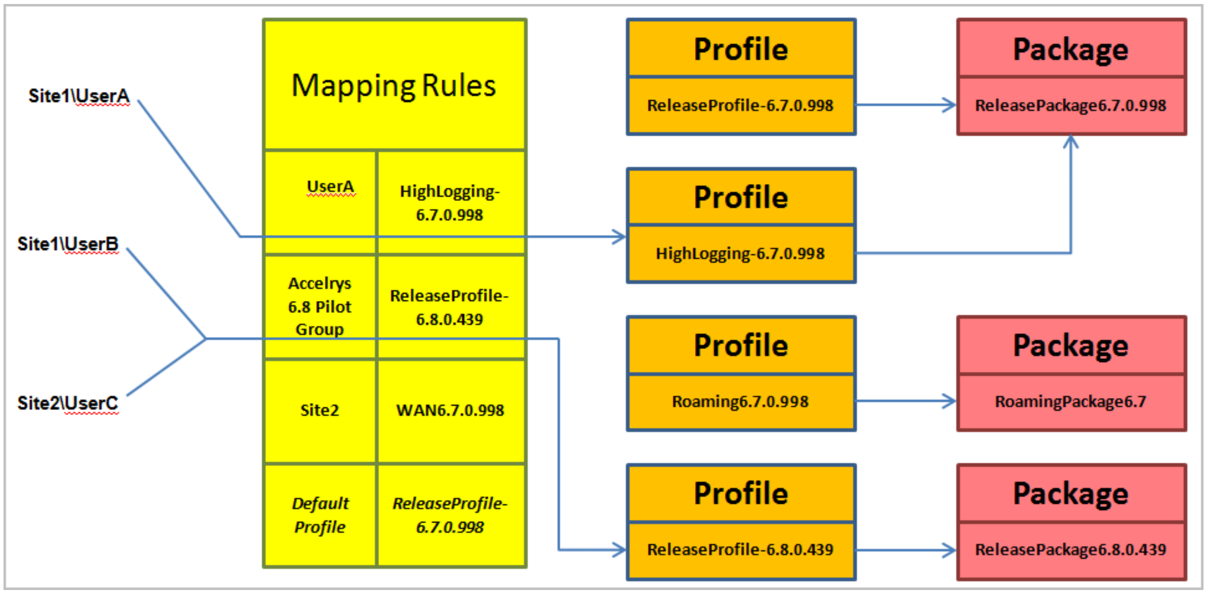
Chapter 1:
Deployment Manager Overview

BIOVIA Workbook Deployment Manager (DM) enables you to manage user configurations and deliver appropriate profiles and versions of the Workbook client application to end users.

You can use DM to:

- Assign users to different profiles at the group or individual level by using mapping rules. You can assign users to default profiles, as well as create custom profiles.
- Push the appropriate profile to each end user.
- Push new versions of DM to each end user.



DM does not change the client system's registry or file system.

Components

You use the Deployment Manager (DM) components listed in the following table to manage and deploy client packages.

Component	Description
DM Service	Web service that provides the correct configuration (Profile) and the correct DM version (Package) to the DM Client users. DM service URL is `http://your-server/accelrys/deployment-manager`.

Component	Description
DM Client	Client Manager shell that resides on the user's client system. When a scientist logs into the shell, the shell contacts the DM service, downloads the correct configuration (Profile), and downloads and unpacks a new DM version (Package) if needed. After installation is complete, the shell launches the DM client. By default, `NotebookClientManager.exe` is located in `C:\Program Files (x86)\BIOVIA\BIOVIA Workbook Client Manager\bin`. **Note:** End users can uninstall BIOVIA Workbook Client Manager by using **Programs and Features** in the Windows **Control Panel**. The DM Client provides options for the following purposes: ■ Customize the login screen. ■ Download a package. You can use the command line option to download the package when users are not working, such as at night or during the weekend. ■ Delete the locally stored copy of a package, which cleans up the client's hard drive and deletes the package from the deployment folder. You can control which users get a new release package by using the Vault Administration Console to assign users to the group that receives the specified profile.
DM Store Manager	The DM Store Manager, `VaultAdmStoreManager.exe`, is a server-side command-line tool that you use to: ■ Create profiles and packages. ■ Change mapping rules that associate profiles with users or groups. You can control the specific package users receive by creating groups and assigning the packages at the group level. The `VaultAdmStoreManager.exe` file resides in a subdirectory of the Vault Deployment Utility's default installation directory.

User Profile Configuration Tasks

Use BIOVIA Deployment Manager to create and manage user configurations, and deploy customized DM clients to users from the Vault server.

Update the settings in the `Symyx.Symyx.Notebook.Application.exe.config.Application.exe.config` file as described in this documentation, or with the advice provided by Dassault Systèmes Customer Support. Back up the file before you change it.

