Revision of RFC 2591 and RFC 2592

- Internet Drafts:
  - draft-ietf-disman-schedule-mib-v2-01.txt
  - draft-ietf-disman-script-mib-v2-01.txt
- Agenda:
  - Resolve any remaining open issues
  - Discuss new standards status of the revised documents
  - Define a schedule for the completion of this work item

DISMAN-SCHEDULE-MIB (RFC 2591 revision)

Issue 04: Automatically disabling failing schedules

The proposal was to add controls that can automatically disable a schedule if the triggered set operation fails for a certain number of times (perhaps in a given time window). It may be useful to emit a notification when this happens.

Discussion:

- Do we want to add such a feature?
- If yes, what are the precise conditions to disable a schedule?
- Shall we emit a notification when a schedule is automatically disabled?

DISMAN-SCRIPT-MIB (RFC 2592 revision)

Issue 05: Index smLangTable and smExtsnTable by name

The MIB would be more consistent if all objects were indexed by their name. Currently, the read-only smLangTable and smExtsnTable are indexed by Integer32 values.

Discussion:

- Requires to replace smLangTable and smExtsnTable.
- Requires to replace the smScriptLanguage column of the smScriptTable.
- Some simplification when installing scripts since a manager can better "guess" a language/extension name.
- Are the benefits worth the costs of the change?

DISMAN-SCRIPT-MIB (RFC 2592 revision)

Issue 06: Dependencies between scripts and script versioning

The MIB does not model dependencies between scripts (e.g. launching script B requires that script A is present and enabled) nor does it provide explicit support for multiple versions of a script.

Discussion:

- Formally expressing dependencies between scripts adds complexity.
- Unclear at this stage whether this complexity is really needed in operational environments.
- Versioning is hard unless we impose our own software versioning model.
- It is possible to encode version numbers in the script name and to rely on conventions to address this problem.
- Should this be left for future work once we have more deployment experience?

DISMAN-SCRIPT-MIB (RFC 2592 revision)

Issue 07: Storage type of script code vs script meta information

There seems to be some confusion if and how the storage type of a script's meta information differs from the storage type of the script code itself. For example, can conditions exist in which the information in the smScriptTable exists in non-volatile storage while the code of the script is volatile?

Discussion:

- Problem may be rooted in the feature which allows to save code fetched from a URL locally.
- The editor does not understand what text is needed to better describe how things are supposed to work.

DISMAN-SCRIPT-MIB (RFC 2592 revision)

Issue 10: Script editing in the smCodeTable

The script MIB provides basic script editing capabilities via SNMP set operations. Implementation of this feature adds costs while the utility of this feature is questionable.

Discussion:

- The capability to edit fragments of a script in the smCodeTable adds some implementation costs.
- The utility of being able to edit scripts in place via SNMP sets is questionable since there is no way to "renumber" fragments or to split "fragments".
- Manager programs (and most humans) are most likely pushing new script instead of trying to edit an installed script.
- Removing the editing feature requires some changes to existing objects in the smCodeTable and the smScriptTable.

DISMAN-SCRIPT-MIB (RFC 2592 revision)

Issue 11: Resource consumption indication in the smRunTable

It would be useful to have resource consumption indications (CPU, memory) for running scripts. The question is whether it is possible to define metrics that are semantically meaningful and generally implementable.

Discussion:

- One approach would be to just augment the smRunTable with the process identifier for the runtime engine executing the script.
- Requires that the manager extracts the statistics from elsewhere and interprets them correctly for a given script MIB implementation.
- Define our own metrics (CPU time consumed and memory used) and require that implementations compute these values.
- Some implementation will have a hard time to compute correct values (e.g. the memory used by a single thread in a Java VM).
- It may be difficult to define the statistics precisely. (Does the CPU time consumed by a process forked from a script account for the script or not?)

DISMAN-SCRIPT-MIB (RFC 2592 revision)

Issue 16: Suspend/resume on runtime engines that do not support it

The current specification says that the running script remains in the suspending (resuming) state when the runtime is not capable to suspend (resume) a script. This makes it impossible to determine on the manager side whether the suspend (resume) just takes a long time to complete or is not supported.

Discussion:

- The first possible solution is to go from the transient suspending state back into the running state when the runtime can not suspend a script.
- This solutions allows a transient suspending state and therefore requires polling to determine if a runtime can suspend a script.
- The second possible solution requires that the agent determines the suspend/resume capability during the processing of the SNMP set operation and to return an SNMP error response when the suspend will not succeed.