



Dialogic® Converged Services Platform - SwitchKit® Development Environment

Intelligent Network & Wireless Protocols Overview

Copyright and Legal Disclaimer

Copyright © [1998-2008] Dialogic Corporation. All Rights Reserved. You may not reproduce this document in whole or in part without permission in writing from Dialogic Corporation at the address provided below.

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. 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 EXPLICITLY SET FORTH BELOW OR 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, Dialogic Pro, Brooktrout, Cantata, SnowShore, Eicon, Eicon Networks, Eiconcard, Diva, SIPcontrol, Diva ISDN, TruFax, Realblobs, Realcomm 100, NetAccess, Instant ISDN, TRXStream, Exnet, Exnet Connect, EXS, ExchangePlus VSE, Switchkit, N20, Powering The Service-Ready

Network, Vantage, Connecting People to Information, Connecting to Growth, Making Innovation Thrive, and Shiva, among others as well as related logos, are either registered trademarks or trademarks of Dialogic.

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

Dialogic Product Line Warranty

Unless otherwise stated in an applicable product purchase agreement between the Customer and Dialogic, Dialogic warrants that during the Warranty Period, products will operate in substantial conformance with Dialogic's standard published documentation accompanying the product. If a product does not operate in accordance therewith during the Warranty Period, the Customer must promptly notify Dialogic. Dialogic, at its option, will either repair or replace the product without charge. The Customer has the right, as their exclusive remedy, to return the product for a refund of purchase price or license fee if Dialogic is unable to repair or replace it.

Warranty Period

In the event that you have no signed agreement setting out a warranty period, the Warranty Period shall be the standard warranty period set out on www.dialogic.com on the date of your purchase of the product.

The Warranty Period begins on the date of shipment of any products or software by Dialogic.

The Warranty Period for repaired, replaced or corrected products and software shall be coterminous to the Warranty Provided for the original products or software purchased.

To report warranty claims, Customer may contact Dialogic via email at techsupport@cantata.com or call (781) 433-9600.

Warranty Provisions

A. During the Warranty Period, Dialogic warrants to Customer only that:

- (i) Products manufactured by Dialogic (including those manufactured for Dialogic by an original equipment manufacturer) will be free from defects in material and workmanship and will substantially conform to specifications for such products;
- (ii) software developed by Dialogic will be free from defects which materially affect performance in accordance with the specifications for such software. With respect to products or software or partial assembly of products furnished by Dialogic but not manufactured by Dialogic, Dialogic hereby assigns to Customer, to the extent permitted, the warranties given to Dialogic by its vendors of such items.

B. If, under normal and proper use, a defect or non conformity appears in warranted products or software during the applicable Warranty Period and Customer promptly notifies Dialogic in writing during the applicable warranty period of such defect or non conformance, and follows Dialogic's instructions regarding return of such defective or non conforming Product or Software, then Dialogic will, at no charge to Customer, either:

- (i) repair, replace or correct the same at its manufacturing or repair facility or
- (ii) if Dialogic determines that it is unable or impractical to repair, replace or correct the product or software, provide a refund or credit not to exceed the original purchase price or license fee.

C. No product or software will be accepted for repair or replacement without the written authorization of and in accordance with instructions from Dialogic. Removal and reinstallation expenses as well as transportation expenses associated with returning such product or software to Dialogic shall be borne by Customer. Dialogic shall pay the costs of transportation of the repaired or replaced product or software to the destination designated in the original Order. If Dialogic determines that any returned product or software is not defective, Customer shall pay Dialogic's costs of handling, inspecting, testing and transportation. In repairing or replacing any product, part of product, or software medium under this warranty, Dialogic may use new, remanufactured, reconditioned, refurbished or functionally equivalent products, parts or software media. Replaced products or parts shall become Dialogic's property.

D. Dialogic makes no warranty with respect to defective conditions or non conformities resulting from any of the following: Customer's modifications, misuse, neglect, accident or abuse; improper wiring, repairing, splicing, alteration, installation, storage or maintenance performed in a manner not in accordance with Dialogic's or its vendor's specifications, or operating instructions; failure of Customer to apply Dialogic's previously applicable modifications or corrections; or items not manufactured by Dialogic or purchased by Dialogic pursuant to its procurement specifications. Dialogic makes no warranty with respect to products which have had their serial numbers removed or altered; with respect to expendable items, including, without limitation, fuses, light bulbs, motor brushes and the like; or with respect to defects related to Customer's data base errors. Improper packaging of product for repair will not be covered under this warranty agreement. No warranty is made that software will run uninterrupted or error free.

E. Warranty does not include:

- a) Dialogic's assistance in diagnostic efforts;
- b) access to Dialogic's Technical Support web sites, databases or tools;
- c) product integration testing;
- d) on-site assistance; or
- e) product documentation updates.

These services are available either during or after warranty at Dialogic's published prices.

F. THE FOREGOING WARRANTIES ARE EXCLUSIVE & ARE GRANTED IN LIEU OF ALL OTHER EXPRESS & IMPLIED WARRANTIES (WHETHER WRITTEN, ORAL, STATUTORY OR OTHERWISE), INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. CUSTOMER'S SOLE AND EXCLUSIVE REMEDY AND DIALOGIC'S SOLE OBLIGATION HEREUNDER, SHALL BE TO REPAIR, REPLACE, CREDIT OR REFUND AS SET FORTH ABOVE.

