Mail Monitoring MIB

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Table of Contents

1. Introduction .................................................. 2
2. The SNMPv2 Network Management Framework ...................... 2
2.1 Object Definitions ........................................... 2
3. Message Flow Model ............................................. 3
4. MTA Objects .................................................... 3
5. Definitions .................................................... 4
6. Acknowledgements ............................................. 19
7. References ..................................................... 19
8. Security Considerations ....................................... 19
9. Authors’ Addresses .......................................... 20
1. Introduction

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, this memo extends the basic Network Services Monitoring MIB [5] to allow monitoring of Message Transfer Agents (MTAs). It may also be used to monitor MTA components within gateways.

2. The SNMPv2 Network Management Framework

The SNMPv2 Network Management Framework consists of four major components. They are:

- RFC 1442 [1] which defines the SMI, the mechanisms used for describing and naming objects for the purpose of management.
- STD 17, RFC 1213 [2] defines MIB-II, the core set of managed objects for the Internet suite of protocols.
- RFC 1445 [3] which defines the administrative and other architectural aspects of the framework.
- RFC 1448 [4] which defines the protocol used for network access to managed objects.

The Framework permits new objects to be defined for the purpose of experimentation and evaluation.

2.1 Object Definitions

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the subset of Abstract Syntax Notation One (ASN.1) defined in the SMI. In particular, each object type is named by an OBJECT IDENTIFIER, an administratively assigned name. The object type together with an object instance serves to uniquely identify a specific instantiation of the object. For human convenience, we often use a textual string, termed the descriptor, to refer to the object type.
3. Message Flow Model

A general model of message flow inside an MTA has to be presented before a MIB can be described. Generally speaking, message flow occurs in four steps:

1. Messages are received by the MTA from User Agents, Message Stores, other MTAs, and gateways.

2. The "next hop" for each message is determined. This is simply the destination the message is to be transmitted to; it may or may not be the final destination of the message. Multiple "next hops" may exist for a single message (as a result of either having multiple recipients or distribution list expansion); this may make it necessary to duplicate messages.

3. Messages are converted into the format that’s appropriate for the next hop.

4. Messages are transmitted to the appropriate destination, which may be a User Agent, Message Store, another MTA, or gateway.

Storage of messages in the MTA occurs at some point during this process. However, it is important to note that storage may occur at different and possibly even multiple points during this process. For example, some MTAs expand messages into multiple copies as they are received. In this case (1), (2), and (3) may all occur prior to storage. Other MTAs store messages precisely as they are received and perform all expansions and conversions during retransmission processing. So here only (1) occurs prior to storage. This leads to situations where, in general, a measurement of messages received may not equal a measurement of messages in store, or a measurement of messages stored may not equal a measurement of messages retransmitted, or both.

4. MTA Objects

If there are one or more MTAs on the host, the following mta group may be used to monitor them. Any number of the MTAs on a host may be monitored. Each MTA is dealt with as a separate application and has its own applTable entry in the Network Services Monitoring MIB.

The MIB described in this document covers only the portion which is specific to the monitoring of MTAs. The network service related part of the MIB is covered in a separate document [5].
5. Definitions

MTA-MIB DEFINITIONS ::= BEGIN

IMPORTS
   OBJECT-TYPE, Counter32, Gauge32
   FROM SNMPv2-SMI
   DisplayString, TimeInterval
   FROM SNMPv2-TC
   mib-2
   FROM RFC1213-MIB
   applIndex
   FROM APPLICATION-MIB;

mta MODULE-IDENTITY
   LAST-UPDATED "9311280000Z"
   ORGANIZATION "IETF Mail and Directory Management Working Group"
   CONTACT-INFO
   " Ned Freed
     Postal: Innosoft International, Inc.
     250 West First Street, Suite 240
     Claremont, CA  91711
     US
     Tel: +1 909 624 7907
     Fax: +1 909 621 5319
     E-Mail: ned@innosoft.com"
   DESCRIPTION
   "The MIB module describing Message Transfer Agents (MTAs)"
   ::= { mib-2 28 }

mtaTable OBJECT-TYPE
   SYNTAX  SEQUENCE OF MtaEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
   "The table holding information specific to an MTA."
   ::= { mta 1 }

