Object Type Classifications

Each actionable message or message content descriptor has its own guid.

|  |  |
| --- | --- |
| Description (debug string) | GUID |
| DEVICE\_INFO | 532ffdda-6f38-41da-9a40-cb0801e6695f |
| DEVICE\_SOFTWARE\_UPGRADE\_PACKAGE / DEVICE\_SOFTWARE | 24637560-edb3-4961-b17d-14e74f74457c |
| DEVICE\_SOFTWARE\_DOWNLOAD\_ACK | 91dec715-f256-421a-8560-e6b015dae5f9 |
| DEVICE\_FIRMWARE\_UPGRADE\_PACKAGE / DEVICE\_FIRMWARE | a582e8a9-282a-4f49-93d1-f11547826a9 |
| DEVICE\_FIRMWARE\_DOWNLOAD\_ACK | 2c0d006f-2282-4417-bf20-2096b56a9ff6 |
| DEVICE\_HARDWARE | fc1f3c2c-16df-4b52-a9ea-99409b131d31 |
| DEVICE\_LIST | 06949c6d-5893-4579-b587-736373685683 |
| DEVICE\_STAT | 4bee9772-adcb-4ec2-b944-3191492b3436 |
| DEVICE\_RAW\_LOG | 2f62d564-a162-440a-a5f6-ed16e7e632d5 |
| DEVICE\_DECODED\_LOG | 8e74db99-f0a3-4de4-aeed-f17afb6896fc |
| RELEASE\_NOTES | 02d7aa7b-25e0-423f-8f98-c15612e6a7c0 |
| SERVICE\_MANUAL | c250c8db-b532-4d18-8956-420c3d637a41 |
| SYSTEMWIDE\_NOTIFICATION | 44972a33-8fc4-4742-a1fc-940f2cc1cc92 |

Other specific identifiers to be added as implemented.

# Pertinent Types

Pub / sub messages are conceptually at a high level and further detail is often necessary to provide context. The pertinent type is a composition of the device classification and the message classification, as follows (without the excess blank spaces):

device classification guid / message classification guid

For example: the device information for a PB980 would read:

7a85f0c9-531e-4754-ad68-04c77ed63657/532ffdda-6f38-41da-9a40-cb0801e6695f

And, within the XML structure of the message, would read:

<pertinent\_type>7a85f0c9-531e-4754-ad68-04c77ed63657/532ffdda-6f38-41da-9a40-cb0801e6695f</pertinent\_type>

As can be seen, the guids defined in the section above are used.

# Device URIs

The specific device is referenced by its device classification guid and relatively unique serial number. Because we cannot control the serial number, a choice was made to wrap each part of the data in curly braces (i.e., { and }). The construct is as follows (without the excess blank spaces):

{ device classification guid } / { device serial number }

For example: the device URI for a PB980 might read:

{7a85f0c9-531e-4754-ad68-04c77ed63657}/{980 abc1234567}

# Object IDs

Communication objects, notifications specifically, are managed by an “object id” (OID). This identifier is provided by the RSA to the Device Manager enabling the device manager the ability to uniquely reference any object (e.g., message notification) of which it has knowledge.

# Specific Commands

Despite the original intent to follow the publish / subscribe model and use the pertinent type for further define the content of a message, several specific commands were specifically identified as follows:

|  |  |
| --- | --- |
| Description | Purpose |
| Createsession | Create a new session |
| Opensession | Open an existing session (session id handed off from an authenticated process) |
| Closesession | Close a session |
| Login | Login a given user (with password) for authentication |
| Logout | Logout the user |
| GetDeviceList | Get the list of devices of which the current user (representing a current facility) is allowed to have knowledge. |
| DeviceStat | Get the device status / statistics of a specific device |
| GetHeaders | Get the message headers that are available for the given user |
| GetNotification | Get the notification associated with a given header |
| PostNotification |  |
| DeleteNotification |  |
| UpdateNotification |  |
| ExpungeNotification |  |

# Message Part Constructs

Certain messages require meaningful construct(s) that may themselves be composed when appropriate. These are defined below.



## Params

A “params” content provides a list of key-value-pairs, where the name attribute in the XML is the name of the param(eter) and is used as the key. The associate text of the XML is then used as the value.

