

Projects to Improve Software Patch Installation Efficiency

Issue

In an effort to offer VAMC sites methods to improve VistA and HealtheVet software patch installation efficiency, several proposals are currently on the table for review. There are generally two groupings of proposals:

- Proposals 'A' and 'B', and now 'enhanced B' from the VistA Maintenance Challenge
- New Service Request (NSR) for an 'Installation Wizard'

Some involved with both proposals raised concerns:

- Due to various features and dependencies can all of these proposals coexist?
- If they cannot coexist, or if resources do not permit developing all of these proposals simultaneously, what priority should be used for development to meet OI management's expectations for these projects?

Background

The VistA Maintenance Challenge was offered to the VHA IT community in an attempt to improve patching efficiency. Two proposals were deemed winners:

- Proposal 'A' submitted from VA Developer.
 - An 'automatic patching system'
- Proposal 'B' submitted by VA Developer.
 - A 'master-servant patching system' to permit one site to facilitate patching at many 'servant' sites

Currently Proposal 'A' has been issued as a New Service Request, with the request to incorporate Proposal 'B' in the not too distant future. Since this began the submitter of Proposal 'B' has added some features from another VAMC IT person and is referred to as 'enhanced Proposal B'. It should be noted that Dennis has now joined HSITES Enterprise VistA Support on the Clinical Systems team.

More details of these proposals and how they may fit together or where concerns will require resolution are described in Attachment A.

The NSR for the Installation Wizard was submitted by HSITES based primarily on concerns and complications involving the installation process raised during the release of Care Management. This NSR is in review and has identified generally 3 areas of concern:

- Single Client Installation
- Multiple Client Installation
- N-Tier Installation

Attachment B is a White Paper for IDMC review further describing this project. Recognizing potential conflicts with the N-tier Installation, the VistA Challenge Proposals were referenced in this Installation Wizard White Paper as follows:

- 4.vi. Coordinate the N-tier analysis with on-going projects, specifically the VistA Maintenance Challenge, to improve KIDS patch installation.

Recommendation

If these proposals are reviewed separately, crosscutting issues may not be readily apparent. Also, a priority of expected outcomes is needed.

Recommend both general types of proposals and all their related components be passed to Development & Infrastructure Support (DaIS) for total analysis to determine what features may coexist, what features may have to be set by parameters or lost due to conflict, and technically, what order the components of the proposals must be done in.

Recommend the service requesters offer a priority listing of desired outcomes from these proposals. The final recommendation to the IDMC concerning all proposals should include such input from the proposal requesters (e.g. OI upper management) as to desired priority.

Attachment A.

VistA Challenge Proposals

1. Proposal A

Proposal A is an 'auto-load' process for patches received via FORUM. Although all patches cannot be auto-loaded, there are site files that can be used to turn certain packages off from being auto-loaded. Therefore, certain patches could auto-load and others would require human intervention.

If certain variables are present, proposal A makes changes to national patching routines to allow the package to 'kick in'. This will allow the package to lie dormant if it is not 'turned on' and patches could be installed as normal. If a local field in the package file were set to enabled, then patches would auto-install when the variables show up in the routines. Proposal A saves copies of the verify process and the install process in a directory on the hard drive for each patch that gets installed and a backup of the install is kept in a mail box of the person running the patch.

2. Proposal B

Proposal B was the master-servant patching process. What the VistA Challenge asked for was a package that could be used by sites to patch other sites remotely. With this understanding, proposal B took it to mean that a person from a site might be responsible to patch several sites remotely. In researching this it was determined that if that were true, and sites could open themselves up and allow other site(s) to patch them, however you should make sure only the authorized site would be able to send patches that would auto-install. To make sure that only one site could do this, proposal B suggests the idea of a master site that could send patches to self-install to a subordinate or 'servant' site, much like master-client servers.

Proposal B worked around this concept being relatively certain that a package could be built that would take care of establishing a master-servant relationship of one site over another with site parameters that would be set up allowing only the master to send patches for install to the servant under pre-specified conditions. The messaging that occurred between the two sites would be recorded in files, and reports could be generated that would allow a user to quickly see which patches had been sent from the master to the servant, and to see the status of the auto-install of those patches. Once this master-servant package was set-up, it was determined that a package could be developed that would run all the components of a standard patch install 'auto-magically' by calling the same entry points in the KIDS package with pre-defined variables, and the patching process would proceed as if the patch were being run interactively.

3. Proposal B Enhancements

The enhancement to proposal B is much like a package that the White River Junction VAMROC currently uses. This package keeps track of patch messages that arrive on station in the message file. A server would populate a file with information about patches when they arrive and would calculate when they were due, and then establish links from this file to the install file when the patches were installed. The purpose of this package is not only to determine when patches are due and to notify IRM personnel when patches are going to be late, but also to come up with statistics at the end of the month to determine IRM's compliance level with patch install dates. As mail messages are purged the patch file would still keep information about the message so a trail of the date the patch arrived on station and when it was installed would still be on record, though the actual message no longer exists and, therefore, the record of arrival died with the message.

