# z/VM GUI Project
## Initial Functionality Documentation
## Version 0.2 01/22/2004

---

# Project Goals

The goals statements will be the basis for every feature of our software product. Everything that is added to this project must be in support of one or more goals and all goals should be met by various features of the product.

### 1. Perform Administrative tasks in z/VM from a GUI (Graphical User Interface)

z/VM is a very powerful and complex virtualization environment which can be used to host many virtual machines on an IBM eSeries z/Server. It is the complexity of z/VM that we wish to minimize by providing a graphical user interface through which an administrator can interact with z/VM to perform routine administrative tasks. The purpose of using an intuitive graphical user interface is to allow the administrator to save time by not having to type the many various (and sometimes lengthy) z/VM commands that are needed to properly manage the hosted virtual environment.

### 2. Little or No CP/CMS command knowledge necessary to operate.

To the novice z/VM user, much of z/VM can seem unfamiliar and even unnecessary. Many z/VM commands have a lot of extra options that can be confusing and often require nothing more than a sane default value. In the design of the GUI, care should be taken to ensure that very little CP/CMS command knowledge is necessary to operate the GUI and perform administrative tasks with z/VM.

### 3. Interface should be self documented

Although we will attempt to reduce the amount of complexity the end user must face in order to administer z/VM, it is impossible to remove all of the complexity involved. There are certain tasks and terms or acronyms that the user will need to understand in order to be sure that he/she knows what is happening within z/VM. In such cases the interface should be documented through the use of tool-tips, in-line help buttons or menus, or through other intuitive, easy and non-intrusive methods.

Examples of such cases include acronyms like OSA, QDIO, DASD and CP (Central Processor). Examples of terms the user will need to know are Guest, Virtual LAN, User Directory and Virtual Device.

**4. Security deserves thought.**

It should be noted that the methods that currently exist for interacting with z/VM are insecure. The z/VM 4.4 System management API is based on the ONC-RPC protocol which transmits plain-text user names, passwords and commands over the network. Using the Telnet client to directly interact with a running 3270 session also results in plain-text network traffic. A scheme to prevent this traffic from being packet sniffed will need to be devised.

**5. Portable Program Operation.**

The GUI should be able to run on many different operating systems as different people prefer to use different operating systems. To facilitate this, the Java programming language will be used.

# Program Functionality

Program Functionality is a high level overview of everything we want our product to do. This section should be as specific as possible as to what features we want to have without getting into implementation detail or specific methods of "doing". The specific methods of "doing" will be addressed in the specification documentation.

## Device management

Device Management refers to configuration and modification of devices that are real or virtual. Exampled of such devices include DASD, OSA devices, CP units, and virtual punch card readers. Initially, we will concentrate on a small but critically important subset of devices. This subset includes: { DASD, OSA, CP }.

### DASD Management

A DASD device can be either real or virtual. For example, if a DASD pack with the real address 1021 is attached to SYSTEM, a guest can have a minidisk residing on that DASD that have a virtual device address of 201. In this case, 201 is considered a virtual DASD device and 1021 is a real DASD device. The z/VM GUI Project will be able to work with both types (virtual and real) of DASD devices where it is appropriate to do so.

#### Formatting

This section only applies to real DASD devices. Before a DASD device can be carved up into minidisks or used by z/VM it must be formatted. The command that is used to do this in z/VM is CPFMTXA. We will emulate some of the functionality of CPFMTXA and allow the user to format DASD packs. The user will be able to choose from the following four possible allocation types: PAGE, PERM, SPOL, or TDSK. The user will also be able to set the volume label of the DASD formatted.

The z/VM GUI Project will also support a timed format queue. The queue can be loaded with format jobs that can be batched and/or set to a specific time of day to run (relative to the system that the GUI is run on). The purpose of this is to allow an administrator to batch queue a large number of DASD format jobs so he/she does not have to wait around to start another format after the completion of the current format. A specific time may be set to allow formatting to be started during times when the system will be under minimum load. This will minimize the impact that larger format jobs will have on other users of the system.

