Proprietary Extension for

COMMON-ISDN-API

Version 2.0

Extension for Fax Paper Formats and Resolutions

October 2007

Dialogic Corporation

COPYRIGHT NOTICE AND LEGAL DISCLAIMER

Sixth Edition (October 2007) 206-412-06

All contents of this document are subject to change without notice and do not represent a commitment on the part of Dialogic Corporation or its subsidiaries (“Dialogic”). Reasonable effort is made to ensure the accuracy of the information contained in the document. However, due to ongoing product improvements and revisions, Dialogic Corporation and its subsidiaries do not warrant the accuracy of this information and cannot accept responsibility for errors or omissions that may be contained in this document.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH DIALOGIC® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN A SIGNED AGREEMENT BETWEEN YOU AND DIALOGIC, DIALOGIC ASSUMES NO LIABILITY WHATSOEVER, AND DIALOGIC DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF DIALOGIC PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT OF A THIRD PARTY.

Dialogic products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

It is possible that the use or implementation of any one of the concepts, applications, or ideas described in this document, in marketing collateral produced by or on web pages maintained by Dialogic Corporation or its subsidiaries may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic Corporation or its subsidiaries do not provide any intellectual property licenses with the sale of Dialogic products other than a license to use such product in accordance with intellectual property owned or validly licensed by Dialogic Corporation or its subsidiaries. More detailed information about such intellectual property is available from Dialogic Corporation's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. The software referred to in this document is provided under a Software License Agreement. Refer to the Software License Agreement for complete details governing the use of the software.

Dialogic Corporation encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.

Dialogic and Diva are registered trademarks of Dialogic Corporation or its subsidiaries (“Dialogic”). Dialogic's trademarks may be used publicly only with permission from Dialogic. Such permission may only be granted by Dialogic's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. Any authorized use of Dialogic's trademarks will be subject to full respect of the trademark guidelines published by Dialogic from time to time and any use of Dialogic's trademarks requires proper acknowledgement.

The names of actual companies and products mentioned herein are the trademarks of their respective owners.

To contact Dialogic Customer Support, visit our Web site at www.dialogic.com.

Motivation:

The current **COMMON-ISDN-API** specification defines paper formats ISO A4, ISO B4 and ISO A3 at standard resolution (R8 x 3.85) and high resolution (R8 x 7.7). Support for B4 and A3 is even optional.

The following document proposes a **COMMON-ISDN-API** extension that enables FAX document transmission and reception with paper formats ISO A4, ISO B4 and ISO A3 and these resolutions as specified in T.30:

- R8 x 3.85
- R8 x 7.7
- R8 x 15.4
- R16 x 15.4
- 200 x 200
- 300 x 300
- 400 x 400

The page format and resolution information is passed via appropriate fields in the SFF page header.

To reduce the additions required in a CAPI application in order to make use of the extension, the new functionality is added directly to the existing SFF format. This means that there is a possibility of conflict with future official extensions of the **COMMON-ISDN-API**. To prevent incompatibilities, support for FAX paper formats and resolutions must be enabled explicitly so as to turn off any future CAPI extension that would cause incompatibility.

Specifically, it has to be enabled by the application for a designated controller through a manufacturer request command 9 (Options request) with bit 6 (Enable FAX paper formats and resolutions) set. It will stay active until either the application releases the **COMMON-ISDN-API** or issues another options request with bit 6 not set.

An application can determine whether the **COMMON-ISDN-API** supports FAX paper formats and resolutions by examining the CAPI profile and searching the manufacturer string for "Eicon" or by examining the CAPI profile and trying the manufacturer command 9 described below.

Changes since the first version from 12.1.2001:

- Added support for resolutions 300 x 600, 400 x 800, 600 x 1200, 600 x 600 and 1200 x 1200 dpi.

