

Jambala Application Platform 3.0

OAM SNMP MIB Reference Guide

Introduction

This document describes the standard MIB implementations for the Jambala Application Platform.

Reason for Reissue

This is the first version of the SNMP MIB Definitions.

Trademark/Copyright Information

UNIX is a registered trademark of The Open Group.

Publication Information

This document is a publication of:

Ericsson Canada Inc.
8400 Decarie Boulevard
Town of Mount Royal, Quebec, Canada
H4P 2N2

All rights reserved. Reproduction in whole or in part is prohibited without the written consent of the copyright owner.

The contents of this document are subject to revision without notice due to the continued progress in methodology, design and manufacturing.

Ericsson shall have no liability from any error or damages of any kind resulting from the use of this document.

Revision History

October 2002 Rev A First release of this document

Introduction

Purpose and Scope of the SNMP MIB Descriptions	7
Target Audience	7
Overview	7
Related Documents.....	7

Introduction to SNMP MIBs

Overview	11
Introduction to SNMP MIBs.....	11
SNMP Management Information Base (MIB)	11
Agent and Manager Communication	11

Ericsson Jambala MIB

Ericsson Jambala MIB Defintion	15
--------------------------------------	----

OpenLDAP-MIB

OpenLDAP-MIB Definitions.....	19
-------------------------------	----

Logging MIB

Logging-MIB Definitions	29
-------------------------------	----

Apache MIB

Apache-MIB Definitions.....	35
-----------------------------	----

1

Introduction

Purpose and Scope of the SNMP MIB Descriptions

The purpose of this document is to describe the standard SNMP MIB implementations which represent the configurable parameters of the Apache, JIM LDAP server and logging modules in Node Management.

Target Audience

This document is intended for system and network administrators working with Jambala Node Management. Knowledge of Node Management and SNMP are required.

It is expected that the reader have prior knowledge about telecommunications. It is further assumed that the reader is familiar with concepts, terminology, and abbreviations, for these areas.

Overview

The SNMP MIB Description contains the following sections:

- Introduction to SNMP MIBs
- Definition of the Ericsson Jambala MIB
- OPENLDAP MIB definition
- Logging MIB definition
- Apache MIB definition

Related Documents

- OAM OMToolBox User Guide, 2/1553-APR 901 265 Uen
- OAM Object Browser User Guide, 4/1553-APR 901 265 Uen
- OAM Node Management Toolbox User Guide, 5/1553-APR 901 265 Uen
- OAM OMToolBox System Administration Guide, 1/1543-APR 901 265 Uen

- OAM Node Management System Administration Guide, 2/1543-APR 901 265 Uen
- OAM Reference Guide, 6/1553-APR 901 265 Uen
- Glossary, 0033-APR 901 265 Uen

2

Introduction to SNMP MIBs

Overview

This chapter consists of the following topics:

- Introduction to SNMP MIBS

Introduction to SNMP MIBs

The currently implemented sub-agents specific to TSP are as follows:

- JIM sub-agents to configure JIM managed objects using SNMP.
- I/O configuration sub-agents for Apache, JIM LDAP server and logging.

SNMP Management Information Base (MIB)

A Management Information Base (MIB) is a collection of information that is organized hierarchically. MIBs are accessed using the network-management protocol SNMP. They are comprised of managed objects and are identified by object identifiers. The MIB hierarchy can be depicted as a tree with a nameless root, the levels of which are assigned by different organizations. Vendors can define private branches that include managed objects for their own products. MIBs that have not been standardized typically are positioned in the experimental branch.

Agent and Manager Communication

The SNMP is based on the agent and manager communication protocol over User Datagram Protocol (UDP). The protocol define operations performed by protocol data units (PDUs) on lists of variable bindings.

The operations are as follows:

- get
- get-next
- getbulk

- get-response
- set-request
- trap

These operations may be collected on three kinds of applications:

- **read:** The read operations (get, get-next and getbuk PDUs) are used by an NMS to monitor managed system. The NMS examines different variables that are maintained by the managed system.
- **write:** The write operation (set-request PDU) command is used by an NMS to control managed system. The NMS changes the values of variables stored within managed system.
- **trap:** The trap (trap PDU) command is used by managed system to asynchronously report events to the NMS. When certain types of events occur, a managed system sends a trap to the NMS.

3

Ericsson Jambala MIB

Ericsson Jambala MIB Defintion

This MIB defines the root tree structure of jambala up to the application level. Any MIB defined for the JAMBALA cabinet should include this MIB in its definitions.