Deployment Profile Modes

Your deployment profiles control whether DM users are permitted to work in *offline* mode or must work in online (*roaming*) mode. By default, working offline is not permitted.

- If a profile uses the default *roaming* mode, users with that profile must connect to the BIOVIA Vault Server server to work on experiments. The users can connect from any client computer, and their checked-out experiments, new experiments, and favorites are stored in an area of BIOVIA Vault

Server that is private to the user.

- If a profile permits *offline* mode, users with that profile can log on using the normal roaming mode, check out experiments that they want to work on while offline, log off, and then select **Offline** from the log-on screen to work on the checked out local copies of the experiments. If security is a concern, BIOVIA recommends not enabling the offline mode.

To change the deployment mode, edit the `Symyx.Notebook.Application.exe.config` file.

Offline Mode Settings

The following settings in the `Symyx.Notebook.Application.exe.config` file are required to enable offline mode:

```
<setting name="AllowOffline" serializeAs="String">
   <value>True</value>
</setting>
<setting name="UserRepositoryStorageOption" serializeAs="String">
   <value>Offline</value>
</setting>
```

Roaming Mode Settings

By default the `Symyx.Notebook.Application.exe.config` disables offline mode as follows:

```
<setting name="AllowOffline" serializeAs="String">
   <value>False</value>
</setting>
<setting name="UserRepositoryStorageOption" serializeAs="String">
   <value>Roaming</value>
</setting>
```

After switching from offline mode back to roaming mode, users must check-in all new and checked-out objects. After switching from offline to roaming mode, the user does not have access to any Favorites saved while working offline.

Local File Caching Configuration

The configuration in the user BIOVIA Deployment Manager profile allows an administrator to configure how the DM client deals with caching files used in file sections on the local computer. Use the enabled setting to use or disable local file caching.

- Set `enabled` to "true" for non-Citrix deployments.
- Set `enabled` to "false" for Citrix deployments.

Note: In the default config file `enabled="true"` is set, however the default value is "false" if the attribute is not defined.

When using local file caching, the installation parameters are as follows:

Parameter	Description	Undefined Default
maxCachableFileSizeBytes	Specifies the maximum size of files allowed in the local file cache. The default value of zero removes the size limit. **Default configuration:** maxCachableFileSizeBytes="0"	50 MB
maxCacheSizeBytes	Specifies the maximum size of the local file cache, in bytes. Mutually exclusive to maxCacheAgeDays. When maxCacheAgeDays and maxCacheSizeBytes are both non-zero, then maxCacheSizeBytes is honored. If maxCacheAgeDays or maxCacheSizeBytes is not greater than zero, cache size is unmanaged. **Default configuration:** maxCacheSizeBytes="5368709120"	1 GB if the attribute is not declared
maxCacheAgeDays	Specifies the maximum allowable number of days that a file in the local file cache can remain unaccessed in the local file cache. Mutually exclusive to maxCacheSizeBytes. **Default configuration:** maxCacheAgeDays="0"	0 if attribute is not declared
greedyCacheTrimming	When greedyCacheTrimming is set to to true, the cache is trimmed after every potentially cache growing operation. If "false", the cache is trimmed once per DM session upon shutdown. **Default configuration:** greedyCacheTrimming="false"	false if attribute is not declared

For example, in the default configuration file:

```
<Symyx.Notebook.Sections.ExternalFile>
   <localCachingConfig enabled="true" maxCachableFileSizeBytes="0"
maxCacheSizeBytes="5368709120" maxCacheAgeDays="0"
greedyCacheTrimming="false"/>
 </Symyx.Notebook.Sections.ExternalFile>
```

IMPORTANT! If you copy sample code from the PDF version of this document and paste it into a text editor, the text might change from the original. To ensure that the text is valid and usable, you might need to restore indents, remove extra spaces, and retype characters for single and double quotes.

Chapter 2:
Release Profiles and Packages

DM administration involves creating release profiles and release packages.

- **Profiles**

  Profiles contain information such as the configuration setting for the object cache and enabling or disabling features. A profile can contain a pointer to the release package.

- **Packages**

  Packages are the mechanism for deploying a version of the application and distributing the installation files to the end user's client computers.

- **`Symyx.Notebook.Application.exe.config`**

  Usually, users start Workbook by using the Windows Start > Programs option. This option executes, `NotebookClientManager.exe`, which downloads the user's profile so that Workbook operations are controlled by the settings in the profile.

  However, users can also start Workbook by executing `Symyx.Notebook.Application.exe` from directly from Windows Explorer. When users start Workbook this way, the settings in the `Symyx.Notebook.Application.exe.config` file are used to control the user's Workbook operations.

Release Profiles

DM administration involves creating release profiles and release packages.

