

# **Dialogic® Open System Release**

## **Linux Host Runtime Libraries**

### **High Level Architecture Document**

## TABLE OF CONTENTS

|   |          |
|---|----------|
| <b>DIALOGIC® OPEN SYSTEM RELEASE .....</b>            | <b>1</b> |
| <b>LINUX HOST RUNTIME LIBRARIES .....</b>             | <b>1</b> |
| <b>HIGH LEVEL ARCHITECTURE DOCUMENT .....</b>         | <b>1</b> |
| 1. <i>Introduction</i> .....                          | 3        |
| 1.1 <i>Scope</i> .....                                | 3        |
| 1.2 <i>Glossary</i> .....                             | 3        |
| 1.3 <i>Revision History</i> .....                     | 3        |
| 2.    DM3 SPECIFIC SOFTWARE.....                      | 4        |
| 2.1    DM3 SOFTWARE OVERVIEW .....                    | 4        |
| 2.2    DM3HRT LIBRARIES .....                         | 4        |
| 2.2.1 <i>cheetah_start</i> .....                      | 4        |
| 2.2.2 <i>liboswin32.so</i> .....                      | 5        |
| 2.2.3 <i>libcheetah.so</i> .....                      | 5        |
| 2.2.4 <i>libdxxdm3.so</i> .....                       | 5        |
| 2.2.5 <i>libdtidm3.so</i> .....                       | 5        |
| 2.2.6 <i>libipm_nettsoc.so</i> .....                  | 5        |
| 2.2.7 <i>libdm3cc.so</i> .....                        | 5        |
| 2.2.8 <i>libdm3csp.so</i> .....                       | 5        |
| 2.2.9 <i>libdm3fax.so</i> .....                       | 6        |
| 2.2.10 <i>libDm3OdiLite.so</i> .....                  | 6        |
| 2.2.11 <i>libdm3dcb.so</i> .....                      | 6        |
| 2.2.12 <i>libdm3msi.so</i> .....                      | 6        |
| 2.2.13 <i>libdm3devmgmt.so</i> .....                  | 7        |
| 2.3    DM3 MESSAGING .....                            | 7        |
| 3    SPRINGWARE SPECIFIC SOFTWARE .....               | 8        |
| 3.1    SPRINGWARE LIBRARIES .....                     | 9        |
| 3.1.1 <i>libd42.so</i> .....                          | 9        |
| 3.1.2 <i>libdti.so</i> .....                          | 9        |
| 3.1.3 <i>libdxxx.so</i> .....                         | 9        |
| 3.1.4 <i>libec.so</i> .....                           | 9        |
| 3.1.5 <i>libfax.so</i> .....                          | 9        |
| 3.1.6 <i>libevmgmt.so</i> .....                       | 9        |
| 3.1.6 <i>libpdkrt.so</i> .....                        | 9        |
| 3.1.6 <i>libgcis.so</i> .....                         | 10       |
| 4.    COMMON SOFTWARE ACROSS DM3 AND SPRINGWARE ..... | 11       |
| 4.1    LIBGC.SO .....                                 | 11       |

## 1. Introduction

This document is written to provide the users of the Dialogic Open System Release Host Runtime Library source code with detailed understanding of the various libraries that are available for download.

### 1.1 Scope

This document provides a high level overview of the Host Runtime Libraries supported under the Dialogic Open System Release project. Each library is outlined with a general architecture overview.