<params>

<session>bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3</session>

<example>sample content</example>

</params>

The following types of messages return params

|  |  |  |
| --- | --- | --- |
| Description | Direction |  |
| Createsession | response |  |
| Opensession | response |  |

## Components

A “components” content provides a list of component details, where each component is defined as having a type and then a collection of internal elements further describing the component.

<component type=”HARDWARE”>

<name>INSPIRATORY</name>

<serial\_number>sample content</serial\_number>

<part\_number>sample content</part\_number>

<revision>sample content</revision>

<timestamp>time stamp</timestamp>

<status>installed</status>

<rights>

-- rights list here

</rights>

</component>

Naturally, the elements of a component may contain some, none, all, or more than exemplified above.

The following types of messages pass or return **params**.

|  |  |
| --- | --- |
| Description | Purpose |
| CreateSession | The resulting “session” or else a “reason” for failure |
| OpenSession | The resulting “session” or else a “reason” for failure |
| CloseSession | A “reason” for failure |
| Login | Request: “username” and “password” Response: a “reason” for failure |
| Logoff | A “reason” for failure |
| CreateDevice | Request: “device\_uri” Response: a “reason” for failure |
| DeleteDevice | Request: “device\_uri” Response: a “reason” for failure |
| StatDevice | Request: “device\_uri” Response: a “reason” for failure |
| GetHeaders | Request: “mailbox” and “device\_uri” Response: a “reason” for failure |
| GetNotification | A “reason” for failure |
| PostNotification | Request: “mailbox” |
| DeleteNotification | Request: “notification\_oid”  Response: a “reason” for failure |
| UpdateNotification | Request: “notification\_oid” (and the full notification?)  Response: a “reason” for failure |
| Expunge | Request: “notification\_oid” (or none if expunging all) Response: a “reason” for failure |
|  |  |

## Rights

A rights list may accompany a component, indicating that the user has, for example, the right to install software.

<rights>

<right>install</right>

<right>remove</right>

</rights>

The list of available rights is not well defined at this point, and ultimately is a contract between the backend service management server and a given device manager. However, here we need only detail the transport mechanism.

## Logs ??

Certain messages require a meaningful construct for their body. These are defined below.

# Connection and Communications Flows

The RSA is a service that will be running on the laptop, listening to socket port number 9999. The Device Manager will connect to the server at the designated port and initiate communications. The following communications flows indicate interaction between the Device Manager and the RSA.



## Basic Connectivity and Session Management

The basic session setup and user login is shown below



## Connectivity and Extended Session Management

The basic session setup and user login is shown below



## Getting Headers and Associated Notifications

Xxx



## Posting Notifications

Xxx



# Sample Messages and Responses

Following are a series of samples of messages and their responses.



## Session Management

### Create Session

A session\_guid id is generated by the laptop agent when the client requests a session. It is used by the laptop agent to distinguish between multiple application sessions and to assigns state to an application

**Call:**

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e">

<request type='createsession' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c' />

</message>

**Response:**

On success, a new session GUID is returned.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<response type='param' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c' >

<params>

**<param name='session'>bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3</param>**

</params>

</response>

</message>

On failure, an error response is returned with an indication of reason.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3' >

<response type='bad' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c'>

<params>

**<param name='reason'>Error:???????</param>**

</params>

</response>

</message>



### Open Session

**Call:**

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e">

<request type='opensession' xaction\_guid='9650e555-810d-487b-94bf-da8d2f0e334e'>

<params>

**<param name='session'>bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3</param>**

</params>

</request>

</message>

**Response:**

On success, a confirmation message indicating that a session has been opened.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3' >

<response type='ok' xaction\_guid='9650e555-810d-487b-94bf-da8d2f0e334e'/>

</message>

On failure, an error response is returned with an indication of reason.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3' >

<response type='bad' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c'>

<params>

**<param name='reason'>Error:???????</param>**

</params>

</response>

</message>



### Close Session

**Call:**

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<request type='closesession' xaction\_guid='9650e555-810d-487b-94bf-da8d2f0e334e'/>

</message>

**Response:**

On success, a confirmation message indicating that the session is closed.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e">

<response type='ok' xaction\_guid='9650e555-810d-487b-94bf-da8d2f0e334e'/>

</message>

On failure, an error response is returned with an indication of reason.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3' >

<response type='bad' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c'>

<params>

**<param name='reason'>Error:???????</param>**

</params>

</response>

</message>

## Authentication

### Login

The Login command is used to identify and authenticate a user, associating him/her to a session.

