EMT: The Next Phase

David Markley

A report prepared for the discussion of further development of EMT

EMT has continually evolved throughout its development. A large part of this evolution is due to the changes in our software development policies and practices. Other factors, such as feature-loading, have also contributed to changes that have taken place. At some time, however, it is necessary to determine a cut-off point for current development and to move the tool fully into production. EMT has reached that point.

1.0 The Current State

The current version of EMT is nearing completion. With the functionality that it will soon have, it should be ready for production use by non-technical staff. Further development may be discussed in the future, but current development should be halted in the near future.

The following sections describe the current functionality available through EMT.

1.1 create

New collections and new versions of collections can be created using the **`create`** command. The command allows for multiple system types; common areas; copied, new, or no source area; afs or local object areas; specification of quota for dest, common, src, and/or object areas; and adding users to the commander's ACL.

1.2 delete

Existing versions of a collection may be deleted through the **`delete`** command. The command allows for multiple system types; common areas; and specification of which version should be deleted in each of the src, dest, and obj areas. This command also ensures that the version being deleted is not in use in any environment of any know tree.

1.3 release

A specific version of a collection can be released through the **`release`** command into any environment known to EMT. The command allows for multiple system types; common areas; running or no running of depot; release of collection volumes; release of environment volumes; and locking of the collections src and dest areas for the released version.

1.4 access

Access to parts of a version of a collection can be given or denied through the **`access`** command. The command allows for multiple system types; addition of users to the commander's ACL; specification of which areas (src, dest, obj) to have their access set; and specification of whether access is being denied (locked) or not.

1.5 examine

The current state of a collection may be viewed through use of the **`examine`** command. The command allows for multiple system types and specification of which areas should be examined.

1.6 update

Environments can be brought up to date through use of the **`update`** command. This command allows for multiple system types; running of depot across all collections in the environment; and release of the environment volumes.

1.7 setquota

The quota for parts of a collection can be set using the **`setquota`** command. This command allows for multiple system types and specification for the quota to be used in each area (src, dest, obj).

1.8 verify

The correctness of all mounts within a given tree can be checked using the **`verify`** command. This command allows for single tree verification or "all" trees. An output file must be specified. This command makes no corrections. It simply reports discrepancies.

1.9 tsdb

Time stamp functions will be available through the **`tsdb`** command. Currently, this is simply a stub supplied for further development.

2.0 Work In Progress

There is a current version of EMT under development that comprises some significant changes from the current version. These changes will result in a more versatile server and client.

2.1 Changes to the server

- Conversion to single user operation
- Change of `pts` commands to `ADM` calls
- Change of `ws` routine to `ADM` call
- Allow for completion of commands that started before time restrictions

2.2 Changes to the client

- Allowance for ^C or ^Z to break out of the current command.

2.3 Changes to functionality

The functionality of various commands is being enhanced and/or simplified.

2.3.1 Create

- Allow for specification of where volumes are created
- Specification of a restricted state for a collection

2.3.2 Release

- Allow for specification of where volumes are replicated
- Specification of a restricted state for a collection

2.3.3 Purge

This is a new command that will allow for the removal of a collection from an environment, and eventually complete removal.

3.0 Move To Production

After the completion of the work in progress, EMT will be ready for movement into full production. This means that someone outside of the development staff will become the gatekeeper and operate EMT. The following would be the steps taken for the move:

1. Complete Work In Progress
2. Select new gatekeeper from outside development staff
3. Train new gatekeeper in operation of EMT
4. Monitor new gatekeeper for some period of time
5. Continue support of EMT by development staff

4.0 Continued Support

Once EMT is fully moved into production, it will still be necessary for the development staff to continue supporting EMT. It will be the responsibility of development staff to ensure that both the EMT and ADM servers are operating correctly, and to determine the cause of any problems outside the scope of the gatekeeper. If a problem occurs with the EMT or ADM servers during non-working hours, the PRIMARY should be beeped. Initially, David Markley would carry the extra beeper and be the first line of support for any EMT problems.

5.0 Further Development

During the use of EMT in full production, more information on the use of EMT will be gathered. What functionality is still required remains to be seen. There is already a list of things that could be added to a future version of EMT.

- Support of “complex” collections (such as atk)
- Database of collection information.
  collections that should be overridden
  related collections
- An X-version of the EMT client