### 1.2 Glossary

| Term or Acronym | Description                        |
|-----------------|------------------------------------|
| DM3HRT          | DM3 Host Runtime                   |
| Springware      | Dialogic older legacy product line |
| VOX             | Voice Device                       |
| DTI             | Digital Telephony Interface        |
| MSI             | Modular Station Interface          |
| IPT             | Internet Protocol Technology       |
| IPM             | Internet Protocol Media            |
| SRL             | Standard Runtime Library           |
| TSC             | Time Slot Component?               |
| FW              | Firmware                           |

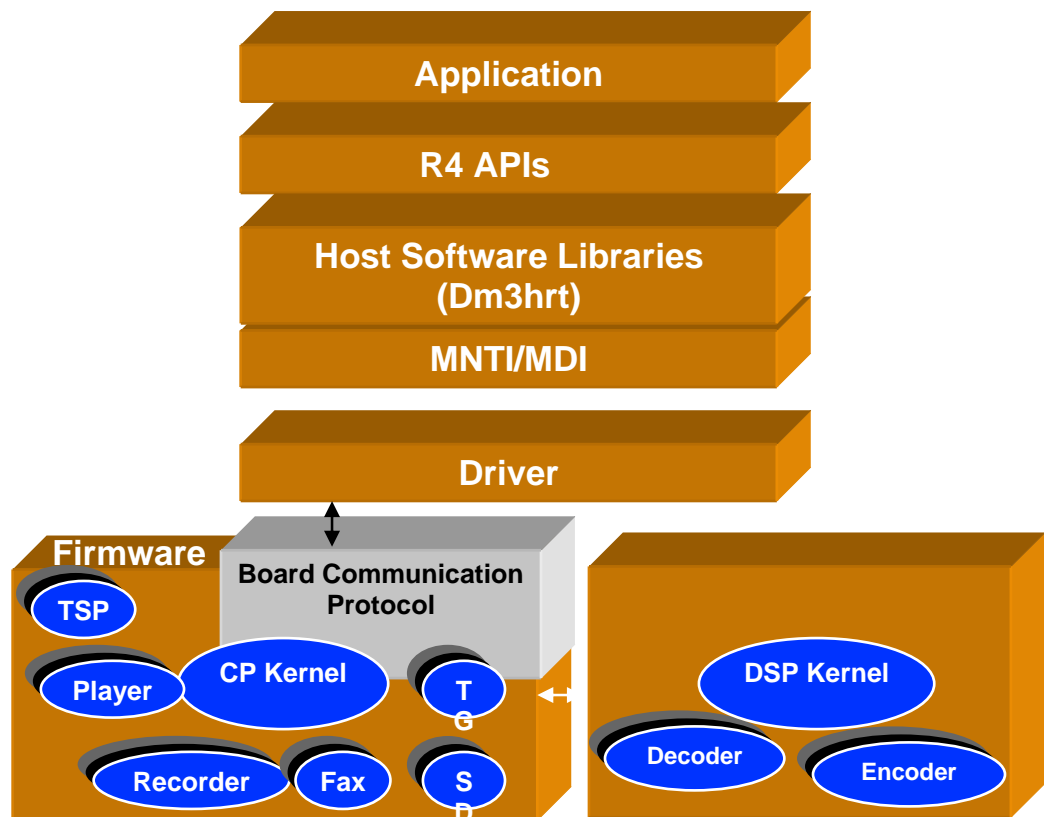
### 1.3 Revision History

| Revision | Author    | Reason for Changes       |
|----------|-----------|--------------------------|
| 0.1      | Host Team | Initial draft 09/20/2006 |

## 2. DM3 Specific Software

### 2.1 Dm3 Software Overview

## DM3 Software Overview Diagram



The DM3HRT libraries were written so that the DM3 boards could be accessible with the same API calls that were used for the Springware products. A common application could be written to interface both cards.

## 2.2 Dm3hrt Libraries

### 2.2.1 cheetah\_start

Cheetah\_Start is the only executable delivered with the DM3HRT VOB. Cheetah start consists of 2 cpp files. Cheetah\_Start.cpp has functions that build and configure a device info list for boards with VOX, DTI, MSI, IPT, IPM or MOH (Music on Hold) devices. It gets the device info data from shared memory. DevMapNode.cpp gets the AUID (Addressable Unique Identifier) associated with the attribute list. It allows the enumeration of the devices. It stores the board condition and the logical id.

