Naming Your Vista Environments & Repositories

On the Road to Automating Vista Distribution & Standardization

Vista Community Meeting 2019-06-04/05
George Mason University
Fairfax, Virginia

Frederick D. S. Marshall
Vista Expertise Network
rick.marshall@vistaexpertise.net
Repository Classes
(draft, 2019-06-05)

Distribution Repo Classes
- **Au** Gold repository
- **Ag** Silver repository
- **Cu** Copper repository

- Forum server (cathedral) → f/lib/gold
- Vista dev server (balance) → d/lib/silver
- Internet server (bazaar) → i/lib/copper

Standardization
[dialects]

Both
[applications/dialects]

Innovation
[applications]

Production Repo Class
- **Pt** Platinum repository

- Vista prod server → p/run
- Vista prod server → q/run

Production
[systems]

Vista production server (target)
- a/lib/platinum/p
- a/lib/platinum/q

Vista analysis server (subscriber)
Mumps Directory: a set of names within which each name is unique, mainly globals & routines, sort of lock-names & devices.

User Class Identifier (UCI): a Mumps environment, containing a directory of globals, one of routines, & a lock-names and device pool.

Volume Set: a small set of related UCIs.

Mumps Virtual Machine (VM): Mumps as operating system for computer, hiding nonportable details, appearing as an abstract computer.

Machine: the computer hardware controlled by the Mumps virtual machine.
Prototype DHCP Configuration: Plessey desktop machine with one UCI per major application, sign in by selecting UCI.

Vista Cookbook: a manual describing how to configure the Mumps VM & machine to support Vista.

Manager UCI: a Mumps environment containing system-management software plus software shared with other UCIs in volume set.

Mumps Implementor: the person or institution providing or selling a standard-compliant version of Mumps.

Job ID: a value that uniquely identifies this process within a pool of jobs determined by the Mumps implementor; typically per machine.

Extended Reference: to access a routine ^RUNME or global ^DATA in another UCI, do ^[“PHA”]RUNME or write ^[“LAB”]DATA. to access it in another volume set, do ^[“PHA”,”ROU”]RUNME or write ^[“LAB”,”ROU”]DATA.
Original Mumps VM 3: Typical Early Configuration
(draft, 2019-06-05)

Early DHCP Configuration: each major application on a separate PDP-11, users segregated by application. Environments named according to the application they contain.
Prototype Lifecycle Configuration: to develop one application, separate live production code in VAH from code being tested in TST from code being developed in DEV. Unfortunately, shared MGR UCI means development of MGR routines can disrupt production. Environment names differ to help clue everyone in about how carefully to tread in the current environment.
**Improved Lifecycle Configuration**: separating development from testing from production insulates production users from development and testing activities. Even a runaway job that consumes or crashes the CPU can’t hurt production in this configuration. Environment names are usually different, to help make the configuration system by ensuring programmers are less likely to be confused about which environment they are in, so they don’t accidentally do development in production.
**Original Mumps VM 6: Replication/Translation Configuration**
(draft, 2019-06-05)

**Incremental Server Upgrade Configuration**: used to add more horsepower by adding more servers, which share the globals and routines through one of two strategies. Environment names are kept identical, along with the rest of the Mumps database.

**Replication**: global sets & kills on machine 1 copied to machine 2.
**Translation**: global sets & kills on machine 2 reach across local network to machine 1.
**Original Mumps VM 7: Failover Configuration**
(draft, 2019-06-05)

**Failover Configuration:** for 24/7/365 reliable uptime, we need to plan for failure. We set up a failover machine as a safety net, with duplicated contents from the primary the users are on. If the primary crashes, it takes only seconds to switch users over to the secondary, a process called failover, where they can pick up exactly where they left off on the primary, since it was a copy kept in synch by replication. The primary machine can then be repaired or replaced and then reintroduced as the failover machine. Environment names are intentionally kept identical, so the secondary looks to the users as though it is the same machine.
Hosted Mumps VM: Original Mumps Virtual Machine + Operating System
(draft, 2019-06-05)

**Operating System**: at first, a set of features to insulate most Mumps jobs from; increasingly, a set of services to provide portably.
Hosted Mumps VM 2: Cluster Configuration
(draft, 2019-06-05)

Cluster: a group of nodes (computers, “boxes”) that share a database, spreading out the load of the jobs across multiple servers, often specialized.

