VIPM 2012 VI-Based API Documentation

All Contents Copyright © 2012 JKI

Table of Contents

Table of Contents

All Contents Copyright © 2012 JKI

Help and Resources

If you need assistance with the VIPM API, please visit our support page:

- **Jki.net/support**

General Operation

The VIPM VI-based API is a collection of VIs that allow you to command and control VIPM (VI Package Manager). The VIs have been compiled in LabVIEW 2009 and can be used in code written in LabVIEW 2009 or newer. The VIPM API VIs only work with VIPM 2012 (or newer) Pro Edition. They do not work with VIPM Free Edition.

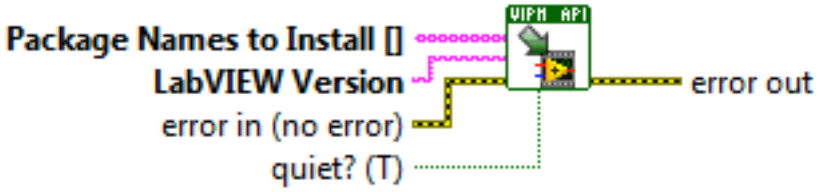
In order for the API to work, you must have the main VIPM application running. It can be minimized, but it must be running.

All Contents Copyright © 2012 JKI

Functions

Main (Root Palette)

Install VI Packages by Name.vi



This VI instructs VIPM to perform an install of packages by name. All the packages to install must already be in the library. If any of the packages are not in the library then an error will be returned and the process will be aborted. You can perform a check to see if the packages are in the library by using the API VI called: "List VIPM Package Library Contents.vi" located under the Library sub palette.

If you need to add the package to the library first, you can use the API VI called: "Add VI Packages to VIPM Package Library.vi" located in the Library sub palette.

If you think the package should be in the Library, but it cannot be found, it's possible that the Library is out of sync with your repository subscriptions. You can perform an update by using the API VI called: "Network Update and Sync VIPM Package Library.vi" located in the Library sub palette.

> ***Note***: if you always install multiple packages as a group, you should consider using VI Package Configuration files also known as VIPC files. This allows you to group your packages into a single file. Then you can install them as a group using the API VI called: "Apply VIPC File.vi" located in the VIPC subpalette.

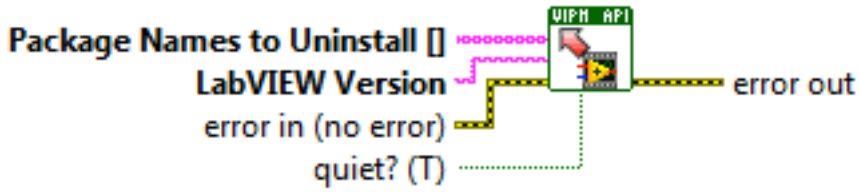
Package Names to Install: This is an array of package names. The package name must be in the following format: package_name-x.x.x.x. For example: acme_lib_gpib_toolkit-1.0.0.1. The best way to figure out the package name is to use the API VI called: "List VIPM Package Library Contents.vi" located under the Library subpalette. You can then perform a search lookup based on display name or other criteria.

LabVIEW Version: This is the version of LabVIEW you want to use for this action. The LabVIEW version you want to use must already be configured and tested with VIPM, otherwise you will get errors. The LabVIEW version must be in the following format: Major. Minor. For example: LabVIEW 8.5 = 8.5, LabVIEW 2009 = 9.0, LabVIEW 2011 = 11.0. The best way to figure out what LabVIEW versions are registered with VIPM on your system is to use the API VI called: "List VIPM LabVIEW Target Versions.vi". If you are installing on 64-bit LabVIEW then you must append the (64-Bit) suffix. For example: 11.0 (64-Bit).

All Contents Copyright © 2012 JKI

Quiet? : This is a Boolean input. If set to TRUE (default), there will not be any dialogs displayed on the VIPM user interface. VIPM will perform the action silently. If you are having problems running the action, you can try setting this to FALSE so you can see any possible dialogs displayed.

Uninstall VI Packages by Name.vi



This VI instructs VIPM to perform uninstallation of packages by name. All the packages to install must already be in the library and installed. If any of the packages are not in the library or not installed then an error will be returned and the process will be aborted. You can perform a check to see if the packages are in the library or installed by using the API VI called: "List VIPM Package Library Contents.vi" located under the Library subpalette.