Profiles contain information such as the configuration setting for the object cache and enabling or disabling features. The profile can contain a pointer to the release package.

Packages are the mechanism for deploying a version of the application and distributing the installation files to the client computers.

The release profiles generated by the Vault Deployment Utility are as follows:

- Offline

  Offline is the default profile. The offline profile stores the client's private repository on the client's local drive.

- Roaming

  The roaming profile stores the private repository in Oracle. In a Citrix environment use the roaming profile with the object cache disabled.

Name	User Repository Storage Location	Local Object Caching Enabled	Comments
Offline	Client computer	Yes	Default profile
Roaming	Vault server	Yes	
Citrix	Vault server	No	

Release Profile with Mapping Rules

Example Rule Set

```
<?xml version="1.0" encoding="utf-8"?>
<Rules xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  defaultProfile="V6.9.0.x-Roaming">
  <!-- SNIPPED: Help text -->
  <Rule userSelector='Citrix Users' profile='V66.0.x-Citrix' />
</Rules>
```

The `defaultProfile` specifies the profile to use when no rule applies to the current user. The `userSelector` maps a user or group to a profile.

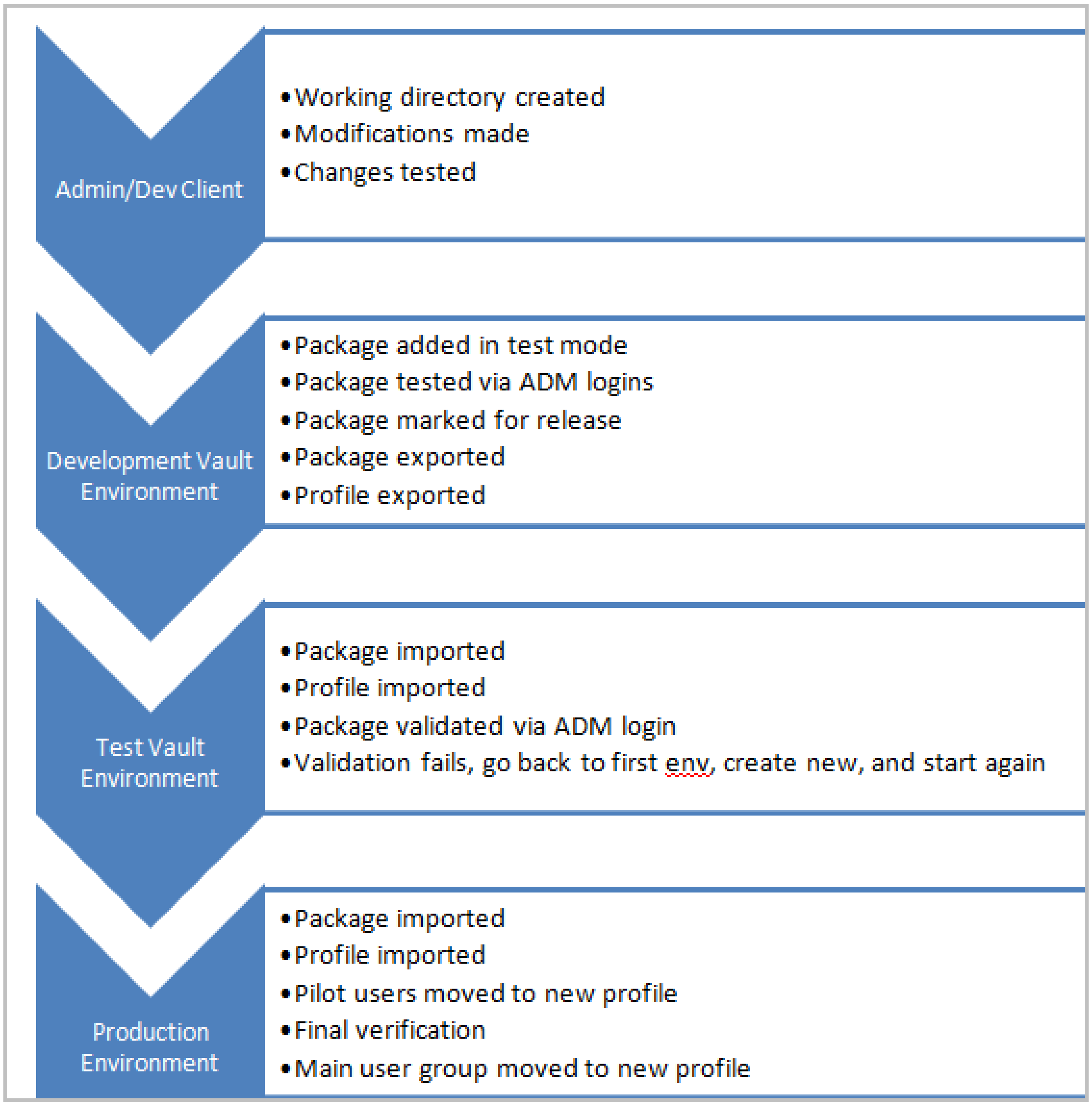
Mapping Rules associate users and groups to Release Profiles.