mtaEntry OBJECT-TYPE
   SYNTAX  MtaEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
   "The entry associated with each MTA."
   INDEX {applIndex}
MtaEntry ::= SEQUENCE {
    mtaReceivedMessages  
        Counter32,
    mtaStoredMessages    
        Gauge32,
    mtaTransmittedMessages  
        Counter32,
    mtaReceivedVolume     
        Counter32,
    mtaStoredVolume       
        Gauge32,
    mtaTransmittedVolume  
        Counter32,
    mtaReceivedRecipients 
        Counter32,
    mtaStoredRecipients   
        Gauge32,
    mtaTransmittedRecipients  
        Counter32
}

mtaReceivedMessages OBJECT-TYPE
    SYNTAX   Counter32
    MAX-ACCESS read-only
    STATUS   current
    DESCRIPTION
        "The number of messages received since MTA initialization."
    ::= {mtaEntry 1}

mtaStoredMessages OBJECT-TYPE
    SYNTAX   Gauge32
    MAX-ACCESS read-only
    STATUS   current
    DESCRIPTION
        "The total number of messages currently stored in the MTA."
    ::= {mtaEntry 2}

mtaTransmittedMessages OBJECT-TYPE
    SYNTAX   Counter32
    MAX-ACCESS read-only
    STATUS   current
    DESCRIPTION
        "The number of messages transmitted since MTA initialization."
    ::= {mtaEntry 3}
mtaReceivedVolume OBJECT-TYPE
SYNTAX Counter32
UNITS "K-octets"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The total volume of messages received since MTA
 initialization, measured in kilo-octets. This volume should
 include all transferred data that is logically above the mail
 transport protocol level. For example, an SMTP-based MTA
 should use the number of kilo-octets in the message header
 and body, while an X.400-based MTA should use the number of
 kilo-octets of P2 data."
 ::= {mtaEntry 4}

mtaStoredVolume OBJECT-TYPE
SYNTAX Gauge32
UNITS "K-octets"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The total volume of messages currently stored in the MTA,
 measured in kilo-octets. This volume should include all
 stored data that is logically above the mail transport
 protocol level. For example, an SMTP-based MTA should
 use the number of kilo-octets in the message header and
 body, while an X.400-based MTA would use the number of
 kilo-octets of P2 data."
 ::= {mtaEntry 5}

mtaTransmittedVolume OBJECT-TYPE
SYNTAX Counter32
UNITS "K-octets"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The total volume of messages transmitted since MTA
 initialization, measured in kilo-octets. This volume should
 include all transferred data that is logically above the mail
 transport protocol level. For example, an SMTP-based MTA
 should use the number of kilo-octets in the message header
 and body, while an X.400-based MTA should use the number of
 kilo-octets of P2 data."
 ::= {mtaEntry 6}
mtaReceivedRecipients OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The total number of recipients specified in all messages
received since MTA initialization. Recipients this MTA
had no responsibility for should not be counted even if
information about such recipients is available."
::= {mtaEntry 7}

mtaStoredRecipients OBJECT-TYPE
SYNTAX Gauge32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The total number of recipients specified in all messages
currently stored in the MTA. Recipients this MTA had no
responsibility for should not be counted."
::= {mtaEntry 8}

mtaTransmittedRecipients OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The total number of recipients specified in all messages
transmitted since MTA initialization. Recipients this MTA
had no responsibility for should not be counted."
::= {mtaEntry 9}

-- MTAs typically group inbound reception, queue storage, and
-- outbound transmission in some way. In the most extreme case
-- information will be maintained for each different entity that
-- receives messages and for each entity the MTA stores messages for
-- and delivers messages to. Other MTAs may elect to treat all
-- reception equally, all queue storage equally, all deliveries
-- equally, or some combination of this.

-- In any case, a grouping abstraction is an extremely useful for
-- breaking down the activities of an MTA. For purposes of labelling
-- this will be called a "group" in this MIB.
-- Each group contains all the variables needed to monitor all aspects
-- of an MTA’s operation. However, the fact that all groups contain
-- all possible variables does not imply that all groups must use all
-- possible variables. For example, a single group might be used to
-- monitor only one kind of event (inbound processing, outbound
-- processing, or storage). In this sort of configuration all unused
-- counters would be inaccessible; e.g., returning either a
-- noSuchName error (for an SNMPv1 get), or a noSuchInstance
-- exception (for an SNMPv2 get).