The format queue will allow each DASD being formatted to have its own allocation type and volume label. It will also support the auto incrementing of volume labels during batch job creation. This will save time as the administrator will not have to repetitively type many volume labels that are similar. This feature can only be used if you volume label ends in a set of numeric characters (decimal or hex). Example: CL000F or VOL256.

### Cloning

This section applies to both real and virtual DASD devices. Cloning is the ability to copy one DASD device (either virtual or real) to another DASD device of exactly the same type and size. A minidisk must be copied to another minidisk and a full pack must be copied to another identical pack. The user will also be able to specify a volume label that the cloning function will apply to the pack after the copy.

Like the format function of the z/VM GUI Program, the Cloning function will also support a timed batch queue that can be used to automate serial cloning a large number of packs. The Queue will also support the auto incrementing of the volume labels as explained in the "Formatting" section above.

### Attach/Detach

This section only applies to real DASD devices. The user will have the ability to attach and detach DASD volumes to/from SYSTEM and guests on the system. To mimic the behavior of CP, a pack should ONLY be allowed to be detached if no users are currently using that pack. However, an easy method should exists for finding out who is currently using a specific pack.

### Status information

It should be possible to easily view status information for real DASD devices. For a real DASD device, the user should be able to query the device's volume label, owner, and number of active references.

## OSA Management

OSA Management refers to the management of real OSA devices. These devices can either be OSA devices from a real OSA card or from a Hipersockets device.

### Attach/Detatch

The user will have the ability to attach and detach OSA devices to/from guests on the system. To mimic the behavior of CP, an OSA device can be detached even if the owner is currently using that pack. An appropriate message should be displayed to the user.

### Status Information

Allow the user to view all OSA devices both free and used. User will be able to see the real addresses of all free OSA devices and all real addresses, virtual addresses and the owner of all non-free OSA devices.

**CPU Management**

The term CPU (Central Processing Unit) is used instead of CP (Central Processor) to avoid confusion with the term CP (Control Program) which is a piece of the z/VM software. CPU management encompasses information about each CPU in the system and the dedication of one or more CPUs to guests.

**CPU Dedication**

We would like to be able to support the option of dedicating CPU's to guests. However, our current hardware setup prohibits testing this functionality as we only have a single CPU enabled. It may be possible to work out arrangements to test this type of functionality on a properly equipped z/Server.

**Status Information**

This will report the CPU ID of the CPU as returned by the "Q CPU" command.

## Guest management

Managing guests is quite possibly the most important and time consuming task that is performed by a z/VM administrator. Guests must be created with a user name, a password, and a set of devices that can be accessed and used in various ways by the guest. It may also be necessary to either extend or revoke privileges to certain devices for a particular guest. When a guest is no longer needed, it must also be deleted and it's resources reclaimed.

**Creation of z/VM Guests**

Guest creation entails the addition of a definition in the z/VM user directory that states the guests name, password, permissions, and the amount storage the guest is allowed to consume. Entries are also needed that deal with certain aspects of customization of the guest and the devices the guest can access. Here is a list of the directory statements that the z/VM GUI Program will be able to add to a guests directory entry at guest creation time. Capital letters are used to denote required keywords, lowercase letters are used to denote variables to be filled in for a specific guest and letters surrounded by [square braces] are used to denote optional parameters. Specific information on these directives can be found in the appropriate IBM documentation.

USER guestname password init-storage max-storage permissions

guestname – Must be a valid z/VM guest name.
password – Must be a valid z/VM password.
init-storage – Must be a valid storage amount in terms of Megabytes
max-storage – Must be a valid storage amount in terms of Megabytes

INCLUDE profile-name

profile-name – Any valid profile name.

IPL CMS [PARM parm]

parm – Values supported: AUTOCR

MACHINE type

type - Values supported: XA, ESA,

DEDICATE v-addr r-addr

v-addr – Any valid hex address that is unique to that guest.
r-addr – Any valid hex address.

MDISK v-addr FB-512 V-DISK size access-mode

v-addr - Any valid hex address that is unique to that guest.
size – Any valid block size.
access-mode – Any valid access mode.

MDISK v-addr 3390 start-cyl size vol-label access-mode [link-passwords]

v-addr - Any valid hex address that is unique to that guest.
start-cyl – Any valid cylinder not overlapping (with respect to this guest only).
size – Any valid size.
vol-label – Any valid volume label.
access-mode - Any valid access mode.
link-passwords – Any valid link password set.