G. IN NO EVENT SHALL DIALOGIC, ITS DIRECTORS, OFFICERS, EMPLOYEES, AGENTS OR AFFILIATES, BE LIABLE FOR ANY COSTS OR DAMAGES ARISING DIRECTLY OR INDIRECTLY FROM YOUR USE OF ANY PRODUCT INCLUDING ANY INDIRECT,

INCIDENTAL, SPECIAL, EXEMPLARY, MULTIPLE, PUNITIVE OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, WHETHER BASED ON CONTRACT, TORT (INCLUDING NEGLIGENCE), STRICT LIABILITY OR OTHER LEGAL THEORY, EVEN IF DIALOGIC, OR ANY OF ITS DIRECTORS, OFFICERS, EMPLOYEES, AGENTS OR AFFILIATES HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN ANY EVENT, DIALOGIC'S CUMULATIVE LIABILITY TO YOU FOR ANY AND ALL CLAIMS RELATING TO THE USE OF ANY PRODUCT SHALL NOT EXCEED THE TOTAL AMOUNT OF THE PURCHASE PRICE OR LICENSE FEES PAID TO DIALOGIC FOR SUCH PRODUCT.

H. CUSTOMER AND DIALOGIC HEREBY WAIVE THEIR RIGHT TO TRIAL BY JURY TO THE FULLEST EXTENT PERMITTED BY LAW IN CONNECTION WITH ALL CLAIMS ARISING OUT OF OR RELATED TO THIS WARRANTY, THE PRODUCTS COVERED HEREBY OR THE PERFORMANCE OF ANY PARTY HEREUNDER.

I. THIS WARRANTY SHALL BE CONSTRUED UNDER AND GOVERNED BY THE LAWS OF THE COMMONWEALTH OF MASSACHUSETTS WITHOUT GIVING EFFECT TO ANY CHOICE OR CONFLICT OF LAW PROVISION OR RULE (WHETHER OF THE COMMONWEALTH OF MASSACHUSETTS OR ANY OTHER JURISDICTION) THAT WOULD CAUSE THE APPLICATION OF THE LAWS OF ANY JURISDICTION OTHER THAN THE COMMONWEALTH OF MASSACHUSETTS. CUSTOMER SPECIFICALLY AND IRREVOCABLY CONSENTS TO THE PERSONAL AND SUBJECT MATTER JURISDICTION AND VENUE OF THE FEDERAL AND STATE COURTS OF THE COMMONWEALTH OF MASSACHUSETTS AND SUCH COURTS SHALL HAVE EXCLUSIVE JURISDICTION WITH RESPECT TO ALL MATTERS CONCERNING THIS WARRANTY OR THE ENFORCEMENT OF ANY OF THE FOREGOING.

J. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM JURISDICTION TO JURISDICTION.

About this Publication

Purpose

This publication provides guidelines for using the Dialogic® CSP.

Safety Labels

The following Safety labels may appear in this information product to alert customers to avoidable hazards. The following are in the order of priority:



DANGER

Danger indicates the presence of a hazard that will cause death or severe personal injury if the hazard is not avoided.



WARNING

Warning indicates the presence of a hazard that can cause death or severe personal injury if the hazard is not avoided.



CAUTION

Caution indicates the presence of a hazard that will or can cause minor personal injury or property damage if the hazard is not avoided. Caution can also indicate the possibility of data loss, loss of service, or that an application will fail.

Conventions used

This information product uses the text conventions explained below. In addition, hexadecimal numbers are preceded by a zero and small “x.” For example, the decimal number 15 is represented in hexadecimal as 0x0F.

Convention	Description
. . .	A horizontal ellipsis in an API message indicates fields of variable length.
:	A vertical ellipsis in an API message indicates that a block of information is repeated or is variable.
<i>n</i>	The letter <i>n</i> is a generic placeholder for a number.
Sans serif mono space	Indicates a command name, option, input, output, non-GUI error, and system messages.
<i>Sans serif monospace italic</i>	Indicates a parameter name in an input message. Example: move *.dot a: c: -s The -s is the parameter.
<i>Serif italic</i>	Indicates the name of a book, chapter, path, file, or API message. Example: <i>UserDirectory/Config.exe</i>
Boldface	Indicates keyboard keys, key combinations, and command buttons Example: Ctrl+Alt+Del
Sans serif boldface	Identifies text that is part of a graphical user interface (GUI). Example: Go to the Configuration menu and select Card->Span Configuration

Contents

Copyright and Legal Disclaimer	2
Dialogic Product Line Warranty	4

1 Intelligent Network and Wireless

Introduction	1-2
Supported Operating Environments and Compatibility	1-8
Intelligent Network and Wireless Protocols.....	1-9
SwitchKit Intelligent Network and Wireless Components	1-14
Benefits of using SwitchKit to develop an IN/Wireless Application.....	1-17
IN and Wireless Call Flows.....	1-19
Wireless Call Flow	1-22
Licensing of Intelligent Network and Wireless Protocols	1-23

1 Intelligent Network and Wireless

Purpose This chapter explains the intelligent network (IN) and wireless protocols offered by Dialogic and how to use them in developing applications.

Introduction

Overview As software developers create more applications that are being deployed and maintained in mobile and intelligent networks, Dialogic is keeping pace by introducing support for intelligent network (IN) and wireless protocols in the CSP. The API supports IN application development enabling API customers to create sophisticated IN applications for deployment on the CSP. For customers who prefer to use SwitchKit as their application development tool, the IN protocols that Dialogic supports can communicate with a CSP using SwitchKit API and the SwitchKit component, Low-Level Communicator (LLC). Using SwitchKit to create IN applications can significantly reduce the development cycle, especially with the availability of two new components that perform abstraction from the signaling foundation technologies.