### **2.2.2 liboswin32.so**

This library allows the Windows specific commands to work on LINUX. In this way we can maintain common code for each operating system.

### **2.2.3 libcheetah.so**

It is responsible for message transport and streaming. The message transport is based on the DM3component address of the firmware. To prevent blocking, all messages to the firmware are asynchronous. Messaging is discussed in more detail in section 2.3.

### **2.2.4 libdxxdm3.so**

This library provides the dm3 implementation of the Voice API(dx\_) that interacts with the firmware voice components on the DM3 board. It maintains the state machine of the player, recorder, tone generator and signal detector components. These components all together make up what is known as a cluster. They are associated with a CT BUS timeslot and for the most part will be routed to a front end. Each voice channel will have all or some of these components depending on the firmware downloaded. This library interacts closely with libcheetah.so which transmits and receives the voice component firmware messages as well as controls the data streams.

### **2.2.5 libdtidm3.so**

This library provides the dm3 implementation of the Network API (dti\_) on the DM3 boards. It Controls the board related messages and alarms on the board, sets and maintains the attributes associated with the board and channels. It manages the timeslots of the digital front end.

### **2.2.6 libipm\_nettsoc.so**

Opens and closes the IP devices. It maintains the Call Control state machine of each IP channel. It manages the timeslots of the IP front end. It can send or receive digits and set the event mask for alarms. This is the embedded implementation of the H.323 call control stack and has been included in the current release, but will be discontinued in the future releases.

### **2.2.7 libdm3cc.so**

This is the call control library for DM3 boards and contains the corresponding dm3 implementation of the GC API (gc\_) for DM3 boards. The call control protocols supported under this library are ISDN, CAS, R2 and Analog protocols. This library maintains the call control and channel state machine. It binds the Call Reference Number CRN on a particular channel with the FW call Id for a given call and unbinds when the call is terminated. In addition to basic call control, advanced call control such as call hold, supervised transfer and blind transfer are also supported. It processes the cause codes or error codes associated with a call. It handles putting a channel in and out of service. It supports setting of various call control related parameters like CPA, CAS/R2 cdp parameters, Line Coding, Framing, etc. in the FW at runtime.

### **2.2.8 libdm3csp.so**

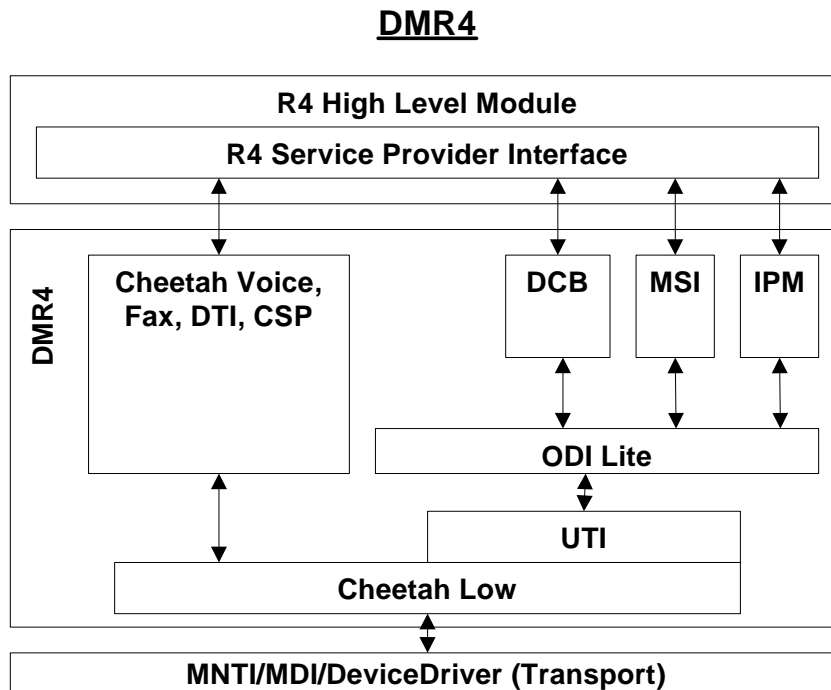
The CSP APIs provide the support for all the EC\_ specific functions calls. The Continuous Speech Processing library allows the player, recorder and signal detector to work together to stop the “prompt” or player component upon the reception of voice activity. It interacts with both libdxxdm3.so and libcheetah.so and maintains the association of voice device and stream. It manages other features such as EC convergence and silence compressed recording.

