

Implementing a Database in Dell Repository Manager and its Advantages

A Dell Technical Paper

Dell, Inc.

Dell Repository Manager Team



Implementing a Database in Dell Repository Manager and its Advantages

Introduction

"Information is a source of learning. But unless it is organized, processed, and available to the right people in a format for decision making, it is a burden, not a benefit." - William Pollard

The IT industry often feels that burden. Much of our efforts in the IT industry are spent organizing data in such a way that it is simple to find, convenient to store, and quick to retrieve.

Dell Repository Manager (DRM) focuses on Dell's software updates for server and client systems. The data it manages consists of software updates and its metadata. It provides a tool for organizing those updates using the metadata.

Those familiar with DRM know that it already provides management for Dell's software updates, but DRM has taken another step to better organize Dell's updates. It has moved the repositories it manages to a database. With the database some changes to the User Interface was also necessary.

XML vs. Database

Why the switch from an XML document to a database? XML documents are formatted text documents. They are marked up with tags to give the data meaning and structure. It is ideal for transporting data, as in the case of the catalog file. It is self-describing. It does not contain any executable code. It is relatively easy for most applications to read.

Databases are a data store that is meant to manage large amounts of related data. It is formatted but a database engine is needed to read the data.

Currently, all of the Dell Update Catalog is found at <ftp://ftp.dell.com/catalog/>. The catalog is found as catalog.xml for server systems and catalogPC.xml for client systems. DRM 1.4 would download that XML file, read the data, and allow users to create their own repositories from the catalog. Subsequently, they would save their repositories as XML files to be read the next time DRM 1.4 opened.

Downloading a XML document from the FTP site is ideal for distributing the catalog, for all of the reasons mentioned above. DRM 1.4 needs to read the catalog.xml and allow the user to manager the data from there. The location of files and other important data can be used to download the necessary files. The fundamental problem is that every time DRM 1.4 loads, it needs to load the repository's catalog file again, which it needs to access from the FTP site. This process can be interrupted by network connectivity problems. It also needs to reinterpret the data again every time.

The solution that DRM offers is a database that stores all the repositories. This gives it the ability to store any catalog it reads into a repository and to download the files it needs. The catalog data does not need to be reloaded or reinterpreted. All changes made to the database stay after exiting DRM.

Changes to Data Manipulation DRM

Moving to a database changes how the data is read, stored, and processed, but much of the same basic functionality found in previous versions of DRM is available in DRM. Yet there are some differences in the User Interface that are worth noting.