**Call:**

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<request type='login' xaction\_guid='e1c38f30-50a0-4ccf-a74a-c7ec11b54544'>

<params>

**<param name='username'>michael.ekaireb</param>**

**<param name='password'>P@ssw0rd1</param>**

</params>

</request>

</message>

**Response:**

On success, a confirmation message indicating that the user has successfully logged in and is in the authenticated state.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<response type='ok' xaction\_guid='e1c38f30-50a0-4ccf-a74a-c7ec11b54544'/>

</message>

On failure, an error response is returned with an indication of reason.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3' >

<response type='bad' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c'>

<params>

**<param name='reason'>Error:???????</param>**

</params>

</response>

</message>

### Logoff

The Logoff command informs the server that a user is no longer associated with a given session.

**Call:**

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<request type='logoff' xaction\_guid='23b2320b-6378-4013-80fb-a05ffe199e9c'/>

</message>

**Response:**

On success, a confirmation that logoff completed, user is no longer associated.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<response type='ok' xaction\_guid='23b2320b-6378-4013-80fb-a05ffe199e9c'/>

</message>

On failure, an error response is returned with an indication of reason.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3' >

<response type='bad' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c'>

<params>

**<param name='reason'>Error:???????</param>**

</params>

</response>

</message>

## Device Management

### Create Device – NOT CURRENTLY IMPLEMENTED

The Create Device command is used to create a device node, allowing the agent to know about the existence of a given device.

**Call:**

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<request type='createdevice' xaction\_guid='55c26878-90fb-4a4a-aa1f-3a689310771e'>

<params>

**<param name='device\_uri'>{7a85f0c9-531e-4754-ad68-04c77ed63657}/{840 34567453}</param>**

</params>

</request>

</message>

**Response:**

On success, a confirmation that the device was created.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<response type='ok' xaction\_guid='55c26878-90fb-4a4a-aa1f-3a689310771e'/>

</message>

On failure, an error response is returned with an indication of reason.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3' >

<response type='bad' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c'>

<params>

**<param name='reason'>Error:???????</param>**

</params>

</response>

</message>

### Delete Device – NOT CURRENTLY IMPLEMENTED

The Delete Device command is used to remove a device node, allowing the agent to forget about the existence of a given device.

**Call:**

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<request type='deletedevice' xaction\_guid='55c26878-90fb-4a4a-aa1f-3a689310771e'>

<params>

**<param name='device\_uri'>****{7a85f0c9-531e-4754-ad68-04c77ed63657}/{840 34567453}</param>**

</params>

</request>

</message>

**Response:**

On success, a confirmation that the device was deleted.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<response type='ok' xaction\_guid='55c26878-90fb-4a4a-aa1f-3a689310771e'/>

</message>

On failure, an error response is returned with an indication of reason.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3' >

<response type='bad' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c'>

<params>

**<param name='reason'>Error:???????</param>**

</params>

</response>

</message>

### Get Device List

### Stat Device

The Stat Device command is used to test the assertion that a device is known by the laptop agent.

**Call:**

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<request type='statdevice' xaction\_guid='55c26878-90fb-4a4a-aa1f-3a689310771e'>

<params>

<param name='device\_uri'>{7a85f0c9-531e-4754-ad68-04c77ed63657}/{840 34567453}</param>

</params>

</request>

</message>

**Response:**

On success, confirmation that a device is known to system.