- You can specify rules for groups or individuals.
- The first rule governs the object.
- The *default* release profile has a lower priority than all other rules.

Example using group membership to control logging levels in a Citrix deployment.

Package Customization Workflow

The following depicts the steps in the process of developing a custom release Package.



Admin/Dev Client

- Working directory created
- Modifications made
- Changes tested

Development Vault Environment

- Package added in test mode
- Package tested via ADM logins
- Package marked for release
- Package exported
- Profile exported

Test Vault Environment

- Package imported
- Profile imported
- Package validated via ADM login
- Validation fails, go back to first env, create new, and start again

Production Environment

- Package imported
- Profile imported
- Pilot users moved to new profile
- Final verification
- Main user group moved to new profile

Chapter 3: Deployment Manager Store Commands

- **Rule Set**
  - `download-rules`

    Downloads the user to profile mapping rules into a file.
  - `upload-rules`

    Uploads the user to profile mapping rules from a file.
- **Profiles**
  - `list-profiles`

    Outputs a line-delimited list of release profiles.
  - `add-profile`

    Creates a new release profile using the same package.
  - `update-profile`

    Updates a release profile from a working directory.
  - `change-profile-package`

    Generates a working directory for a profile with a new package.
- **Package Management**
  - `add-package`

    Creates a new package and profile.
  - `list-package`

    Outputs a line-delimited list of release packages list-profiles
  - `release`

    Marks an unreleased package as released.
  - `update-package`
- **Working Directories**
  - `generate`

    Generates a working directory.
  - `get-folder-status`

    Outputs the state of a working directory.
  - `patch`

    Applies one or more patches to the working directory in a specified order.
- **Package Security**
  - `add-namespace`

    Creates a namespace.
  - `list-namespaces`

    Outputs a line-delimited list of package namespaces.
  - `create-key`

    Creates a key pair.

- `get-public-key`

  Gets the public key from key pair file and saves it to another file.

- **Promotion**
  - `export-profile`

    Creates a file containing a released profile.
  - `import-profile`

    Creates a profile based on the contents of a package export file.
  - `export-package`

    Creates a file containing a released package.
  - `import-package`

    Creates a package based on the contents of a package export file.

DM Store Manager Command Syntax

The VaultADMStoreManager is a command line utility that manages Deployment Manager entities (packages, profiles, and rules).

Syntax

```
VaultADMStoreManager [/store | /local<store-folder> | /vault <server_name>
<user> <password>] <command><arguments>
```

For example to release a package: `VaultADMStoreManager /vault <server> <user> <password> release MyCorpNamespaceMyCorpPackage`

Command Options

If you do not designate a storage option, the command defaults to the `/nostore` option.

Command Option	Description
`/nostore`	■ Does not connect to local or a Vault storage folder. ■ Use with key-pair generation and folder status commands.
`/local`	■ Use or create a Deployment Manager store in <*store-folder*>. ■ Use to test Deployment Manager automation scripts or package construction without a Vault server.
`/vault <server> <user> <password>`	■ Use the Deployment Manager store in the specified Vault server and connect with the specified credentials. If the password string is empty, you are prompted to enter the password. ■ Use `/console` command with the login to create a Deployment Manager console session, and eliminate the need to provide credentials with each command.

Syntax Help

To view help in the VaultADMStoreManager, type the following at the command prompt:

```
VaultADMStoreManager /help
```

or

```
VaultADMStoreManager /help command
```

Assign Deployment Profiles to Users

If your organization has a site preference for the private repository to use roaming storage, modify the Deployment Manager default rules as follows.

To assign a profile to users:

1. On your BIOVIA Vault Server, navigate to the folder that contains the Vault Store Manager, by default `C:\Program Files (x86)\BIOVIA\Vault\Utilities`.
2. Instruct users to check their documents.
3. List the available profiles by executing the following command, where username is the Vault administrator account:

   ```
   VaultADMStoreManager /vault server_name domain\username "" list-profiles
   ```

   > **Note:** Using the double-quotes after the user name causes the system to prompt for the password. The other option is to type the password instead of "".

