Ministry of Defence
Defence Standard 00-74

Issue 2 Publication Date 19 December 2008

ASAAC Standards

Part 2:

Rationale Report for Software Standards

AMENDMENT RECORD

Amd No	Date	Text Affected	Signature and Date

REVISION NOTE

HISTORICAL RECORD

Interim Defence Standard 00-74 Part 2 Issue 1 – Published 14 January 2005

Table of Contents

List of Figures

PREFACE

a. This standard specifies the requirements to define and validate a set of open architecture standards, concepts & guidelines for Advanced Avionics Architectures.

b. This standard has been produced on behalf of the Standardization Advisory Group (SAG)

c. This standard has been agreed by the authorities concerned with its use and is intended to be used whenever relevant in all future designs, contracts, orders etc. and whenever practicable by amendment to those already in existence. If any difficulty arises which prevents application of the Defence Standard, UK Defence Standardization (DStan) shall be informed so that a remedy may be sought.

d. Any enquiries regarding this standard in relation to an invitation to tender or a contract in which it is incorporated are to be addressed to the responsible technical or supervising authority named in the invitation to tender or contract.

e. Compliance with this Defence Standard shall not in itself relieve any person from any legal obligations imposed upon them.

f. This standard has been devised solely for the use of the Ministry of Defence (MOD) and its contractors in the execution of contracts for the MOD. To the extent permitted by law, the MOD hereby excludes all liability whatsoever and howsoever arising (including, but without limitation, liability resulting from negligence) for any loss or damage however caused when the standard is used for any other purpose.

This page is intentionally left blank

1 Introduction

1.1 Scope of this Document

This document is produced under the contract ASAAC Phase II Contract [6]. It is the second deliverable associated with Work Package 32410, “Final Draft of Proposed Standards for Software” and is the Rationale report for this Standard, which is included in Poste 3D of the contract.

1.2 Work Package Objectives

The objective of work package WP32400 is to produce the final draft of the Standards that define an IMA system, its architecture, software and the Common Functional Modules (CFMs) to operate within it.

In order to obtain a set of software standards for an IMA core-processing system, it is not sufficient merely to define a set of standards without giving a justification as to their selection or their content, which is the objective of this document.

1.3 Software Standards

During ASAAC a common software model based on the concept of a layered software architecture has been defined. Within this model, the layers are separated by standardised interfaces in order to provide independence of these layers. Interfaces encapsulate a lower software layer and provide a type of virtual machine view to a higher software layer. In this context, each interface provides a generic set of services and resources. This supports the following top-level requirements as identified in [7]:

Number	Requirement
TLR_2	Modules Applicable to Wide range of platforms
TLR_3.1	Re-use of Software
TLR_3.2	Module replaceable at first line
TLR_3.3	No base and depot level maintenance
TLR_3.4	Deferred maintenance
TLR_3.5	Comprehensive BIT and Testability
TLR_8	Interoperability
TLR_9	Interchangeability
TLR_10	Technology Transparency
TLR_11	Use of Commercial components, technologies and processes
TLR_12	Maximise digital processing of functions
TLR_13.1	General system requirements and performance
TLR_13.2	Sensors and sub-system
TLR_13.3	Interface definitions
TLR_13.4	Criticality of functions
TLR_14.1	Growth Capability

Number	Requirement
TLR_14.2	Modularity and configurability
TLR_15	Certification and qualification
TLR_16	Security
TLR_17	System management

1.4 Abbreviations

API	Application Programming Interface
AC	Aircraft
APOS	Application to Operating System interface
ASAAC	Allied Standard Avionics Architecture Council
BIT	Built in Test
CFM	Common Functional Module
GLI	Generic system management Logical Interface
GSM	Generic System Management
HW	Hardware
IA	Integration Area
ITM	Integrated Test and Maintenance
MLI	Module Logical Interface
MMM	Mass Memory Module
MOS	MSL to Operating System interface
MSL	Module Support Layer
OLI	Operating system Logical Interface
OS	Operating System
OSL	Operating System Layer
PCM	Power Conversion Module
PE	Processing Element
RE	Resource Element
SMBP	System Management to Blueprint interface
SMLI	System Management Logical Interface
SMOS	System Management to Operating System interface
SW	Software
TLR	Top Level Requirement
VC	Virtual Channel