4.2.2.7 CAPI_GET_PROFILE

Applications call CAPI_GET_PROFILE to retrieve capability information from **COMMON-ISDN-API**. **COMMON-ISDN-API** copies information about implemented features, the total number of controllers, and protocols supported by the requested controller to a 64-byte buffer passed by the calling application. The application must ignore unknown bits. **COMMON-ISDN-API** sets reserved fields to zero. CAPI_GET_PROFILE fills the buffer with the following structure:

Type	**Description**
2 bytes	number of installed controllers, least significant byte first
2 bytes	number of supported B-channels, least significant byte first
4 bytes	Global Options (bit field): [0]: internal controller supported [1]: external equipment supported [2]: Handset supported (external equipment must also be set) [3]: DTMF supported [4]: Supplementary Services (see Part III) [5]: channel allocation supported (leased lines) [6]: parameter *B channel operation* supported [7]: Line Interconnect supported [8]...[31]: reserved
4 bytes	B1 protocol support (bit field): [0]: 64 kbit/s with HDLC framing, always set. [1]: 64 kbit/s bit-transparent operation with byte framing from the network [2]: V.110 asynchronous operation with start/stop byte framing [3]: V.110 synchronous operation with HDLC framing [4]: T.30 modem for fax group 3 [5]: 64 kbit/s inverted with HDLC framing. [6]: 56 kbit/s bit-transparent operation with byte framing from the network [7]: Modem with all negotiations [8]: Modem asynchronous operation with start/stop byte framing [9]: Modem synchronous operation with HDLC framing [10]...[31]: reserved
4 bytes	B2 protocol support (bit field): [0]: ISO 7776 (X.75 SLP), always set [1]: Transparent [2]: SDLC [3]: LAPD in accordance with Q.921 for D-channel X.25 (SAPI 16) [4]: T.30 for fax group 3 [5]: Point-to-Point Protocol (PPP) [6]: Transparent (ignoring framing errors of B1 protocol) [7]: Modem error correction and compression (V.42 *bis* or MNP5) [8]: ISO 7776 (X.75 SLP) modified supporting V.42 *bis* compression [9]: V.120 asynchronous mode [10]: V.120 asynchronous mode supporting V.42 *bis* [11]: V.120 bit-transparent mode [12]: LAPD in accordance with Q.921 including free SAPI selection [13]...[31]: reserved

4 bytes	B3 protocol support (bit field): [0]: Transparent, always set [1]: T.90NL with compatibility to T.70NL in accordance to T.90 Appendix II. [2]: ISO 8208 (X.25 DTE-DTE) [3]: X.25 DCE [4]: T.30 for fax group 3 [5]: T.30 for fax group 3 with extensions [6]: reserved [7]: Modem [8]...[31]: reserved
24 bytes	reserved for **COMMON-ISDN-API** use
4 bytes	Private options (bit field): [0]...[5]: reserved [6]: FAX paper formats and resolutions [7]...[31]: reserved
16 bytes	Manufacturer-specific information

CAPI_GET_PROFILE information structure

Manu ID (dword)

The purpose of the parameter *Manu ID* is to communicate a dword which identifies the manufacturer in MANUFACTURER messages. Every manufacturer supplying MANUFACTURER messages should choose a unique value (such as an abbreviation of the company name).

The manufacturer ID used by Dialogic is:

0x44444944

This information element appears in:

MANUFACTURER_REQ
MANUFACTURER_RESP
MANUFACTURER_CONF
MANUFACTURER_IND

Manufacturer Specific

The purpose of the parameter *manufacturer specific* is to exchange manufacturer-specific information.

Manufacturer specific information for MANUFACTURER_REQ:

word	manufacturer command	Manufacturer specific operation requested.
struct	manufacturer command parameters	Command-dependent parameters for manufacturer request.

Manufacturer specific information for MANUFACTURER_CONF:

word	manufacturer command	Manufacturer-specific operation that was requested.
word	info	Result of the operation according to **COMMON-ISDN-API** definition of Info.