The CSP currently operates effectively as a Service Node (SN) in many production networks. The addition of IN protocols allows the CSP to also serve effectively as:

- A Service Control Point (SCP)
- An Intelligent Peripheral in emerging Intelligent Networks.

IN and Wireless Protocols Dialogic's Platform initially supports five protocols, developed by IntelliNet Technologies, to access IN and wireless networks starting with release 8.2.2 CI. Additional capabilities will be included in future releases.

ITU/ETSI standard

- Customized Application for Mobile network Enhanced Logic (CAMEL)
- Universal Mobile Telecommunications System- Mobile Application Part (UMTS- MAP) which includes Global System for Mobile communication - Mobile Application Part (GSM- MAP)
- Intelligent Network Application Protocol (INAP)

TIA/ANSI standard

- ANSI - 41
- Wireless Intelligent Network (WIN)
- AIN (planned for future availability)

Advanced Intelligent Network (AIN) conforms to BELLCORE Generic Requirements AIN0.2 GR 1299 (Issue 4, 09/97).

For more details on IntelliNet's IN protocols see, the following documents (Available from Dialogic):

- CAMEL User's Guide, document part number P_EXCL_CAMEL_UG_2.0
- UMTS MAP User's Guide, document part number P_EXCL_UMTS_MAP_UG_2.0
- WIN User's Guide, document part number P_EXCL_WIN_UG_2.0
- ANSI 41D User's Guide, document part number P_EXCL_AS41D_UG_2.0
- INAP User's Guide, document part number P_EXCL_INAP_UG_2.0

The IN protocols are implemented on the CSP through the offering of IntelliNet's protocol-specific encoder/decoder software (codecs). See *Table 1-3, Codecs (1-9)* for a description of the codecs used with each IN protocol.

SwitchKit IN and Wireless Components

Dialogic offers two optional SwitchKit layers that perform abstraction/adaptation from the signaling foundation technologies:

- SwitchKit TCAP Abstraction Layer (SKTAL)
- SwitchKit Interface Module (SKIM)

Use of the SwitchKit TCAP Abstraction Layer and the SwitchKit Interface Module is intended to simplify the work required for the application developer. Use of these components is optional; however, SKIM requires the use of SKTAL.

SwitchKit TCAP Abstraction Layer

The SwitchKit TCAP Abstraction Layer is used to create a generic view of the CSP's TCAP layer.

SwitchKit Interface Module

The Switchkit Interface Module provides an API interface to abstract services from the TCAP and SCCP layers to a transaction-based application.

The IN application, codecs, and SwitchKit reside on the host computer, using the SS7 services of the CSP.

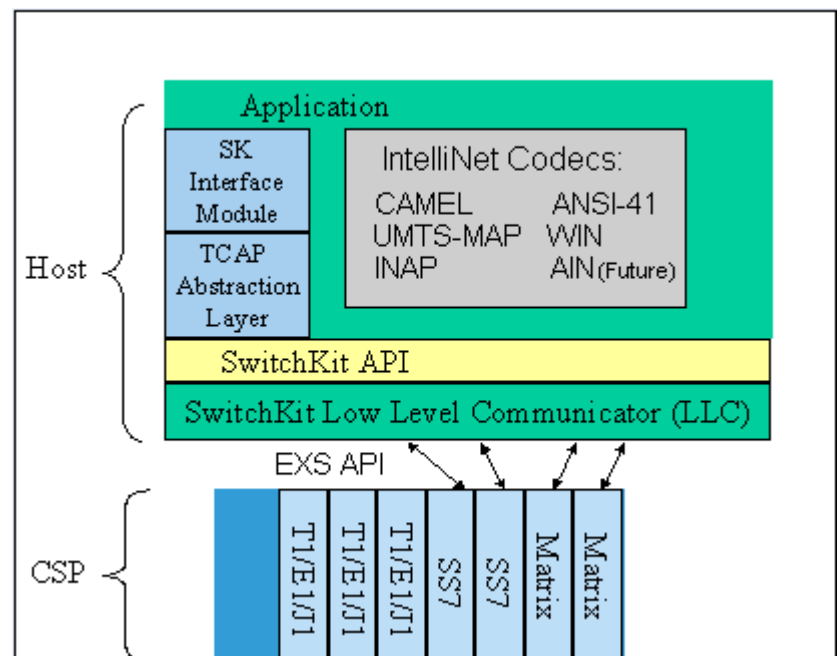
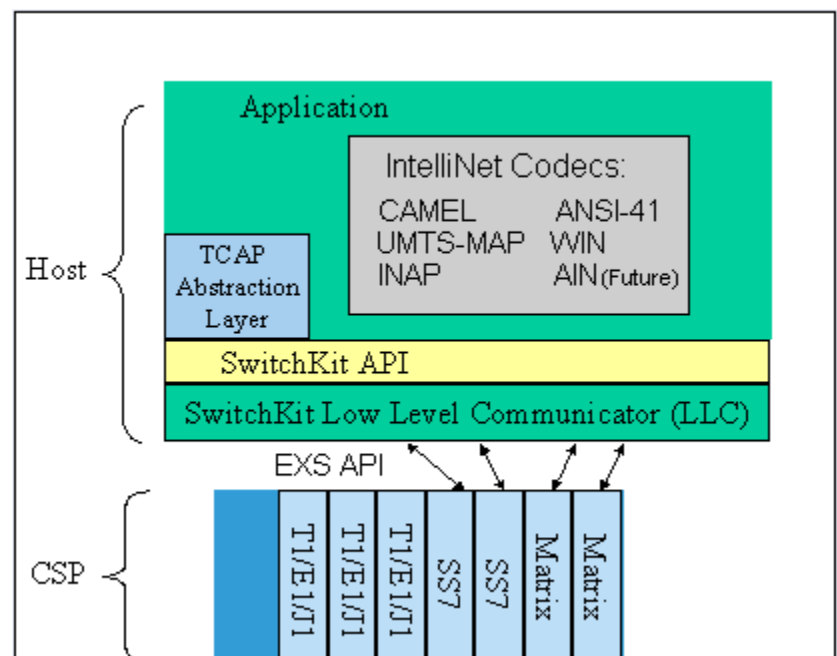
The next two diagrams show where the IntelliNet IN and wireless protocols and abstraction layers fit into the CSP architecture.