<?xml version=“1.0” encoding=“utf-8”?>

<message schema\_version=“3644767c-2632-411a-9416-44f8a7dee08e” session\_guid='SESSION\_ID'>

<response type='statdevice' xaction\_guid='55c26878-90fb-4a4a-aa1f-3a689310771e'>

<params>

<device\_uri>{7a85f0c9-531e-4754-ad68-04c77ed63657}/{980 112358}</device\_uri>

</params>

<body>

<is\_accessible>true</is\_accessible>

<last\_update>08/17/2012 12:00:00</last\_update>

<components>

<component type=“software”>

<name>fedora</name>

<revision>15</revision>

<part\_number>097531</part\_number>

<rights>

<right>install</right>

<right>remove</right>

</rights>

</component>

<component type=“software”>

<name>bluetoothdvr</name>

<revision>2.4.5</revision>

<part\_number>097235</part\_number>

<rights>

<right>install</right>

<right>remove</right>

</rights>

</component>

</components>

</body>

</response>

</message>

On failure, an error response is returned with an indication of reason.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3' >

<response type='bad' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c'>

<params>

**<param name='reason'>Error:???????</param>**

</params>

</response>

</message>

## Notification Management

### Get (Notification) Headers

**Call:**

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<request type='getheaders'

xaction\_guid='55c26878-90fb-4a4a-aa1f-3a689310771e' >

<params>

<param name='mailbox'>34caa93b-6ed3-4a93-9be7-4e31eb4b22d5</param>

<param name='device\_uri'>{bc108d24-1cd7-403f-8dd4-e8e0ad816a20}/{840 34567453}</param>

</params>

</request>

</message>

**Response:**

On success, a list of notifications that is available for a given device\_uri.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<response type='headers'

xaction\_guid='55c26878-90fb-4a4a-aa1f-3a689310771e'>

<headers>

<notification oid='344aa359-f0ba-4f5a-b6d2-790bfd1ad95e'>

<header notification\_type='command' notification\_action='set'>

<pertinent\_type>7a85f0c9-531e-4754-ad68-04c77ed63657/86fd6b83-219a-4482-8813-191c477ccd7e</pertinent\_type>

<pertinent\_identifier>840 351545678</pertinent\_identifier>

<notification\_id>344aa359-f0ba-4f5a-b6d2-790bfd1ad95e</notification\_id>

</header>

</notification>

<notification oid='344aa359-f0ba-4f5a-b6d2-790bfd1ad95e'>

<header notification\_type='command' notification\_action='get'>

<pertinent\_type>7a85f0c9-531e-4754-ad68-04c77ed63657/ac2b174e-f6f7-45ed-997c-35c71f74ef5b</pertinent\_type>

<pertinent\_identifier>840 351545678</pertinent\_identifier>

<notification\_id>344aa359-f0ba-4f5a-b6d2-790bfd1ad95e</notification\_id>

</header>

</notification>

</headers>

</response>

</message>

On failure, an error response is returned with an indication of reason.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3' >

<response type='bad' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c'>

<params>

**<param name='reason'>Error:???????</param>**

</params>

</response>

</message>

### Get Notification

**Call:**

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<request type='getnotification' xaction\_guid='55c26878-90fb-4a4a-aa1f-3a689310771e' >

<params>

<param name='notification\_oid'>bc108d24-1cd7-403f-8dd4-e8e0ad816a20</param>

</params>

</request>

</message>

**Response:**

On success, a notification with both body and header. The body element is base64 encoded.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3' >

<response type='notifications'

xaction\_guid='55c26878-90fb-4a4a-aa1f-3a689310771e'>

<notifications>

<notification oid='344aa359-f0ba-4f5a-b6d2-790bfd1ad95e'>

<header notification\_type='command' notification\_action='set'>

<pertinent\_type>ddf96625-cb25-49c2-a59b-07da0f48d46c/86fd6b83-219a-4482-8813-191c477ccd7e</pertinent\_type>

<pertinent\_identifier>840 351545678</pertinent\_identifier>

<notification\_id>344aa359-f0ba-4f5a-b6d2-790bfd1ad95e</notification\_id>

</header>

<body />

</notification>

</notifications>

</response>

</message>

On failure, an error response is returned with an indication of reason.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3' >

<response type='bad' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c'>

<params>

**<param name='reason'>Error:???????</param>**

</params>

</response>

</message>

### Post Notification

**Call:**

The body element is base64 encoded.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3' >