   For example:

   ```
   VaultADMStoreManager /vault vm-avs69.3ds.com 3ds\vault.admin "" list-
   profiles
   Enter password: ******
   ```

   Output:

   ```
   V21.x-Offline
   V21.x-Roaming
   V21.x-Citrix
   ```

4. Download the mapping rules file from the server, using the following command.

   ```
   VaultADMStoreManager /vault server name domain\username "" download-
   rules C:\Temp\rules.xml
   ```

5. Edit the mapping rules file, located in `C:\Temp\rules.xml`, using a text editor to specify the default profile.

   For example, if your DM version is 2021, you might change

   ```
   defaultProfile="V21.x-Offline">
   ```

   to

   ```
   defaultProfile="V21.x-Roaming">
   ```

6. Upload the edited mapping rules file to your Vault server using the following command:

   ```
   VaultADMStoreManager /vault server name domain\username "" upload-rules
   C:\Temp\rules.xml
   ```

7. Inform your end-users that they can log in to BIOVIA Workbook. When a user logs in, DM Client Manager automatically uses the new profile.

Deploying a Custom Profile

To deploy a new release to users with customized profiles:

1. Publish the zip file with a Release Package, for example, put the *NewRelease*.admpkg package in your Vault environment using the following command:

   ```
   VaultADMStoreManager /vault <server> <domain>\<user> <password> import-
   package NewRelease.admpkg
   ```

2. Generate a working directory that represents a Release Profile assigned to a new Release Package, using the change-profile-package command:

   ```
   VaultADMStoreManager /vault <server> <domain>\<user> <password> change-
   profile-package YourProfileName System ReleasePackage6.9.0.818
   C:\Temp\MoveProfileTo69
   ```

3. In the working folder, represented as `C:\Temp\MoveProfileTo69`, modify the configuration files.
4. Double-click the `Symyx.Notebook.Application.exe` file to start the DM client to test your changes.

   When the configuration works the way you want, update the profile.

5. Update the profile using the following command:

   ```
   VaultADMStoreManager /vault <server> <domain>\<user> <password> update-
   profile C:\Temp\MoveProfileTo69
   ```

   Each user who is assigned to the custom *YourProfileName* automatically receives the specified Release Package to download on the next log in to DM.

Create a Client Profile

To create a customized DM profile you should:

- Create a working directory based on an existing package.
- Modify the configuration files in the existing package.
- Add the new release profile to the working directory.

Chapter 4:
Create a Release Profile

To build your new release profile, you can alter an existing profile:

1. In the Deployment Manager working directory, modify the configuration file, `Symyx.Notebook.Application.exe.config`.
2. Double-click `Symyx.NotebookClientManager.exe` to start the DM client.
3. Do one of the following:
   - Add the new profile using the following command:

     ```
     VaultADMStoreManager /vault <server> <domain>\<user> <password> add-
     profile NewProfileName <NewPackageDir>
     ```

     For example:

     ```
     VaultADMstoreManager /vault vm-avs69 vm-avs69\vault.admin abc123 add-
     profile MyPackage1.V6.9.0.x-Roaming C:\TEMP\Rules_package
     ```

   - Update an existing profile using the following command:

     ```
     VaultADMStoreManager /vault <server> <domain>\<user> <password>
     update-profile C:\TEMP\Rules_package
     ```

   > **Note:** Do not use a plus (+) character or other special characters in package or profile names. If you do so, when a user whose account uses that package or profile attempts to log on, the system displays an message that says it cannot log them on to Vault due to an error on the server.

4. Download the current mapping rules by using the following command:

   ```
   VaultADMStoreManager /vault <server> <domain>\<user> <password>
   download-rules <directory\filename>
   ```

   For example:

   ```
   VaultADMstoreManager /vault vm-avs69 vm-avs69\vault.admin abc123
   download-rules C:\temp\rules.xml
   ```

5. List the available profiles using the following command:

   ```
   VaultADMStoreManager /vault <server> <domain>\<user> <password> list-
   profiles
   ```

   For example:

   ```
   VaultADMstoreManager /vault vm-avs69 vm-avs69\vault.admin abc123 list-
   profiles
   ```

   Example Results:

   ```
   V6.9.0.x-Offline
   V6.9.0.x-Roaming
   V6.9.0.x-Citrix
   V6.9.0.x-Offline
   V6.9.0.x-Roaming
   V6.9.0.x-Citrix
   MyPackage1.V6.9.0.x-Roaming
   ```

6. Update the mapping rules using the following example:

```
<?xml version="1.0" encoding="utf-8"?>
    <Rules xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:xsd="http://www.w3.org/2001/XMLSchema"
       defaultProfile="V6.9.0.x-Roaming">
    </Rules>
```

   Example Results:

```
<?xml version="1.0" encoding="utf-8"?>
     <Rules xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:xsd="http://www.w3.org/2001/XMLSchema"
       defaultProfile="MyPackage1.V6.9.0.x-Roaming">
     </Rules>
```

7. Upload the changed rules for testers using the following command:

```
VaultADMStoreManager /vault <server> <domain>\<user> <password> upload-
rules <directory\filename>
```

   For example:

```
VaultADMstoreManager /vault vm-avs69 vm-avs69\vault.admin abc123 upload-
rules C:\temp\rules.xml
```

Create a Deployment Manager Working Directory

To create a working directory for the Deployment Manager and modify the Symyx.Notebook.Application.exe.config file.