Data Server: a computer dedicated more or less full time to serving up translated globals to other servers.

Compute Servers: a group of computers dedicated to running Mumps user jobs; almost all global accesses are translated to data server(s).

Print Servers: one or more computers dedicated to running Mumps background jobs (tasks); most globals are translated.

Load Balancing: a system of ensuring no server is underutilized or overutilized.
No More MGR UCI: Eventually, surprisingly late, we noticed there was no longer any reason to have a MGR UCI, because these more powerful computers and Mumps implementations are perfectly capable of storing all applications in the same UCI. Indeed, we had been doing so for some time - we had to - because in the more modern Vista architecture the applications are integrated and communicate closely, which is awkward if they are siloed into separate UCIs. With only one production UCI, there's no point any more to MGR UCIs, nor to %-routines and -globals, so in the late 1990s we finally noticed and eliminated the MGR UCI. %-routines and -globals are slowly being phased out.
Hosted Mumps VM 4: Native Configuration (Datatree & Greystone Implementations)
(draft, 2019-06-05)

The Native Configuration Revolution: what if the Mumps job contained its own virtual machine, instead of the other way around?

Mumps Job: an operating-system process that is running the Mumps VM software.
Directory: an actual operating-system directory containing routine files or global dataset files.
Routine Source File: the source text for a Mumps routine stored in a text file, e.g., RUNME.m or _wfhform.m.
Routine Object File: the compiled binary file for a Mumps routine, e.g., RUNME.o or _wfhform.o.
Global Dataset File: a binary file containing one or more global variables, often grouped into a database segment, e.g., default.gld.

Insert diagram here
**Standard Mumps VM: Jobs, Environments, Systems, & Machines**

draft, 2019-06-05

**Mumps 1995 Virtual Machine:** the new Mumps standard significantly changed the original Standard Mumps Virtual Machine, with an eye toward connecting Mumps environments with each other and with external software written in other languages. The old conventional implementor terminology & syntax was overlaid with new standards. But although this extended the original Mumps virtual machine, it did not take into account the inside-out transformation of the Native Configuration, nor its re-centering in the operating system. Mumps 1995 focused on portability, with hiding the operating system except through defined interfaces, which left Mumps with two incompatible models.

**Mumps Environment:** directory of names, set of names within which each name is unique, applies to globals, lock-names, routines, & devices. **Mumps System:** a job id environment, a set of jobs within which each job id is unique.

**Extended Reference:** to access a routine ^RUNME or global ^DATA in another environment, do ^|“LIV”|RUNME or write ^|“LIV”|DATA. there are no UCIs or volumes, but names can look like them, such as do ^|“LIV,VOL”|RUNME or write ^|“LIV,VOL”|DATA.
Mash VM: Jobs, Environments, Systems, & Operating Systems
(draft, 2019-06-05)

Mumps Advanced Shell Virtual Machine: the new Mumps infrastructure layer needs to marry the Mumps 1995 model with the native-configuration model, to make the operating system visible and central to Mumps computing, to help wrap up and provide access to operating-system services. Mumps needs to fully embrace the larger computing world it is part of, so it needs to standardize how Mumps is configured within that world, to produce predictable, automatable architecture those features can be built upon.

But it also needs to preserve enough of the terminology of the original Mumps Virtual Machine so existing infrastructure software continues to work and make sense to sysadmins. This model maps the original Mumps, Mumps 1995, & Native Configuration models onto a new virtual machine. These are therefore the semantic elements our new Mash standard needs to assign unique names to.
Mash Naming 1: Dialect & Developer
(draft, 2019-06-05)

**Dialect:** Vista codebase managed as a whole & partly compatible w/other dialects; gold repos transport entire dialects; need codes for names