Package Names to Uninstall: This is an array of package names. The package name must be in the following format: package_name-x.x.x.x. For example: acme_lib_gpib_toolkit-1.0.0.1. The best way to figure out the package name is to use the API VI called: "List VIPM Package Library Contents.vi" located under the Library subpalette. You can then perform a search lookup based on display name or other criteria.

LabVIEW Version: This is the version of LabVIEW you want to use for this action. The LabVIEW version you want to use must already be configured and tested with VIPM, otherwise you will get errors. The LabVIEW version must be in the following format: Major. Minor. For example: LabVIEW 8.5 = 8.5, LabVIEW 2009 = 9.0, LabVIEW 2011 = 11.0. The best way to figure out what LabVIEW versions are registered with VIPM on your system is to use the API VI called: "List VIPM LabVIEW Target Versions.vi". If you are installing on 64-bit LabVIEW then you must append the (64-Bit) suffix. For example: 11.0 (64-Bit).

Quiet? : This is a Boolean input. If set to TRUE (default), there will not be any dialogs displayed on the VIPM user interface. VIPM will perform the action silently. If you are having problems running the action, you can try setting this to FALSE so you can see any possible dialogs displayed.

All Contents Copyright © 2012 JKI

Library Subpalette

Add VI Packages to VIPM Package Library.vi

This VI instructs VIPM to add packages (by path) to the library.

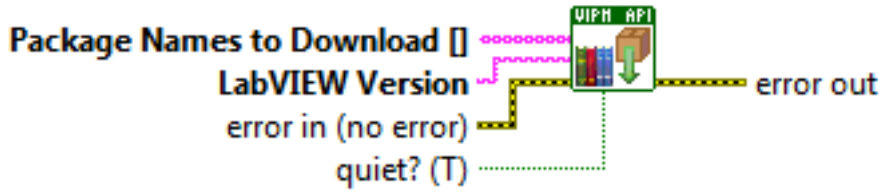
Package to add to VIPM Library (vipc, vip, ogp): This is an array of package source paths. This can be a mixture of *.vip, *.ogp or *.vipc files. When operating on VIPC files, VIPM will pull out the packages from the VIPC file and add them to the library.

> ***Note***: if you are using the API VI called: "Apply VIPC File.vi", you do not need to add the contents of the VIPC file to the library since VIPM does this automatically as part of the "Apply" process.

Quiet? : This is a Boolean input. If set to TRUE (default), there will not be any dialogs displayed on the VIPM user interface. VIPM will perform the action silently. If you are having problems running the action, you can try setting this to FALSE so you can see any possible dialogs displayed.

Package Names Added: This will return a list of the package names that were just added to the library. You can use this output to perform other actions on the packages downstream.

Download VI Packages.vi



This VI instructs VIPM to download packages by name. This will pull packages from the remote repositories and download the *.vip files to the package library. This is useful (for example) if you want to install packages while you are offline at a later time.

Package Names to Download: This is an array of package names. The package name must be in the following format: package_name-x.x.x.x. For example: acme_lib_gpib_toolkit-1.0.0.1. The best way to figure out the package name is to use the API VI called: "List VIPM Package Library Contents.vi" located under the Library subpalette. You can then perform a search lookup based on display name or other criteria.

All Contents Copyright © 2012 JKI

LabVIEW Version: This is the version of LabVIEW you want to use for this action. The LabVIEW version you want to use must already be configured and tested with VIPM, otherwise you will get errors. The LabVIEW version must be in the following format: Major. Minor. For example: LabVIEW 8.5 = 8.5, LabVIEW 2009 = 9.0, LabVIEW 2011 = 11.0. The best way to figure out what LabVIEW versions are registered with VIPM on your system is to use the API VI called: "List VIPM LabVIEW Target Versions.vi". If you are installing on 64-bit LabVIEW then you must append the (64-Bit) suffix. For example: 11.0 (64-Bit).

Quiet? : This is a Boolean input. If set to TRUE (default), there will not be any dialogs displayed on the VIPM user interface. VIPM will perform the action silently. If you are having problems running the action, you can try setting this to FALSE so you can see any possible dialogs displayed.

Network Update and Sync VIPM Package Library.vi

This VI instructs VIPM to check the repository subscriptions for updates to installed packages.