SwitchKit Users

In addition to implementing the required service logic, the application developer must handle the message flow between the IntelliNet codecs and the SS7 protocol stack on the CSP, using SKIM and SKTAL (as desired) and the SwitchKit API and LLC components.

Threadsafe

All elements of this offering support threadsafe operations.

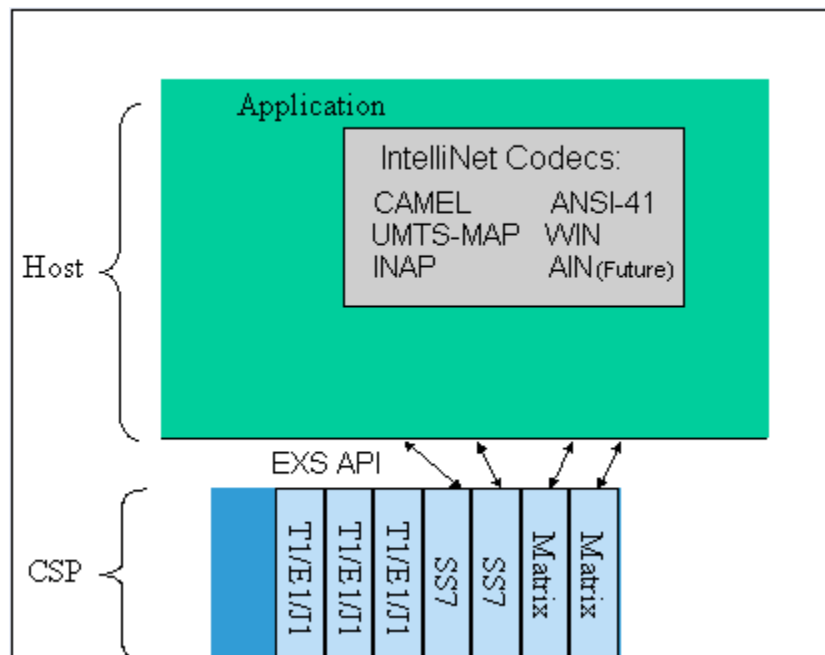
Scenario for SwitchKit Users with SKIM and SKTAL**Scenario for SwitchKit Users with SKTAL Only**

API Users Application developers who do not use the SwitchKit Interface Module and or the SwitchKit TCAP Abstraction Layer must implement equivalent functionality to manage communication with the CSP, handle the TCAP dialogues, and format messages to and from the codec APIs.

The IN and wireless application and codecs reside on the host computer using the SS7 services of the CSP.

Scenario for API Users

In addition to implementing the required service logic, the application developer must handle the message flow between the IntelliNet codecs and the SS7 protocol stack on the CSP, using API.



Codecs The IntelliNet IN and wireless codecs are provided to developers as application layer APIs in the form of header files and software libraries of protocol-specific ASN.1 encoders/decoders. The encoders/decoders are delivered to customers in binary format, available for linking with the application developer's software and (optionally) SwitchKit to form an executable image. These libraries execute on a host computer, not the CSP.

Performance & Reliability The IN and wireless protocols are capable of operating in redundant, partially redundant, and non-redundant CSP environments. This includes optional redundancy for each of the following components:

- Host machine
- LLC
- CSP Matrix Series 3 Card
- SS7 cards
- Line cards

Supported Operating Environments and Compatibility

Purpose This section describes the operating environments that are supported by Intelligent Network (IN) and wireless protocols.

Compatibility Each of the IN and wireless codecs is supported in the following host hardware/operating system environments:

Table 1-1 Hardware/Operating System Requirements

Operating System	IN & Wireless Protocols are supported on...
Solaris	Version 8.0 for the SPARC platform
Linux	Red Hat Linux version 8.0 for INTEL platform
HP-UX	Version 11.0

The next table shows the compilers supported by IN and wireless protocols in a development environment.

Table 1-2 Compiler Requirements

Operating System	Applications must be built using...
Solaris	Sun Forte (Sun Workshop 6, update 2 C++ 5.3)
Linux	GNU gcc compiler packaged with Red Hat Linux 8.0 (gcc 3.2)
HP-UX	HP aC++ compiler 3.3

Versions of the IN and wireless protocols are independent of SwitchKit and CSP system software versions, so that updates to the IN and wireless protocols will not require updates to either SwitchKit or the CSP system software to support the same level of functionality. Similarly, updates to SwitchKit and CSP system software versions will not require updates to the IN and wireless protocols to support the same level of functionality.

The IN and wireless protocols require CSP system software and hardware, Release 8.2.2 CI or higher. The IN and wireless protocols are supported by both the SS7 Series 3 and the SS7 PQ cards.

