OpenClovis
Software Development Kit (SDK)
Overview for Service Description and API Reference Documents

For OpenClovis SDK Release2.3 V0.5

Document Revision Date: January 03, 2007

Contents

Chapter 1

Introduction

The Service Description and API Reference documents define the functions, Model, Management Information Model, Notifications, and Configuration of the OpenClovis Application Service Platform (ASP) services. This document is written for system integrators, designers, and system architects. It is also intended for use by application developers who would use the OpenClovis ASP Services to develop applications that must be highly available and manageable. It helps the developer understand the capabilities of the OpenClovis ASP services and the APIs that are provided by each service.

You should be familiar with the OpenClovis SDK User's Guide and a good knowledge of C programming language is required as the APIs are defined using C language.

1.1 Service Availability Forum (SA Forum)

The Service Availability Forum (SA Forum) is a consortium of industry-leading communications and computing companies that work together to develop and publish high availability and management software interface specifications.

A standardization of the interface between middleware and heterogeneous resources that need to be managed can be established by writing software that conforms to SA Forum interface specifications.

The following are the OpenClovis ASP components that are conformant with SA Forum B.01.01 specifications.

- Availability Management Framework (AMF)
- Checkpoint Service (CKPT)
- Event Manager (EM)
- Group Membership Service (GMS) component is conformant with the SA Forum CLM version B.01.01.

1.2 Using OpenClovis Libraries

The following are basics for using OpenClovis ASP Libraries described in this document:

1.2.1 Include File and Library Name

To use the APIs provided by an ASP service, the statement containing declarations of data types and function prototypes must be included.
For example, if your application is using a Checkpoint service API you must include the following:

```
#include <clCkptApi.h>
```

To use a particular Service API, an application must link with the corresponding library.
For example, to use Log service API, your application must link with the following library:

```
libClLogClient.a or .so
```

1.2.2 Library Life Cycle

Before you use an OpenClovis ASP library it is required to initialize the library. You can initialize the library using the initialize function.
For example, the log service provides the following function to initialize the log service library.

```
clLogInitialize()
```

After initializing the library you can perform the required operations. When you no longer need to use a library it is required to finalize the library. You can finalize the library using the finalize function.
For example, the log service provides the following function to finalize the log service library.

```
clLogFinalize()
```

The initialize and finalize functions are provided by all OpenClovis ASP libraries.

1.2.3 Type Definitions

The type definitions specific to a library are described within the Type Definitions section. The type definitions that are common to all libraries are described in the Common Type Definitions chapter which is a part of this document.

1.3 Header Files Naming Convention

This section provides the ASP services acronym list. You can identify the header files and the library files you need using this list. The naming convention used for the OpenClovis header files is

```
cl<Service Acronym>Api.h
```

Acronym	*ASP Service*
Alarm	Alarm Service
Ams	Availability Management Service
Buffer	Buffer Management Service
Ckpt	Checkpointing Service
Cksm	Checksum Management Service
Clist	Circular List Management Service
Cm	Chassis Management Service
Cnt	Container Management Service
Cor	Clovis Object Registry (COR) Service
Cpm	Component Management Service
Dbal	Database Abstraction Layer (DBAL) Service
Debug	Debug Service
Eo	Execution Object (EO) Service
Event	Event Service
Fault	Fault Service
Gms	Group Membership Service
Heap	Heap Management Service
Idl	Interface Definition Language (IDL) Service
Ioc	Intelligent Object Communication (IOC) Service
Log	Log Service
Med	Mediation Management Service
Name	Name Service
Osal	Operating System Abstraction Layer (OSAL) Service
Queue	Queue Management Service
Rmd	Remote Method Dispatch (RMD) Service
Rule	Rule Based Engine (RBE) Service
Timer	Timer Service
Txn	Transaction Management Service

1.4 Document Organization

Most of the Service Description and API Reference documents are organized into the following chapters:

Chapter 1, Functional Overview: Provides a brief description of the functionality of the service.

Chapter 2, Service APIs: Describes all the APIs within a service and the type definitions specific to the APIs.