LabVIEW Version: This is the version of LabVIEW you want to use for this action. The LabVIEW version you want to use must already be configured and tested with VIPM, otherwise you will get errors. The LabVIEW version must be in the following format: Major. Minor. For example: LabVIEW 8.5 = 8.5, LabVIEW 2009 = 9.0, LabVIEW 2011 = 11.0. The best way to figure out what LabVIEW versions are registered with VIPM on your system is to use the API VI called: "List VIPM LabVIEW Target Versions.vi". If you are installing on 64-bit LabVIEW then you must append the (64-Bit) suffix. For example: 11.0 (64-Bit).

Quiet? : This is a Boolean input. If set to TRUE (default), there will not be any dialogs displayed on the VIPM user interface. VIPM will perform the action silently. If you are having problems running the action, you can try setting this to FALSE so you can see any possible dialogs displayed.

New Package Updates: This output will contain the names of the packages that have updates. You can use this information to perform an install of the updates downstream.

List VIPM Package Library Contents.vi

All Contents Copyright © 2012 JKI

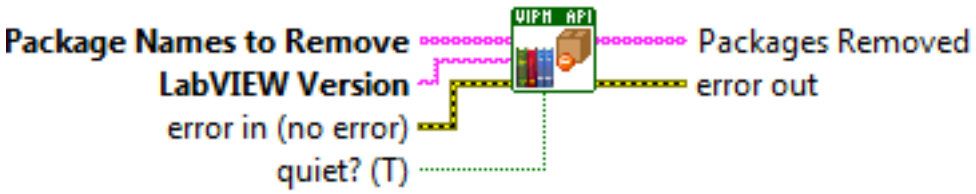
This VI lists the entire contents of the VI Package Library that is managed by VIPM. The contents are filtered based on the LabVIEW version specified.

LabVIEW Version: This is the version of LabVIEW you want to use for this action. The LabVIEW version you want to use must already be configured and tested with VIPM, otherwise you will get errors. The LabVIEW version must be in the following format: Major. Minor. For example: LabVIEW 8.5 = 8.5, LabVIEW 2009 = 9.0, LabVIEW 2011 = 11.0. The best way to figure out what LabVIEW versions are registered with VIPM on your system is to use the API VI called: "List VIPM LabVIEW Target Versions.vi". If you are installing on 64-bit LabVIEW then you must append the (64-Bit) suffix. For example: 11.0 (64-Bit).

Library Contents: This is the library contents output. It's an array of clustered data. The following info is in the cluster:

> ***Package Name:*** This is the full package name. This is the only name that should be used for all other functions in this API.
>
> ***Version:*** This is the package version.
>
> ***Display Name:*** This is the package display name that is shown in the package list. This is also known as the product name.
>
> ***Repository name:*** This is the name of the repository in which the package originated. If the package is unpublished (does not belong to a repository), then "Unpublished" is returned in this field.
>
> ***Company Name:*** This is the company name for the package.
>
> ***License Type:*** This is the license type name for the package.
>
> ***Installed:*** This will be TRUE if the package is installed.
>
> ***Deprecated:*** This will be TRUE if the package is deprecated.
>
> ***Dependency or Conflict Problem:*** This will be TRUE if the package has a conflict or dependency problem.
>
> ***Downloaded:*** This will be true if the package is downloaded to the local package library.

Remove VI Package from VIPM Package Library.vi



This VI will remove packages from the VI Package library that VIPM manages. Only packages that are unpublished can be removed. Published packages will not be removed. Also, if the package is installed on any LabVIEW version, it will not be removed.

All Contents Copyright © 2012 JKI

The packages that are removed successfully will be listed in the "Packages Removed" output.

Package Names to Remove: This is an array of package names. The package name must be in the following format: package_name-x.x.x.x. For example: acme_lib_gpib_toolkit-1.0.0.1. The best way to figure out the package name is to use the API VI called: "List VIPM Package Library Contents.vi" located under the Library subpalette. You can then perform a search lookup based on display name or other criteria.

LabVIEW Version: This is the version of LabVIEW you want to use for this action. The LabVIEW version you want to use must already be configured and tested with VIPM, otherwise you will get errors. The LabVIEW version must be in the following format: Major. Minor. For example: LabVIEW 8.5 = 8.5, LabVIEW 2009 = 9.0, LabVIEW 2011 = 11.0. The best way to figure out what LabVIEW versions are registered with VIPM on your system is to use the API VI called: "List VIPM LabVIEW Target Versions.vi". If you are installing on 64-bit LabVIEW then you must append the (64-Bit) suffix. For example: 11.0 (64-Bit).