- **vac** CASS Mumps Systems (Veterans Affairs) [FOIA]
- **vad** DHCP (Veterans Affairs) [FOIA]
- **vav** VA Vista (Veterans Affairs) [none]
- **vaf** FOIA Vista (Veterans Affairs) [FOIA]
- **vbr** IHS RPMS (Indian Health Service) [none]
- **vbf** FOIA RPMS (Indian Health Service) [FOIA]
- **vcd** CHCS (Department of Defense/SAIC) [???]
- **vho** Hui Open Vista (Pacific Hui) [???]
- **vmu** MUSTI (Finnish Universities) [???]
- **vmi** MINPHIS (Nigeria) [private]
- **vos** Osehra Vista (Open Source EHR Alliance) [Apache 2.0]
- **vso** Medsphere Open Vista (Medsphere) [???]
- **vst** GFT ("Medsphere") Fileman (gft) [Apache 2.0]
- **vvp** VAPALS (Vista Expertise Network) [Apache 2.0]
- **vvn** Angel (Vista Expertise Network) [VCL 1.0]
- **vvf** FLAP (Vista Expertise Network) [Apache 2.0]
- **vwo** Open Vista (WorldVistA) [???]
- **vwh** Vista Office EHR (Health & Human Services/WorldVistA) [???]
- **vwe** World Vista EHR (WorldVistA) [AGPL 3.0]
- **vxd** vx Vista (Document Storage Systems) [private]
- **vxo** Open vx Vista (Document Storage Systems) [Apache 2.0]
- **vzz** vista dialect undeclared or unknown
- **mdc** Mumps Development Committee (Vista Expertise Network) [Apache 2.0?]
- **mgy** YottaDB (YottaDB) [AGPL 3.0]
- **zzz** not vista-related

**Developer/Development Shop:** Development organization that manages an entire Vista dialect; used to create dialect names

- **va** Veterans Affairs
- **vb** Indian Health Service
- **vo** OSEHRA
- **vs** Medsphere
- **vw** WorldVistA
- **vx** Document Storage Systems

- **va** Veterans Affairs
- **vs** Medsphere
- **vw** WorldVistA
- **vx** Document Storage Systems
**Mash Naming 2: Application, Adopter, Site, & System**  
(draft, 2019-06-05)

**VNNS**: The Vista Name- and Number-space System (VNNS), managed by OSEHRA's Database Administration Committee (DBAC), assigns namespaces & numberspaces to each Vista application, as well as to adopters, sites, vendors, and individual hardhats, which can be used to help develop unique names for UCIs, volumes, and systems. For example, SAMI is the namespace for the VAPALS-ELCAP application.

<table>
<thead>
<tr>
<th><strong>Adopter</strong>: runs one or more Vista production sites, each with one or more Vista production systems</th>
<th><strong>Site</strong>: Vista production organization or location, with one or more Vista production systems</th>
<th><strong>System</strong>: Vista production computer or cluster, running one or more Vista environments</th>
</tr>
</thead>
<tbody>
<tr>
<td>A* Veterans Affairs</td>
<td>APG VA Phoenix</td>
<td>APGSAMI VA Phoenix VAPALS-ELCAP Server</td>
</tr>
<tr>
<td>B* Indian Health Service</td>
<td>BZX IHS Phoenix</td>
<td>BZXSAMI IHS Phoenix VAPALS-ELCAP Server</td>
</tr>
<tr>
<td>G*/I* Department of Defense</td>
<td>??? DOD Hospital X</td>
<td>DOD CHCS Server X</td>
</tr>
<tr>
<td>J* Nonfederal adopter</td>
<td>JVV Vista Expertise Network</td>
<td>JVVSAMI Avicenna (VAPALS-ELCAP Dev Server)</td>
</tr>
<tr>
<td>K* Individual Vista hardhat</td>
<td>KBAF Frederick D. S. Marshall (toad)</td>
<td>KBAF HAN Hanbalur (toad Workstation)</td>
</tr>
<tr>
<td>U* International adopter</td>
<td>UJO Kingdom of Jordan</td>
<td>UJOMOH Jordan Ministry of Health Server</td>
</tr>
<tr>
<td>ZZ* Unknown adopter</td>
<td>ZZZ Unknown site</td>
<td>ZZZZZZ Unknown system</td>
</tr>
</tbody>
</table>

$**System**: Mumps intrinsic special variable uniquely distinguishing this Mumps system from all others

47,avicenna,jvvsami,vista,1 $system for Avicenna (VAPALS-ELCAP Dev Server)

