Notes on Job Monitoring MIB Objects and Tables

Scott Isaacson

During the JMP meeting in NYC on 10/3/96, there was some discussion about how to partition the proposed objects into various groups and tables. It appears that most of the proposed objects apply to both spooling and non-spooling instances of service providers. Randy suggested that we partition the Job Monitoring (JM) MIB into groups and tables that could easily be implemented by agents in both spooling and non-spooling "devices", but keep all of the spooling related objects in a conditionally mandatory group or table. That is, if the device or server on which the agent is implemented has spooling capabilities then it would implement the spooling related objects, otherwise it would not.

I took the action item to write up our discussion and propose solutions in the areas that were more grey. My proposal is as follows. See Figure 1 for a Graphical View of the proposed tables.

1. There should be a jmJobTable that is indexed by a jmJobIndex objet. An entry is created in this table for every new job as it becomes known (submitted to) the service provider. It stays in this table at least until the job reaches the completed state but for as long as the local policy dictates (see General Group).

2. There should be a General Group with an object that represents the local policy pertaining to how long jobs are kept in the jmJobTable.

3. Every agent should implement a Spooling Group with at least three objects in the group:
- Is Spooling Implemented (Will PresentOnOff syntax work??)
- Current Scheduling Algorithm object (FIFO, Shortest, None (if no spooling),etc.)
- jmQueueTable (see #4 below)

4. There should be jmQueueTable that contains indexes into the jmJobTable. This table is conditionally mandatory and is implemented only if the service provider for which the agent exists has a spooling capability. The entries in this table are jmJobIndicies back into the full job information in the correct row of the jmJobTable. This table gives order to the jobs that are in some state other than "completed". The order is based on the current scheduling algorithm. I added two objects to this table:
- Priority
- Print After

since these only apply to spooling/scheduling. The information in these objects is not needed after the job completes; it is only useful for scheduling of spooled jobs.

5. In the figure below I show a jmCompTable that contains indexes into the jmJobTable. This table is mandatory and shows the jobs in the order in which they were actually processed. The

entries in this table are jmJobIndicies back into the full job information in the correct row of the jmJobTable. I am not so certain that this table is needed/

6. There should be jmDocTable that contains jmDocEntries. Each entry in this table is doubly indexed by both a jmJobIndex and a jmDocSequenceNumber. This table is needed for multiple documents per job. Each entry contains the document name and the PDL that this document is in. *Issue: Is there more than one PDL per document?*

7. There should be jmResTable (Resource) that contains jmResEntries. Each entry in this table is doubly indexed by both a jmJobIndex and a jmResSequenceNumber. This table is needed for multiple resources per job. Each entry contains the resource type, name, units in which it is measured, and then units consumed for accounting purposes. If the consumed it 0, then it is a resource that was requested but never used.

8. There should be jmAccountTable (Accounting) that contains jmAccountEntries. Each entry in this table is doubly indexed by both a jmJobIndex and a jmAccountSequenceNumber. This table is needed for multiple accounting messages per job. Each entry simply contains some text that can be used for accounting purposes other than explicit resources used.

9. There should be jmEWRTable (Error, Warning, and Report) that contains jmEWREntries. Each entry in this table is doubly indexed by both a jmJobIndex and a jmEWRSequenceNumber. This table is needed for multiple error, warning, and report messages per job. Each entry simply contains some text that can be used for reporting specific events (errors, warnings, reports) that apply to a specific job.

10. When a jmJobEntry is removed from the jmJobTable, all corresponding entries from jmEWRTable, jmQueueTable, jmCompTable, jmAccountTable, jmResTable, and jmDocTable must be removed as well.

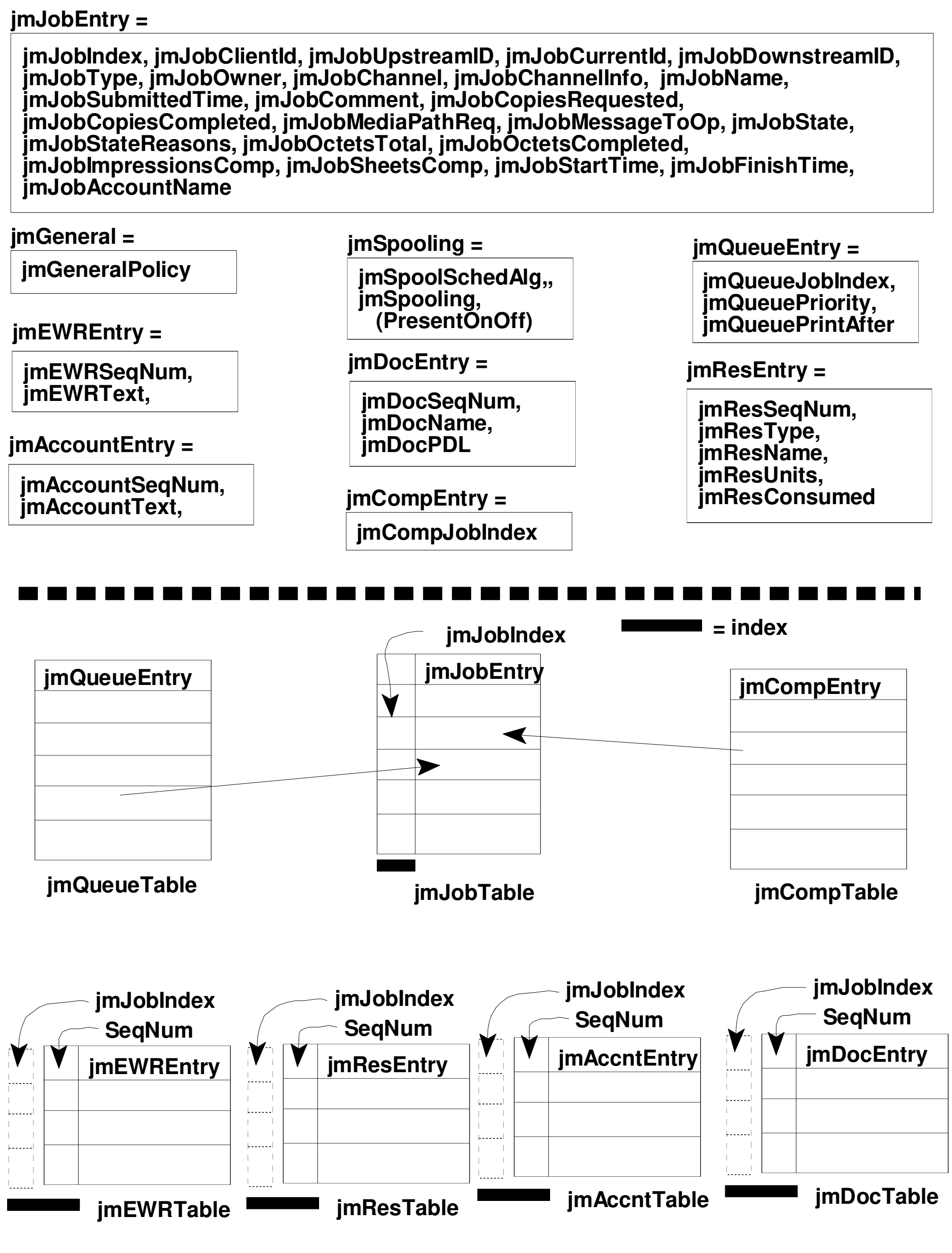

Figure 1 JM MIB Graphical View