Quiet? : This is a Boolean input. If set to TRUE (default), there will not be any dialogs displayed on the VIPM user interface. VIPM will perform the action silently. If you are having problems running the action, you can try setting this to FALSE so you can see any possible dialogs displayed.

Packages Removed: This output will list all the package names that were actually removed successfully.

All Contents Copyright © 2012 JKI

Package Building Subpalette

Build VI package.vi

This VI allows you to perform a package build. VIPM will use the existing vipb build spec located at the path specified.

Path to VIPB File: This is a path to the vipb build spec. For example: "c:\my project\.vipb".

Build Output: This is a path to the final built package file.

Read VI Package Build Spec.vi

This VI allows you to read some information from the package build spec. VIPM will use the existing vipb build spec located at the path specified.

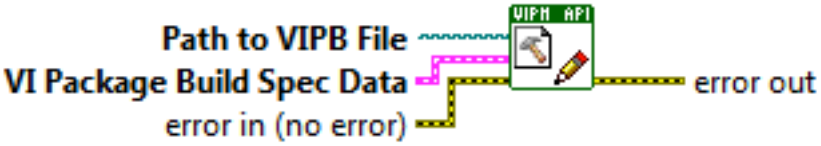
Path to VIPB File: This is a path to the vipb build spec. For example: "c:\my project\.vipb".

VI Package Build Spec Data: This is a cluster containing some package build spec information:

- Package Version
- Product Name
- Company Name
- Author Name (Person or Company)
- Product Homepage (URL)
- Legal Copyright
- License Agreement Name
- Product Description Summary
- Product Description
- Release Notes - Change Log

All Contents Copyright © 2012 JKI

Write VI Package Build Spec.vi



This VI allows you to read some information from the package build spec. VIPM will use the existing vipb build spec located at the path specified. If you specify a path to a vipb file and the file does not exist, VIPM will create a default file using the source file contents located at the base folder and the provided VI Package Build Spec Data input.

Path to VIPB File: This is a path to the vipb build spec. For example: "c:\my project\.vipb".

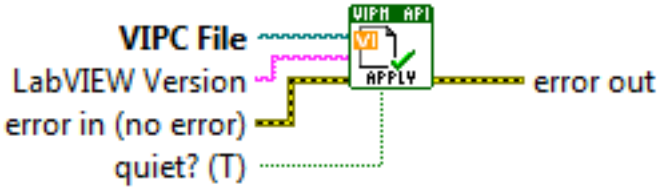
VI Package Build Spec Data: This is a cluster containing some package build spec information that will be written to the file:

- Package Version
- Product Name
- Company Name
- Author Name (Person or Company)
- Product Homepage (URL)
- Legal Copyright
- License Agreement Name
- Product Description Summary
- Product Description
- Release Notes - Change Log

All Contents Copyright © 2012 JKI

VI Package Configuration (VIPC) Subpalette

Apply VIPC File.vi



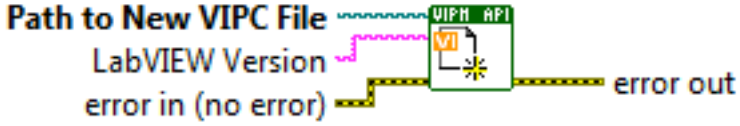
This VI will apply a package configuration (VIPC) file specified in the VIPC File input path. If no LabVIEW version is specified then the LabVIEW version defined in the VIPC file is used.

VIPC File: This is a path to an existing VI Package Configuration (VIPC) file.

LabVIEW Version: This is the version of LabVIEW you want to use for this action. The LabVIEW version you want to use must already be configured and tested with VIPM, otherwise you will get errors. The LabVIEW version must be in the following format: Major. Minor. For example: LabVIEW 8.5 = 8.5, LabVIEW 2009 = 9.0, LabVIEW 2011 = 11.0. The best way to figure out what LabVIEW versions are registered with VIPM on your system is to use the API VI called: "List VIPM LabVIEW Target Versions.vi". If you are installing on 64-bit LabVIEW then you must append the (64-Bit) suffix. For example: 11.0 (64-Bit).