Intelligent Network and Wireless Protocols

Purpose This section describes Dialogic's software support for mobile and wireline communications networks. The Intelligent Network (IN) and wireless protocols offered by Dialogic were developed by IntelliNet Technologies, a leading provider of IN infrastructure software. Dialogic's platform supports five IntelliNet protocols to access IN and wireless networks:

Table 1-3 Codecs

Codec Name	Description	Standard
Customized Application for Mobile network Enhanced Logic (CAMEL)	Phase 2, 3, & 4	3GPP TS 29.078 (v5.1.0 Release 5)
Universal Mobile Telecommunications System - Mobile Application Part (UMTS-MAP) which includes Global System for Mobile communication - Mobile Application Part (GSM-MAP)	Phase 1,2,2+, & 3	UMTS MAP - 3GPP TS 29.002 V4.2.1 (2000-12) 3G TS 29 002 v3.4.0 (2000-3)
Wireless Intelligent Network (WIN)	Phase I & II	TIA/EIA/IS 771 TIA/EIA/IS 826
ANSI - 41	ANSI - 41D	TIA/EIA-41.(1-6) D Dec 1997
Intelligent Network Application Protocol (INAP)	CS-1 CS-2	ITU-T Q.1218, release 1095 ITU-T Q.1228, release 997

Matrix The matrix below shows the IN wireline and wireless protocols supported by telecommunications standards.

Standard Network	ITU	ANSI
WIRELINE	INAP	AIN (Future)
WIRELESS	CAMEL UMTS/GSM MAP	WIN ANSI-41

CAMEL A European Telecommunications Standards Institute (ETSI) standard messaging protocol for including IN functions into GSM mobile networks. CAMEL is used when roaming between networks, allowing the home network to monitor and control calls made by its subscribers. CAMEL API allows roaming subscribers access to their full portfolio of Intelligent Network (IN) services. CAMEL is a relatively inexpensive method of allowing telecom operators to add new services to the existing network infrastructure.

A few typical applications include:

- Pre-Paid Calling
- Personal Numbering
- Location dependent services

UMTS/GSM-MAP An ETSI standard messaging protocol used in UMTS/GSM wireless networks to communicate among network elements to support user authentication, equipment identification, and roaming:

Mobile Switching Center (MSC)

Home Location Register (HLR)

Visitor Location Register (VLR)

Equipment Identity Register (EIR)

Short Message Service Center (SMSC)

Authentication Center (AuC)

Typical applications include:

- Intelligent Peripheral (IP)
- Service Control Point (SCP)
- Enhanced Services Platform

The IntelliNet MAP protocol layer provides an easy to use library of C++ classes that can be utilized to develop GSM Mobility Applications.

WIN A Telecommunications Industry Association/American National Standards Institute (TIA/ANSI) standard messaging protocol that enables subscribers in ANSI-41 based wireless networks to use intelligent network services. WIN also supports the network capabilities to provide wireless activities such as automatic roaming, incoming call screening, and voice-controlled services.

Intellinet's WIN protocol facilitates the development of platform-independent, transport-independent and vendor-independent WIN services such as:

- Hands-Free
- Voice-Controlled Services
- Voice-Controlled Dialing (VCD)
- Voice-Controlled Feature Control (VCFC)
- Voice-Based User Identification (VUI)
- Incoming Call-Restriction/Control
- Calling Name Presentation (CNAP)
- Password Call Acceptance (PCA)
- Selective Call Acceptance (SCA)

ANSI - 41 A TIA/ANSI standard messaging protocol used in Code-Division Multiple Access (CDMA) and Time Division Multiple Access (TDMA) wireless networks primarily in the Americas and parts of Asia to communicate among network elements (MSC, HLR, VLR, EIR, SMSC) to support inter-system hand-off, automatic roaming, authentication, and supplementary call features. The ANSI 41D specification (formerly known as IS-41) is primarily used in the wireless network to provide services such as automatic roaming,

authentication, intersystem hand-off, and short message service. All wireless network elements use this messaging protocol to communicate.

Typical applications include:

- Intelligent Peripheral (IP)
- Service Control Point (SCP)
- Enhanced Services Platform

INAP

Intelligent Network Application Protocol (INAP), an ITU-T specification, allows applications to communicate between various nodes/functional entities of a wireline Intelligent Network. The protocol defines the operations required to be performed between nodes/functional entities for providing Intelligent Network services.

A few typical applications include:

- Call Center solutions requiring handling of special service numbers (800 & 900 services)
- Local Number Portability
- Calling Card registration and authentication including charging and fraud management capabilities
- Interactive Voice Response (IVR) systems for small and large business segments
- Calling Name delivery
- Service Management systems for study of traffic patterns as well as generating call reports and billing records at central administration and billing center.

Future availability of AIN protocol

Dialogic plans to offer the Advanced Intelligent Network (AIN) protocol to support ANSI-based, wireline Intelligent Networks in future releases.

Some examples of applications using AIN include:

- Toll-free dialing and FreePhone facilities for subscribers
- Virtual Private Network Services for closed user groups operating over geographically distributed facilities
- Universal Access Number (UAN)
- Split Charging capability enabling the subscriber to separately charge personal and business calls made from the same instrument

- Call Rerouting and Redistribution based on traffic volume and/or time of day suitable for telemarketing businesses and reservation centers with multiple locations.
- Prepaid and Calling Card services
- Televoting, whereby franchisees may cast their choice over secure voice response systems, preserving privacy, possible travel time as well as avoiding human tampering of results and other malpractices.

SwitchKit Intelligent Network and Wireless Components

Overview To integrate the IntelliNet Intelligent Network (IN) and wireless protocol stacks with the CSP, application developers can make use of two basic layers to reduce their development cycle when using SwitchKit as their application development tool. These abstraction layers are:

- SwitchKit TCAP Abstraction Layer (SKTAL):
Formats data for the IntelliNet codecs interfacing with the CSP TCAP stack.
- SwitchKit Interface Module (SKIM):
Hides the TCAP stack from the application developer.