To create a working directory for Deployment Manager:

1. Launch a command prompt in the folder containing VaultADMStoreManager.exe.
2. Run the following command:

```
VaultADMStoreManager /vault <server> <domain>\<user> <password> generate
<profile> <directory>
```

   For example:

```
VaultADMstoreManager /vault vm-avs69 vm-avs69\your_vault.admin abs123
generate V6.9.0.x-Test C:\TEMP\Rules_package
```

   Where the user variables inserted into the example represent:

   - vm-avs69 is the server.
   - vm-avs69 is the domain.
   - your_vault.admin is the user.
   - abc123 is the password.
   - V6.9.0.x-Test is the name of an existing profile.
   - C:\TEMP\Rules_package is the working directory.

Applying a Hotfix to an ADM Working Directory

The VaultADMStoreManager.exe has a /patch command that updates existing working directories with specified *.admpatch files.

To apply a hotfix to a working directory:

1. Launch a command prompt in the folder containing `VaultADMStoreManager.exe`.
2. Run the following command:

   ```
   VaultADMStoreManager /patch <working-directory> <first patch file>
   <second patch file> <third patch file>
   ```

   For example:

   ```
   VaultADMStoreManager /patch C:\<MyWorkingFolder> hotfix.admpatch
   hotfix2.admpatch anotherpatch.admpatch
   ```

   You can order hotfixes to apply in the order you type them into the command line. This applies multiple patches in one command line.

Quick File Settings

The DM Experiment Editor provides as Quick File window that users can add files and folders and access those resources to include in experiments. DM users can drag and drop the files saved in the Quick File window into experiments.

Users can configure the Quick File window to display folders. The folders are stored with a combination of the user ID and the specific computer identifier to enable configuring a different set of folders for different computers. If your users want to access the same files independent of the specific computer, set the `MachineIndependentQuickFilePaths` to *True* to store paths on a per-user basis.

Citrix users frequent connect using identically configured computers and need to access the same files. For Citrix deployments, you might want to set the `MachineIndependentQuickFilePaths` to *True*.

You must create a Citrix profile and set the `MachineIndependentQuickFilePaths` property to *True* to enable using the Quick File window. Set the `MachineIndependentQuickFilePaths` setting to *False* for other profiles such as roaming and offline profiles. To implement the settings changes for Citrix users, make the modifications to the existing Citrix profile using step 3 in Create a Release Profile in the `Symyx.Notebook.Application.exe.config` file.

The following XML example is an example of the user setting section showing the `MachineIndependentQuickFilePaths` setting.

```
<userSettings>
 <Symyx.Notebook.Applications.Properties.Settings>
 <setting name="MachineIndependentQuickFilePaths" serializeAs="String">
 <value>False</value>
 </setting>
 </Symyx.Notebook.Applications.Properties.Settings>
</userSettings>
```

To create a roaming or offline profile with `MachineIndependentQuickFilePaths` property set to true, the administrator must:

- Create a working directory using the `Deployment Manager GENERATE` command from an existing profile.
- Edit the `Symyx.Notebook.Application.exe.config file` using an ASCII file editor to change the `MachineIndependentQuickFilePaths` setting from *False* to *True*.
- Create a new profile using the DM ADD-PROFILE command.

  The `ADD-PROFILE` command takes the content of the Quick File working directory and creates a new profile with the specified name.

  A standard DM installation includes a Citrix profile already with this setting set to true.

The Quick File window is always available if the user selects the View option. The `MachineIndependentQuickFilePaths` setting control the Quick File window behavior, to use one set of folders for the user all the time, or use a set of folders specific to the user and the computer where the DM client is installed.

ForceCheckInOnClose Setting

You can set user preferences to force the user to check in an experiment when the experiment is closed. The user is not given the option to keep the experiment checked out when the `ForceCheckInOnClose` property is set to true.

Modify the `Symyx.Notebook.Application.exe.config` file to change the `ForceCheckInOnClose` property's value. The default value in the `ForceCheckInOnClose` property is *False* to enable users to select their preference settings in DM. Use the instructions in Create a Release Profile step 1 to change the `ForceCheckInOnClose` property to *True*.

Reconcile Private Repository

If an incomplete check-in occurs because of an issue such as an interruption in network connectivity or premature shutdown of a client computer during the check-in process, files in a user's local repository and those in the Vault repository can become out of sync.

The user can reconcile differences by using a **Reconcile** dialog box that identifies each conflict and provides options for resolving each conflict.