This information element appears in:

MANUFACTURER_REQ
MANUFACTURER_RESP
MANUFACTURER_CONF
MANUFACTURER_IND

Manufacturer Command

The purpose of the parameter *manufacturer command* is to specify the kind of operation requested in a MANUFACTURER_REQ.

The following manufacturer commands are defined:

1:	**Assign PLCI**
2:	**Advanced Codec control**
3:	**DSP control**
4:	**Signaling control**
5:	**RXT control**
6:	**IDI control**
7:	**Configuration control**
8:	**Remove Codec**
9:	**Options request**

This information element appears in:

Manufacturer Specific

Manufacturer Command Parameters

The purpose of the parameter *manufacturer command parameters* is to specify command dependent parameters.

Parameters for manufacturer command 9: Options Request:

dword	Options mask	Manufacturer specific options that have to be enabled: [Bit 0..5]: reserved, must be set to 0 [Bit 6]: Enable FAX paper formats and resolutions [Bit 7..31]: reserved, must be set to 0

This information element appears in:

Manufacturer Command

ANNEX B (NORMATIVE): SFF FORMAT

B.1 Introduction

SFF (**S**tructured **F**ax **F**ile) is a representation specific to fax group 3 documents. As shown below in Fiugre 6 it consists of information concerning the page structure and compressed line data of the fax document. An SFF-formatted document always starts with a header, which is valid for the complete document. Every page starts with a page header. This is followed by the pixel information, line by line. As the SFF format is a file format specification, some entries in header structures (e.g. double-chaining of pages) may not be used or supported by **COMMON-ISDN-API**.

document header	page 1 header	page 1 data	page 2 header	page 2 data	...	page n data

Figure 6: SFF format

B.2 SFF coding rules

The following type conventions are used:

byte	8-bit unsigned
word	16-bit unsigned integer, least significant octet first
dword	32-bit unsigned integer, least significant word first

B.2.1 Document header

Parameter	Type	Comment
SFF_Id	dword	Magic value (identification) of SFF Format: coded as **0x66666653** ("**SFFF**")
Version	byte	Version number of SFF document: coded **0x01**
reserved	byte	Reserved for future extensions; coded **0x00**
User Information	word	Manufacturer-specific user information (not used by **COMMON-ISDN-API**, coded as **0x0000**)
Page Count	word	Number of pages in the document. Must be coded **0x0000** if not known (as in the case of receiving a document).
OffsetFirstPageHeader	word	Byte offset of first page header from start of document header. This value is normally equal to the size of the document header (**0x14**), but there could be additional user-specific data between the document header and the first page header. **COMMON-ISDN-API** ignores and does not provide such additional data.
OffsetLastPageHeader	dword	Byte offset of last page header from start of document header. Must be coded **0x00000000** if not known (as in the case of receiving a document).
OffsetDocumentEnd	dword	Byte offset of document end from start of document header. Must be coded **0x00000000** if not known (as in the case of receiving a document).