MDISK v-addr 3390 start-cyl END vol-label access-mode [link-passwords]

v-addr - Any valid hex address that is unique to that guest.
start-cyl – Any valid cylinder not overlapping (with respect to this guest only).
vol-label – Any valid volume label.
access-mode - Any valid access mode.
link-passwords – Any valid link password set.

LINK owner owner-addr v-addr access-mode

owner - Any valid guest name.
owner-addr – Any valid hex address.
v-addr - Any valid hex address that is unique to that guest.
access-mode - Any valid access mode.

SPECIAL v-addr QDIO 3 owner lan-name

v-addr - Any valid hex address that is unique to that guest.
owner – Any valid guest name, or keyword SYSTEM.
lan-name – Any valid guest lan name.

ACCOUNT account-num [dist-code]

account-num – Any valid account string.
dist-code – Any valid string

A user of the z/VM GUI Program will be able to create a guest using all of these directives provided that the use of each directive is correct.

**Modification of z/VM Guests**

The z/VM GUI Program will allow modification of guests directory statements. Deletion and addition of supported statements can be performed graphically while deletion and addition of any unsupported statements must be done manually.

**Deletion of z/VM Guests**

The user will be able to completely remove a guest from the user directory. This removes all directory statements associated with the guest. Optionally, the user may choose to format the guest's DASD packs to prevent someone else from recovering it after the pack is reassigned to another user.

**Guest Performance Management**

The user will have the ability to manage the important of guests in terms of system resources. Through the use of QUICKDSP (quick dispatch) and z/VM's ability to prevent a guest from using more than a specified amount of CPU time and I/O bandwidth, guests performance and access to system resource can be moderated.

# Live System Status.

Live system status will allow the user to see a system summary that will give him/her a sense of how the system and it's guests are behaving. This includes real time monitoring of free storage, used storage, free and used page space and spool space, the number of guests logged on (and their names), and CPU utilization.

Support will exist for non-real time monitoring of channel paths to devices through the use of the Q PATHS and Q CHPID commands.

**Alert Conditions**

Alert conditions may be set. An alert condition is a "statistic / condition" pair that can be used to perform an action that will alert an administrator if the system needs attention. For example, an administrator could monitor storage (the statistic) for the condition that if the amount free drops below 10% then an action should be performed. The type of action that will be supported will be the execution of an external program. In a Linux environment, this external program could be a shell script that is set to send an email to an address that resolves to a cell phone. The end result is that the administrator will receive a phone call from his overly stressed out z/Server.

# Global System Management

This section encompasses everything that applies to the system as a whole.

**System Security**

The user should have the ability to quickly "ban" a guest from the system that he/shee deems a threat to system security. The guest is question will be instantly forced off of the system and the guests password will optionaly be changed to "NOLOG"to prevent that guest from restarting until the administrator lifts the ban.

**Page/Spool Management**

The user will be able to add/remove VM page and spool packs as needed. These additions will be placed in the SYSTEM CONFIG to make them permanent. The z/VM GUI Program will show the administrator how many slots are open for CP OWNED packs. If the number reaches zero, further CP OWNED packs will not be accepted until the number of slots is increased and VM is re-IPLed.

**Scheduler Queue Management**

Through the Q SRM and SET SRM commands, the user will be able to view and set values for the Q1, Q2, Q3 and Q4 dispatch Queues. The user will not be able to input values that are beyond z/VM's limits.

**System Shutdown**

The user will be able to Shutdown and re-IPL z/VM.

**System Wide Message**

The user will be able to send messages to all guests.

**Guest LAN Management**

The user will be able to create and delete Guest Lans. Lans that are created can be optionally entered in the SYSTEM CONFIG to make their presence permanent. If a restricted guest LAN has been created, the user will be able to define users who are able/unable to access the LAN.

**Vswitch Management**

The user will be able to create and delete Vswitches . Vswitches can also be entered into the SYSTEM CONFIG file to make them permanent. The user will be able to specify which guests are allowed to couple to the Vswitch as well as which real NIC couples to the Vswitch if desired.