2 WARNING

The Ministry of Defence (MOD), like its contractors, is subject to both United Kingdom and European laws regarding Health and Safety at Work, without exemption. All Defence Standards either directly or indirectly invoke the use of processes and procedures that could be injurious to health if adequate precautions are not taken. Defence Standards or their use in no way absolves users from complying with statutory and legal requirements relating to Health and Safety at Work.

3 Related Documents

3.1 The publications shown below are referred to in the text of this Standard. Publications are grouped and listed in alphanumeric order.

A) References to published standards

None.

B) References to standards in preparation

[1]	ASAAC2-STA-32410-001-SWG Issue 01	Final Draft of proposed Standards for Software
[2]	ASAAC2-STA-32420-001-HWG Issue 01	Final Draft of proposed Standards for Communications / Network
[3]	ASAAC2-STA-32430-001-HWG Issue 01	Final Draft of proposed Standards for Common Functional Module
[4]	ASAAC2-STA-32440-001-HWG Issue 01	Final Draft of proposed Standards for Packaging
[5]	ASAAC2-STA-32460-001-CPG Issue 01	Final Draft of Proposed Standards for Architecture

C) References to other documents

[6]	N°26/97/SPAé/ST/AVI du 26/06/97	Clauses Techniques Annexées au marché 97/86.066, ASAAC Phase II
[7]	ASAAC2-RPT-52100-010-TMG-I01 Issue 01	Stage 1 Final Report
[8]	ASAAC-STA-32420-002-HWG Issue 01	Rationale Report for Communications / Network Standards
[9]	ASAAC-STA-32430-002-HWG Issue 01	Rationale Report for Common Functional Module Standard

D) References to documents from other organizations

None.

3.2 Reference in this Standard to any related document means in any Invitation to Tender or contract the edition and all amendments current at the date of such tender or contract unless a specific edition is indicated.

3.3 In consideration of clause 3.2 above, users shall be fully aware of the issue and amendment status of all related documents, particularly when forming part of an Invitation to Tender or contract. Responsibility for the correct application of standards rests with users.

3.4 DStan can advise regarding where related documents are obtained from. Requests for such information can be made to the DStan Helpdesk. How to contact the helpdesk is shown on the outside rear cover of Def Stans.

4 ASAAC Software Architecture

4.1.1 Software Architecture Overview

The ASAAC Software Architecture is based on a three-layer stack with each layer being described in terms of it dependency/independency on both the aircraft system and the underlying hardware as shown by Figure 1.

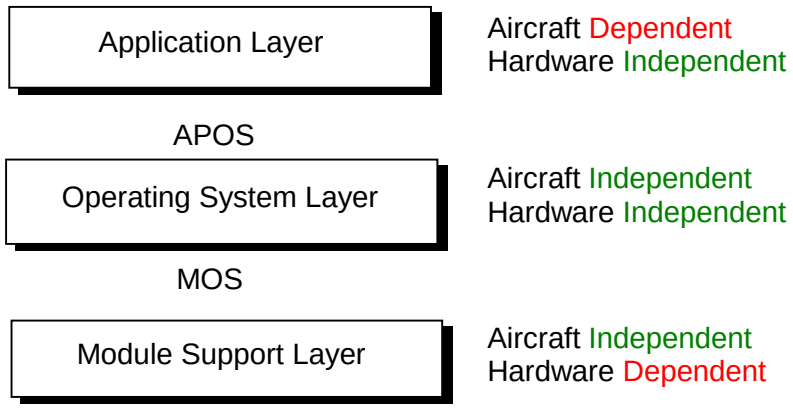

Figure 1 - ASAAC Three Layer Software Architecture

The full ASAAC Software Architecture is more complex than that shown in Figure 1 and includes a number of standardised interfaces, both direct and logical, and a number of software components that have a standardised functional behaviour.