-- Groups are not necessarily mutually exclusive. A given event may
-- be recorded by more than one group, a message may be seen as
-- stored by more than one group, and so on. Groups should be all
-- inclusive, however: if groups are implemented all aspects of an
-- MTA’s operation should be registered in at least one group. This
-- freedom lets implementors use different sets of groups to
-- provide different "views" of an MTA.

-- The possibility of overlap between groups means that summing
-- variables across groups may not produce values equal to those in
-- the mtaTable. mtaTable should always provide accurate information
-- about the MTA as a whole.

-- The term "channel" is often used in MTA implementations; channels
-- are usually, but not always, equivalent to a group. However,
-- this MIB does not use the term "channel" because there is no
-- requirement that an MTA supporting this MIB has to map its
-- "channel" abstraction one-to-one onto the MIB’s group abstraction.

mtaGroupTable OBJECT-TYPE
SYNTAX SEQUENCE OF MtaGroupEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The table holding information specific to each MTA group."
::= {mta 2}

mtaGroupEntry OBJECT-TYPE
SYNTAX MtaGroupEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The entry associated with each MTA group."
INDEX {applIndex, mtaGroupIndex}
::= {mtaGroupTable 1}
MtaGroupEntry ::= SEQUENCE {
    mtaGroupIndex
        INTEGER,
    mtaGroupReceivedMessages
        Counter32,
    mtaGroupRejectedMessages
        Counter32,
    mtaGroupStoredMessages
        Gauge32,
    mtaGroupTransmittedMessages
        Counter32,
    mtaGroupReceivedVolume
        Counter32,
    mtaGroupStoredVolume
        Gauge32,
    mtaGroupTransmittedVolume
        Counter32,
    mtaGroupReceivedRecipients
        Counter32,
    mtaGroupStoredRecipients
        Gauge32,
    mtaGroupTransmittedRecipients
        Counter32,
    mtaGroupOldestMessageStored
        TimeInterval,
    mtaGroupInboundAssociations
        Gauge32,
    mtaGroupOutboundAssociations
        Gauge32,
    mtaGroupAccumulatedInboundAssociations
        Counter32,
    mtaGroupAccumulatedOutboundAssociations
        Counter32,
    mtaGroupLastInboundActivity
        TimeInterval,
    mtaGroupLastOutboundActivity
        TimeInterval,
    mtaGroupRejectedInboundAssociations
        Counter32,
    mtaGroupFailedOutboundAssociations
        Counter32,
    mtaGroupInboundRejectionReason
        DisplayString,
    mtaGroupOutboundConnectFailureReason
        DisplayString,
    mtaGroupScheduledRetry
        TimeInterval,
    mtaGroupMailProtocol
}
OBJECT IDENTIFIER,
mtaGroupName
    DisplayString

mtaGroupIndex OBJECT-TYPE
    SYNTAX INTEGER (1..2147483647)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The index associated with a group for a given MTA."
    ::= {mtaGroupEntry 1}

mtaGroupReceivedMessages OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The number of messages received to this group since MTA initialization."
    ::= {mtaGroupEntry 2}

mtaGroupRejectedMessages OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The number of messages rejected by this group since MTA initialization."
    ::= {mtaGroupEntry 3}

mtaGroupStoredMessages OBJECT-TYPE
    SYNTAX Gauge32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The total number of messages currently stored in this group’s queue."
    ::= {mtaGroupEntry 4}

mtaGroupTransmittedMessages OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The number of messages transmitted by this group since MTA initialization."
    ::= {mtaGroupEntry 5}
mtaGroupReceivedVolume OBJECT-TYPE
SYNTAX Counter32
UNITS "K-octets"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The total volume of messages received to this group since MTA initialization, measured in kilo-octets. This volume should include all transferred data that is logically above the mail transport protocol level. For example, an SMTP-based MTA should use the number of kilo-octets in the message header and body, while an X.400-based MTA should use the number of kilo-octets of P2 data."
::= {mtaGroupEntry 6}

mtaGroupStoredVolume OBJECT-TYPE
SYNTAX Gauge32
UNITS "K-octets"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The total volume of messages currently stored in this group’s queue, measured in kilo-octets. This volume should include all stored data that is logically above the mail transport protocol level. For example, an SMTP-based MTA should use the number of kilo-octets in the message header and body, while an X.400-based MTA would use the number of kilo-octets of P2 data."
::= {mtaGroupEntry 7}