These abstraction layers are supported for both ANSI and ITU TCAP standards.

SwitchKit TCAP Abstraction Layer

The SwitchKit TCAP Abstraction Layer presents a generic view of the CSP TCAP layer by translating between PPL Events and TCAP messages. This is a modular layer that is directed to applications that require explicit control over TCAP operations. With the SwitchKit TCAP Abstraction Layer the following is provided:

- A consistent programming paradigm to an existing SwitchKit user
- A simple API to handle TCAP operations
- Dialogue ID allocation and management capabilities
- Error handling and detection capability for transaction level protocol errors that can be handled transparently
- Notification and alarm generation mechanism to the application

The SwitchKit TCAP Abstraction Layer is delivered in binary format. To form an executable image using the defined functionality this layer is required to be linked with:

- The application developer's software
- The required codecs
- SwitchKit
- SKIM (optional)

SKTAL APIs The following is a list of APIs used with the SwitchKit TCAP Abstraction Layer.

sktal_initializeTCAP()
sktal_terminateTCAP()
sktal_registerTCAPSSNHandler()
sktal_unregisterTCAPSSNHandler()
sktal_allocateTCAPDialogID()
sktal_setTCAPDialogueHandler()
sktal_clearTCAPDialogHandler()
sktal_sendTCAPDialog()
sktal_recvTCAPDialog()
sktal_sendTCAPComponent()
sktal_recvTCAPComponent()

The SwitchKit TCAP Abstraction Layer employs the handler function SwitchKit model for event handling.

SwitchKit Interface Module

The SwitchKit Interface Module differs from the SwitchKit TCAP Abstraction Layer primarily in its ability to abstract details of the TCAP protocol for an application.

The SwitchKit Interface Module provides a consistent programming paradigm to existing SwitchKit users. This module's APIs abstract details of the SS7 TCAP protocol. This interface can be used by application developers to reduce their development cycle. The SwitchKit Interface Module has a default class for each type of operation and knows which operations have linked operations. If the module is unable to handle a certain message flow, the application can use the SwitchKit TCAP Abstraction Layer. The capabilities of the SwitchKit Interface Module are summarized as follows:

- Abstracts details of the TCAP protocol for an application.
- Formats the TCAP messages into a message list
- Keeps track of active TCAP dialogs
- Invokes entities, such as, IDs, operation types, and operation codes.
- Provides error handling and detection capability for transaction level protocol errors transparently.

- Provides notification and alarm generation mechanism to the application.
- Tracing

The SwitchKit Interface Module is delivered in binary format. To form an executable image using the defined functionality this module is required to be linked with:

- The application developer's software
- The required codecs
- SwitchKit API
- SKTAL

SKIM APIs

The following is a list of APIs used with the SwitchKit Interface Module. These APIs employ the handler function SwitchKit model for event handling.

- SKIM_Api::Initialize()
- SKIM_Api::Terminate()
- SKIM_Api::Send()
- SKIM_Api::SendSSNInService()
- SKIM_Api::SendSSNOutOfService()
- SKIM_Api::SendReject()
- SKIM_Api::CancelOperation()
- SKIM_Api::SendPreArrangedEnd()
- SKIM_Api::SetTransHandler()
- SKIM_Api::ClearTransHandler()
- SKIM_Api::EnableTracing()
- SKIM_Api::DisableTracing()

Benefits of using SwitchKit to develop an IN/Wireless Application

Overview SwitchKit provides a comprehensive, high-level and open programming environment for the application development and maintenance of the CSP. SwitchKit includes a feature-rich operation, administration, maintenance, and provisioning (OAM&P) system and a high-level API suite, freeing developers to concentrate on revenue-generating applications and services.

SwitchKit greatly reduces the time and cost of deploying and maintaining switch-based solutions - providing a quantitative edge in markets that are increasingly competitive, dynamic, global, and deregulated. SwitchKit enables the development and implementation of custom, highly-differentiated services, providing a qualitative edge in response to the demands of the telecommunications marketplace.

Application Development IN/wireless application developers can use either the SwitchKit Interface Module and or the SwitchKit TCAP Abstraction Layer, in conjunction with the SwitchKit API. Without these layers, developers must create applications that can process PPL Event messages to and from the CSP. The application would use normal SwitchKit API procedures to connect to the LLC and register for messages. Then, through the initialization of the SwitchKit Interface Module, the application would register a default handler for all incoming IN Messages and notifications.

SwitchKit API The SwitchKit API is a library of function calls used to communicate to both the LLC and the CSP. Both the SwitchKit TCAP Abstraction Layer and the application will use the SwitchKit API to communicate to LLC.

The Role of LLC The LLC is responsible for connection to the CSP Platform, application redundancy, channel and channel group management, as well as message routing to multiple applications. See the diagrams on Page 1-5

for additional information. For IN applications the LLC is responsible for routing TCAP messages (PPL Events) to the proper application based on Stack ID (local PC) and subsystem number (SSN).

OAM&P Operation, administration, maintenance, and provisioning (OAM&P) for all aspects of all components required for the IN and wireless protocols is handled by SwitchKit.