ERICSSON-JAMBALA-MIB DEFINITIONS ::= BEGIN

IMPORTS

OBJECT-TYPE FROM RFC1212

enterprises FROM RFC1155-SMI;

ericsson OBJECT IDENTIFIER ::= { enterprises
193 }

telORB OBJECT IDENTIFIER ::= { ericsson 62 }

jambala OBJECT IDENTIFIER ::= { telORB 12 }

applicationOBJECT IDENTIFIER ::= { jambala 1 }

platform OBJECT IDENTIFIER ::= { jambala 2 }

4

OpenLDAP-MIB

OpenLDAP-MIB Definitions

MIB for the configurable parameters of the JIM LDAP Server

```
ERICSSON-OPENLDAP-MIB DEFINITIONS ::= BEGIN
IMPORTS
    OBJECT-TYPE FROM RFC1212
    IPAddress FROM RFC1155-SMI
    platform FROM ERICSSON-JAMBALA-MIB;
DisplayString ::= OCTET STRING
openldap OBJECT IDENTIFIER ::= { platform 1 }
schemacheck OBJECT-TYPE
    SYNTAX  INTEGER { On(1), Off(2) }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "This option turns schema checking on or off. If
        schema checking is on, entries added or modified are
        checked. This ensures that they obey the schema rules
        implied by their object class(es) as defined by the
        corresponding objectclass option(s). If schema
        checking is off this check is not done. Default:
        Schemacheck off."
    ::= { openldap 1 }

schemasTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF SchemasEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "File names of the schemas used by the server."
    ::= { openldap 2 }
```

```
schemasentry OBJECT-TYPE
    SYNTAX  SchemasEntry
    ACCESS   not-accessible
    STATUS   mandatory
    DESCRIPTION
        "Entry corresponding to each schema in the
        schemasTable."
    INDEX { schemasIndex }
    ::= { schemasTable 1 }
```

```
SchemasEntry ::= SEQUENCE {
    schemasIndex
        INTEGER,
    schemas
        DisplayString
}
```

```
schemasIndex OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS   read-only
    STATUS   mandatory
    DESCRIPTION
        "Index for the schemas table"
    ::= { schemasentry 1 }
```

```
schemas OBJECT-TYPE
    SYNTAX  DisplayString
    ACCESS   read-write
    STATUS   mandatory
    DESCRIPTION
        "Schema full path name"
    ::= { schemasentry 2 }
```

referral OBJECT-TYPE

SYNTAX DisplayString

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This option specifies the referral to pass back when slapd cannot find a local database to handle a request. For example, referral ldap:ldap.foo.com. This refers non-local queries to the LDAP server at foo.com. Smart LDAP clients can re-ask their query at this server, but **note** that most of these clients are only going to know how to handle simple LDAP URLs that contain a host part and optionally a distinguished name part."

::= { openldap 3 }

sizelimit OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This option specifies the maximum number of entries to return from a search operation"

::= { openldap 4 }

database OBJECT-TYPE

SYNTAX DisplayString

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Name of the backend used to connect to JIM. The value is 'jim'."

::= { openldap 5 }

suffix OBJECT-TYPE

SYNTAX DisplayString

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Suffix used to connect to any property on jim"

::= { openldap 6 }

server OBJECT IDENTIFIER ::= { openldap 7 }**serveraddress OBJECT-TYPE**

SYNTAX IPAddress

ACCESS read-write

STATUS mandatory

DESCRIPTION

"IP address of the JIM server used as the backend. For load sharing purposes, it should contain the IP address of the internal VirtualIP address of the cabinet"

::= { server 1 }

serverport OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"TCP port number used by the JIM server which the ldap server uses as the backend"

::= { server 2 }

backupServerTable OBJECT-TYPE

SYNTAX SEQUENCE OF IpAndPortAddressEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"Table of redundant JIM servers. If the main server is not accessible, the ldap server will try to connect to this server. The first entry should contain the secondary address of the internal VirtualIP."

::= { openldap 8 }

backupserverentry OBJECT-TYPE

SYNTAX IpAndPortAddressEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"Entry definition for each redundant JIM server"

INDEX { serverIndex }

::= { backupServerTable 1 }

IpAndPortAddressEntry ::= SEQUENCE {

serverIndex

INTEGER,

address

IpAddress,

port

INTEGER

}

serverIndex OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Backup JIM Server Index"

::= { backupserverentry 1 }

address OBJECT-TYPE

SYNTAX IpAddress

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Backup JIM Server address"

::= { backupserverentry 2 }

port OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Backup JIM Server port"

::= { backupserverentry 3 }

reconnectperiod OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Time in seconds before trying to reconnect to the
main JIM server address."

