

PowerSNMP for .NET 4.6 Upgrade Guide

Overview:

This minor version release includes interface changes that improve core functionality:

1. Manager Inform requests have been generalized for use by Agents.
2. Some events and properties were moved from the Agent and Manager classes into a common base class to extend functionality. For example the Log event is no available on the Agent class.
3. Some usability issues associated with naming were addressed. For example, TrapMessage is now Trap1Message, NotificationMessage is now Trap2Message.
4. Added single-threaded support so simple console and service applications can be built using blocking communications.
5. Security features have been enhanced for SNMPv3.

Feature Enhancements:

1. Traps - Added ability to send traps from Agent without starting primary Agent functionality first (Agent.Send() without Agent.Start()).
2. Informs - Added support for sending Informs from an Agent to a Manager.
3. Log event added to Agent (already existed in Manager) to provide robust logging of all messages.
4. Scripting support added to Agent (messages from managers can now be processed synchronously using a single thread).
5. Scripting support added to Manager (traps and informs can now be processed synchronously using a single thread).
6. Usability:
 1. Version 1 TrapMessage class renamed to Trap1Message to better indicate relationship with Trap2Message.
 2. Version 2/3 NotificationMessage class renamed to Trap2Message to indicate functional similarities with Trap1Message.
 3. Dart.Snmp.Trap namespace renamed Dart.Snmp.Trap1 to better indicate relationship with Dart.Snmp.Trap2.
 4. Dart.Snmp.Notification namespace renamed Dart.Snmp.Trap2 to better indicate relationship with Dart.Snmp.Trap1.
 5. ManagerSlave class renamed and refactored SnmpSocket to generalize it so Agent can use it for sending Informs.

6. Agent.CreateDefaultResponse deprecated and renamed to SnmpBase.CreateResponse.
 7. Password class changed to User class, with the addition of a Name property.
 8. MibNodes.Populate() - removed unused 2nd parameter.
 9. Manager.Security.TrapUsers is now used to contain user details for SNMPv3 trap decoding, instead of Manager.AgentEngines.
 10. SysUpTime moved from MessageBase to Agent, and changed from a static to instance variable to support multiple agents in a single application. This prevented us from keeping TrapMessage and NotificationMessage as deprecated classes in this release.
7. Security properties (Username, AuthenticationPassword, AuthenticationPassword, PrivacyPassword, PrivacyProtocol) grouped together as a single User property.
 1. Enhances encryption performance by supporting caching of encrypted intermediate results.
 2. Links use of AuthoritativeEngines to user identification and processing.
 3. Optimizes specification and reuse of security parameters.
 8. Security decoding operations have been relaxed to provide a decoded message whenever possible.
 1. Previously, the username included in a v3 packet was required to be in the AuthoritativeEngine or a remote AuthoritativeEngine, or a DecodingException was thrown. Now, no username check is made. The decoded Message.Security.User.Name property may be checked for a valid value.
 2. Previously, if the message did not pass the authentication check it would throw a DecodingException. Now, no DecodingException is ever thrown if authentication fails. The new Message.AuthenticatedState property indicates whether the authentication process passed.
 3. Consequently, only decryption failures will typically cause a DecodingException. The username from the packet is included in the DecodingException to help trouble-shoot configuration problems.

Guidelines for Upgrading from version 4.5.

Note: Although this list is not short, most applications will only need slight modifications.

1. Replace "Dart.Snmp.Notification" namespace references with "Dart.Snmp.Trap2".
2. NotificationMessage class deprecated. Replace with Trap2Message.
3. Dart.Snmp.SlaveThreadStart renamed Dart.Snmp.ThreadStart, and the first parameter changed from type ManagerSlave to SnmpSocket.

4. `Manager.Start(new SlaveThreadStart(userFunction) ...)` changed to `Manager.Start(new Dart.Snmp.ThreadStart(userFunction) ...)`.
5. `ManagerSlave` removed and replaced with `SnmpSocket` that derives from `Socket` base class.
6. `Manager.AgentEngines` deprecated. Replace with `SnmpBase.Security.EngineCache`.
 - Used to cache SNMPv3 security information (`EngineId`, `EngineBoots`, `EngineTime`) for `InformMessage` requests sent by either `Manager` or `Agent` to other managers.
7. `Agent.Start(ThreadStart ...)` added. Use for sending `InformMessage` request to a `Manager`.
8. `ManagerSlave.Manager` property (of type `Manager`) changed to `SnmpSocket.Entity` (of type `SnmpBase`).
9. Replace references to `SnmpBase.AuthoritativeEngine` with `SnmpBase.Security.AuthoritativeEngine`.