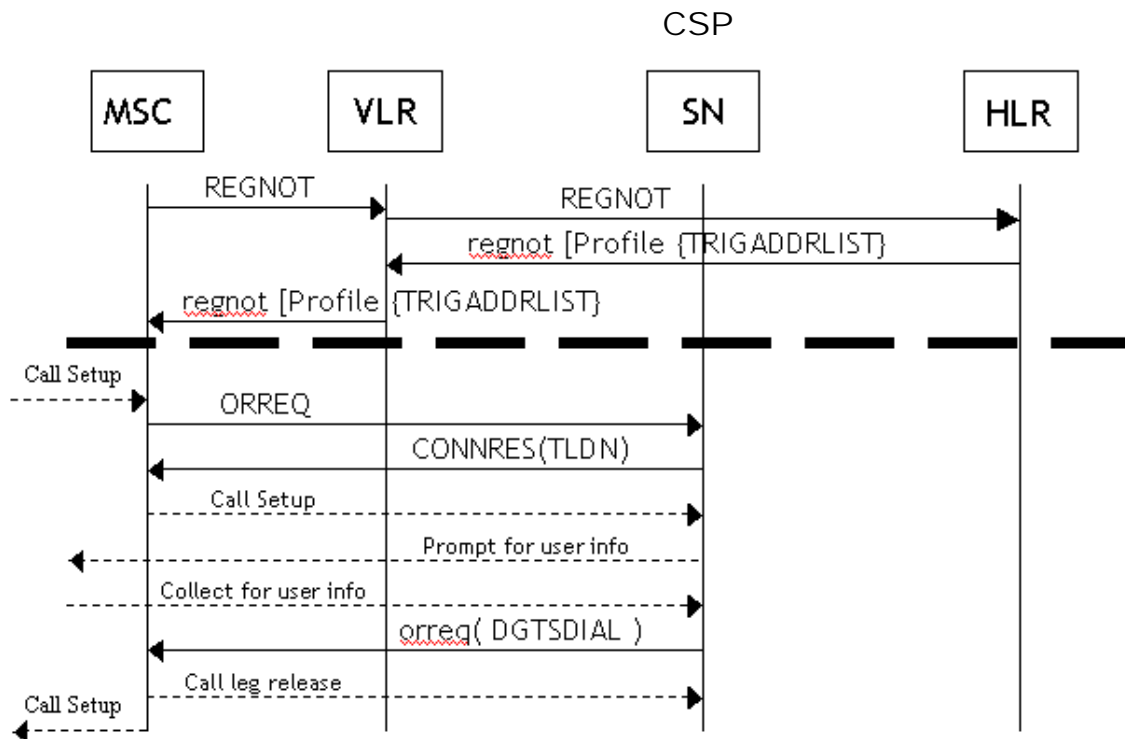
IN and Wireless Call Flows

Overview The following call flows provide examples of an application handling prepaid Intelligent Network (IN) services and voice controlled dialing using WIN and CAMEL. The following acronyms are used in the calls flows:

Table 1-4 Call Flow Acronyms

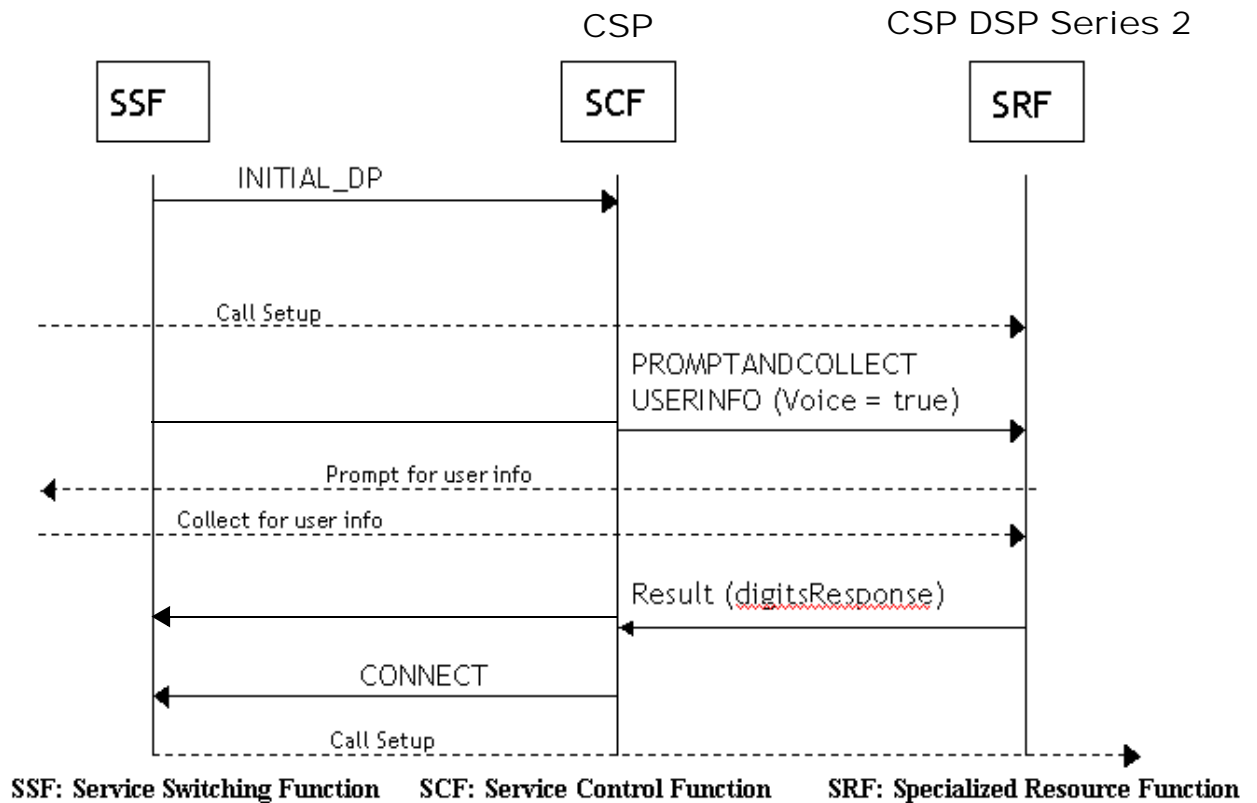
Acronym	Description
BCSM	Basic Call State Model
CCF	Call Control Function Function that sets up and tears down connections
HLR	Home Location Register
MSC	Mobile Switching Center
SCF	Service Control Function The service control point (SCP)
SN	Service Node
SRF	Specialized Resource Function Interactive Voice Response system, to play announcement (for example, Intelligent Peripheral)
SSF	Service Switching Function The switch (typically a Class5 or MSC for example)
UMSC	U(niversal Mobile Telecommunications System) Mobile Switching Center
VLR	Visitor Location Register