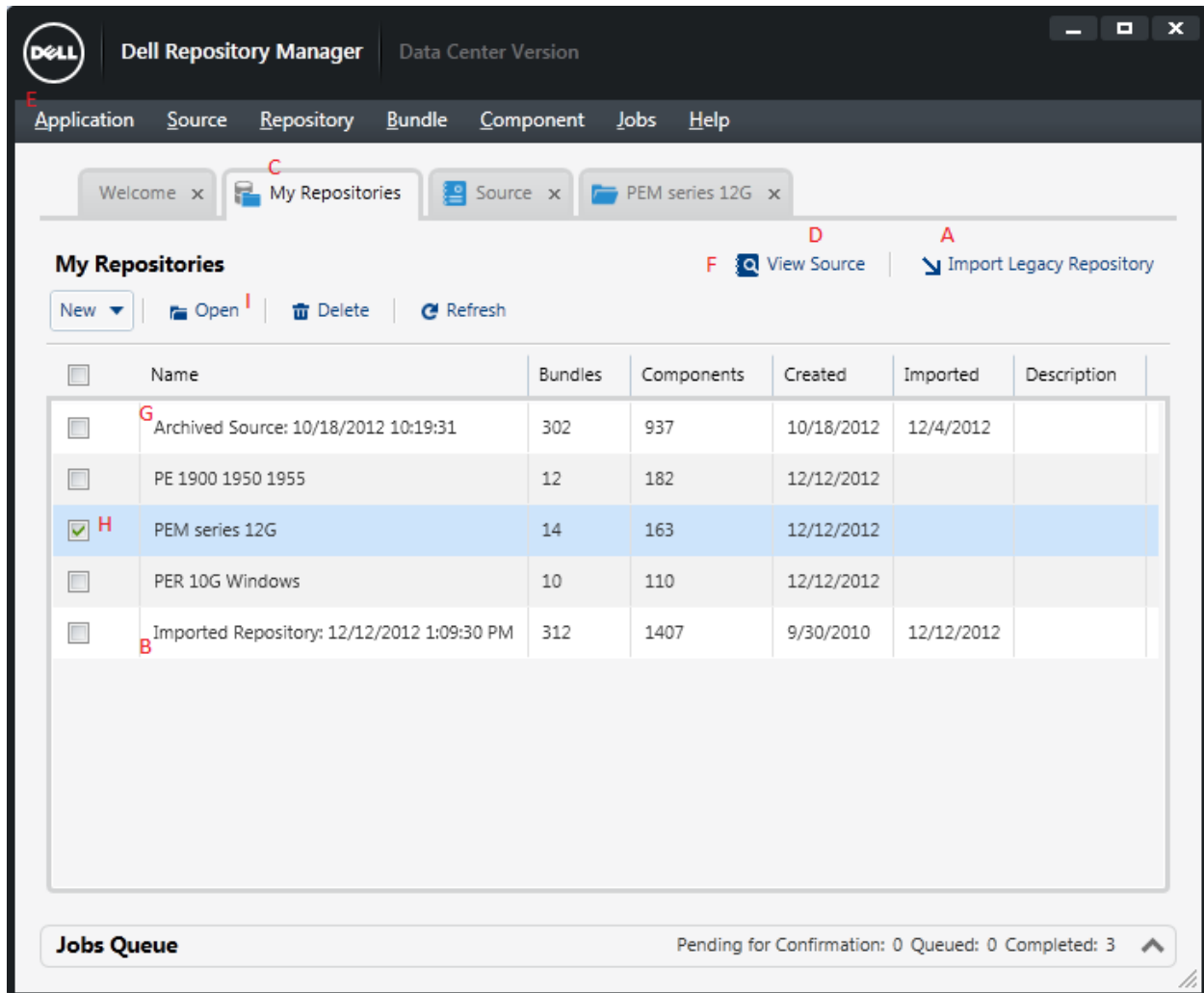


Figure 1 DRM 'My Repositories' Tab

Importing a Catalog

The data shown in DRM is no longer loaded from an XML file, rather it is imported (Fig 1 A) and the repository is stored in the database. A catalog only needs to be imported once, after that the repository is part of the database. (Fig 1 B) Repositories can be changed, read, updated, and deleted. When the repository is imported, there is also an option to download all of the update files at that time. But on the other hand, those files will be downloaded when those specific files are exported.

My Repositories vs. Source

Right after launching DRM, you see a tab for 'My Repositories'. (Fig 1 C) After the initial setup, it will be empty. Top right there is a button to 'View Source'. (Fig 1 D) 'My Repositories' lists the name of the

repositories that you have created. These repositories are read/write, and you can create as many as needed. There is no need to keep track of several catalog.xml files for each repository. The data is now in the database.

Source is a read-only repository from which all other repositories can be created from, hence the name source. By default, this is set to the Dell online ftp catalog. Once you click the 'View Source' button (Fig 1 D); DRM will open this source repository. If it is not in the database, it will be imported from the FTP site into the database. The source can be changed through the application settings (Fig 1 E), if another catalog file will be used as the source repository.

Once there is a newer source available, a notification icon (Fig 1 F) next to the 'View Source' button will appear on the source tab, and the latest source can be imported. So as a new source is imported, the previous source will be archived and then will be added to 'My Repositories'. (Fig 1 G)

This concept of a source repository is woven throughout DRM 1.5. When selecting a repository to get components or bundles from, the source will be one of the options.

Opening a Repository

After you have repositories in the 'My Repositories' tab, you can select and open them. (Fig 1 H I) The repositories are already in the database, so opening them does not load or download a DUP. Rather; it loads the data from the database into a new tab for that repository. Once it is loaded you can manage that repository.

Referencing a Repository

In DRM 1.4, components and bundles could be moved to and from open repositories. When prompted in various places for repositories, you would be given a list of open ones. In DRM, instead of you will be given a list of repositories in the database, you can open the repositories. (Fig 2 J)

Create Repository

REPOSITORY MANAGER | **Create Repository**

Name and Description ✓ Choose the repository that you would like to use as a base for your new repository.

Base Repository

Form Factor

Select OS

Select Models

Select Bundles

Additional Components

Summary

☐ Source Repository

☒ My Repositories

Archived Source: 10/18/2012 10:19:31
PE 1900 1950 1955
PEM series 12G
PER 10G Windows
Imported Repository: 12/12/2012 1:09:30 PM

Cancel | < Back | Next >

Figure 2 Creating a Repository

Saving a Repository

In DRM 1.4, if the repository was changed while loaded into memory, you would be prompted to save it when exiting DRM. In DRM, these repositories are in the database instead of in memory. Any changes to the repository are changed instantly in the database; therefore there is no need to save a repository. There will be the need to extract the catalog file, as well as the update files defined by the catalog, from a repository for other system management consoles' usage. This can be done by clicking the 'Export' button on repository task bar. (Fig 3 K) There you will have options to save a generated catalog.xml file or the full repository including the catalog.xml and the update files.

