

DSR Reproducible Harvest Functional Specification

Overview

- This functional specification outlines the expected user interaction and experience with the DSR Reproducible Harvest application. It will define the different actors and will walk through examples of how each actor will interact during the process.

Summary

- In summary, the DSR Reproducible Harvest is an application that will perform the following actions:
 - Install on the UF Dev, Staging and Production web servers by downloading a package from Source Forge, unpacking, and following and installation and configuration procedure
 - Automatically harvest UF DSR data on a scheduled basis
 - Email group of end users the status of each harvest

Roles

Name	Description
Developer	Patches software
System Administrator	Installs, configures, monitors
Data Curator	Reviews data for accuracy
End User	Views their data

- **Developer**
 - Jim is the initial of the DSR Reproducible Harvester project. He wrote the software and may or not maintain the software beyond its release. However, the software is open source and publicly available on the VIVO SourceForge web site, which allows anyone to contribute changes to it as necessary. Kelly, a developer at University of Washington, is running VIVO and wants to implement Reproducible Harvesting of DSR data. However, she has found that some modifications to the software might be a good idea to meet her needs, as well as the needs of others interested in using the software. Kelly then joins the project, contributes her patches to the source code, and tags it as a new release. If the community feels they conflict with the requirements of the tool, then they may require her to fork the software. Or, they may simply accept the patch for future releases.
- ***System Administrator ***
 - Rank is the system administrator at UW and will install and configure both the VIVO Harvester and the DSR Reproducible Harvester Scripts. Frank finds documentation on the SourceForge wiki about a utility that will harvest DSR data in an automated fashion. He decides to research the utility by looking at the install scripts. The installation and configuration looks fairly easy. He already has the Harvester 1.x installed, so this should be a breeze. He shells into his server and follows the installation instructions. After installing and configuring the program, Frank decides to execute the DSR harvest manually instead of waiting for the cron job to run. After it runs, he gets an email showing the status of the harvest he just ran. He finds an error and modifies his configuration settings to repair the process. He then runs the script again manually. It runs successfully. He logs into VIVO through the web interface and sees that his DSR data was indeed harvested. He then waits for the cron job to run to verify that it runs in an automated fashion. After it runs successfully, he continuously monitors future harvesting runs and ensures the system continues functioning properly
- **Data Curator**
 - Tom is the data curator and reviews the data after every new harvest for accuracy and quality. He is responsible for reporting any problems back to the system administrator or the developer. He simply logs into the VIVO web interface to ensure data is being harvested correctly.
- **End User**
 - Dr. Smith is a faculty member at UW. He hears about VIVO at his institution and uses VIVO to look for grants related to his research. He also uses VIVO to share information about his grants with other faculty members. Dr. Smith logs into VIVO, loads his profile and sees that he has a grant automatically added to his profile that he is key personnel on.

Data Flow