B.2.2 Page header

Parameter	Type	Comment
PageHeaderID	byte	**254** (Page data record type)
PageHeaderLen	byte	**0:** Document end **1...255**: byte offset of first page data from the *Resolution Vertical* field of the page header. This value is normally equal to the size of the remainder of the header (**0x10**), but there could be additional user-specific data between page header and page data. **COMMON-ISDN-API** ignores and does not provide such additional data.
Resolution Vertical	byte	Definition of vertical resolution; different resolutions in one document may be ignored by **COMMON-ISDN-API** **0:** 98 lpi (standard) **1:**: 196 lpi (high resolution) **2...254:** reserved **255:** end of document (should not be used: *PageHeaderLen* should be coded **0** instead) If FAX paper formats and resolutions are enabled **0x00:** 98 lpi (3.85 lines per millimeter) **0x01:** 196 lpi (7.7 lines per millimeter) **0x02..0x7f:** reserved **0x80:** reserved **0x81:** 100 lpi **0x82:** reserved **0x83:** 200 lpi **0x84:** reserved **0x85:** 300 lpi **0x86:** 392 lpi (15.4 lines per millimeter) **0x87:** 400 lpi **0x88..0x8a:** reserved **0x8b:** 600 lpi **0x8c..0x8e:** reserved **0x8f:** 800 lpi **0x90..0x96:** reserved **0x97:** 1200 lpi **0x98..0xff:** reserved
Resolution Horizontal	byte	Definition of horizontal resolution **0:** 203 dpi (standard) **1...255:** reserved If FAX paper formats and resolutions is enabled **0x00:** 203 dpi (R8, 8 pels per millimeter) **0x01..0x7f:** reserved **0x80:** reserved **0x81:** reserved **0x82:** reserved **0x83:** 200 dpi **0x84:** reserved **0x85:** 300 dpi **0x86:** 406 dpi (R16, 16 pels per millimeter) **0x87:** 400 dpi **0x88..0x8a:** reserved **0x8b:** 600 dpi **0x8c..0x96:** reserved **0x97:** 1200 dpi **0x98..0xff:** reserved
Coding	byte	Definition of pixel line coding **0:** modified Huffman coding **1...255**: reserved
reserved	byte	Coded as **0**

Line Length	word	Number of pixels per line **1728:** standard fax G3 **2048:** B4 (optional) **2432:** A3 (optional) Support for additional values is optional for **COMMON-ISDN-API** If FAX paper formats and resolutions are enabled **1728:** ISO A4 at R8 x 3.85, R8 x 7.7, R8 x 15.4, 200 x 200 **2048:** ISO B4 at R8 x 3.85, R8 x 7.7, R8 x 15.4, 200 x 200 **2432:** ISO A3 at R8 x 3.85, R8 x 7.7, R8 x 15.4, 200 x 200 **2592:** ISO A4 at 300 x 300, 300 x 600 **3072:** ISO B4 at 300 x 300, 300 x 600 **3648:** ISO A3 at 300 x 300, 300 x 600 **3456:** ISO A4 at R16 x 15.4, 400 x 400, 400 x 800 **4096:** ISO B4 at R16 x 15.4, 400 x 400, 400 x 800 **4864:** ISO A3 at R16 x 15.4, 400 x 400, 400 x 800 **5184:** ISO A4 at 600 x 600, 600 x 1200 **6144:** ISO B4 at 600 x 600, 600 x 1200 **7296:** ISO A3 at 600 x 600, 600 x 1200 **10368:** ISO A4 at 1200 x 1200 **12288:** ISO B4 at 1200 x 1200 **14592:** ISO A3 at 1200 x 1200
Page Length	word	Number of pixel lines per page. If not known, coded as **0x0000**.
OffsetPreviousPage	dword	Byte offset to previous page header or **0x00000000**. Coded as **0x00000001** if first page.
OffsetNextPage	dword	Byte offset to next page header or **0x00000000**. Coded as **0x00000001** if last page.

B.2.3 Page data

Page data are coded line by line: data describes each pixel row. Lines are coded as records of variable length; each line is coded according to the element *coding* in the page header. At present, only modified Huffman coding is supported. MH coding is bit-oriented: the pixel bits are stored in the bits of code words least significant first. No EOL code words or fill bits are included. If the data include EOL code words, **COMMON-ISDN-API** ignores these.

Each record is identified by the first byte:

- **1...216:** a pixel row with 1...216 MH-coded bytes follows immediately
- **0:** escape code for a pixel row with more than 216 MH-coded bytes. In this case, the following word in the range **218...32767** defines the number of MH-coded bytes which follow.
- **217...253**: white space, skip 1...37 empty lines
- **254**: start of page header (see above)
- **255:** if followed by a byte with value **0**, illegal line coding. Applications may choose whether to interpret this line as empty or as a copy of the previous line. If this byte is followed by a byte with a value **1...255**, then 1...255 bytes of additional user information follow (reserved for future extensions).