> **Note:** If a user's **OnStartup** preference is set to **Notebook Explorer**, the **Reconcile** window cannot be displayed automatically or made available on the right-click menu, either. To access the **Reconcile** window, the user must change their OnStartup preference to a different option, log out, and then log back in to Workbook.

An administrator can control whether and when the Reconcile dialog box is displayed by changing the value of the following setting in the client configuration file, `Symyx.Notebook.Application.exe.config`.

Setting	Description
`ReconcilePrivateRepository`	Controls when the Reconcile Check In Conflicts dialog appears. The options are: ■ **Never** - Users are not allowed to reconcile check-in conflicts. ■ **OnDemand** *(Default)* - Users can right-click a repository and select **Reconcile Check In Conflicts**. ■ **OnStartup** - If conflicts exist, DM displays the dialog on startup, as well as allowing a user to right-click the repository and select Reconcile Check In Conflicts. **Note:** If you choose **OnStartup**, be aware that the start-up process can take longer.

Document Checkout Settings

Workbook provides the following settings for controlling checkouts and preventing potential version conflicts that can occur when an experiment is opened for editing on two different computers at the same time:

- `OpenForEditWhenCheckedOutToMeOnAnotherMachine`:

  When a scientist checks out an experiment on one computer and then views it on another computer, this setting controls the color of the check-out state icon Workbook and what happens if the scientist attempts to open the experiment on the other computer. Possible values for this setting are:

  - **Info**: Display a green icon and allow the scientist to open and edit the experiment.

    > **Note:** The default value for the Citrix profile is **Info**, because a separate session management feature is used to prevent users from opening the same experiment on multiple computers at the same time in a Citrix environment.

  - **Warn**: Display a yellow icon and allow the scientist to open the experiment, but only after responding to a warning message about the check-out state.
  - **Error** (the default setting): Display a red icon and do not allow the scientist to check out the experiment.

- `AllowCheckInWhenCheckedOutToMeOnAnotherMachine`

  This setting is a Boolean value that controls whether the scientist can check in an experiment on one computer that is already checked out to that scientist on another computer. The default setting is False in all profiles except the standard Citrix profile.

Performance Monitoring

In the Deployment Manager working directory, create a profile and enable performance logs in the `Symyx.Notebook.Application.exe.config` file.

To enable performance monitoring:

- Set `LoginPerformanceCounters` and `OtherPerformanceCounters` to true in the `Symyx.Notebook.Application.exe.config` file.

Enable Debug Logging

In the BIOVIA Workbook Deployment Manager working directory, create a profile and enable debug logging in the `Symyx.Notebook.Application.exe.config` file.

To enable debug logging:

1. Create a backup copy of the `Symyx.Notebook.Application.exe.config` file.
2. Open the `Symyx.Notebook.Application.exe.config` file in a text editor.
3. Navigate to the following section in the file:

```
<root>
  <level value="OFF" />
  <appender-ref ref="Symyx.Tools.FileLogger" />
</root>
```

4. In `level value`, change the `OFF` to *`ALL`*.

   The section should look like the following:

```
<root>
  <level value="ALL" />
  <appender-ref ref="Symyx.Tools.FileLogger" />
</root>
```

Chapter 5:
Create a Client Package

To create a customized version of the DM client for users:

1. Create a Namespace.

   If your company does not already have a namespace for packages, you must create one. Create a new public/private key pair using the following command:

   ```
   VaultADMStoreManager /vault <server> <domain>\<user> <password> create-
   key MyOrganization.keypair
   ```

   For example:

   ```
   VaultADMstoreManager /vault vm-avs69 vm-avs69\vault.admin abc123 create-
   key C:\Temp\MYCompany.keypair
   ```

   The newly-created MYCompany.keypair file has a public and private key for signing packages, and a public key for validating them. Store the key pair in a secure location to use when creating a release package.

2. Generate a public-key-only file using the following command:

   ```
   VaultADMStoreManager /vault <server> <domain>\<user> <password> get-
   public-key MyOrganization.keypair MyOrganization.publickey
   ```

   For example:

   ```
   VaultADMstoreManager /vault vm-avs69 vm-avs69\vault.admin abc123 get-
   public-key C:\Temp\MYCompany.keypair C:\Temp\MYCompany.publickey
   ```

3. Store the public key file to use each time you create your organization’s package namespace in new Vault environments.

   You must reuse the key-pair files for each usage of the namespace in the development process.