**System Name**: unusual or unique, assigned by site, user friendly  
27 avicenna, vista  

**Mumps DBA Code**: uniquely identifying database administration authority/community, registered with MDC

**Mumps Implementor #**: uniquely assigned by MDC  
47  

**System Version #**: increment when upgrade would affect system performance  
47,avicenna,jvvsami,vista,1 1

43 = Micronetics  
44 = MGlobal  
45 = PFCS Corporation  
46 = Intersystems Corporation  
47 = Greystone  
48 = ESI Technology  
50 = Mumps v1  

**System Code**: user-friendly code uniquely identifying system within DBA community

47,avicenna,jvvsami,vista,1
Mash Naming 3: UCI & Volume 
(draft, 2019-06-05)

**UCI Names:** Short UCI names will follow the old conventions. Long UCI names will be uci,vol and serve as environment names.

**Volume Names:** Using dialect names, adopter names, application namespaces, & version #s, we can construct unique volume names.

**UCI:** old word for an environment, short name of environment, pick from or extend list, describes lifecycle role

- **liv:** live/production environment
- **tst:** test environment
- **dev:** development environment
- **fld:** field-test environment
- **rel:** release environment
- **dem:** demo environment
- **ver:** verification environment

**Volume:** collection of UCIs/environments, aka “volume set”

- **apg:** VA Phoenix Vista environments
- **apgsami:** VA Phoenix VAPALS-ELCAP environments
- **jvva:** VAPALS-ELCAP VA field environments
- **jvref:** VAPALS-ELCAP Vista reference environments
- **jvvhop:** VAPALS-ELCAP Vista Forum environments (Hopper system)
- **jvsvami-18.0-vvp:** VAPALS-ELCAP v18.0 development environments
- **jvsvami:** VAPALS-ELCAP Vista reference environments
- **jvsvami-18.0-vvp:** VAPALS-ELCAP v18.0 development environments
- **jvfhop:** VAPALS-ELCAP Vista Forum environments (Hopper system)
- **jvvc9c:** VEN Central Regional Hospital field environments

**Dialect Source’s Site Prefix:** “j” for nonfederal sites, “vv” for Vista Expertise Network

**Vista Application Namespace:** Vista application developed in these environments

**System Code:** user-friendly code uniquely identifying system within DBA community

**Dialect Code:** target dialect for software, “vvp” = VAPALS-ELCAP

**Application Version #:** version of application developed in these environments
Under the Mash Virtual Machine model, naming the environment includes standardizing how home directories and paths are named.

<table>
<thead>
<tr>
<th>Environment Name</th>
<th>Environment Description</th>
<th>Home Directory</th>
<th>Home Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>liv,apg</td>
<td>VA Phoenix Vista live environment</td>
<td>liv-apg</td>
<td>Now: /home/osehra/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Soon: /home/liv-apg/</td>
</tr>
<tr>
<td>tst,apg</td>
<td>VA Phoenix Vista test environment</td>
<td>tst-apg</td>
<td></td>
</tr>
<tr>
<td>dev,apg</td>
<td>VA Phoenix Vista development environment</td>
<td>dev-apg</td>
<td></td>
</tr>
<tr>
<td>liv,apgsami</td>
<td>VA Phoenix VAPALS-ELCAP live environment</td>
<td>liv-apgsami</td>
<td>Now: /home/osehra/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Soon: /home/liv-apgsami/</td>
</tr>
<tr>
<td>vav,jvva</td>
<td>VAPALS-ELCAP VA Vista field environment</td>
<td>vav-jvva</td>
<td></td>
</tr>
<tr>
<td>liv,jvva</td>
<td>VAPALS-ELCAP VA Vista field live environment</td>
<td>liv-jvva</td>
<td>Soon: /home/liv-jvva/</td>
</tr>
<tr>
<td>tst,jvva</td>
<td>VAPALS-ELCAP VA Vista field test environment</td>
<td>tst-jvva</td>
<td></td>
</tr>
<tr>
<td>rel,jvva</td>
<td>VAPALS-ELCAP VA release environment</td>
<td>rel-jvva</td>
<td></td>
</tr>
<tr>
<td>dev,jvvsami-18.0-vvp</td>
<td>VAPALS-ELCAP v18.0 development environment</td>
<td>dev-jvvsami-18.0-vvp</td>
<td>Now: /home/osehra/</td>
</tr>
<tr>
<td>tst,jvvsami-18.0-vvp</td>
<td>VAPALS-ELCAP v18.0 test environment</td>
<td>tst-jvvsami-18.0-vvp</td>
<td>Soon: /home/dev-jvvsami-18.0-vvp/</td>
</tr>
<tr>
<td>fld,jvvsami-18.0-vvp</td>
<td>VAPALS-ELCAP v18.0 field environment</td>
<td>fld-jvvsami-18.0-vvp</td>
<td></td>
</tr>
<tr>
<td>dev,jvvsami-19.0-vvp</td>
<td>VAPALS-ELCAP v19.0 development environment</td>
<td>dev-jvvsami-19.0-vvp</td>
<td></td>
</tr>
<tr>
<td>dev,kbafvvp</td>
<td>Toad’s VAPALS-ELCAP development environment</td>
<td>dev-kbafvvp</td>
<td>Now: /home/osehra/</td>
</tr>
<tr>
<td>dev,kbafmdc</td>
<td>Toad’s Mash development environment</td>
<td>dev-kbafmdc</td>
<td>Soon: /home/kbafvvp/</td>
</tr>
<tr>
<td>dev,kbafvvn</td>
<td>Toad’s Angel Taskman development environment</td>
<td>dev-kbafvvn</td>
<td></td>
</tr>
</tbody>
</table>
Phoenix VAPALS-ELCAP System “akimel”
generic id = “47 ,akimel,apgsami,vista,1”