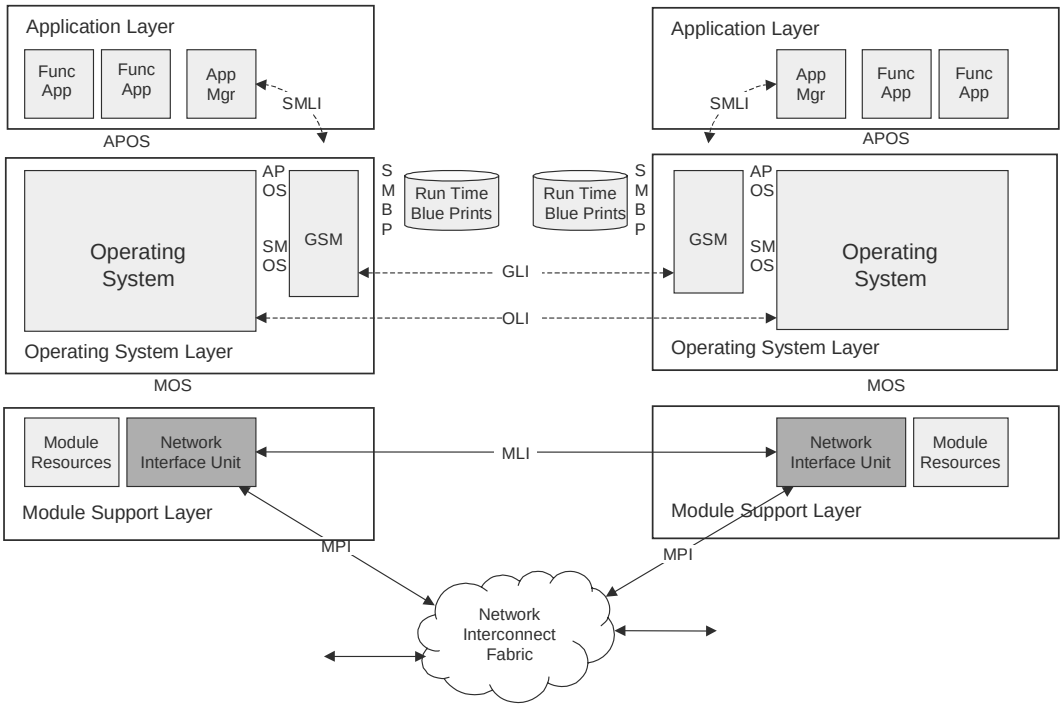

Figure 2 - The Software Architecture Model

Table 1 shows the standardised components that comprise the full ASAAC Software Architecture and the interfaces between them. The interfaces in the application domain are not standardised, while the interfaces in the domain of OSL and MSL are standardised. The components and their relationship are shown in Figure 2 as well.

Table 1 - ASAAC Software Architecture Components and Interfaces

from (rows) \ to (columns)	**Functional Application**	**Application Manager**	**OS**	**GSM**	**RTBP**	**MSL**
Functional Application	*not standard-ised*	*not standard-ised*	**APOS**	null	null	null
Application Manager	*not standard-ised*	*not standard-ised*	**APOS**	**SMLI**	null	null
OS	null	null	**OLI**	null	null	**MOS**
GSM	null	**SMLI**	**SMOS**	**GLI**	**SMBP**	null
RTBP	null	null	null	null	null	null
MSL	null	null	null	null	null	**MLI**

The remaining sections in this document provide the rationale behind including each of these components within the ASAAC Software Architecture.

