

Minutes from National iMedConsent™ VANTS Call

Wednesday, September 3, 2008

Ray Frazier, National Center for Ethics in Health Care

1. Upcoming Release

Beta volunteers are being sought for the “Otter” release. This release will include some fairly extensive content updates and new features such as a “Content Request” link on the main page which will allow end-users to submit content requests and questions directly to Dialog Medical. There are also major changes to the chemotherapy consent as well as a new education documentation tool for most education documents. Refer to the forthcoming Release Notes for details.

2. Policy and Form Update

No updates available on the timeline for release of the following forms, publications, and enhancements:

- Informed Consent Handbook revisions (including blank consent form templates and changes to the consent form expiration, witness requirement, and “who can sign”)
- Opioid Agreement (aka Pain Agreement)
- CCOW enhancement (eliminate iMed sign-on)
- Signature for receipt of drugs in pharmacy

Oxygen safety documents will be released in the “Python” update (the patch after “Otter”)

3. iMedConsent Handbook

This Handbook is nearing entrance into the formal concurrence process in which other VACO program offices will have the opportunity to comment on the proposed Handbook.

4. Signature Pad Issues

Wireless signature pads have not yet been approved by OI&T. As soon as this process is complete and pads are available for purchase, I will announce it on the listserv.

I have made yet another request that the guidance for purchase of signature pads with medical money (as opposed to IT funds) be formalized in a memo or other directive to the field. Again, word at the national level is that signature pads, like barcode scanners, should be purchased with medical funds.

5. Technical Updates From Bill Taylor

Bill Taylor from Dialog Medical was kind enough to summarize a response to several technical issues that were raised on the September call. I am including this response below (thanks Bill!!):

Upon SaveToChart, the workstation is supposed to transfer a .txw file and an .xml file from the client to the server's DropBox. If the transfer of the pair is successful, the consent processes normally. If the transfer is unsuccessful, the pair are supposed to be cached on the workstation, under the currently-logged-on user's profile at ...\\%username%\Application Data\Dialog Medical\iMedDoc*.xml (or .txw) until such time as the same user, same workstation next does a consent, at which time they're supposed to be flushed with the next data transfer and allow the original consent to be processed.

This can be problematic if/when: a) there is an immediate need for the consent (e.g., same day surgery) and the current user may not do another consent on that same workstation for days/weeks; b) the user is logged into a thin client server which does not retain the user's profile, and/or c) only one of the pair is transmitted. What we see most often is that the .txw file transfers successfully, but the .xml file does not and is left on the client machine. This requires the site to have to do a search of the client workstation and manually find and reunite the pair before the consent will process. We first thought that this might be more of a wireless issue, but we have seen it occur in many wired scenarios as well. From a support standpoint, even if we are remoted to the server, we do not have administrative access to search and retrieve the file from the user's profile, thus requiring administrator assistance. We usually clear the .txw files from the DropBox (to make it easier to see what is currently being received), but place the orphaned .txw files in a /Temp folder until they can be reunited with their missing .xml counterpart.

I'm currently working with one of our developers to see if we can improve the reliability of the transfer (i.e., it should only be considered 'complete' if/when it successfully transfers both of the files, not if it successfully transfers one or the other, and to see if we can improve the visual/aural notification to the user so that they can immediately re-attempt the transfer at the moment the transfer fails. This will improve the workflow from a standpoint of having all parties present in case they need to re-consent/re-sign, providing the consent in the most timely fashion, reducing the time and effort that administrators are having to spend to search for the missing file(s), and reinforcing the best practices in the event that something the user did contributed to the issue (e.g., you're more likely to associate the fact that moving the Flo-cart back from the bed after saving to chart breaks the wireless connection if you receive an alarm at the time rather than trying to remember what you did a week later when it's discovered that the consent never came through.).

Second issue... Boston is seeing a large number of job failures, especially during the hours of 9 – 11 AM EDT (and more especially, it seems, on Tuesdays). They (or we) resubmit the jobs, and they usually go through on first resubmission... but if not on first, certainly on second or third. They are, admittedly, doing a higher volume of consents than ever before, now that they've begun using the Receipt of Pharmacy form, and they are using multi-threading (the server works on more than one consent

at a time) – but we’re not seeing the actual server resources taxed beyond limits, and the issue continues to occur even if/when I turn multi-threading off. We’ve explored other possible causes: bumping against a maximum number of concurrent users (they have more than adequate ceiling to handle this), packet monitoring to see how/when packets are being dropped, etc., but have yet to pinpoint the exact cause. I’ve talked it over with David [Sommers] and a few of our developers, and they feel that it’s just that VistA handles very well the occasional user authenticating against the server when they log on... but handles very poorly the steady barrage of authentications supplied by our document service during peak hours. We’re exploring how we might improve the way we connect, or even reduce the number of connections per consent we make to try and ease the pain.

Lastly, because not all sites use Interlink Electronics ePads and we didn’t want the program to slow while it searched for the correct driver for the signature capture device to use, we designed, within the administrative section of the application, a pull-down whereby the site could set which type of signature device their site most predominantly used. This selection would move that particular driver to the top of the list, so to speak, and make it that much faster for the application to find and initialize that driver. This would not prevent the site from also using some other type of signature capture device with its own driver; merely, it would take just a little longer for the application to have to search through the list of drivers to find the appropriate driver for the non-primary device. This setting can be found in the application by going to Maintenance > Preferences > System Options > Documents and using the Signature Device pull-down (see figure below).

