Definitions of Managed Objects for the Delegation of Management Scripts

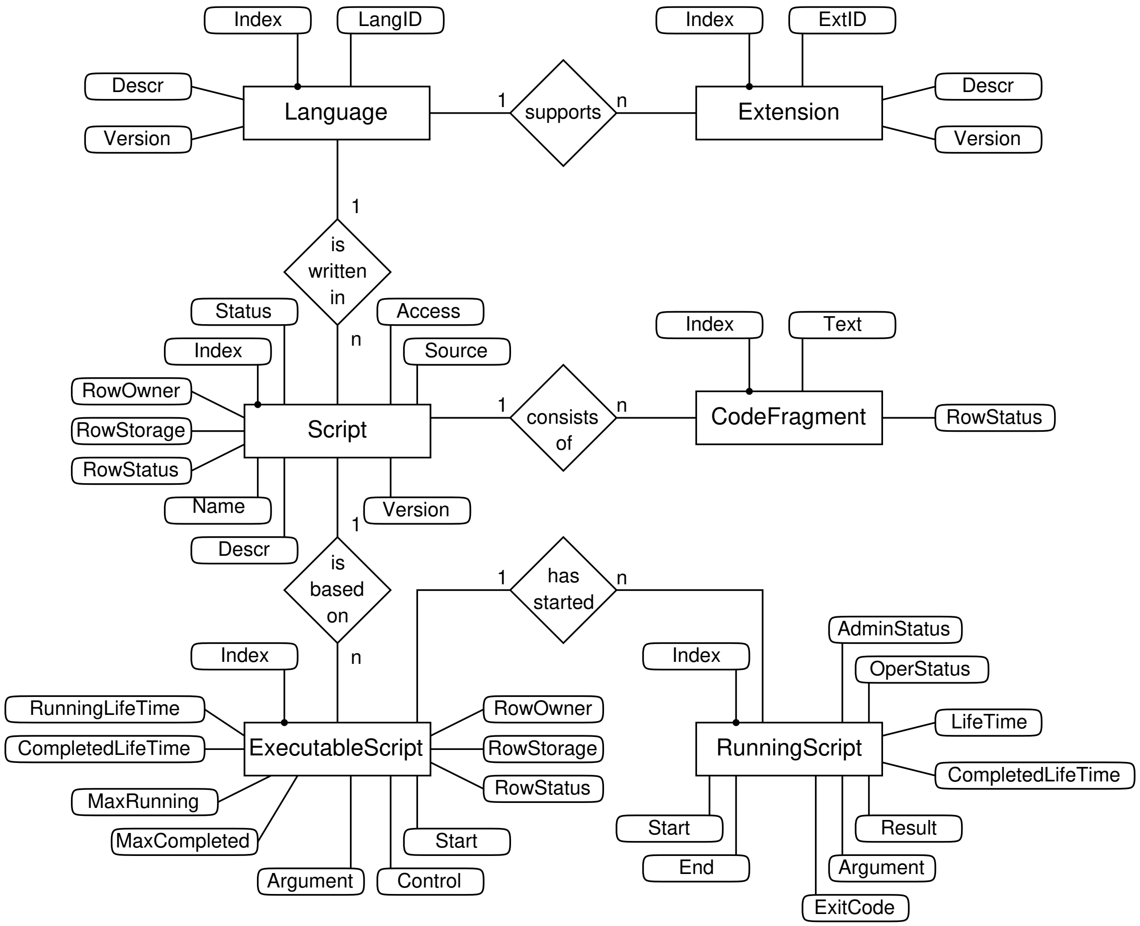
March 12, 1997

<draft-ietf-disman-script-mib-01.txt>

David B. Levi
SNMP Research, Inc.
levi@snmp.com

Juergen Schoenwaelder
University of Twente
schoenw@cs.utwente.nl

Script MIB Overview



Planned Enhancements

1. Remove smScriptAccess object.
2. Add state change table to define script readability (smCodeTable).
3. Register smExec as sub-identifier 3 instead of 4.
4. Register smRunNext as sub-identifier 3 instead of 4.
5. Register smRunTable as sub-identifier 4 instead of 5.
6. Register smLanguageID as sub-identifier 2 and smLanguageVersion as 3.
7. Avoid DisplayString sub-typing in TCs.
8. Define that the normal value of smRunAdminStatus is 'continue'.
9. Clarification that changing smExecRunningLifeTime or smExecCompletedLifeTime does not affect running or completed scripts.
10. Value of smRunLifeTime after script termination must be 0.

smExecRunningLifeTime	$\longrightarrow$	smExecLifeTime
smRunLifeTime	$\longrightarrow$	smRunLifeTime