5 Software Components

5.1 Functional Applications

The term "Functional Applications" relates to all functions that handle the processing of operational data, e.g.

- Radar Applications,
- Mission Management,
- Stores Management,
- Vehicle Management System,
- Communication, Navigation and Identification.

5.1.1 Justification for Functional Applications

These are the applications that comprise the operational functionality of the system.

5.2 Application Management

Application Management covers that aspect of system management whose purpose is to control the mission moding selection and the selection of modes within a particular mission.

5.2.1 Justification for Application Management

Some aspects of system management are directly related to the aircraft or mission being flown so, according to the definition of the three layers in Figure 1, cannot reside in the Operating System Layer. Thus, application management becomes another instance of a functional application, but one that can communicate directly with the Generic System Management resident in the OSL and thus either instigate mode changes during a mission or respond to changes in available resources due to faults and failures.

5.3 Operating System

The Real-Time OS provides the particular part of OSL functionality that controls the real-time behaviour of the Processing Element and its associated resources.

5.3.1 Justification for Operating System

The Real-time Operating System provides the particular part of OSL functions the management of PE resources, especially the handling the real-time scheduling of threads. In detail, it comprises the following functionality:

- Process Management

  Responsible for assigning memory segments to processes and ensuring integrity of memory segments.

- Communication Services

  Responsible for the Virtual Channel (VC) communication

- Synchronisation Services

  Responsible for the control of semaphores and events

- Time Management

This includes the management of:

- Time-out periods for blocking services
- The time interval between periodic threads

- Thread Management

  This includes the management of:

  - Scheduling
  - State transition
  - Deadline overrun detection
  - Thread queuing discipline

- Error Handling

  Responsible to detect PE errors and to provide the error reporting functions.

- File Management Services

5.4 Generic System Management

The GSM provides the basic functions that enable the system management at the RE, IA and AC levels within a system to control the resources and behaviour of the ASAAC core.

5.4.1 Justification for Generic System Management

The functionality offered by the GSM is generic and therefore this function resides in the Operating System Layer of the basic three-layer model and comprises:

- Health Monitoring

  Required to detect the occurrence of faults.

- Fault Management

  Required to identify, localise and contain any faults that occur during a mission as well as log data for maintenance purposes.

- Configuration Management

  Required to initialise and shutdown a system as well as performing reconfiguration due to the receipt of mode change requests or failures.

- Security Management

  Required to protect the integrity, availability and confidentiality of ‘protectively marked' data as it is uploaded to, downloaded from and processed within a system.

5.5 Runtime Blueprints

The RTBP contain the data (e.g. process description, routing information, fault management data) required to configure and manage the core processing on which it is hosted.

5.5.1 Justification for Runtime Blueprints

The system configuration is described in terms of configuration states and transitions between configuration states. A configuration state is characterised as each stable state of a GSM entity (i.e. an individual AC, IA or RE level GSM function).

It is ultimately the Runtime Blueprints that provide the data that describe these states and the transitions between them. This data comprises:

- Configuration Description

  Defines a configuration status.

- Transition Definition

  Provides a sequence of actions to be performed in order to transfer a GSM entity from one configuration status into another configuration status.

- Configuration Data

  Define the description of atomic configuration items.

- Fault Management Description

  Defines the fault management policy and corrective actions.

- Security Management Description

  Defines the security management policy.

5.6 Module Support Layer

The Module Support Layer (MSL) implements the Module to Operating System Interface (MOS), which encapsulates the hardware architecture of a given Common Functional Module (CFM).

5.6.1 Justification for Module Support Layer

The Module Support Layer encapsulates the details of the underlying hardware and provides generic, technology independent access to low-level resources. The encapsulation is provided using defined services. These services comprise communication and board resource services.

In case an operating systems needs additional optional MOS services, the MOS is defined in a way that there is no need for the layer above to have any knowledge about the implementation of the HW provided of the MOS interface.