<request type='postnotification'

xaction\_guid='55c26878-90fb-4a4a-aa1f-3a689310771e'>

<params>

**<param name='mailbox'>34caa93b-6ed3-4a93-9be7-4e31eb4b22d5</param>**

**<notification>**

**<header notification\_type='command' notification\_action='set'>**

**<pertinent\_type>7a85f0c9-531e-4754-ad68-04c77ed63657/86fd6b83-219a-4482-8813-191c477ccd7e</pertinent\_type>**

**<pertinent\_identifier>840 351545678</pertinent\_identifier>**

**<notification\_id>344aa359-f0ba-4f5a-b6d2-790bfd1ad95e</notification\_id>**

**</header>**

**<body />**

**</notification>**

</params>

</request>

</message>

**Response:**

On success, the OID of the notification is returned.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<response type='param' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c' >

<params>

**<param name='**notification\_oid**'>bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3</param>**

</params>

</response>

</message>

On failure, an error response is returned with an indication of reason.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3' >

<response type='bad' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c'>

<params>

**<param name='reason'>Error:???????</param>**

</params>

</response>

</message>

### Delete Notification

Mark a notification deleted.

**Call:**

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<request type='deletenotification' xaction\_guid='55c26878-90fb-4a4a-aa1f-3a689310771e' >

<params>

<param name='notification\_oid'>bc108d24-1cd7-403f-8dd4-e8e0ad816a20</param>

</params>

</request>

</message>

**Response:**

On success, a confirmation that the notification was deleted.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<response type='ok' xaction\_guid='55c26878-90fb-4a4a-aa1f-3a689310771e'/>

</message>

On failure, an error response is returned with an indication of reason.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3' >

<response type='bad' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c'>

<params>

**<param name='reason'>Error:???????</param>**

</params>

</response>

</message>

### Update Notification

**Call:**

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3' >

<request type='updatenotification'

xaction\_guid='55c26878-90fb-4a4a-aa1f-3a689310771e'>

<params>

<notification oid='344aa359-f0ba-4f5a-b6d2-790bfd1ad95e'>

<header notification\_type='command' notification\_action='set'>

<pertinent\_type>ddf96625-cb25-49c2-a59b-07da0f48d46c/86fd6b83-219a-4482-8813-191c477ccd7e</pertinent\_type>

<pertinent\_identifier>840 351545678</pertinent\_identifier>

<notification\_id>344aa359-f0ba-4f5a-b6d2-790bfd1ad95e</notification\_id>

</header>

<body />

</notification>

</params>

</request>

</message>

**Response:**

On success, confirmation that a notification has been updated.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<response type='ok' xaction\_guid='55c26878-90fb-4a4a-aa1f-3a689310771e'/>

</message>

On failure, an error response is returned with an indication of reason.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3' >

<response type='bad' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c'>

<params>

**<param name='reason'>Error:???????</param>**

</params>

</response>

</message>

### Expunge (Notification)

**Call:**

Remove a single notification marked for deletion.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<request type='expunge' xaction\_guid='55c26878-90fb-4a4a-aa1f-3a689310771e' >

<params>

<param name='notification\_oid'>bc108d24-1cd7-403f-8dd4-e8e0ad816a20</param>

</params>

</request>

</message>

Remove notifications marked deletion from system.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<request type='expunge' xaction\_guid='55c26878-90fb-4a4a-aa1f-3a689310771e' />

</message>

**Response:**

On success, a confirmation that the notification was undeleted is returned.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3'>

<response type='ok' xaction\_guid='55c26878-90fb-4a4a-aa1f-3a689310771e'/>

</message>

On failure, an error response is returned with an indication of reason.

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

<message schema\_version="3644767c-2632-411a-9416-44f8a7dee08e" session\_guid='bd86a4a2-a894-4be5-b0e4-8b9b24ba6ed3' >

<response type='bad' xaction\_guid='0f16a851-a043-498d-a857-77ee5c59499c'>

<params>

**<param name='reason'>Error:???????</param>**

</params>

</response>

</message>

# Sample Messages / Notifications

Following are a variety of samples using.

