OpenClovis
Software Development Kit (SDK)
Service Description and API Reference for Chassis Management Service

For OpenClovis SDK Release 2.3 V0.4

Document Revision Date: March 13, 2007

DRAFT

Copyright © 2007 OpenClovis Inc.

All rights reserved
This document contains proprietary and confidential information of OpenClovis Inc., and may not be used, modified, copied, reproduced, disclosed or distributed in whole or in part except as authorized by OpenClovis Inc. This document is intended for informational use and planning purposes only. All planned features, specifications, and content are subject to change without notice.

Third-Party Trademarks
Sun, Sun Microsystems, and Java are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. UNIX is a registered trademark of The Open Group. Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries. CLEI is a trademark of Telcordia Technologies, Inc. Adobe, Acrobat, and Acrobat Reader are registered trademarks of Adobe Systems, Inc. All other trademarks, service marks, product names, or brand names mentioned in this document are the property of their respective owners.

Government Use
Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR 12.212 (Commercial Computer Software-Restricted Rights) and DFAR 227.7202 (Rights in Technical Data and Computer Software), as applicable.

Note: This document is not subject of the GPL license, even if you have obtained this document as a part of the GPL-ed version of OpenClovis SDK.

Contents

1 Functional Overview 1

2 Service Model 3

3 Service APIs 5

3.1 Type Definitions 5

3.1.1 ClCmFruOperationT 5

3.2 Functional APIs 6

3.2.1 clCmVersionVerify 6

3.2.2 clCmBladeOperationRequest 7

4 Service Management Information Model 9

5 Service Notifications 11

6 Configuration 13

7 Debug CLI 15

Chapter 1

Functional Overview

The OpenClovis Chassis Manager (CM) communicates with the chassis to obtain and control hardware platform status information. The CM uses Hardware Platform Interface (HPI) to:

- Interact with the hardware platform and thereby, monitor its status.
- Control the entities contained in it and Field Replaceable Units (FRUs), such as blades, AMCs, PMCs, and so on.

Chassis Manager is designed to use HPI B.01.01 specification's implementation of OpenHPI (Open source of HPI implementation), and `libhcl` (Radisys HPI Client Library), and provide the platform management services to ASP. CM depends on the Shelf Manager to monitor and control operations on the chassis. The Shelf Manager is responsible for monitoring and control to ensure correct operation of boards and other Shelf components. CM resides on the Shelf Manager to provide this system management functionality. The Chassis Manager controls the activation and de-activation of the blades based on Shelf Manager's implementation during hotswap state transition, when a blade is inserted and removed.

Chassis Manager detects arrival and departure of cards through HPI/IPMI events and other platform hardware related events, and reports it to the appropriate ASP components.

It also takes management requests from various sources such as Availability Management. For instance, CM supports hot-swapping, where the entities running on a node can shutdown before the node is physically removed.

Chapter 2

Service Model

TBD

Chapter 3

Service APIs

3.1 Type Definitions

3.1.1 ClCmFruOperationT

```
typedef enum {
            CL_CM_POWERON_REQUEST,
            CL_CM_POWEROFF_REQUEST,
            CL_CM_POWER_CYCLE_REQUEST,
            CL_CM_INSERT_REQUEST,
            CL_CM_EXTRACT_REQUEST,
            CL_CM_RESET_REQUEST,
            CL_CM_WARM_RESET_REQUEST,
            CL_CM_RESET_ASSERT_REQUEST,
            CL_CM_RESET_DEASSERT_REQUEST
} ClCmFruOperationT;
```

The enumeration, `ClCmFruOperationT`, contains the various chassis manager operations.

3.2 Functional APIs

3.2.1 clCmVersionVerify

clCmVersionVerify

Synopsis:

Verifies the version of the library supported by the CM.

Header File:

clCmApi.h

Syntax:

```
ClRcT clCmVersionVerify(
                  ClVersionT *version  );
```

Parameters:

version (in/out): As an input parameter, it accepts the version being verified. As an output parameter, it returns the version that is supported.

Return values:

CL_OK: The API executed successfully.

CL_ERR_VERSION_MISMATCH: The version provided is not supported.

Description:

This function is used to verify if the version of the client is supported. If the version is not supported, an error is returned with the value of the supported version populated in the `out` parameter. Typically, this function must be invoked at the initialization of the application.

Library File:

ClCm

Related Function(s):

None.

3.2.2 clCmBladeOperationRequest

clCmBladeOperationRequest

Synopsis:

Performs an operation on the blade or any platform hardware.

Header File:

clCmApi.h

Syntax:

```
ClRcT clCmBladeOperationRequest(
            CL_IN  ClUint32T ChassisId,
            CL_IN  ClUint32T SlotId,
            CL_IN ClCmFruOperationT request);
```

Parameters:

ChassisId: (in) `ChassisId` of the blade.

SlotId: (in) `SlotId` of the blade.

request: (in) Operation requested on the hardware.

Return values:

CL_OK: The API executed successfully.

CL_ERR_INVALID_PARAMETER: An invalid parameter has been passed to this function. A parameter is not set correctly.

CL_ERR_VERSION_MISMATCH: The version of the client and server incompatible.

CL_ERR_CM_HPI_ERROR: An HPI error has occurred at the server.

CL_ERR_TRY_AGAIN: CM is busy and cannot service the request immediately.

Description:

This API is used to operate on a BLADE or platform hardware. It accepts three arguments: `ChassisId`, `SlotId` of the FRU to operate, and the requested operation. The various operations are listed in the enumeration, `ClCmFruOperationT`.

Warning:

Setting-up the hotswap state of the BLADE must conform to the HPI hotswap state transition sequence. Only valid state transitions are allowed and any invalid state transition requests are rejected as `invalid` requests.

Note:

Refer to `DBG_PRINTS` on the Chassis Manager console or Log file for HPI errors.

Library File:

ClCm

Related Function(s):

None.

Chapter 4

Service Management Information Model

TBD

Chapter 5

Service Notifications

TBD

Chapter 6

Configuration

TBD

Chapter 7

Debug CLI

TBD

Index

clCmBladeOperationRequest, 7
ClCmFruOperationT, 5
clCmVersionVerify, 6