Make(Report)

Environment Maintenance Tool
emt

@Chapter(Goals of emt)

The concept of an application like emt was brought about by the current difficulty that exists in maintaining and releasing applications within several different environments. It was quickly realized that the tools needed for such maintenance were available, but the process of using them was large, time consuming, and complex. This lead to errors because the maintainer easily lost track of where in the process he thought he was. To be extra careful often meant to backtrack the steps that had been taken, costing the maintainer more time. The goal of emt is to relieve as much of this mundane and tedious work as possible, by providing a common application that will provide all the needed tools and monitor the status of the environment.

@Section(The Tools) To provide the maintainer with the ability to perform his work within the framework of a single application will require the inclusion of the functionality of several other applications through a common interface. To increase the speed at which the maintainer can perform his tasks, several high level tools will need to be created that carry out the steps of his tasks using the base tools combined into a single step.

@Section(The Database)

To allow the maintainer the confidence that all is going well, and to provide him with the current status of the environment, a database of all activity will be kept. This will provide information on each level of the environment, from individual volumes to whole applications. It will also keep track of any errors that occurred, so the maintainer can quickly fix the problem, instead of having to look for it.

@Chapter(Functionality of emt) To the user, emt will be a very interactive program. After he has chosen the operation he wishes to perform, emt will prompt the user for all the necessary information to complete that operation. This will greatly simplify the process of system maintainers. There will be default values that can be set for each choice.

To choose the operation that he wishes, the user will simply enter the name of the operation in the emt shell, or press the correct buttons in the emt window interface. These are the two interfaces that will be developed for emt: command line and windowed.

The command line version of emt will enter the user into an "emt shell." The user will see a different prompt, indicating that he is in emt. At this point, he can enter any of the commands available. There will be a help command at this level to assist the user in the use of the commands.

The windowed version of emt will be a motif based interface that will present the user with pull-down menus for the commands he can issue, and a log window displaying the results of all the operations performed. This will be a very user friendly interface, as each choice is made by simply pressing buttons.

@Section(Volume Maintenance) The volume maintenance commands will allow direct manipulation of volumes by the user. In these operations, the user will have to supply three main pieces of information: volume name, server, and partition. With this information, and default guidelines set by the user, each operation can be performed by emt.

@SubSection(Volume Creation) Once the volume creation operation is selected, the user will be prompted for the name of the volume to create, the server on which to create it, and the partition on that server in which to create it. The user will also be queried as to whether the information is correct or not. If all information is correct, emt proceeds to create the volume as specified, and reports the results to the user. It also creates the appropriate entry in the emt's volume database, and sends a note to the correct bboard.

@SubSection(Volume Deletion) When the volume deletion operation is chosen, the user will be prompted for the name of the volume to delete and the server and partition in which it resides. The user will be prompted about the correctness of this information. If the user has indicated that he is sure this is the correct volume to delete, emt will first make its own check on the volume. If it finds the volume to still be mounted anywhere (within its database), it will first verify the mount, and then inform the user he will be deleting a mounted volume (if the mount does exist). The user will be queried if he wants to remove the mount and continue. If no mounts exist in the database, or they did exist but were in error, the volume will be removed and emt's volume database updated to be correct.

@SubSection(Volume Mounting) The user will be required to supply the name of the volume to be mounted, and the full path name of the mount (including the name of the mount as the end of the path). Emt will then attempt to make the mount, and report any errors (and why they took place) to the user.

@SubSection(Volume Unmounting) The user will need to supply the name of the volume to be unmounted, and the full path name to the mount (including the name of the mount as the end of the path). Emt will then attempt to remove the mount, and report any errors (and why they took place) to the user.

@SubSection(Volume Replication) When the volume replication operation is chosen, the user will be prompted for the name of the volume to replicate. The user will have to acknowledge that this is the correct volume name. Emt will attempt to add the appropriate sites, move the original volume, and release the volume according to defaults set up by the user. Emt will report any errors (and why they took place) to the user.

@SubSection(Volume Releasing) When the volume release operation is selected, the user will have to provide the name of the volume to release. The user will be queried about the correctness of the volume name. If the the volume name is indicated to be correct, emt will attempt to release the volume and report any errors (and why they took place) to the user.

@Section(Application Maintenance) In the application maintenance section, the user will be required to supply various information beyond what was required in the volume maintenance section to perform the operation selected. It will be necessary to know what application, under which tree, in which version, and for which system types an operation needs to be performed. The user will be operating on an application as a whole, not as separate pieces.