6 Direct Interfaces

6.1 APOS: Application to Operating System Interface

6.1.1 Justification for APOS

The Application to Operating System Interface provides the Functional Application Software developer with the following services:

- Communication Services (Virtual Channels)
- Time Services
- Thread Control Services
- Intra-process Synchronisation Services (Semaphores, Events)

- Fault Reporting Services
- File Handling Services
- Power Control Services

Thus, the APOS establishes means to couple applications to the ASAAC core and allow them to share all system properties provided by the ASAAC software architecture.

6.1.2 Justification for Identified Services

The ASAAC software architecture presents an increasing level of abstraction from the resources level towards the higher layers. This is especially reflected by the concept of virtual channels for all communications between processes and the use of configuration information contained in the run-time blueprints. Because all interactions with system management are restricted to the SMLI, all the APOS services are provided by Operating System solely. The justification for the APOS services is:

- Communication Services

  The use of virtual channels and the blueprint information enforces the modularity and re-usability of application functions as virtual channels provide applications a way to communicate transparent to the underlying resources and the kind and number of co-operating application functions.

- Time Services

  The time services provide basic time information on module, system and global time in order to facilitate time synchronisation within a process, between application functions and with external events.

- Thread Control Services

  Threads represent the temporal characteristics of an application. The need to prioritise processing within a single process justifies the utilisation of multiple threads.

- Synchronisation Services

  The synchronisation services are required in order to support the implementation of multi-threaded algorithms.

- Error Handling Services

  Support the handling of errors in the application domain and synchronisation with the handling of errors and faults in the domain of the ASAAC core.

- File Handling Services

  Provide access to files on MMM.

- Power Control Services

  Provide control of power switches on PCM.

6.2 MOS: Module Support Layer to Operating System Interface

6.2.1 Justification for MOS

The Module to Operating System Interface presents a common set of standardised low-level services to the operating system. Thus, the MOS encapsulates the underlying processing hardware and in this context provides transparency to the operating system layer.

It is important to mention, that the set of MOS services supports all ASAAC system properties required by the Operating System to provide sufficient services to the applications via the APOS and to the System Management functions via SMOS.

The MOS interface also provides for services and their behaviour, which allow the Operating System Layer to be implemented and ported on an arbitrary hardware platform. Therefore, the MOS is divided into a ‘Core MOS’ and ‘Optional MOS’ subsets. Additionally there are module specific subsets of MOS services for PCM and MMM modules.

6.2.2 Justification for Identified Services

The MOS services have been divided into three groups of services:

- Processor Support Services

  The Processor Support Services define an interface, which encapsulates processor dependent features such as execution context handling.

- Board Services

  The Board Services define an interface, which encapsulate board specific device accesses.

- Communication Services

  The Communication Services define an interface, which encapsulate network services such as receiving and sending messages on Transfer Connections.

6.3 SMOS: System Management to Operating System Interface

6.3.1 Justification for SMOS

The SMOS, encapsulated within the OSL, describes the services provided by the Operating System to the Generic System Management. It establishes a bound between the resource management functions that are provided by the operating system, and the system management functions that are provided by the Generic System Management.

The SMOS interface ensures the technology transparency for the operating system.

6.3.2 Justification for Identified Services

The SMOS services provide an interface between generic system management and the operating system for command, control, and configuration purposes. To cope with the system management the SMOS provides services necessary to 'instantiate' a system according to the information contained in the blueprints:

- Resource Configuration Control Services

These services provide control of the standardised configuration items for a particular PE. They include the set-up of processes, threads, their associated scheduling information, virtual channels, transfer connections network ports and clock in accordance with the RTBP configuration data.

- Control of remote Modules

  This includes the set-up of the network, the initialisation and the monitoring of remote modules in accordance with the RTBP configuration description.

- BIT Management Services

  Provide the control and access on local BIT functions required for the fault management within the ASAAC core and ITM activities.

- CFM Information Services

  Provide status and characteristics of the local module.