1.5 Related Documentation

Service Description and API Reference document: is written for the knowledgeable application programmer who understands the basic architecture and working of the OpenClovis Application Service Platform. To achieve this, you can refer the following documents:

OpenClovis SDK User's Guide: OpenClovis SDK User's Guide provides information about OpenClovis ASP architecture, various OpenClovis ASP components, and their interactions. This guide helps you to configure the OpenClovis ASP components, compile, and execute the OpenClovis ASP code to build your custom application.

OpenClovis IDE Reference Guide: OpenClovis IDE Reference Guide describes the usage of IDE, a graphical development environment that compliment the ASP platform. This guide helps you to configure, compile, and execute IDE to build the application.

1.6 Documentation Conventions

The Service Description and API Reference documents use different fonts and symbols to differentiate between document elements and types of information. Following are the convention details:

Notation	Description
`code`	This font denotes the C code provided in various usage scenarios.
Cross reference	This font denotes a hyperlink. You can click on the hyperlink text to access the reference location, which can be either a section within the API Guide or a URL link. A cross reference refers to a section name accesses the first page of that section.
Italic text	Functions and parameters are indicated in italics.

Chapter 2

Common Type Definitions

2.1 Defines

Define	*Value*	*Description*
CL_BIT	(X) (0x1 << (X))	Creates a bit mask where x bit is set.
CL_BITS_PER_BYTE	8	Specifies number of bits per byte.
CL_CID_OFFSET	16	16 bit positions.
CL_ERROR_CODE_MASK	0xffff	16 bits.
CL_FALSE	0	false value for a boolean.
CL_FORCED_TO_16BITS	0xffff	Mask to force a number to be 16 bits.
CL_FORCED_TO_32BITS	0xffffffff	Mask to force a number to be 32 bits.
CL_FORCED_TO_8BITS	0xff	Mask to force a number to be 8 bits.
CL_GET_CID(RC)	((ClUint32T) ((RC) » CL_CID_OFFSET))	Get a component ID from a returned error code.
CL_GET_ERROR_CODE(RC)	(RC) ((ClUint32T) ((RC) & CL_ERROR_CODE_MASK))	Get the actual error code from a returned error code.
CL_IN		Input argument to a function.
CL_INOUT		Used for both input and output argument to a function.
CL_MAX_NAME_LENGTH	256	Maximum number of characters in a Name type.
CL_NO	0	no value for a boolean.
CL_FORCED_TO_16BITS	0x00	The API executed successfully.
CL_OUT		Output argument to a function.
CL_RC	(CID, ERROR_CODE)	Get the returned code from the component ID and the error code.

Define	*Value*	*Description*
CL_TRUE	1	true value for a boolean.
CL_YES	1	yes value for a boolean.
DISABLE	0	disable value for a boolean.
ENABLE	1	enable value for a boolean.

2.1.1 Error Codes

Define	*Value*	*Description*
CL_ERR_ALREADY_EXIST	0x10	An entry is already existing.
CL_ERR_BUFFER_OVERRUN	0x0d	Buffer over run.
CL_ERR_COMMON_MAX	0xff	2^8-1, the max for common errors.
CL_ERR_DOESNT_EXIST	0x13	An entry does not exist.
CL_ERR_DUPLICATE	0x08	Duplicate entry.
CL_ERR_INITIALIZED	0x0c	Already initialized.
CL_ERR_INUSE	0x15	Resource is in use.
CL_ERR_INVALID_BUFFER	0x06	The buffer passed in is invalid.
CL_ERR_INVALID_HANDLE	0x05	The handle passed is invalid.
CL_ERR_INVALID_-PARAMETER	0x02	Input parameters are invalid.
CL_ERR_INVALID_STATE	0x12	Invalid State.
CL_ERR_MUTEX_ERROR	0x18	Thread mutex error.
CL_ERR_NO_CALLBACK	0x17	No callback available for request.
CL_ERR_NO_MEMORY	0x01	Memory is not available.
CL_ERR_NO_OP	0x19	Null operation.
CL_ERR_NO_RESOURCE	0x0b	No resources.
CL_ERR_NOT_EXIST	0x04	Requested resource does not exist.
CL_ERR_NOT_IMPLEMENTED	0x07	The function not yet implemented.
CL_ERR_NOT_INITIALIZED	0x0e	Component not initialized.
CL_ERR_NULL_POINTER	0x03	Input parameter is a NULL pointer.
CL_ERR_OP_NOT_-PERMITTED	0x20	Requested operation is not permitted.
CL_ERR_OUT_OF_RANGE	0x0a	The value of one of the parameters passed is out of range.
CL_ERR_TIMEOUT	0x14	Timeout.
CL_ERR_TRY_AGAIN	0x16	Component is busy , Try again.
CL_ERR_UNSPECIFIED	0x11	Unknown/Unspecified error.
CL_ERR_VERSION_-MISMATCH	0x0f	Version mismatch.