Quiet? : This is a Boolean input. If set to TRUE (default), there will not be any dialogs displayed on the VIPM user interface. VIPM will perform the action silently. If you are having problems running the action, you can try setting this to FALSE so you can see any possible dialogs displayed.

Create New VIPC File.vi



This VI will create a blank VIPC file configured for the LabVIEW version specified. This is useful when using it with the VIPM API VI called: "Scan Project.vi" to allow creation of new project configurations.

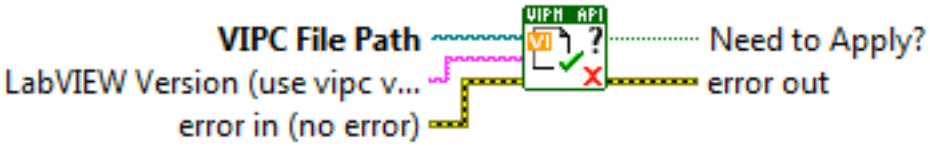
Path to New VIPC File: This is a path to where you want the VIP file saved.

LabVIEW Version: This is the version of LabVIEW you want to use for this action. The LabVIEW version you want to use must already be configured and tested with

All Contents Copyright © 2012 JKI

VIPM, otherwise you will get errors. The LabVIEW version must be in the following format: Major. Minor. For example: LabVIEW 8.5 = 8.5, LabVIEW 2009 = 9.0, LabVIEW 2011 = 11.0. The best way to figure out what LabVIEW versions are registered with VIPM on your system is to use the API VI called: "List VIPM LabVIEW Target Versions.vi". If you are installing on 64-bit LabVIEW then you must append the (64-Bit) suffix. For example: 11.0 (64-Bit).

Test VIPC Apply Needed.vi

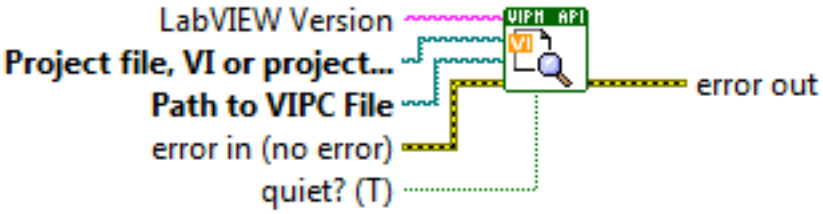


This VI will perform a check to see if the configuration file specified in the 'VIPC File Path' input needs to be applied. If no LabVIEW version is specified then the version configured inside the VIPC is used.

VIPC File Path: This is a path to the VIPC file you want to check.

LabVIEW Version: This is the version of LabVIEW you want to use for this action. The LabVIEW version you want to use must already be configured and tested with VIPM, otherwise you will get errors. The LabVIEW version must be in the following format: Major. Minor. For example: LabVIEW 8.5 = 8.5, LabVIEW 2009 = 9.0, LabVIEW 2011 = 11.0. The best way to figure out what LabVIEW versions are registered with VIPM on your system is to use the API VI called: "List VIPM LabVIEW Target Versions.vi". If you are installing on 64-bit LabVIEW then you must append the (64-Bit) suffix. For example: 11.0 (64-Bit).

Scan Project.vi



This VI will perform a scan of your project file (lvproj), source folder or toplevel VI. The scanning results are saved to the VIPC file path specified. The LabVIEW version is optional since VIPM will use the version found in the files scanned.

Project file, VI or project directory path: This is a path to the scan source.

VIPC File Path: This is a path to the VIPC file you want to write the results to.

All Contents Copyright © 2012 JKI

LabVIEW Version: This is the version of LabVIEW you want to use for this action. The LabVIEW version you want to use must already be configured and tested with VIPM, otherwise you will get errors. The LabVIEW version must be in the following format: Major. Minor. For example: LabVIEW 8.5 = 8.5, LabVIEW 2009 = 9.0, LabVIEW 2011 = 11.0. The best way to figure out what LabVIEW versions are registered with VIPM on your system is to use the API VI called: "List VIPM LabVIEW Target Versions.vi". If you are installing on 64-bit LabVIEW then you must append the (64-Bit) suffix. For example: 11.0 (64-Bit).