@SubSection(Application Creation) When the user chooses to create a new application, he will be prompted for the name of the application, and for the server and partition the src and rcs volumes should be placed on. The program will then create the src and rcs volumes on the server and in the partition given. It will also create the object directory, and the group for that application. After creating each area, emt will set the access list accordingly. Emt will report any errors encountered (and the reason) to the user. Emt will update its application database with the appropriate information, upon a successful creation.

@SubSection(Application New Version Creation) The user will be asked for the name of the application, the system types, and the version number for the set of new volumes the user wants for an application. Emt will then create the new volumes on the user set default server and partition, mount them in the user set default tree, and set their access lists appropriately. Emt will report any errors encountered (and the reason) to the user. Emt will update its application database with the appropriate information, upon successful creation.

@SubSection(Application Releasing) The user will be asked to supply the name of the application, the trees in which it is to be released, and the version which should be released there. The program will then release all necessary volumes, and notify the user if chosen volumes are not replicated. It will allow the user to decide to replicate them at that time. Emt will report any errors encountered (and the reason) to the user. Emt will update its application database with the appropriate information, upon a successful release.

@SubSection(Application Deletion) For this option, the user will only have to supply the application name. The program will then remove all items related to that application found in its database. Emt will report any errors encountered (and the reason) to the user. Emt will update its application database with the appropriate information, upon a successful deletion.

@Section(Environment Control) The user will be able to select what environment they are working on, and access information for applications and volumes that reside withing that environment. There will also be the ability to combine the information from different environments to determine items such as system wide disk usage, etc.

@Section(Interactive Database Control) The user will be able to access the information about any particular volume or whole application through the databases that emt will maintain. Emt will maintain a database that contains information on each volume, such as its creation date, if it is replicated, where it is mounted, etc. Emt will also maintain a database for whole applications, which contains what volumes are part of an application, where and what version of an application is mounted in which trees, when releases of the application have taken place and for which versions, etc. This will allow maintainers the ability to quickly see the state of an individual volume or an entire application.

@Chapter(Design of emt)

@Section(Default Setting Routines) With these routines, the defaults of certain parts of emt will be able to be set by the user. These defaults are such items as the defaults server and partition for new volumes, the default servers and partitions for volume replication, the default tree for new application volumes, and the environment being worked on.

@SubSection(New Server and Partition)

The user will be queried for a server name and a partition that should be used as the default. The user will also be given the choice as to whether the supplied server and partition should be used as the default in future sessions. The global variables for default server and default partition will be set.

@SubSection(Replication Sites)

This section will require more thought.

@SubSection(Current Environment)

@Section(Low Level Routines) These routines are the base level at which emt operates. They are the system calls that create, delete, move, add sites, and release volumes. In general, all calls that directly affect the system are part of the low level routine section.

@SubSection(Create a Volume)

@SubSection(Delete a Volume)

@SubSection(Move a Volume)

@SubSection(Add a Site For a Volume)

@SubSection(Release a Volume)

@SubSection(Mount a Volume)

@SubSection(Unmount a Volume)

@SubSection(Run Depot on a Mounted Volume)

@SubSection(Examine a Volume)

@Section(High Level Routines) These routines are for speeding up larger operations by the user. Things such as replicating volumes and releasing applications would be considered too time consuming if the user had to complete each step himself, even through the emt low-level interface.

@SubSection(Replicate a Volume)

@SubSection(Create an Application)

@SubSection(Create the New Volumes for a New Version of an Application)

@SubSection(Release an Application)

@SubSection(Delete an Application)

@Section(Database Routines) These are the calls that modify the database. They are the only way in which the database should be modified. There is a capability to rebuild the database, but it is a long process. The database will attempt to correct itself, and will notify the maintainer of errors it encounters. Part of these routines are dedicated towards the maintainers ability to correct the database instead of rebuild it.

@SubSection(Add a New Volume Entry)

@SubSection(Add a New Mount Site to a Volume Entry)

@SubSection(Remove a Mount Site from a Volume Entry)

@SubSection(Delete a Volume Entry)

@SubSection(Add a New Application Entry)

@SubSection(Add a New Version to an Application Entry)

@SubSection(Modify the Tree an Application Version is Located in)

@SubSection(Note the Release of an Application Version)

@Chapter(Code of emt)