Summary of Benefits:

1. Previously, `Agent.Start()` had to be used to create a `Socket` before `Agent.Send()` could be used to send a trap. Now, `Agent.Send()` can be used to send a trap (a `Socket` is automatically created for use).
2. Previously, there was no `Agent` support for sending an `InformMessage` from an `Agent` to a `Manager`
 1. `Manager.Start()` functionality has been moved to the common base class `SnmpBase`, so `Agent.Start()` can be used the same way `Manager.Start()` works.
 2. `ManagerThreadStart` delegate renamed to `ThreadStart` delegate to reflect its newly generalized applicability to `Agents`.
 3. `ManagerSlave` class renamed `SnmpSocket` to reflect its newly generalized applicability to `Agents` (additionally, derived `SnmpSocket` from `Socket` and removed `Socket` property to better indicate the functionality of the class).
3. Previously, `TrapMessage` indicated a version 1 trap and `Notification` indicated a version 2 trap. This naming convention caused customer confusion. Renamed `Trap` version 1 to `Trap1` and `Notification` to `Trap2` (applied to message classes and namespaces).
4. Previously, `Manager.AgentEngines` was used to cache remote engine information like `EngineId`, `Reboots`, and `Uptime`.
 1. Deprecated `Manager.AgentEngines` in favor of new `SnmpBase.Security.EngineCache` to reflect its newly generalized applicability to `Agents` (supporting `Inform` requests), and its use sending `Inform` requests to other managers.
 2. Moved `SnmpBase.AuthoritativeEngine` to `SnmpBase.Security.AuthoritativeEngine`.
 3. Added new `SnmpBase.Security.TrapUsers` list that better supports the decoding of SNMPv3 traps.

4. SnmpBase.Security, of type SessionSecurity, can now be used to easily serialize all session parameters, preserving all values found during "discovery" and password caching.

Interface Tables:

The following tables map the interface changes required to upgrade from a previous version of PowerSNMP for .NET to PowerSNMP for .NET 4.6.

Namespace

Previous	New
Dart.Snmp.Trap	Dart.Snmp.Trap1
Dart.Snmp.Notification	Dart.Snmp.Trap2

Classes

Previous	New
ManagerSlave	SnmpSocket
NotificationMessage (removed due to structural changes)	Trap2Message
Passwords	User (or the new TrapUser for decoding traps – See Manager.Security.TrapUser)
TrapMessage (removed due to structural changes)	Trap1Message

Delegates

Previous	New
ManagerMessageReceived	NotificationReceived
SlaveThreadStart	ThreadStart

Agent Class Members

Previous	New
Agent.AuthoritativeEngine (deprecated)	Agent.Security.AuthoritativeEngine
Agent.CreateDefaultResponse() (deprecated)	Agent.CreateResponse()
Agent.Send() (deprecated)	Agent.Send() (inherited from SnmpBase)
Agent.Send(Messagebase, String)	Agent.Send(MessageBase, new Dart.Snmp.IPEndPoint (String, Manager.DefaultPort))
Agent.Send(Messagebase, String, Int32)	Agent.Send(MessageBase, new Dart.Snmp.IPEndPoint(String, Int32))

Manager Class Members

Previous	New
Manager.AgentEngines (deprecated)	Manager.Security.EngineCache
Manager.AuthoritativeEngine (deprecated)	Manager.Security.AuthoritativeEngine
Manager.Start(SlaveThreadStart...)	Manager.Start(ThreadStart...)
Manager.Start (ManagerMessageReceived...)	Manager.Start(NotificationReceived...)
Manager.Log	Manager.Log (inherited from SnmpBase)

ManagerSlave Class Members

Previous	New
ManagerSlave.Manager	SnmpSocket.Entity

MibTraps Class Members

Previous	New
MibTraps.CreateGenericTrapMessage (Dart.Snmp.TrapName, string)	MibTraps.CreateGenericTrapMessage (Dart.Snmp.TrapName, string, long)*
MibTraps.CreateSpecificTrapMessage(Enum)	MibTraps.CreateSpecificTrapMessage(Enum, long)*

*Use Agent.SysUpTime for the last argument

Security Class Members

Previous	New
Security.AuthenticationPassword	Security.User.AuthenticationPassword
Security.AuthenticationProtocol	Security.User.AuthenticationProtocol
Security.PrivacyPassword	Security.User.PrivacyPassword
Security.PrivacyProtocol	Security.User.PrivacyProtocol
Security.Username	Security.User.Name

Trap Message Class Members

Previous	New
Trap.coldStart(string)	Trap1.coldStart(string, long)*
Trap.warmStart(string)	Trap1.warmStart(string, long)*
Trap.linkDown(string)	Trap1.linkDown(string, long)*
Trap.linkUp(string)	Trap1.linkUp(string, long)*
Trap.authenticationFailure(string)	Trap1.authenticationFailure (string, long)*
Trap.egpNeighborLoss(string)	Trap1.egpNeighborLoss(string, long)*
Trap.SpecificTrap(int, string)	Trap1.SpecificTrap(int, string, long)*
Trap.Encode()	Removed

*Use Agent.SysUpTime for the last argument

Notification Class Members and Methods

Previous	New
Notification.coldStart()	Trap2.coldStart(long)*
Notification.warmStart()	Trap2.warmStart(long)*
Notification.authenticationFailure()	Trap2.authenticationFailure(long)*
Notification.Encode()	Removed

* Use Agent.SysUpTime for the last argument