2.2 Typedefs

Typedef	*Value*	*Description*
signed long long ClUint64T	__attribute__ ((__aligned__-(8)))	Unsigned Long Long, 64-bit.
ClInt8T	* ClAddrT	This can be used for storing address.
ClUint16T	ClBoolT	This can be used for storing boolean datatype.
char	ClCharT	This can be used for strings.
ClInt32T	ClHandleT	This can be used for storing handle.
signed short	ClInt16T	Signed integer, 16-bit.
signed int	ClInt32T	Signed Integer, 32-bit.
signed char	ClInt8T	Signed Integer, 8-bit.
ClUint64T	ClInvocationT	This can be used for storing invocation type.
ClUint64T	ClNtfIdentifierT	This can be used for storing identifier.
ClUint64T	ClOffsetT	This can be used for storing offset.
void *	ClPtrT	This can be used for storing pointer.
ClUint32T	ClRcT	OpenClovis return code type.
ClUint64T	ClSelection-ObjectT	This can be used for storing selection object.
ClUint64T	ClSizeT	This can be used for storing size.
ClInt64T	ClTimeT	This can be used for storing time.
unsigned short	ClUint16T	Unsigned, short integer, 16-bit.
unsigned int	ClUint32T	Unsigned Integer, 32-bit.
unsigned char	ClUint8T	Unsigned Integer, 8-bit.
union cl_u64_u	ClUnion64T	Union having different representations of 64 bit datatype.

2.3 Enumeration *ClClovisComponentT*

Enumeration	*Value*	*Description*
CL_CID_UNSPECIFIED	0x0	Start.
CL_CID_OSAL	0x01	OS Abstraction Layer.
CL_CID_HAL	0x02	Hardware Abstraction Layer.
CL_CID_DBAL	0x03	Database Abstraction Layer.
CL_CID_EO	0x04	Execution Object.
CL_CID_IOC	0x05	Intelligent Object Communication.
CL_CID_RMD	0x06	Remote Method Dispatch.
CL_CID_NAMES	0x07	Name Service.
CL_CID_TIMER	0x08	Timer.
CL_CID_SHM	0x09	Shared Memory Support.
CL_CID_DSHM	0x0a	Distributed Shared Memory.
CL_CID_LOG	0x0b	Logging.
CL_CID_TRACE	0x0c	Tracing.
CL_CID_DIAG	0x0d	Diagnostics.
CL_CID_DEBUG	0x0e	Debug.
CL_CID_CPM	0x0f	Component Management.
CL_CID_CAP	0x10	Capability Management.
CL_CID_RES	0x11	Resource Management.
CL_CID_GMS	0x12	Group Membership Service.
CL_CID_EVENTS	0x13	Event Messaging.
CL_CID_DLOCK	0x14	Distributed Locking.
CL_CID_TXN	0x15	Transactions.
CL_CID_CKPT	0x16	Checkpointing.
CL_CID_COR	0x17	Clovis Object Registry.
CL_CID_CNT	0x18	Containers.
CL_CID_DCNT	0x19	Distributed Containers.
CL_CID_RCNT	0x1a	Resilient Containers.
CL_CID_ALARMS	0x1b	Alarm.
CL_CID_POLICY	0x1c	Policy Engine .
CL_CID_RULE	0x1d	Rule Base Engine.
CL_CID_SCRIPTING	0x1e	Scripting Engine.
CL_CID_CM	0x1f	Chassis Manager.
CL_CID_HPI	0x20	Hardware Platform Interface.
CL_CID_FAULTS	0x21	Fault Management.
CL_CID_AMS	0x22	Availability Management.
CL_CID_MED	0x23	Mediation.
CL_CID_BUFFER	0x24	Buffer Management.
CL_CID_QUEUE	0x25	Queue Management.
CL_CID_CLIST	0x26	Circular List Management.
CL_CID_SNMP	0x27	SNMP Agent.
CL_CID_NS	0x28	Name Service.
CL_CID_OM	0x29	Object Manger.
CL_CID_POOL	0x2a	Pool management.
CL_CID_CD	0x2b	Common Diagnostics.
CL_CID_DM	0x2c	Diagnostics Manager.
CL_CID_OAMP_RT	0x2d	OAMP RT parser.
CL_CID_PROV	0x2e	provisioning Manger.
CL_CID_UM	0x2f	upgrade manager.