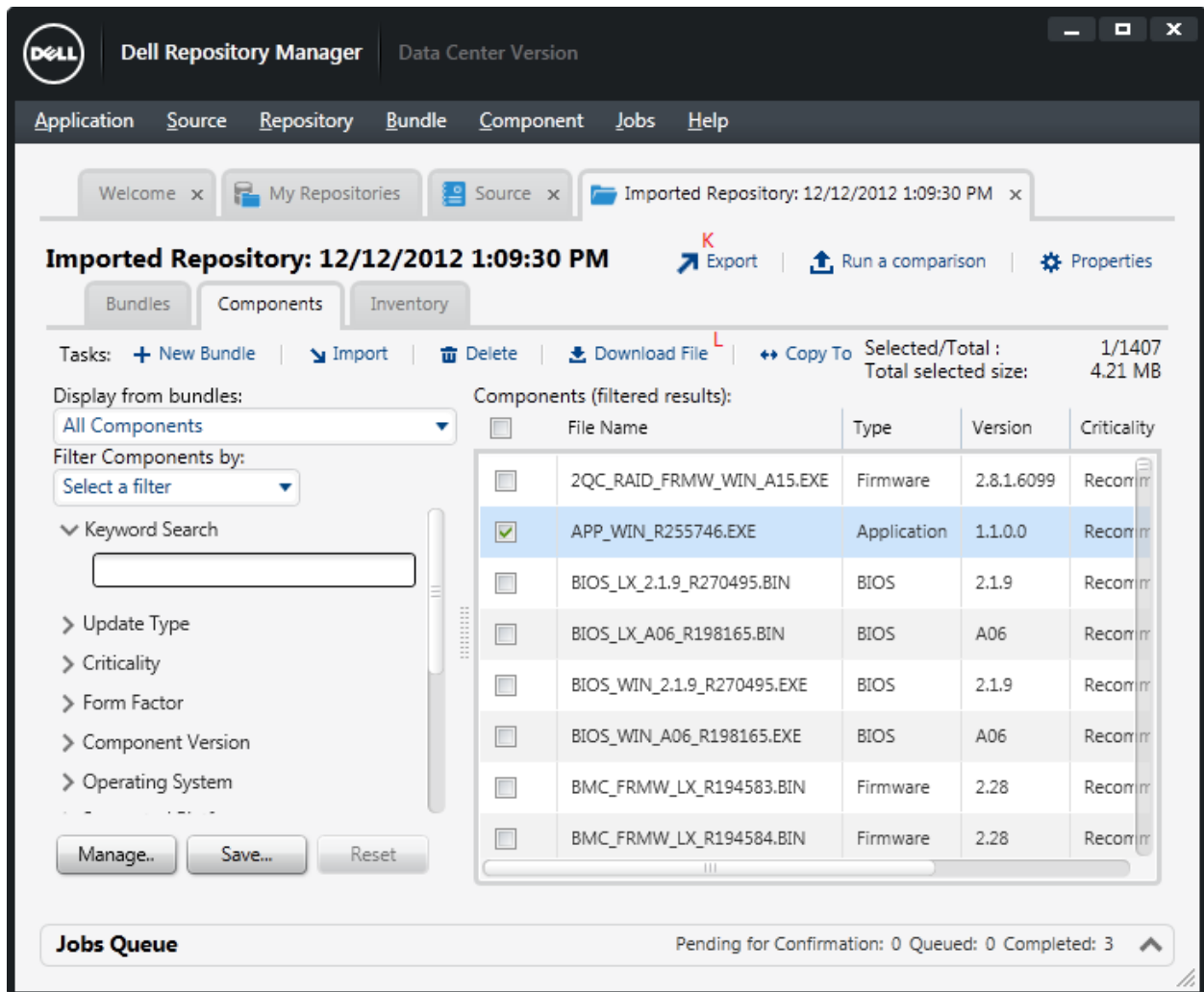


Figure 3 Repository Tab

Downloading Files

Along with a database, DRM uses a file store which keeps all downloaded files on your local storage. When you export bundles or components (Fig 3 L) to any of the various export types, DRM will keep a copy of the file that was downloaded. Next time if the same file is needed, the file will be copied from the file store rather than download the file again.

Maintaining the Database

The database used in DRM is Microsoft SQL Compact Edition (SQLCE). Unlike other Microsoft SQL servers, this is a single application database. It has a limit of 2GB file size. It is only active when the application is loaded, and only the application can access it. DRM is the only user and manager of the database.

In case DRM cannot access the database, there is a simple fix. The database file is located at <user's folder>\AppData\Local\RepositoryManager\Database\RepositoryManager.sdf. That file can be renamed while no DRM instance running and DRM will generate a new database file when launched next time.