## Software Download (Notification)

<?xml version=“1.0” encoding=“UTF-8” standalone=“yes”?>

<message schema\_version=“3644767c-2632-411a-9416-44f8a7dee08e” session\_guid="123">

<response type=“notifications” xaction\_guid="456">

<notifications><notification oid=“1”>

<header notification\_action=“get” notification\_type=“response”>

<notification\_id>3644767c-2632-411a-9416-44f8a7d11235</notification\_id>

<pertinent\_identifier>980 112358</pertinent\_identifier>

<pertinent\_type>7a85f0c9-531e-4754-ad68-04c77ed63657/24637560-edb3-4961-b17d-14e74f74457c</pertinent\_type>

</header>

<body>

<components>

<component type=“software”>

<location>file:///C:/LaptopAgent/Downloads/Output.pre.gz</location>

<length>14976866</length>

<md5sum>29f896d8ff7c6eb3dcf1dd89d7b30a62</md5sum>

<name>PB980\_10090951X00</name>

<part\_number>10090951</part\_number>

<revision>X00</revision>

</component>

<component type=“document”>

<location>file:///C:/LaptopAgent/Downloads/4\_075609\_00\_rev g\_ops\_840\_useng.pdf</location>

<md5sum>29f896d8ff7c6eb3dcf1dd89d7b30a62</md5sum>

<part\_number>075609</part\_number>

<name>user manual</name>

<revision>00-g</revision>

</component>

</components>

</body>

</notification>

</notifications>

</response>

</message>

## Log File (Notification)

Xxx

xx

## Device Info (Notification)

Xxx

xx

# Sample Log File

Following are samples of raw and decoded log files.

## Raw Log File

The following is a sample raw log file

<LogFiles>

<LogFile name="estsstdiag" >

<head>

<meta http-equiv="Content-Type" content="application/pbVentLog-diag;ver=1.0" />

<meta http-equiv="Transfer-Encoding" content="base64" />

</head>

<body>

AAAABBEGEwEAAABRAAAAAAAAADEAAAAAAAAAAAAACZ0AAABwAAAAIAAAAQEAACCQAAAA5gAAAOYAAAEAAAAAAAAAAAAAAAAwAAAAMQAAAAAAAAAxAAAAAAwFFAglEQFbDAUUCggDA28AAAAAAAAGYwwFEAwSJgE7AAAAAAAAAKGOBg

AEAAAAAAAAAAAAAAHaDAUQDBMkAhcAAAAAAAAAfY4EAA4AAAAAAAAAAAAAAZoMBRAMFDcAmQAAAAAAAAERjgAAAQAAAAAAAAAAAAABsgwFEAwbGgMQAAAAAAAAAHWOBAAOAAAAAAAAAAAAAAGKDAUQDBs5ACkAAAAAAAAAqo4EAA4A

AAAAAAAAAAAAAfQMBRAMHB4BlAAAAAAAAAD8jgQADgAAAAAAAAAAAAACmAwFEAwdAgPGAAAAAAAAARWOBAAOAAAAAAAAAAAAAAHLDAUQDB0wAigAAAAAAAAApI4GAAQAAAAAAAAAAAAAAeAMBRAMIi8AawAAAAAAAADpjgQADgAAAA

AAAAAAAAACcgwFEAwjCgD8AAAAAAAAAVaOBAAOAAAAAAAAAAAAAAJNDAUQDCMfA64AAAAAAAABII4EAA4AAAAAAAAAAAAAAeEMBRAMJAADgQAAAAAAAADVjgYABAAAAAAAAAAAAAACQgwFEAwkOgL0AAAAAAAAAYGOBAAOAAAAAAAA

AAAAAAKjDAUQDCUWAMoAAAAAAAABMo4GAAQAAAAAAAAAAAAAAf0MBRAMJR0CuAAAAAAAAAEpjgAAAQAAAAAAAAAAAAAB4gwFEAwlMwFRAAAAAAAAANeOAQACAAAAAAAAAAAAAAI/DAUQDCYPA1UAAAAAAAAAuo4EAA4AAAAAAAAAAA

AAAhQMBRAMLA4DQQAAAAAAAACrjgQADgAAAAAAAAAAAAAB9gwFEAwsJgNlAAAAAAAAAOeOBAAOAAAAAAAAAAAAAAJuDAUQDCw6AgMAAAAAAAAAmI4EAA4AAAAAAAAAAAAAAdAMBRAMLRcBkwAAAAAAAAEFjgYABAAAAAAAAAAAAAAB

owwFEAwuAANpAAAAAAAAAMeOBAAOAAAAAAAAAAAAAAIuDAUQEDIeAqMAAAAAAAABJhQBAAAAAAAAAAAAAAAAAWIMBRAQODsB9gAAAAAAAAGbFAEAAAAAAAAAAAAAAAACTAwFEBEADgH7AAAAAAAAATwUAQAAAAAAAAAAAAAAAAGODA