Mash VM 2: Sample 1, Phoenix VAPALS-ELCAP Server
(draft, 2019-06-05)

Job 1: VAPALS-ELCAP User
Phoenix VAPALS-ELCAP Primary Server “akimel1”
$system="47,akimel1,apgsami1,vista,1"

Job 2: VAPALS-ELCAP User
Phoenix VAPALS-ELCAP Secondary Server “akimel2”
$system = “47,akimel2,apgsami2,vista,1”

Operating System (Linux)
Phoenix VAPALS-ELCAP Volume “apgsami”
Phoenix VAPALS-ELCAP Live Environment “liv,apgsami”
/home/liv-apgsami/
Phoenix VAPALS-ELCAP Test Environment “tst,apgsami”
/home/tst-apgsami/

Job 1: VAPALS-ELCAP Tester
Mash VM 3: Sample 2, VAPALS-ELCAP Development & Field Servers
(draft, 2019-06-05)
## Repo Naming: A Complete Lack of Standards
(draft, 2019-06-04)

<table>
<thead>
<tr>
<th>Repository Name</th>
<th>Repository Description</th>
<th>Where/How</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILEMAN</td>
<td>Fileman</td>
<td>where should I look to find the application repo?</td>
</tr>
<tr>
<td>VA FILEMAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KERNEL-GTM</td>
<td>Kernel GTM</td>
<td>how are these related to official patches?</td>
</tr>
<tr>
<td>BROKER</td>
<td>RPC Broker</td>
<td>sort in 3 different places in big directory</td>
</tr>
<tr>
<td>XWB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M Web Server</td>
<td>m-web-server</td>
<td>which of these 3 does my environment depend on?</td>
</tr>
<tr>
<td>M-Web Server</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M-UNIT</td>
<td>M-Unit</td>
<td>what syntax should repo names have?</td>
</tr>
<tr>
<td>Random Stuff</td>
<td></td>
<td>what is even in these?</td>
</tr>
<tr>
<td>Random Vista Utilities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Important</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Imagine a directory containing hundreds or even thousands of repository clone directories, from many dozens of sources, inside and outside the Vista community.

Some are the official software distribution directories our environment depends on. Others are local extensions or development directories. They come from countless developers from many different repo hosting sites.

How can we tell which is which? Where do we look for the software we need?

