



IMM Persistent back end

Hans Feldt
Senior Software Engineer, Ericsson AB
Hans.Feldt@ericsson.com

Developer Days May 2010

IMM in general - recapturing

- IMM - Information Model Management
- Manages the SA Forum Information Model
- A database with configuration and runtime objects
- Two distinct APIs; object management and object implementer
- Objects identified with a LDAP distinguished name
- Configuration changes
- IMM acts a broker for administrative operations on objects between the client and the implementer
- Persistent entities stored in an internal persistent repository

IMM in OpenSAF 3.0 - recapturing...

- IMM database is fully replicated to all nodes
- Mostly for configuration data thus optimized for reading
- Implements most of the A.02 specification
- Additional non standard shell commands to support simple management
- Persistence has to be maintained by external functions
- For example dumping the IMM DB to file before/after configuration changes (backup/restore system functions)
- For more information see previous presentations from developer days and
`osaf/services/saf/immsv/README`

What does the IMM specification say about persistence?

- "The IMM service maintains a copy of all its persistent entities within an internal persistent repository kept on stable storage."
- "The nature of this internal repository is implementation-specific."
- "During startup of the IMM Service, the contents of its internal repository may be over-written (or initialized if the internal repository was empty) from the contents of an XML file."
- "The configuration parameter *salmRepositoryInit* of the *SalmMngt* object class specifies whether to overwrite or not the contents of the IMM Service internal repository at startup of the IMM Service."

IMM Persistence in OpenSAF 4.0

- PBE - Persistent Back End, new feature to meet the persistence requirement
- PBE uses SQLite
- "**SQLite** is an ACID-compliant embedded relational database management system contained in a relatively small C programming library." (Wikipedia)
- PBE requires that the sqlite DB file is accessible from both controllers thus replicated in some manner
- By default disabled!

IMM PBE - Configuring

- We have a weird configure bug:

```
$ export PKG_CONFIG=`which pkg-config`
```

- Enable building PBE using configure:
 - `configure --enable-imm-pbe ...`
- Build and install opensaf
- Configure PBE by editing `immnd.conf`
 - `IMMSV_ROOT_DIRECTORY` - directory for IMM files
 - `IMMSV_LOAD_FILE` - name of XML initial load file
 - `IMMSV_PBE_FILE` - name of sqlite DB file

IMM PBE - Enabling

- Disabled by default, enable it!
- IMM has a management object with DN
safRdn=immManagement,safApp=safImmService
- Attribute *saImmRepositoryInit* determines action at start
- Edit imm.xml or change attribute value in runtime
- SA_IMM_KEEP_REPOSITORY(1) enables PBE:
\$ immcfg -a saImmRepositoryInit=1
safRdn=immManagement,safApp=safImmService
- SA_IMM_INIT_FROM_FILE(2) disables PBE:
\$ immcfg -a saImmRepositoryInit=2
safRdn=immManagement,safApp=safImmService

IMM PBE - Wrap up

- PBE Status can be followed in tickets 858, 859, 860 & 861
- Weaknesses with this solution is that file system problems will block writes to IMM of CCBs and updates of persistent runtime attributes
- Remember that the IMMSV_ROOT_DIRECTORY file system location needs to be accessible from both controllers!
- Stability test missing of this new function!

IMM PBE

Questions?