All Contents Copyright © 2012 JKI

Utility Subpalette

Exit VIPM.vi

This VI will exit VIPM

Minimize VIPM.vi

This VI will minimize VIPM

List VIPM LabVIEW Target Version.vi

This VI will list all the LabVIEW versions that are configured with VIPM.

List Invalid Targets? (F): When set to TRUE, this indicates if the returned target list includes invalid targets. Invalid targets are the ones that have not been verified with VIPM or are disabled. You cannot perform any VIPM function on invalid targets.

LabVIEW Target Info: This is an array of cluster data containing the following info:

> ***Version:*** The LabVIEW version configured
>
> ***Connection Tested***: Whether VIPM has gone through the connection test process.
>
> ***Disabled:*** Whether the LabVIEW version has been disabled in the VIPM configuration.

All Contents Copyright © 2012 JKI

Switch Target.vi

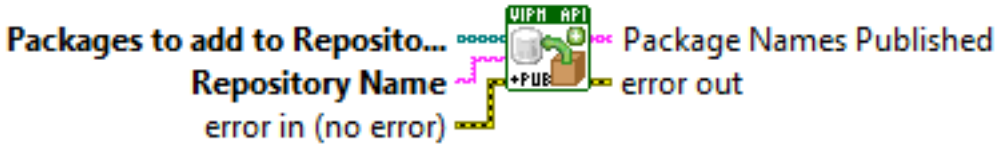
This VI will switch VIPM to use the LabVIEW version specified on the LabVIEW Version input.

LabVIEW Version: This is the version of LabVIEW you want to use for this action. The LabVIEW version you want to use must already be configured and tested with VIPM, otherwise you will get errors. The LabVIEW version must be in the following format: Major. Minor. For example: LabVIEW 8.5 = 8.5, LabVIEW 2009 = 9.0, LabVIEW 2011 = 11.0. The best way to figure out what LabVIEW versions are registered with VIPM on your system is to use the API VI called: "List VIPM LabVIEW Target Versions.vi". If you are installing on 64-bit LabVIEW then you must append the (64-Bit) suffix. For example: 11.0 (64-Bit).

All Contents Copyright © 2012 JKI

Repository Subpalette

Publish VI Packages to Repository.vi



This VI will publish packages to the repository specified in the Repository Name input.

Packages to Publish to Repository: This is a list of package paths to publish to the repository.

Repository Name: This is the name of the repository you wish to publish to. You can find out what the names of the repositories are by running the VIPM API VI called: "List VIPM Managed Repositories.vi"

Package Names Published: This will return the package names of the packages published.

Unpublish VI Packages from Repository.vi

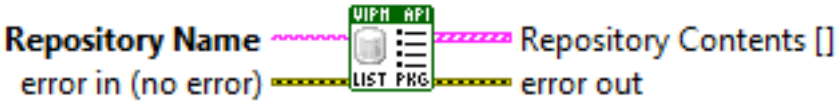
This VI will unpublish packages from the repository specified in the Repository Name input.

Packages to publish to Un-Publish: This is a list of package paths to publish to the repository.

Repository Name: This is the name of the repository you wish to Un-Publish from. You can find out what the names of the repositories are by running the VIPM API VI called: "List VIPM Managed Repositories.vi"

Package Names Un-Published: This will return the package names of the packages un-published.

List Repository Contents.vi



This VI will list the contents of the repository specified in the Repository Name input. This is the name of the repository that you are managing.

All Contents Copyright © 2012 JKI

Repository Name: This is the name of the repository you wish to Un-Publish from. You can find out what the names of the repositories are by running the VIPM API VI called: "List VIPM Managed Repositories.vi"

Repository Contents: This is an array of clustered data that contains the following info:

> ***Package Name:*** This is the package name of the package in the repository.
>
> ***Version:*** This is the version of the package in the repository.
>
> ***Display Name:*** This is the display name (product name) of the package in the repository.
>
> ***Release Date:*** This is the date that the package was published to the repository***.***
>
> ***Deprecated:*** This will be TRUE, if the package is deprecated.

List VIPM Managed Repositories.vi

This VI will list all the repositories that are managed with VIPM.

Managed Repositories Info: This is an array of clustered data with the following info:

> ***Repository Name***: This is the name of the repository.
>
> ***Repository Folder Location***: This is the physical location of the repository on disk.

All Contents Copyright © 2012 JKI