## 2.2.9 libdm3fax.so

This library provides the dm3 implementation of the Fax API (fx\_) on the DM3 boards. The Dm3 Fax Library maintains the state machine of fax components. It connects the fax device with the transmit device and receive streams associated with the fax.

## 2.2.10 libDm3OdiLite.so

The Object Direct Interface Lite library handles messages between libdm3dcb.so, libdm3msi.so, libipm\_nettsc.so, OAM components (clusterpkg.exe, regvox.exe, libdm3be.so and MSI line activate). Voice, Fax, DTI and CSP libraries talk directly to libcheetah.so.



At download time the library is used to set and get attributes of the board. The CDm3Conf.cpp module handles the conferencing component messages for libdm3dcb.so. The CDm3Tsc.cpp module handles the TSC component messages for the libdm3msi.so. The CDm3IPVsc.cpp module handles the IP component messages for libipm\_nettsc.so.

## 2.2.11 libdm3dcb.so

This library provides the dm3 implementation of the Conferencing API (dcb\_) on the DM3 boards. The Dialogic Conferencing Bridge library handles the conferencing component on the board. The library can create and delete conferences as well as add and remove devices from a conference. Conferencing resource management and bridging across DSP is handled through this library.

## 2.2.12 libdm3msi.so

This library provides the dm3 implementation of the MSI API (msi\_) on the DM3 boards. The Modular Station Interface handles the TSC component on the board. The library maintains the hook state of the phone associated with the MSI channel. It also generates rings as well as monitors the ring state. It can set various parameters associated with the station device such as message waiting indicators and tones.

### **2.2.13 libdm3devmngmt.so**

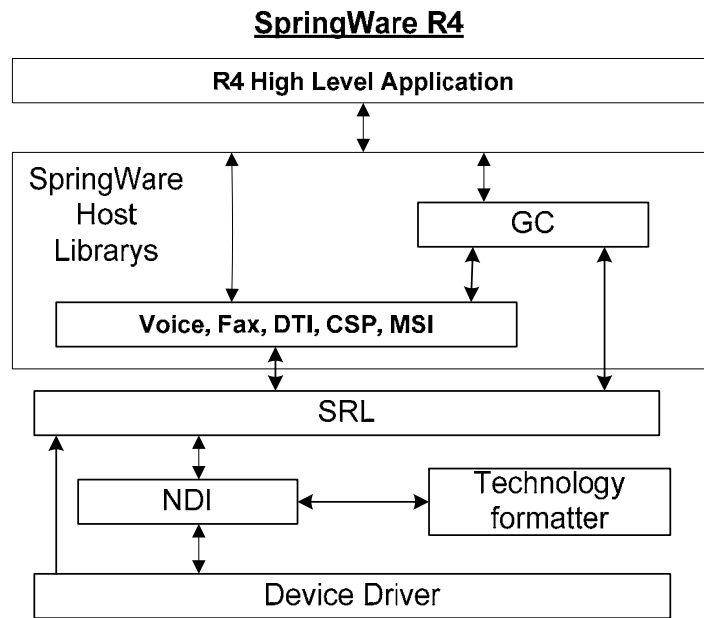
This library provides the dm3 implementation of the board API (brd\_\_\_) on the DM3 boards. This library maintains the keep alive messages to the DM3 board. If the host and firmware disconnect, everything can be reset to regain the connection. It keeps track of the physical and virtual boards in the system.