This is a teeny tiny sampling of such a directory to illustrate a few of the issues with the status quo.
Mash Naming 5: Gold, Silver, & Copper Repo & Clone Names  
(draft, 2019-06-05)

Home Path to Repos

<table>
<thead>
<tr>
<th>Gold Repo</th>
<th>Silver Repo</th>
<th>Copper Repo</th>
</tr>
</thead>
<tbody>
<tr>
<td>/home/dev-jvvsami-18.0-vvp/lib/</td>
<td>standardized, automated distribution of entire vista dialects for production installation</td>
<td>non-standardized, manual distribution of software for any purpose</td>
</tr>
</tbody>
</table>

Gold Repo & Clone Names

Dialect Code: FOIA Vista (Veterans Affairs) [FOIA]  
Dialect Name: FOIA Vista (Veterans Affairs) [FOIA]  

Silver Repo Clone Names

Software Namespace: computable code describing repo contents, namespace  
Software Name: human-readable name of repo contents, lowercase, hyphens, short  
Action Code: r = read-only, a = active development (r/w)  
Software Source Code: hardhat namespace or vista dialect source code or etc., vo = osehra, kban= sam habiel, etc.  
Software Source System: name of repo host system.  
Software Source Username: username of user at repo hosting system.

Copper Repo Clone Names

Remote Repo Name: because the remote repo is not named _ut-m-unit, the actual name is included
## VAPALS Repo Clone Names

*(draft, 2019-06-05)*

<table>
<thead>
<tr>
<th>Home Path to Repos</th>
<th>Repo Type</th>
<th>Repo Clone Directories</th>
</tr>
</thead>
<tbody>
<tr>
<td>/home/dev-jvvsami-18.0-vvp/lib/</td>
<td>gold/</td>
<td>none at present, coming in a little over a year for vvp-vapals dialect repo</td>
</tr>
<tr>
<td></td>
<td>silver/</td>
<td>a-mdc-mash--vv-git-mumps</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a-sami-vapals-elcap--vo-osehra-github</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a-vvp-vapals--vv-git-mumps</td>
</tr>
<tr>
<td></td>
<td>copper/</td>
<td>r-vos-dist-osehra-vista--vista--vo-osehra-github</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r-_web-m-web-server--kban-shabiel-github</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r-xugtm-kernel-gtm--kban-shabiel-github</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r-xv-vpe--kban-shabiel-github</td>
</tr>
</tbody>
</table>

The goal in a nutshell is a schema that can:

1. manage thousands of closely related repos
2. without any confusion about what's in each one or how we use it
3. support visually scanning a massive directory full of clone names
4. with names that lend themselves to automation
5. to support software knowing which clones contain the software updates they need, without human intervention
6. while ensuring we can get to tab-completion in the fewest possible characters
7. to keep the names practical for use by sysadmins
8. while also ensuring we can unerringly locate the remote repo
9. and that we know what person or institution is responsible for the repo's contents.
10. It must also store its contents in a standardized structure, to support automation (see next slide).
Mash Naming 6: Repository Structure
(draft, 2019-06-05)

Status Quo: Vista repos follow no detectable standard in terms of where package elements are stored nor in the syntax used for directory and file names. Processing of such free-for-all repos is therefore strictly manual, labor- and expertise-intensive. To replace this state of affairs with automation therefore requires standardizing repo structures as well as names.

Repo Path

/home/<env>/lib/<category>/<repo-name>/

Intermediate Standard

- admin/ unix scripts
- kids/ kids distribution files
- docs/ documentation, notes, diagrams
- elements/ non-routine software elements
- globals/ global distribution files
- java/ java source files
- routines/ mumps source files

Emerging Standard

- dist/ sami-18/ semver release 1.18.0
- sami-18-s1-p1/ semver release 1.18.1
- sami-18-s2-p3/ semver release 1.18.2
- sami-18-s3-p2/ semver release 1.18.3
- src/ docs/ www/
- elements/ java/ routines/
- unix/ bin/ etc/

Copper: no standard structure, bazaar, freedom, manual
Silver: as described below, balanced, automatable
Gold: each subdirectory named & structured like a silver repo, cathedral, automated
Platinum: like src directory below, balanced, automated
**Status Quo**: Automating Vista distribution also requires standardizing the home directory structure of Vista environments.