Enumeration	*Value*	*Description*
CL_CID_HANDLE	0x30	handle database.
CL_CID_VERSION	0x31	version checker library.
CL_CID_XDR	0x32	xdr library.
CL_CID_IDL	0x33	IDL.

2.4 Data Structures

2.4.1 ClNameT

typedef struct {
ClUint16T length;
ClCharT value [CL_MAX_NAME_LENGTH];
} ClNameT;

The structure *ClNameT* contains the name of the service. Its attributes are:

- *length* - Length of the name in bytes excluding the terminating NULL character.
- *value* - Actual name.

2.4.2 ClVersionT

typedef struct {
ClUint8T releaseCode;
ClUint8T majorVersion;
ClUint8T minorVersion;
} ClVersionT; The structure *ClVersionT* contains the version information for various services. Its attributes are:

- *releaseCode* - Single ASCII capitol letter "A-Z."
- *majorVersion* - Major Number in range of [01-255].
- *minorVersion* - Minor Number in range of [01-255].

Chapter 3

List of Service Description and API Reference Documents

Service Description and API Reference Documents	***Version***
Service Description and API Reference for Alarm Service	0.3
Service Description and API Reference for Availability Management Service	0.3
Service Description and API Reference for Buffer Management Service	0.3
Service Description and API Reference for Checkpointing Service	0.3
Service Description and API Reference for Checksum Management Service	0.3
Service Description and API Reference for Chassis Management Service	0.3
Service Description and API Reference for Circular List Management Service	0.3
Service Description and API Reference for Container Management Service	0.4
Service Description and API Reference for Clovis Object Registry (COR) Service	0.3
Service Description and API Reference for Component Management Service	0.3
Service Description and API Reference for Database Abstraction Layer (DBAL) Service	0.3
Service Description and API Reference for Debug Service	0.3
Service Description and API Reference for Execution Object (EO) Service	0.3
Service Description and API Reference for Event Service	0.3
Service Description and API Reference for Fault Service	0.3
Service Description and API Reference for Group Membership Service	0.4
Service Description and API Reference for Heap Service	0.5
Service Description and API Reference for Interface Definition Language (IDL) Service	0.3
Service Description and API Reference for Intelligent Object Communication (IOC) Service	0.4
Service Description and API Reference for Log Service	0.4

Service Description and API Reference Documents	***Version***
Service Description and API Reference for Mediation Management Service	0.3
Service Description and API Reference for Name Service	0.3
Service Description and API Reference for Operating System Abstraction Layer (OSAL) Service	0.3
Service Description and API Reference for Queue Management Service	0.3
Service Description and API Reference for Remote Method Dispatch (RMD) Service	0.3
Service Description and API Reference for Rule Based Engine (RBE) Service	0.3
Service Description and API Reference for Timer Service	0.3
Service Description and API Reference for Transaction Management Service	0.3

Index