## **2.3 Dm3 Messaging**

Dm3 uses command/reply message based communication to talk with the board. All messages are asynchronous. Command messages are sent from the host to the board and reply messages are sent from the board to the host.

There are different types of messages that are classified in DM3. With each DM3 message the message type, destination address, source address and optional Transaction ID is included in the message header. Each firmware instance or “voice device” has a unique destination address which is signified by the following format: node:board:processor:component:instance. Each host component for example, dxxxB1C1, has a unique source address which is also signified by the following format: node:board:processor:component:instance. By default, the driver sends the reply message back based upon the source address from where the message is sent. A transaction ID or other user criteria may be supplied in the header, so that multiple commands can be sent for a particular instance and reply messages do not have to be received back in the same order.

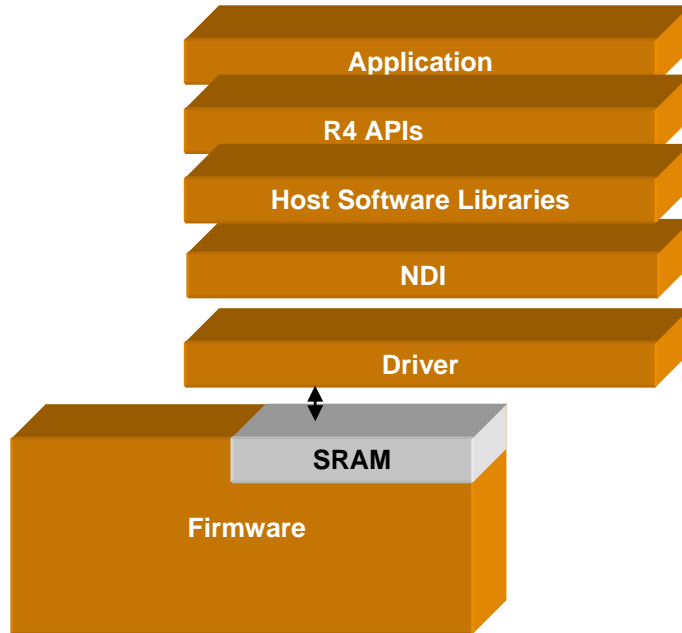
### 3 Springware Specific Software



These libraries provide the APIs for the applications to run on the springware platforms. This is a streams based architecture and use the LIS for the streams functionality. Whenever calls are made to the APIs provided by the Springware specific libraries on the DM3 boards, the calls are routed internally to the DM3 libraries through the Springware libraries. The following Springware specific libraries provide the same functional APIs to the user apps as the DM3 Libraries that were mentioned above.



## SpringWare Software Overview Diagram



### 3.1 Springware Libraries

#### 3.1.1 libd42.so

This library provides the D42 board specific functional APIs to applications on the D42 boards

#### 3.1.2 libdti.so

This library provides the Network API (dti\_) to applications on the Springware boards.

#### 3.1.3 libdxxx.so

This library provides the Voice API(dx\_) to applications on the Springware boards.

#### 3.1.4 libec.so

The CSP APIs provide the support for all the EC\_ specific functions calls.

#### 3.1.5 libfax.so

This library provides the Fax API (fx\_) to applications on the Springware boards.

#### 3.1.6 libevmgmt.so

This library maintains the keep alive messages to the Springware board. If the host and firmware disconnect, everything can be reset to regain the connection. It keeps track of the physical and virtual boards in the system.

#### 3.1.6 libpdkrt.so

This is the control library which is used for Analog and CAS/R2 call control on Springware boards. This library is used in conjunction with the Global Call library and Global Call PDK Protocols.

### **3.1.6 libgcis.so**

This is the translation library between the Global Call library and the Springware ISDN call control library.

## **4.     *Common Software across DM3 and Springware***

### **4.1    *libgc.so***

This library contains the implementation of the Global Call API (gc\_) that is used for call control. The GC library interacts with several call control libraries to provide call control functionality for ISDN, CAS/R2, SIP, H323, Analog and SS7 protocols.