EVMS Command Line Grammar

Notes:

Segments - Segments represent blocks of physical disk space.

Regions - Regions represent blocks of logical disk space.

Containers - Containers in EVMS are the equivalent of Volume Groups in the current Linux LVM.

Compatibility Volume - a Volume (which may or may not be created by EVMS) which is compatible with the Volume Manager from another operating system.

Storage Objects - Storage Objects represent blocks of logical disk space which have one or more EVMS Features applied to them.

EVMS Volume - a Volume which was created by EVMS using the native EVMS format. EVMS Volumes are not compatible with the volume managers of other operating systems.

EVMS command keywords will NOT be case sensitive.

EVMS names ARE case sensitive. This includes plug-in names, volume names, container names, storage object names, region names, segment names, and disk names.

EVMS operations involving plug-ins often require that options for the plug-ins be specified. This is done using name - value pairs. The name portion of the name - value pair is NOT case sensitive. The value portion, if the value is not numeric (i.e. a name), IS case sensitive.

The EVMS command line interpreter does not accept EVMS commands as part of the command line used to invoke it. The options on the command line used to invoke it control how the interpreter actually interprets EVMS commands. Once invoked, the EVMS command line interpreter will prompt the user for EVMS commands. To exit the EVMS Command Line Interpreter, press the enter key without having entered a command at the prompt.

Notation:

Items in <> represent non-terminals - they will be defined elsewhere in the grammar
Items in [] are optional
() are used for grouping
()+ means that the item in the () appears 1 or more times
()* means that the item in the () appears 0 or more times
’(’ and ’)’ indicate that the parenthesis are not being used as part of the grammar, but are instead characters that appear on the command line.
’[’ and ’]’ indicate that the brackets are not being used as part of the grammar, but are instead characters that appear on the command line.
’<’ and ’>’ indicate that the angle brackets are not being used as part of the grammar, but are instead characters that appear on the command line.
’’’ means that the quote is not being used as part of the grammar, but is instead a character which must appear on the command line.
| means "or"
::= defines the meaning of a non-terminal.
::== defines the meaning of a non-terminal in terms of something external to the grammar.

Grammar for Invoking the EVMS Command Line Interpreter:

EVMS [<Command Line Parameters>]

<Command Line Parameters> ::= ([-c] [-d <Debug Level>] [-F[ile] : <File Name>] [-s] [-p] [-v | -v0 | -v1 | -v2]) | [-h]

where:

-c indicates that changes are not committed to disk until the user exits the EVMS Command Line Interpreter. Normally, changes are committed to disk after each individual command is completed, even if there are multiple commands specified on the same command line.
-d sets the debug level, which controls how much information is logged to the evmsEngine log file.
-F tells the EVMS command line interpreter to take its input from <FileName>.
-s The EVMS command line interpreter normally prompts for commands until an empty command line is encountered. This switch tells the EVMS command Line Interpreter to prompt for one command line only.
-p tells the EVMS command line interpreter to parse commands only. Errors found during parsing will be reported. Commands will NOT be executed.
-v, -v0, -v1, -v2 tells the EVMS command line interpreter how to handle status messages from the EVMS Engine while the EVMS Engine is committing changes to disk. -v0 indicates that EVMS Engine status messages should be discarded. -v, which is the same as -v2, indicates that EVMS Engine status messages should be reported on one line, with each message being erased before a new message is displayed. -v1 indicates that EVMS Engine status messages should be displayed with one message per line.
-h invokes the initial EVMS help for the EVMS command line interpreter.

Grammar for the commands accepted at the EVMS command prompt:

<Command> ::=
C[reate]: <Creation Parameters>
| A[llocate]: <Specify Freespace> (, <Name Value Pair>)+
| As[sign]: <Plugin Identifier> = { [<Name Value Pair> (, <Name Value Pair>)*] } , (<Specify Disk> | <Specify Segment>)
| Ch[eck]: <Specify Volume> (, <Name Value Pair>)+
| D[elete]: <Specify Volume> | <Specify Container> | <Specify Storage Object> | <Specify Region> | <Specify Segment>
| E[xpand]: <Specify Storage Object> [(, <Name Value Pair>)+] (, <Specify Storage Object> | <Specify Region> | <Specify Segment>)+
| De[fragment]: <Specify Volume> (, <Name Value Pair>)+
| F[ormat]: <Specify Filesystem Interface Module> = { [<Name Value Pair> (, <Name Value Pair>)*] } , < Specify Volume >
| H[elp] [: Allocate | Assign | Check | Create | Delete | Defragment | Expand | Format | Probe | Query | Set | Remove | Rename | Revert | Shrink | Unformat]
| P[robe]
| Q[uery]: <Query Type>
| Rem[ove]: <Specify Disk> | <Specify Segment>
| Ren[ame]: <Specify Volume>, N[ame] = " <EVMS Name> "
| Rev[ert]: <Specify Volume> | <Specify Storage Object>
| Set : <Specify Storage Object> (, <Name Value Pair>)+
| S[hrink]: <Specify Storage Object> [(, <Name Value Pair>)+] (, <Specify Storage Object> | <Specify Region> | <Specify Segment>)+
| U[nformat] : <Specify Volume>