UQEiYmAfAAAAAAAAABcBQBAAAAAAAAAAAAAAAAAfYMBRATIhIBRAAAAAAAAACtFAEAAAAAAAAAAAAAAAABbwwFEBQDFQCMAAAAAAAAANkUAQAAAAAAAAAAAAAAAAHHDAUTCSMkATIAAAAAAAAAp3wBAAMAAAAAAAAAAAAAAc4MBRMJ

IyUDMgAAAAAAAACqfAEAAwAAAAAAAAAAAAAB1AwFEwkjJwM2AAAAAAAAALB+AQACAAAAAAAAAAAAAAHhDAUTDQ8jAq8AAAAAAAABFBQBAAAAAAAAAAAAAAAAAT4MBRMNDyMCsAAAAAAAAAEVFAEAAAAAAAAAAAAAAAABQAwFFAgmHw

BJAAAAAAAAALsUAQAAAAAAAAAAAAAAAAGLDAUUCCYfAEoAAAAAAAAAvBQBAAAAAAAAAAAAAAAAAY0MBRQIOiwCyQAAAAAAAAFeFAEAAAAAAAAAAAAAAAAB0gwFFAkDNAH2AAAAAAAAAVyCHQEZAAAAAAAAAAAAAAJyDAUUCQgPAecA

AAAAAAABLRQBAAAAAAAAAAAAAAAAAXAMBRQJCSgDpwAAAAAAAAEJih0AAQAAAAAAAAAAAAABuwwFFAkJKAPDAAAAAAAAASWKHQABAAAAAAAAAAAAAAHzDAUUCQsLAs8AAAAAAAABFXoIAAEAAAAAAAAAAAAAAa4MBRQJCwsC2gAAAA

AAAAEgeggAAQAAAAAAAAAAAAABxAwFFAkNLwICAAAAAAAAAG5+AAAEAAAAAAAAAAAAAAFeDAUUCREeAqsAAAAAAAABCn4AAAQAAAAAAAAAAAAAAZcMBRQJEzoDTAAAAAAAAADKfgAABAAAAAAAAAAAAAACFgwFFAk1GAAcAAAAAAAA

AJd8AQADAAAAAAAAAAAAAAGuDAUUCTUZAB4AAAAAAAAAmnwBAAMAAAAAAAAAAAAAAbQMBRQJNRoCWwAAAAAAAADafAEAAwAAAAAAAAAAAAACNAwFFAk1HABNAAAAAAAAAMx8AQADAAAAAAAAAAAAAAIY

</body>

</LogFile>

</LogFiles>

## Decoded Log File

The following is a sample decoded log file

<LogFiles><LogFile name='sysdiag'><head><meta http-equiv='content-type' content='application/pbVentLog-diag-json' /><renderHeaderHead><thead><tr><th>Meta Data Item</th><th>Value</th></tr></thead></renderHeaderHead><renderHeaderBody><headRows><row><tr><td>File Id</td><td>%1%</td></tr></row><row><tr><td>File Revision</td><td>%2%</td></tr></row><row><tr><td>Status</td><td>%3%</td></tr></row><row><tr><td>File Created</td><td>%23%-%24%-%25% %26%:%27%:%28%</td></tr></row><row><tr><td>Last Updated</td><td>%30%-%31%-%32% %33%:%34%:%35%</td></tr></row></headRows></renderHeaderBody><renderBodyHead><thead><tr><th>#</th><th>Date</th><th>Time</th><th>Patient Id</th><th>Diag Code</th></tr></thead></renderBodyHead><renderBodyBody><bodyRows><row><tr><td align="right">%0%</td><td>%2%-%3%-%1%</td><td align="right">%4%:%5%:%6%.%7%</td><td>%8%</td><td>%10%</td></tr></row></bodyRows></renderBodyBody></head><body><head>[ 0,2,285610753,81,0,52,0,0,1331,112,32,257,8336,230,230,256,0,0,51,52,0,52,0,'12','05','20','08','37','17','225','12','05','20','10','08','03','868',0,1657 ]</head><body>[ [ 0,'12','05','19','09','40','59','559',0,193,[ 'ZB0000',4,2,0,71,256,0,0 ],530 ],