::= { openldap 9 }

OpenLdapAction OBJECT-TYPE

SYNTAX DisplayString

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Process control command line parameters, used to
start, restart and stop the JIM Ldap server."

::= { openldap 9999 }

END

5

Logging MIB

Logging-MIB Definitions

MIB for the logging module configurable parameters in Jambala.

ERICSSON-LOGGING-MIB DEFINITIONS ::= BEGIN

IMPORTS

OBJECT-TYPE FROM RFC1212

platform FROM ERICSSON-JAMBALA-MIB;

DisplayString ::= OCTET STRING

logging OBJECT IDENTIFIER ::= { platform 3 }

general OBJECT IDENTIFIER ::= { logging 1 }

loggingArchiver OBJECT IDENTIFIER ::= { logging 2 }

loggingDBTableCreator OBJECT IDENTIFIER ::= {
 logging 3 }

loggingDBWriter OBJECT IDENTIFIER ::= { logging 4 }

queryTool OBJECT IDENTIFIER ::= { logging 5 }

applogClient OBJECT IDENTIFIER ::= { logging 6 }

LoggingDatabase OBJECT-TYPE

SYNTAX DisplayString

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Name of the database where logged entries are
persisted."

::= { general 1 }

LoggingDBUserName OBJECT-TYPE

SYNTAX DisplayString

ACCESS read-write

STATUS mandatory

DESCRIPTION

"User name used by the different logging modules to access the log database"

::= { general 2 }

LoggingDBPassword OBJECT-TYPE

SYNTAX DisplayString

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Password used by the different logging modules to access the log database"

::= { general 3 }

LoggingDatabaseBackup OBJECT-TYPE

SYNTAX DisplayString

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Name of the database used to restore archived logged entries"

::= { general 4 }

nbOfDaysBeforeArchive OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Number of days used as a criteria to archive logged entries. Entries older than number will be archived."

::= { loggingArchiver 1 }

HourTimeArchive OBJECT-TYPE

SYNTAX INTEGER (0..24)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Hour of the day at which the archive process should take place."

::= { loggingArchiver 2 }

MinuteTimeArchive OBJECT-TYPE

SYNTAX INTEGER (0..60)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Minute of the hour at which the archive process should take place."

::= { loggingArchiver 3 }

LoggingDBWriterSleepTime OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Delay between polling of raw text format log files (files generated by ApplogClient)."

::= { loggingDBWriter 1 }

DefaultMaxResults OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Default value for the number of results per page displayed after each query."

```
::= { queryTool 1 }
```

MAX_DIGIT_FILE OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Maximum number of raw text files created before the first file is re-used."

```
::= { applogClient 1 }
```

SIZE_OF_FILES OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Maximum size of each raw text file."

```
::= { applogClient 2 }
```

LoggingAction OBJECT-TYPE

SYNTAX DisplayString

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Process control command line parameters, used to start, restart and stop logging processes."

```
::= { logging 9999 }
```

END

6

Apache MIB

Apache-MIB Definitions

MIB definitions for Apache configurable parameters in Jambala.

ERICSSON-APACHE-MIB DEFINITIONS ::= BEGIN

IMPORTS

OBJECT-TYPE FROM RFC1212

platform FROM ERICSSON-JAMBALA-MIB;

DisplayString ::= OCTET STRING

apache OBJECT IDENTIFIER ::= { platform 2 }

startServers OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Number of servers to start initially. It should be a reasonable approximate figure."

::= { apache 1 }

maxClients OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Limit on total number of servers running. For example, it indicates the limit on the number of clients who can simultaneously connect. If this limit is ever reached, clients are locked out. Therefore, it should not be set too low. The main goal of the limit is to impede a runaway server from taking down the system with it."

::= { apache 2 }

keepAlive OBJECT-TYPE

SYNTAX INTEGER { On(1), Off(2) }

ACCESS read-write

STATUS mandatory

DESCRIPTION

"KeepAlive is defined as whether or not to allow persistent connections (more than one request per connection). Set to Off(2) to deactivate."

::= { apache 3 }

keepAliveTimeout OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"KeepAliveTimeout is defined as the number of seconds to wait for the next request from the same client on the same connection."

::= { apache 4 }

ApacheAction OBJECT-TYPE

SYNTAX DisplayString

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Process control command line parameters, used to start, restart and stop apache processes."

::= { apache 9999 }

END

M

Management Information Base (MIB) 11

MIB hierarchy 11

P

Protocol data units 11

T

Trap 12

U

User Datagram Protocol 11