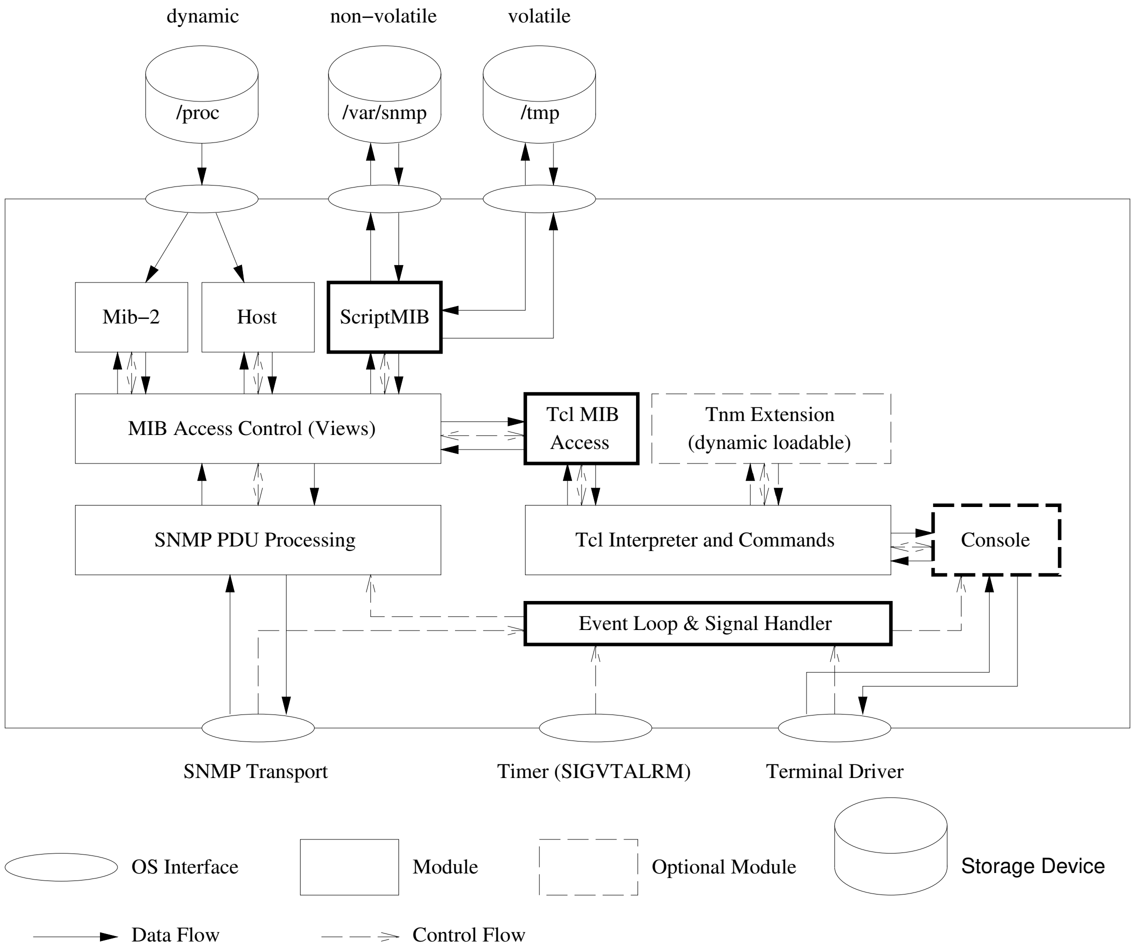
11.

smExecCompletedLifeTime	$\longrightarrow$	smExecExpireTime
smRunCompletedLifeTime	$\longrightarrow$	smRunExpireTime
smExec*	$\longrightarrow$	smLaunch*

Implementation Overview & Status

- Goal: Script MIB implemented on top of the Linux CMU agent.
- Goal: Separate the runtime system from the agent by a well defined API.
- Goal: Address security issues.
- Approach: Use secure Tcl as a scripting language.
  - Well-known and documented scripting language with nice C interface.
  - Secure Tcl allows to run each script in a separate Tcl interpreter.
  - Dynamic loading of Tcl allos to load manager functionality (e.g. Tnm extension) if required.
- Status: Code is beginning to work, but lots of stuff missing.

Implementation Structure



Security Issues

- Every script executes in its own safe Tcl interpreter.
- A trusted Tcl interpreter is only used internally.
- The default security profile removes/hides all "unsafe" commands from the Tcl interpreter.
- The smExecRowOwner value can be mapped on a security profile which is evaluated by the trusted interpreter before script execution starts. (profile file)
- The smExecRowOwner value can be used to define a default setup within a safe Tcl interpreter. (initialization file)
- A virtual timer is used to interrupt the Tcl system regularily to make sure that SNMP requests are processed even if a Tcl script enters a "`while 1 {}`" loop.

Security Questions

- Is it a good idea to bind the security profile to smExecRowOwner?
- Or should there a another object (e.g. smExecProfile) which is only used to identify the security profile?
- Or should the security profile only be determined by the "security options" used SNMP set messages?
- Relationship of security profiles to access control: The usual mechanism to define SNMP views for access control makes it difficult to express the following rule:

  Remove all rows from a view accessible by an SNMP session where the *RowOwner objects do not match a given value.

- We need a standard way to handle security profiles within DISMAN (and this needs to be aligned with SNMPv3). How do we get there?