<table>
<thead>
<tr>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/home/dev-jvvsami-18.0-vvp/</code></td>
<td>Home directory root.</td>
</tr>
<tr>
<td><code>data/</code></td>
<td>Holds all the environment's live running data, including the backups and journals used as the safety net for that data, though those two are often mount points for other drives.</td>
</tr>
<tr>
<td><code>backup/</code></td>
<td></td>
</tr>
<tr>
<td><code>globals/</code></td>
<td></td>
</tr>
<tr>
<td><code>journals/</code></td>
<td></td>
</tr>
<tr>
<td><code>lib/</code></td>
<td>Lib directory holds all incoming software; if this is a development site, then some of these repos are for outgoing software, while others are bidirectional; standardizes Vista distribution, so it can be automated.</td>
</tr>
<tr>
<td><code>copper/</code></td>
<td></td>
</tr>
<tr>
<td><code>gold/</code></td>
<td></td>
</tr>
<tr>
<td><code>silver/</code></td>
<td></td>
</tr>
<tr>
<td><code>run/</code></td>
<td>Run directory holds the running software, which is then pushed out to the environment's platinum repository (excluding the object folder) to support environment-to-environment, as well as doing version control on what is and has been actually running in this environment; standardizes Vista standardization, so it can be automated.</td>
</tr>
<tr>
<td><code>docs/</code></td>
<td></td>
</tr>
<tr>
<td><code>&lt;app-namespace&gt;/</code></td>
<td></td>
</tr>
<tr>
<td><code>objects/</code></td>
<td></td>
</tr>
<tr>
<td><code>java/</code></td>
<td></td>
</tr>
<tr>
<td><code>&lt;app-namespace&gt;/</code></td>
<td></td>
</tr>
<tr>
<td><code>routines/</code></td>
<td></td>
</tr>
<tr>
<td><code>unix/</code></td>
<td></td>
</tr>
<tr>
<td><code>bin/</code></td>
<td></td>
</tr>
<tr>
<td><code>etc/</code></td>
<td></td>
</tr>
<tr>
<td><code>www/</code></td>
<td></td>
</tr>
<tr>
<td><code>&lt;app-namespace&gt;/</code></td>
<td></td>
</tr>
<tr>
<td><code>tmp/</code></td>
<td>Temp directory is for scratch data.</td>
</tr>
<tr>
<td><code>var/</code></td>
<td>Var directory is for highly variable data files, such as logs, print queues, and so on.</td>
</tr>
<tr>
<td><code>log/</code></td>
<td></td>
</tr>
<tr>
<td><code>queues/</code></td>
<td></td>
</tr>
<tr>
<td><code>&lt;printer-name&gt;/</code></td>
<td></td>
</tr>
</tbody>
</table>
1. **VAPALS Increased Compliance**: In the past couple weeks we significantly improved the compliance of the sami-vapals-elcap repo with this emerging Mash standard. During June 2019, we will bring it and the home directory fully into compliance, before distribution to our first VA system at Phoenix.

2. **Automation for Renaming Environments**: We are partway through writing a new routine, ZTMRENAME, to run while an environment is shut down to change the value of every field in Vista that holds a UCI, volume, or system name, to bring them into compliance with the environment’s proper name. We will then write a script to move the home directory, change the startup scripts, and so on to bring its unix elements into compliance with the proper system and environment names. We will then develop files and routines to ask the sysadmin a few questions and use those answers plus a database of namespaces to calculate the proper name for the UCI, volume, environment, and system. Together, these pieces of automation will make it practical for the first time to rename an environment without causing innumerable negative side effects - and without having to be an expert in Mash’s naming standards. This software will be distributed in the VAPALS Dialect Repo (vvp-vapals) and MDC Mash Repo (mdc-mash) and VAPALS-VA Dialect Repo (vva-vapals-va) in time for the release of patch SAMI*18*1, before distribution to Phoenix.

3. **Automation for Upgrading Environments**: Over the course of the next year and a half, to be released incrementally, we are iteratively developing new versions of KIDS and the Patch Module to support keeping our VAPALS servers up to date with Vista patches and new application versions without requiring a Vista expert to do the patching. In support of this software, we will be creating and operating what as far as we know is the first fully automated Vista patch stream outside VA, for our VAPALS dialect, which will distribute upgrades to our SAMI software and upgrades to other Vista applications (including preserving our dialect-specific Vista modifications) for installation on our VAPALS servers. We will simultaneously create and operate an MDC-specific patch stream for distribution of upgrades to the Mumps Advanced Shell, for installation on our VAPALS servers.
Coming Soon 2: Patch Stream Architecture
(draft, 2019-06-05)
Thank You!

Vista Community Meeting 2019-06-04/05
George Mason University
Fairfax, Virginia

Frederick D. S. Marshall
Vista Expertise Network
rick.marshall@vistaexpertise.net