Proposal A differs from the proposal B in that you would only turn on auto-patching or not turn it on. If turned on, patches would be installed as they came out from FORUM.

4. Combination of all three proposals

Proposal B would work nicely for someone who is trying to catch a system up if it had fallen behind on patches. Patches could be sent off to the site and loaded automatically saving time over manual installation of one or more sites. A trail of the correspondence between the sites is kept and a site could remotely check on another site's patch status.

The enhancement to proposal B tracks local patching, keeping a site wise to due dates and would be able to come up with numbers during or well after the patching fact. This package does not install patches, only keeps track of them.

Proposal A would be great to allow a site that is already patched to get a number of their patches installed automatically and, thereby, allow the site to focus on only the more complicated patches that need human intervention.

The optimum package would be a combination of all three packages. You could have the proposal B enhancement package, or a package like it, to keep track of local patching, and then have the choice of whether you would like to use auto-patching to automatically do your patching, or if you want another site to do patches remotely, or if you want to install the patch yourself.

5. Issues

As anyone who has done patching could probably tell you, either of the two self-installing packages described above would not be perfect. There are many patches that would have to be manually installed. For one thing, any kernel patch that would require a running of ZTMGRSET would have to be hand installed. Any patches that would require routines to be moved to another UCI would have to be hand installed. Any patches that require files to be manually populated would need to be hand installed. Patches that

need mail groups to be set up, domains to be created, site parameters to be entered, options queued up in Taskman, etc. could not be auto installed or remotely installed.

Using a term heard in IT shops, this practice has been referred to as Patch Slamming, because you are in effect 'slamming' a patch in rather than installing it. A large number of patches could be 'slammed' but a certain percentage would not want to be 'slammed'. For this reason, a person would have to read up on patches prior to slamming them in. For test accounts that are way behind on patches, slamming might be perfectly acceptable. Many test accounts are not fully used and if a mail group or two was unpopulated or a job not scheduled up in Taskman it would probably not cause any dire consequences. Live systems on the other hand would be a different issue.

Another issue that was written into proposal B is a patch sequencer to make sure that patches are queued up in sequence order. If not they might be installed out of order. Proposal A by virtue of the way it installs patches as they come in would take care of this as long as patches come out in the proper sequence.

These and similar issues would need to be resolved regardless of individual or combinations of proposals as all were not considered in the original proposals.

Attachment B.

Installation Wizard Proposal (Draft)

1. Overview

The purpose of this document is to solicit a high level review by the Informatics & Data Management Committee (IDMC) Screening Committee of this new Information Technology (IT) Service Request. Upon review and discussion, the committee should determine if the information in this paper supports immediate approval or disapproval of one or more of the analysis projects outlined in this White Paper. If the information provided is not sufficient to make such a determination, the committee should instruct Planning and Analysis to proceed with a full analysis of the request, which shall be presented to the Screening Committee at a later date.

2. Description of Request

Health System Implementation, Training, and Enterprise Support (HSITES), is requesting the creation of a generic installation wizard to ease installation of all re-hosted and re-engineered HealthVet products. This request was submitted in response to recent experiences with the installation process for Care Management (Java client/M server dual installation) as it was found to be far more complicated than the typical Kernel Installation and Distribution System (KIDS) (M only) installation process. While some partial tools were available, they were not complete. The ultimate goal is to make it easier for sites to install HealthVet products. This White Paper does not propose a solution, but rather identifies three areas that need further analysis to reach this goal.

3. Pertinent History and Background

Both client and n-tier installation issues need to be addressed. In terms of client installation, the process for installation of the application on one client needs to be addressed as well as subsequent installation on multiple clients. These can be conceived of as either two separate projects, or one project with two phases. Improvement of the installation on one client is seen by the Requester as the first priority and will be easier to identify a solution to this issue.

HSITES is conducting an evaluation of Microsoft's System Management Server (SMS) 2003, a remote tool that can be used to deliver (push) updates to multiple clients. The evaluation of multiple client installation issues should not be limited to an evaluation of SMS as any single installation method has limitations. For example, CPRS is not backwards compatible. Installation of client components may not work properly unless server side components are also updated. This must be accomplished in a timely manner and in a particular order. To ensure compatibility of all components an additional tool that will update clients on demand is also needed. Multiple approaches should be evaluated such as the enhancement of the Module Update Manager (MUM) developed

for the Care Management project, use of a commercial tool such as Java Web Start, or development of an in-house application programmer interface (API).

Installation on multiple clients does not necessarily mean that installation is needed on all machines at all sites. Sometimes a patch is installed as a test and it only goes to a sub-set of sites, and sometimes to only a sub-set of the equipment on those sites. Some software is Food and Drug Administration (FDA) controlled and can only be installed after certain administrative issues have been addressed. These are examples of situations that need to be considered in the evaluation.

In terms of the n-tier installation process, there are many interdependencies that need to be evaluated between Vista, M, Kernel, relational database (Oracle), Java 2 platform, Enterprise Edition (J2EE), web server and a repository for client pieces. There is a need to look at how all of these pieces fit together. The n-tier project area will likely be addressed later than client installation, once more of the issues about the n-tier have been resolved.