- Fault and Logging Management Services

  Provide services to complement the GSM fault management function.

- Security Management Services

  Provide services to complement the GSM security management function.

The SMOS provides for a replication of MOS services, because context switching, scheduling and the control of access rights are dealt with within the OS. Because the Generic System Management consists of a set of processes under the control of the OS, no direct access to the MOS is feasible. Therefore those replicated services allow the GSM access to the relevant MOS services through the SMOS.

6.4 SMBP: System Management to Blueprint Interface

6.4.1 Justification for SMBP

The standardisation of this interface separates the generic GSM functions from the system specific blueprints data.

The SW standard (see [1]) defines the grammar, but not the format of the Runtime Blueprints. The SMBP interface hides the project specific Runtime Blueprint implementation from the GSM. It assumes that the implementation can be mapped to a tree and can be parsed with the standard operations of a tree.

6.4.2 Justification for the RTBP Grammar

The standardised RTBP grammar provides the basic tree of blueprint data required for every ASAAC core. This set of data is therefore open for project specific extension of the RTBP information.

6.4.3 Justification for Identified Services

The SMBP services are independent from the contents of the RTBP but only dependent on the tree grammar of the RTBP referring to a single root. There are two kinds of services:

- Tree Traversal Services

  Navigate within the structure of runtime blueprint data.

- Information Access Services

  Access the configuration data.

7 Logical Interfaces

7.1 OLI: Operating System Logical Interface

7.1.1 Justification for OLI

The OLI addresses the interaction between different instances of the Operating System.

It defines the necessary aspects and protocols necessary to promote interoperability of the modules with one another and the provision of access to MMM localised files.

7.1.2 Justification for OLI Services

The OLI covers:

- Data Equivalence Services

  The OLI provides the information required for the translation between the standardised network representation of virtual channel data and the data representation used by the processes of the local PE.

- Remote File Access

  The OLI provides access to files required by the Generic System Management for process initialisation and handles requests for module initialisation. It ensures transparency with respect to the location of a file.

7.2 GLI: Generic System Management Logical Interface

7.2.1 Justification for GLI

The GLI addresses the interaction between adjacent hierarchical levels of Generic System Management functions.

It defines the aspects and protocols necessary to promote interoperability of the GSM entities with one another.

7.2.2 Justification for GLI Services

The GLI covers the protocols for the actions taken according to RTBP transition definitions between GSM entities of adjacent management layers. This is necessary for the achievement of interoperability and consistency between different implementations of GSM.

The protocols for the actions within one GSM entity, i.e. within one management hierarchy level are left at the designers' discretion. The GLI covers:

- Configuration / Reconfiguration Management

  Control of the configuration of sub-ordinate GSM instances.

- Module Management

  The allocation and de-allocation of module resources to integration areas.

- Fault Management

  Health monitoring for sub-ordinate GSM instances and escalation of fault handling to the super-ordinate GSM instance.

- Security Management

  Key management between adjacent GSM management layers.

7.3 SMLI: System Management Logical Interface

7.3.1 Justification for SMLI

Application Control involves co-operation between Application Management (AM) in the application layer and Generic System Management (GSM) in the operating system layer. The term "Application Management" relates to the part of applications, which manage the Logical Configurations.

The Application Management as well as the Generic System Management consists of a set of processes. The communication and synchronisation between these specific and generic parts of the system management, which not necessarily need to be co-located on a PE is performed by the use of virtual channels. This interface sets forth necessary protocols for the communication between Application Management and GSM.

7.3.2 Justification for SMLI Services

As the algorithms to handle logical configurations on the higher system management levels and the algorithms to handle distant errors are of aircraft dependent nature they cannot be handled by the Generic System Management exclusively.

The SMLI therefore covers:

- Change of Logical Configurations

  The change of a logical configuration can be initiated by the application functions or requested by the generic system management due to a degradation of available core resources.