mtaGroupTransmittedVolume OBJECT-TYPE
SYNTAX Counter32
UNITS "K-octets"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The total volume of messages transmitted by this group since MTA initialization, measured in kilo-octets. This volume should include all transferred data that is logically above the mail transport protocol level. For example, an SMTP-based MTA should use the number of kilo-octets in the message header and body, while an X.400-based MTA should use the number of kilo-octets of P2 data."
::= {mtaGroupEntry 8}
mtaGroupReceivedRecipients OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The total number of recipients specified in all messages received to this group since MTA initialization. Recipients this MTA had no responsibility for should not be counted."
::= {mtaGroupEntry 9}

mtaGroupStoredRecipients OBJECT-TYPE
SYNTAX Gauge32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The total number of recipients specified in all messages currently stored in this group’s queue. Recipients this MTA had no responsibility for should not be counted."
::= {mtaGroupEntry 10}

mtaGroupTransmittedRecipients OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The total number of recipients specified in all messages transmitted by this group since MTA initialization. Recipients this MTA had no responsibility for should not be counted."
::= {mtaGroupEntry 11}

mtaGroupOldestMessageStored OBJECT-TYPE
SYNTAX TimeInterval
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Time since the oldest message in this group’s queue was placed in the queue."
::= {mtaGroupEntry 12}
mtaGroupInboundAssociations OBJECT-TYPE
SYNTAX Gauge32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of current associations to the group, where the
group is the responder."
::= {mtaGroupEntry 13}

mtaGroupOutboundAssociations OBJECT-TYPE
SYNTAX Gauge32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of current associations to the group, where the
group is the initiator."
::= {mtaGroupEntry 14}

mtaGroupAccumulatedInboundAssociations OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The total number of associations to the group since MTA
initialization, where the group is the responder."
::= {mtaGroupEntry 15}

mtaGroupAccumulatedOutboundAssociations OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The total number of associations from the group since MTA
initialization, where the group was the initiator."
::= {mtaGroupEntry 16}

mtaGroupLastInboundActivity OBJECT-TYPE
SYNTAX TimeInterval
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Time since the last time that this group had an active
inbound association for purposes of message reception."
::= {mtaGroupEntry 17}
mtaGroupLastOutboundActivity OBJECT-TYPE
   SYNTAX TimeInterval
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
     "Time since the last time that this group had an
      outbound association for purposes of message delivery."
   ::= {mtaGroupEntry 18}

mtaGroupRejectedInboundAssociations OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
     "The total number of inbound associations the group has
      rejected, since MTA initialization."
   ::= {mtaGroupEntry 19}

mtaGroupFailedOutboundAssociations OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
     "The total number associations where the group was the
      initiator and association establishment has failed,
      since MTA initialization."
   ::= {mtaGroupEntry 20}

mtaGroupInboundRejectionReason OBJECT-TYPE
   SYNTAX DisplayString
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
     "The failure reason, if any, for the last association this
      group refused to respond to. An empty string indicates that
      the last attempt was successful. If no association attempt
      has been made since the MTA was initialized the value
      should be 'never'."
   ::= {mtaGroupEntry 21}
mtaGroupOutboundConnectFailureReason OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS read-only
STATUS current
DESCRIPTION
   "The failure reason, if any, for the last association attempt
this group initiated. An empty string indicates that the last
attempt was successful. If no association attempt has been
made since the MTA was initialized the value should be
'never'."
 ::= {mtaGroupEntry 22}

mtaGroupScheduledRetry OBJECT-TYPE
SYNTAX TimeInterval
MAX-ACCESS read-only
STATUS current
DESCRIPTION
   "The time when this group is scheduled to next attempt to
make an association."
 ::= {mtaGroupEntry 23}

mtaGroupMailProtocol OBJECT-TYPE
SYNTAX OBJECT IDENTIFIER
MAX-ACCESS read-only
STATUS current
DESCRIPTION
   "An identification of the protocol being used by this group.
For an group employing OSI protocols, this will be the
Application Context. For Internet applications, the IANA
maintains a registry of the OIDs which correspond to well-known
message transfer protocols. If the application protocol is
not listed in the registry, an OID value of the form
(applTCPProtoID port) or (applUDPProtoID port) are used for
TCP-based and UDP-based protocols, respectively. In either
case 'port' corresponds to the primary port number being
used by the group. applTCPProtoID and applUDPProtoID are
defined in [5]."
 ::= {mtaGroupEntry 24}
mtaGroupName OBJECT-TYPE
   SYNTAX DisplayString
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
   "A descriptive name for the group. If this group connects to
   a single remote MTA this should be the name of that MTA. If
   this in turn is an Internet MTA this should be the domain name.
   For an OSI MTA it should be the string encoded distinguished
   name of the managed object using the format defined in RFC-1485.
   For X.400(1984) MTAs which do not have a Distinguished Name,
   the RFC-1327 syntax 'mta in globalid' should be used."
   ::= {mtaGroupEntry 25}