Voice Controlled Dialing with WIN



1. MS subscriber register to HLR and VLR and arm triggers.
2. Call comes in, ORREQ message is sent to SN.
3. SN ask MSC to connect resource to temporary local directory number (TLDN).
4. Call is setup to SN, prompt and collect user information.
5. SN send back ORREQ message with called party number (DGSTDIAL).
6. Call is setup to final destination.

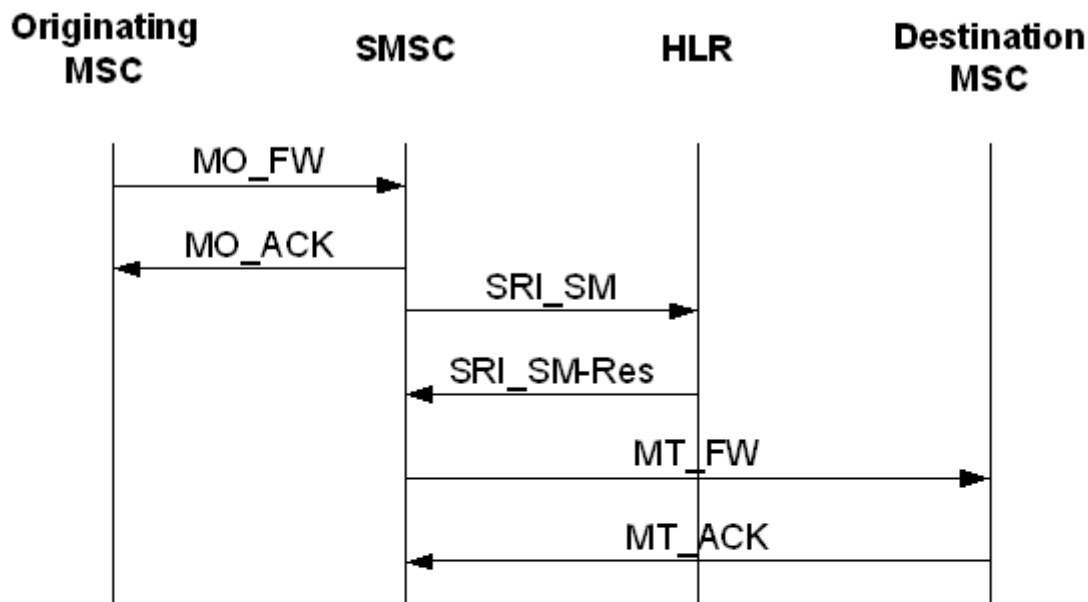
Voice Controlled Dialing with CAMEL



1. TDP-R trigger must be armed on the switch
2. Call comes in, InitialDP is sent to SCF.
3. SCF ask SSF to connect call to the SRF.
4. PromptAndCollectUserInfo is sent to the SRF.
5. SRF plays an announcement and performs speech recognition.
6. SRF return the results (DigitsResponse) to the previous invoke method (PromptAndCollectUserInfo).
7. SCF send a CONNECT message to the SSF.
8. Call is setup to final destination.

Wireless Call Flow

Purpose The next call flow provides an example of an SMSC application sending and receiving SMS.



MO_FW – the MSC forwards an SMS to the SMSC that has been submitted by a subscriber.

MO_ACK – The SMSC acknowledges the reception of the SMS.

SRI_SM – The SMSC sends a “Send Routing Info for Short Message” message to the HLR, requesting the destination’s location.

SRI_SM Res – The HLR responds with the routing info.

MT_FW – the SMSC route the SMS to the destination MSC.

MT_ACK – The destination MSC acknowledges the reception of the SMS.

Licensing of Intelligent Network and Wireless Protocols

Overview Operation of the Intelligent Network (IN) protocols on the CSP requires two separate license keys:

- SS7 Product License SCCP/TCAP
- IntelliNet Codec License

These license keys are orderable through Dialogic Sales, and are based on the following customer-supplied information:

- Chassis ID
Unique serial number of the specific CSP chassis running the SCCP/TCAP stack
- Host ID
Manufacturer's unique identification number for the customer-supplied host computer
- Number of Active SS7 Cards
This is the number of SS7 cards that are licensed to execute the SCCP/TCAP stack concurrently in the specified CSP chassis. A redundant SS7 card pair counts as 1 Active SS7 Card. This number has a range of 1 to 4.
- Codecs
Required IN protocol stacks. Valid values are UMTS MAP, CAMEL, WIN, and ANSI-41 and INAP. Future values will include AIN.

These license keys are independent from the codec version, host operating system, and CSP System Software release.

Upgrades Upgrades for additional Active SS7 Cards or Codecs will result in new license keys for both the SS7 Product License SCCP/TCAP and the IntelliNet codec license, superseding the previous license keys for the specified Chassis ID and Host ID.

Pricing Pricing for these license keys is available from Dialogic Sales.