- Signalling of Distant Errors

  The event of a distant error occurring in a separate integration area may require a reaction of application management.

7.4 MLI: Operating System Logical Interface

7.4.1 Justification for MLI

The MLI defines the logical interactions between modules so as to meet the module interoperability requirement. The justification of the Module Logical Interface is defined in reference [8] and reference [9].

The MLI provides interoperability between modules. To meet the interoperability requirement the MLI services as defined in ref. [1] are deemed necessary.

7.4.2 Justification for MLI Services

The MLI services are split into the following groups:

- CFM Resource Management Services

  CFM Resource Management Services are needed to provide a way of getting information on the condition of the CFM such as status reports and test results.

- Download Management Services

  The Download Management Services provide the means to transfer software images and configuration data from the MMM over the network onto the CFM.

- Time Management Services

  The time distribution and synchronisation between the CFM's is catered for by the Time Management Services.

- Network Management Services

  Network Management Services are needed to allow the CFM's to communicate with the NSM. The configuration data field has been left unspecified so that it does not limit the choice of network technology.

- Power Switches Management Services

  Power Switches Management Services are to provide the means by which the CFM's can communicate with the PCM.

8 Conclusion

The document has justified the contents of the final draft of ASAAC Software Standards for the aspects of Software Components and Software Interfaces.

The coverage of top-level requirements already listed in section 1.3 is provided by Table 2.

Table 2 – TLR Coverage

Number	Components	Interfaces
TLR_2		APOS
TLR_3.1		APOS SMLI SMOS

Number	Components	Interfaces
		SMBP
TLR_3.2		MLI
TLR_3.3	Module Support Layer	MOS SMOS
TLR_3.4	Generic System Management Runtime Blueprints	APOS MOS SMOS SMBP
TLR_3.5	Module Support Layer	MOS SMOS MLI
TLR_8	Operating System Generic System Management Module Support Layer	OLI GLI MLI
TLR_9		OLI GLI SMLI MLI
TLR_10		APOS MOS SMOS SMBP
TLR_11	Operating System	MOS
TLR_12		APOS
TLR_13.1		OLI
TLR_13.2		APOS
TLR_13.3		SMBP
TLR_13.4		APOS SMOS OLI GLI
TLR_14.1		APOS MOS SMBP
TLR_14.2		GLI

Number	Components	Interfaces
TLR_15	Functional Application Application Management Runtime Blueprints	
TLR_16	Runtime Blueprints Generic System Management Operating System	SMBP SMOS GLI
TLR_17	Application Management Generic System Management Runtime Blueprints	SMOS SMBP GLI SMLI

Inside Rear Cover

Defence Standards are Published by and Obtainable from:

Defence Procurement Agency
An Executive Agency of The Ministry of Defence
UK Defence Standardization
Kentigern House
65 Brown Street
GLASGOW G2 8EX

DStan Helpdesk

Tel 0141 224 2531/2
Fax 0141 224 2503
Internet e-mail enquiries@dstan.mod.uk

File Reference

The DStan file reference relating to work on this standard is D/DStan/21/74/2.

Contract Requirements

When Defence Standards are incorporated into contracts users are responsible for their correct application and for complying with contractual and statutory requirements. Compliance with a Defence Standard does not in itself confer immunity from legal obligations.

Revision of Defence Standards

Defence Standards are revised as necessary by up issue or amendment. It is important that users of Defence Standards should ascertain that they are in possession of the latest issue or amendment. Information on all Defence Standards is contained in Def Stan 00-00 Standards for Defence Part 3 , Index of Standards for Defence Procurement Section 4 'Index of Defence Standards and Defence Specifications' published annually and supplemented regularly by Standards in Defence News (SID News). Any person who, when making use of a Defence Standard encounters an inaccuracy or ambiguity is requested to notify the Directorate of Standardization (DStan) without delay in order that the matter may be investigated and appropriate action taken.