mtaGroupAssociationTable OBJECT-TYPE
   SYNTAX SEQUENCE OF MtaGroupAssociationEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
   "The table holding information regarding the associations
   for each MTA group."
   ::= {mta 3}

MtaGroupAssociationEntry ::= SEQUENCE {
   mtaGroupAssociationIndex
   INTEGER
   }

mtaGroupAssociationIndex OBJECT-TYPE
   SYNTAX INTEGER (1..2147483647)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
   "Reference into association table to allow correlation of
   this group’s active associations with the association table."
   ::= {mtaGroupAssociationEntry 1}
-- Conformance information

mtaConformance OBJECT IDENTIFIER ::= {mta 4}

mtaGroups OBJECT IDENTIFIER ::= {mtaConformance 1}
mtaCompliances OBJECT IDENTIFIER ::= {mtaConformance 2}

-- Compliance statements

mtaCompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION
   "The compliance statement for SNMPv2 entities which
    implement the Mail Monitoring MIB for basic
    monitoring of MTAs."
   MODULE -- this module
      MANDATORY-GROUPS {mtaGroup}
      ::= {mtaCompliances 1}

mtaAssocCompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION
   "The compliance statement for SNMPv2 entities which
    implement the Mail Monitoring MIB for monitoring of
    MTAs and their associations."
   MODULE -- this module
      MANDATORY-GROUPS {mtaGroup, mtaAssocGroup}
      ::= {mtaCompliances 2}
-- Units of conformance

mtaGroup OBJECT-CLASS
  OBJECTS {
    mtaReceivedMessages, mtaStoredMessages,
    mtaTransmittedMessages, mtaReceivedVolume, mtaStoredVolume,
    mtaTransmittedVolume, mtaReceivedRecipients,
    mtaReceivedRecipient, mtaTransmittedRecipient,
    mtaGroupReceivedMessages, mtaGroupRejectedMessages,
    mtaGroupStoredMessages, mtaGroupTransmittedMessages,
    mtaGroupReceivedVolume, mtaGroupStoredVolume,
    mtaGroupTransmittedVolume, mtaGroupReceivedRecipients,
    mtaGroupReceivedRecipient, mtaGroupTransmittedRecipient,
    mtaGroupOldestMessageStored, mtaGroupInboundAssociations,
    mtaGroupOutboundAssociations,
    mtaGroupAccumulatedInboundAssociations,
    mtaGroupAccumulatedOutboundAssociations,
    mtaGroupLastInboundActivity, mtaGroupLastOutboundActivity,
    mtaGroupRejectedInboundAssociations,
    mtaGroupFailedOutboundAssociations,
    mtaGroupRecentRejectionReason,
    mtaGroupOutboundConnectFailureReason,
    mtaGroupScheduledRetry, mtaGroupMailProtocol, mtaGroupName
  }
  STATUS current
  DESCRIPTION
    "A collection of objects providing basic monitoring of MTAs."
  ::= {mtaGroups 1}

mtaAssocGroup OBJECT-CLASS
  OBJECTS {
    mtaGroupAssociationIndex
  }
  STATUS current
  DESCRIPTION
    "A collection of objects providing monitoring of MTA associations."
  ::= {mtaGroups 2}

END
6. Acknowledgements

This document is a product of the Mail and Directory Management (MADMAN) Working Group. It is based on an earlier MIB designed by S. Kille, T. Lenggenhager, D. Partain, and W. Yeong.

7. References


8. Security Considerations

Security issues are not discussed in this memo.
9. Authors' Addresses

Steve Kille, WG Chair
ISODE Consortium
The Dome, The Square
Richmond TW9 1DT
UK

Phone: +44 81 332 9091
EMail: S.Kille@isode.com

Ned Freed, Editor
Innosoft International, Inc.
250 West First Street, Suite 240
Claremont, CA 91711
USA

Phone: +1 909 624 7907
Fax: +1 909 621 5319
EMail: ned@innosoft.com