[ 1,'12','05','19','09','40','59','559',0,193,[ 'ZB0000',4,2,0,71,256,0,0 ],530 ],

[ 2,'12','05','19','11','30','06','151',0,234,[ 'LB0000',4,1,6,52,256,0,0 ],592 ],

[ 3,'12','05','19','11','32','23','463',0,310,[ 'LB0000',4,1,0,54,256,0,0 ],488 ],

[ 4,'12','05','19','11','34','01','292',0,119,[ 'LB0000',4,1,6,52,256,0,0 ],362 ],

[ 5,'12','05','19','11','36','17','609',0,199,[ 'LB0000',4,1,0,54,256,0,0 ],521 ],

[ 6,'12','05','19','11','42','21','134',0,244,[ 'LB0000',4,1,6,52,256,0,0 ],612 ],

[ 7,'12','05','19','11','44','25','722',0,328,[ 'LB0000',4,1,0,54,256,0,0 ],524 ],

[ 8,'12','05','19','11','46','30','503',0,371,[ 'LB0000',4,1,6,52,256,0,0 ],611 ],

[ 9,'12','05','19','11','53','37','910',0,282,[ 'LB0000',4,1,0,54,256,0,0 ],432 ],

[ 10,'12','05','19','12','09','31','503',0,336,[ 'LB0000',4,1,6,52,256,0,0 ],541 ],

[ 11,'12','05','19','12','13','22','439',0,267,[ 'LB0000',4,1,0,54,256,0,0 ],402 ],

[ 12,'12','05','19','12','14','54','711',0,317,[ 'LB0000',4,1,6,52,256,0,0 ],503 ],

[ 13,'12','05','19','12','22','35','641',0,236,[ 'LB0000',4,1,0,54,256,0,0 ],595 ],

[ 14,'12','05','19','12','23','01','027',0,99,[ 'LB0000',4,1,6,52,256,0,0 ],322 ],

[ 15,'12','05','19','12','23','32','365',0,213,[ 'LB0000',4,1,0,54,256,0,0 ],549 ],

[ 16,'12','05','19','12','26','25','912',0,246,[ 'LB0000',4,1,6,52,256,0,0 ],616 ],

[ 17,'12','05','19','12','29','07','274',0,103,[ 'LB0000',4,1,0,54,256,0,0 ],329 ],

[ 18,'12','05','19','12','29','46','732',0,345,[ 'LB0000',4,1,6,52,256,0,0 ],559 ],

<snip>

[ 42,'12','05','20','09','05','52','160',0,263,[ 'LB0000',4,1,6,52,256,0,0 ],395 ],

[ 43,'12','05','20','09','08','15','486',0,300,[ 'LB0000',4,1,0,54,256,0,0 ],468 ],

[ 44,'12','05','20','09','08','15','487',0,301,[ 'E00000',1,1,1,0,0,0 ],368 ],

[ 45,'12','05','20','09','54','07','740',0,337,[ 'LB0000',4,1,6,52,256,0,0 ],543 ],

[ 46,'12','05','20','09','55','17','286',0,149,[ 'LB0000',4,1,0,54,256,0,0 ],421 ],

[ 47,'12','05','20','09','55','18','173',0,292,[ 'ZB0000',4,2,6,53,256,0,0 ],458 ],

[ 48,'12','05','20','09','55','18','217',0,336,[ 'ZB0000',4,2,0,55,256,0,0 ],545 ],

[ 49,'12','05','20','09','58','41','226',0,371,[ 'LB0000',4,1,6,52,256,0,0 ],611 ],

[ 50,'12','05','20','10','01','14','568',0,120,[ 'LB0000',4,1,0,54,256,0,0 ],363 ],

[ 51,'12','05','20','10','03','37','650',0,227,[ 'LB0000',4,1,6,79,256,0,0 ],605 ] ]</body></LogFile></LogFiles>

Xx