<Creation Parameters> ::=
C[ontainer] , <Plugin Identifier> = { [<Name Value Pair> (, <Name Value Pair>)*] } (, <Specify Segment> | <Specify Region> | <Specify Disk>)+
| O[bject] , <Feature Identifier> = { [<Name Value Pair> (, <Name Value Pair>)*] } (, (<Specify Storage Object> | <Specify Region> | <Specify Segment> | <Specify Disk>))+
| R[egion] , <Plugin Identifier> = { [<Name Value Pair> (, <Name Value Pair>)*] } (, <Specify Segment> | <Specify Region> | <Specify Disk>)+
| S[egment] , <Specify Freespace Segment> (, <Name Value Pair>)+
| V[olume] , (<Specify Storage Object> | <Specify Segment> | <Specify Region> | <Specify Disk>) , ((N[ame] = " <EVMS Name> ") | C[ompatibility])

```
<Query Type> ::=
        P[lugins] [ , ( (T[ype] = <Plugin Types>) | ( P[lugin] = <Plugin Identifier> ) |  ( V[olume] = <Specify Volume> ) | ( O[bject] = <Specify Storage Object> ) |
            ( C[ontainer] = <Specify Container> ) ) ]  [ , L[ist ]O[ptions] ]
        | V[olumes] ( ( [ , <Size Parameters> ] [ , P[lugin] = <Plugin Identifier> ]  [ , C[ontainer] = <Specify Containert> ]  [ , ( E[xpandable] | S[hrinkable] ) ] ) |
            ( ,  R[egion] = <Specify Region> ) | ( , O[bject] = <Specify Storage Object> ) ) [ , L[ist ]O[ptions] ]
        | R[egions] [ , <Size Parameters> ] [ , V[olume] = <Specify Volume> ] [ O[bject] = <Specify Storage Object> ] [ C[ontainer] = <Specify Container> ]
            [ U[nclaimed] ] [ , P[lugin] = <Plugin Identifier> ] [ , ( E[xpandable] | S[hrinkable] ) ] [ , L[ist ]O[ptions] ]
        | O[bjects]  [ , <Size Parameters> ] [ , P[lugin] = <Plugin Identifier> ] [ ,V[olume] = <Specify Volume> ] [ , C[ontainer] = <Specify Container> ]
            [ , ( E[xpandable] | S[hrinkable] ) ]  [ , U[nclaimed] ] [ , R[egion] = <Specify Region> ] [ , L[ist ]O[ptions] ]
        | C[ontainers] ( ( [ , <Size Parameters> ] [ , P[lugin] = <Plugin Identifier> ]  [ , F[reespace]  [ , <Size Parameters> ] ]  ) | ( , R[egion] = <Specify Region> ) ) [ , L[ist ]O[ptions] ]
        | D[isks] [ , <Size Parameters> ] [ , P[lugin] = <Plugin Identifier> ] [ , U[nclaimed] ] [ , L[ist ]O[ptions] ]
        | S[egments] [ , <Size Parameters> ] [  , P[lugin] = <Plugin Identifier> ]  [ , C[ontainer] = <Specify Container> ] [ , D[isk] = <Specify Disk> ]
            [ , U[nclaimed] ] [ , ( E[xpandable] | S[hrinkable] ) ] [ , L[ist ]O[ptions] ]
        | A[cceptable] , ( ( C[reate] , <Object To Create> , <Plugin Identifier> = { [ <Name Value Pair> (, <Name Value Pair> )* ] } ) | (  E[xpand] ,  <Specify Storage Object> ) )
            ( , (<Specify Storage Object> | <Specify Region>  | <Specify Segment> | <Specify Disk>) )*
        | Chi[ldren] ,  <Specify Storage Object> | <Specify Volume> | <Specify Region> | <Specify Segment> | <Specify Container>
        | Par[ent] ,  <Specify Storage Object> | <Specify Disk> | <Specify Region> | <Specify Segment> | <Specify Container>
        | E[xpand ]P[oints] , <Specify Storage Object> | <Specify Volume> | <Specify Region> | <Specify Segment>
        | S[hrink ]P[oints] , <Specify Storage Object> | <Specify Volume> | <Specify Region> | <Specify Segment>
        | E[xtended ]I[nfo] , (  <Plugin Identifier> | <Specify Storage Object> | <Specify Disk> | <Specify Region> | <Specify Segment> | <Specify Container> ) [ , <Field Name> ]

<Plugin Types> ::=
        D[evice Manager]
        | F[eature]
        | R[egion Manager]
        | S[egment Manager]
        | F[ile]S[ystem ]I[nterface ]M[odule]
        | D[istributed ]L[ock ]M[anagement]
        | C[luster ]M[anagement]

<Object To Create> ::=
        V[olume]
        | O[bject]
        | R[egion]
        | S[egment]
        | C[ontainer]

<Feature Identifier> ::=
        "  <EVMS Feature Name> "
        | <EVMS Feature ID>

<Plugin Identifier> ::=
        " <EVMS Plugin Name> "
        | <EVMS Plugin ID Number>

<Field Name> ::= <String>
```

```
<Size Parameters> ::=
        LT <Size>
        | GT <Size>
        | EQ <Size> (,  < Precision> )
        | LT <Size> , GT <Size>
        | GT <Size> , LT <Size>

<Specify Freespace>  ::== the EVMS name of a segment or region representing a block of freespace.

<Specify Freespace Segment> ::== the EVMS name of a segment representing a block of freespace.

<Specify Region> ::== the EVMS name of a region

<Specify Segment> ::== the EVMS name of a segment

<Specify Disk> ::== the EVMS name of a disk.

<Specify Storage Object> ::== the EVMS name of a storage object.

<Specify Container> ::== the EVMS name of a container.

<Specify Volume> ::== the EVMS name of a Volume.

<Specify Filesystem Interface Module> ::== the EVMS name of a filesystem interface module.

<Name Value Pair> ::=
        <Parameter Name> = <Value>
        | <Parameter Name> = '(' <Value> ( , <Value> )* ')'

<Value ::=
        <Integer> [ ( KB | MB | GB | TB  | s[econds] | sectors | h[ours] | min[utes] | m[illi]s[econds] | us | microseconds ) ]
        | <Real Number> [ ( KB | MB | GB | TB  | s[econds] | sectors | h[ours] | min[utes] | m[illi]s[econds] | us | microseconds ) ]
        | <String>

<Size> ::= <Cardinal> ( MB | GB | KB | TB )

<Precision> ::= <Cardinal>

<EVMS Feature Name> ::== This is provided to the EVMS by the feature plugin and must be in the format specified by the EVMS.

<EVMS Feature ID> ::== This is provided to the EVMS by the feature plugin and must be in the format specified by the EVMS.

<Parameter Name> ::== Any parameter name provided to the EVMS by a plugin that is in the format specified by the EVMS.

<EVMS Plugin Name> ::== This is provided to the EVMS by the plugin and must be in the format specified by the EVMS.

<EVMS Plugin ID Number> ::== This is provided to the EVMS by the plugin and must be in the format specified by the EVMS.
```

<Cardinal> ::== A cardinal number according to the mathematical definition of such.

<Integer> ::== An integer number according to the mathematical definition of such.

<Real Number> :== A floating point number.

<File Name> ::== A Linux acceptable filename

<String > ::== Any sequence of ascii characters except for the quote mark. A string should be enclosed in quote marks if it contains any non-alphabetic or non-numeric characters or spaces. If a quote mark is embedded in the string as a part of the string, then two quote marks should be used, as in the following example create command:

Create:Volume,hda7,Name ="This is the name of an ""EVMS"" test volume."

<EVMS Name> ::== a character string the length of which is yet to be determined. This character string should be one provided by EVMS for something created by, or under, EVMS control.