4. Description of Project

The purpose of this request is to identify ways to install new software and software updates in a timely manner, to reduce the amount of time information technology (IT) and clinical staff spend on installations, and to find ways to validate that new installations and updates have occurred on all desired machines. Analysis of client and n-tier installations is proposed. The high-level issues that need to be evaluated are identified.

a) Client Installation:

- i. Conduct a thorough evaluation of the current client implementation process to identify shortcomings.
- ii. Gather requirements to address shortcomings in client installation identified through the evaluation process.
- iii. Coordinate this analysis with the pilot evaluation of Microsoft's SMS 2003 being conducted by HSITES. Determine the need to evaluate other platforms such as Virtual Memory System (VMS).
- iv. Possibly identify criteria for selection of a commercial product(s) that could be used for the development of single and multiple client installation wizards.
- v. Possibly determine what in-house development will be needed, as a substitute for or in addition to a commercial product(s), to create single and multiple client installation wizards. This could include evaluation of the MUM to determine if it can or should be enhanced for multiple client installation. Evaluation to determine if development of an in-house API would be a viable alternative to purchase of commercial product(s) might also be addressed.
- vi. Possibly conduct a market survey to determine which product(s) would be most appropriate for the development of single and multiple client

installation wizards. The extent to which Windows tools can be leveraged would need to be included in this evaluation.

- vii. Possibly conduct laboratory testing, if appropriate, on candidate products.
- viii. Develop a report with recommendations on ways to improve single and multiple client installation processes.

b) N-tier Installation:

- i. Conduct a thorough evaluation of the current n-tier implementation processes to identify shortcomings.
- ii. Gather requirements to address shortcomings in the n-tier installation identified through the evaluation process.
- iii. Identify the tiers that will be included in the evaluation, such as VistA, database server(s), and application server(s).
- iv. Evaluate interdependencies between the tiers. This would likely include, but not be limited to VistA, M Kernel, relational database (Oracle), J2EE, web server and a repository for client pieces.
- v. Identify the features needed for n-tier installation wizard tools such as dependency checking, user interfaces, specific file formats, and version checking. A high-level review of the features currently available in KIDS may assist with the identification process.
- vi. Coordinate the n-tier analysis with on-going projects, specifically the VistA Maintenance Challenge, to improve KIDS patch installation.
- vii. Possibly identify criteria for selection of a product(s) that could be used for the development of the n-tier installation wizard(s).
- viii. Possibly conduct a market survey to determine which product(s) would be most appropriate for the development of the n-tier wizard(s).
- ix. Possibly conduct laboratory testing, if appropriate, on candidate product(s).
- x. Develop a report recommending ways to improve n-tier installation processes.

5. Assumptions/Issues

- a) Even with tools, installation will not be simple. Additional training/skills outside of HSD&D will be needed to support appropriate system administration for new technologies such as J2EE and Relational Database Management System (RDMBS) management. Either staff with the necessary skills will need to be hired and/or our staff will need training to perform the installations.
- b) Development & Infrastructure Support (DaIS) has been the group historically involved in this sort of development, however the needs of HealthVet cross OI efforts. Creation of a full solution for automation will require active involvement from all parts of OI.
- c) It is probably not possible to create a “one size fits all” software packaging and distribution process through the use of one product. VistA and HealthVet VistA applications contain an extensive mix of technologies. What works well for one system may be cumbersome for another. It is expected that several different

products/approaches will be identified to streamline the HealtheVet installation process.

6. Proposed Approach

Evaluation of the HealtheVet Installation request indicates that the level of analysis required by this project is beyond the scope of a Planning & Analysis Service project. A thorough analysis of the client and n-tier installation issues is needed before decisions can be made about what installation tools should be added and if they should be bought and/or built. The areas that need to be analyzed are briefly described in the Description of Project section of this document. Other areas may be identified through this analysis.

The VistA Foundations Team, with responsibility for meeting J2EE application tool and infrastructure needs, is one group that could coordinate this analysis effort. They estimate that the client implementation would take approximately six months to complete; analysis of n-tier implementation is estimated at between six months to one year. These estimates are based upon the assumption that dedicated staff are assigned to conduct the analysis. If this project were given to the VistA Foundations Team it would impact work on HealtheVet Desktop and VistA Link projects.

The Infrastructure Team is another group that might be assigned responsibility for coordinating this analysis project. It should be noted that the Infrastructure Team did not participate in the preliminary analysis or the development of this White Paper. Another possible candidate is the Common Services Team, however this project does not seem to fit their focus on business services, such as Person/Patient service and middleware that supports the transfer of data.

In addition to assigning a team responsibility for the overall coordination of the project, due to the complexity of the proposed analysis, wide representation across the Office of Information is required. Participants could include, but not be limited to include the ad hoc Development Technical Advisory Committee (DTAC), Enterprise Management Center representatives coordinating the evaluation of SMS, development teams such as VistA Imaging, CPRS and Scheduling, as well as Enterprise VistA Support representatives and users.