4. Set up a namespace on your test or development server:

   ```
   VaultADMStoreManager /vault <server> <domain>\<user> <password> add-
   namespace MyOrganization MyOrganization.publickey
   ```

   For example:

   ```
   VaultADMstoreManager /vault vm-avs69 vm-avs69\vault.admin abc123 add-
   namespace MYCompany C:\Temp\My Company Name.publickey
   ```

   The new package namespace is added to your Vault server and is associated with the public key you provided. Packages that are not signed with the corresponding key pair are rejected by the Vault server. You can create as many packages as you want with no risk of colliding with packages released by BIOVIA or other parties.

Create and Test a Custom Package

To add the new package and profile:

1. Use the command:

   ```
   VaultADMStoreManager /vault <server> <domain>\<user> <password> add-
   package <NewProfileName> <namespace> <NewPackageName>
   <MyOrganization.keypair> <NewPackageDir>
   ```

For example:

```
VaultADMstoreManager /vault vm-avs69 vm-avs69\vault.admin abc123 add-
package MyPackage1.V6.9.0.818-Roaming MYCompany MyPackage1.V6.9.0.818-
Roaming C:\Temp\MYCompany.Keypair C:\TEMP\Rules_package
```

Note: Do not use a plus (+) character or other special characters in package or profile names. If you do so, when a user whose account uses that package or profile attempts to log on, the system displays an message that says it cannot log them on to Vault due to an error on the server.

The newly created package is automatically in test mode. Packages in test mode are refreshed by clients each time the package logs into the DM client. Test mode enables updating the package and prevents exporting the package to another server. You can only update test packages. You cannot update release packages.

2. Download the current mapping rules by using the following command:

   ```
   VaultADMStoreManager /vault <server> <domain>\<user> <password>
   download-rules <directory\filename>
   ```

   For example:

   ```
   VaultADMstoreManager /vault vm-avs69 vm-avs69\vault.admin abc123
   download-rules C:\temp\rules.xml
   ```

3. List the available profiles, using the following command:

   ```
   VaultADMStoreManager /vault <> <domain>\<user> <password> list-profiles
   ```

   For example:

   ```
   VaultADMstoreManager /vault vm-avs69 vm-avs69\vault.admin abc123 list-
   profiles
   ```

   Example Results:

   ```
   V6.9.0.x-Offline
   V6.9.0.x-Roaming
   V6.9.0.x-Citrix
   V6.9.0.x-Offline
   V6.9.0.x-Roaming
   V6.9.0.x-Citrix
   MyPackage1.V6.9.0.818-Roaming
   ```

4. Update the mapping rules, using the following example as a guide.

   Example:

   ```
   <?xml version="1.0" encoding="utf-8"?>
     <Rules xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            defaultProfile="V6.9.0.818-Roaming">
     </Rules>
   ```

   Example Results:

   ```
   <?xml version="1.0" encoding="utf-8"?>
     <Rules xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            defaultProfile="MyPackage1.V6.9.0.818-Roaming">
     </Rules>
   ```

5. Upload the changed rules for testers.

   ```
   VaultADMStoreManager /vault <server> <domain>\<user> <password> upload-
   rules <directory\filename>
   ```

   For example:

   ```
   VaultADMstoreManager /vault vm-avs69 vm-avs69\vault.admin abc123 upload-
   rules C:\temp\rules.xml
   ```

6. Select a pilot group to test the new release profile and release package.

Release a Custom Package

When you are ready to promote the profile and package to a validation environment, mark the package released.

A package is read-only when in release mode. The package content is safe to export and import into other BIOVIA Vault Server environments. You can export a release package using the export-package command. You can export a release profile using the export-profile command.

To release a custom package:

1. Use the following command:

   ```
   VaultADMStoreManager /vault <server> <domain>\<user> <password> release
   MyOrganization NewPackageName
   ```

   For example:

   ```
   VaultADMstoreManager /vault vm-avs69 vm-avs69\vault.admin abc123 release
   MYCompany MyPackage1.V6.9.0.818-Roaming
   ```

2. Create a portable package and profile for installation on other machines, using the following command:

   ```
   VaultADMStoreManager /vault <server> <domain>\<user> <password> export-
   package MyOrganization PackageName package-file
   ```

   or

   ```
   VaultADMStoreManager /vault <server> <domain>\<user> <password> export-
   profile NewProfileName profile-file
   ```

   For example:

   ```
   VaultADMstoreManager /vault vm-avs69 vm-avs69\vault.admin abc123 export-
   package MYCompany MyPackage1.V6.9.0.818-Roaming C:\Temp\MYCompany
   ```

   You can promote the assets through your validation environments and release to the production environment.

3. Create ZIP file for transport to other machines, for example containing:

   - `C:\Temp\MYCompany.admprof`
   - `C:\Temp\MYCompany.admpkg`
   - `MYCompany.publickey`

Install Custom Package on Clients

To deploy customized profiles:

1. (Optional) Set up a namespace on your test or development server when a namespace has not been initialized.

   - Run the following command to check for a valid namespace, and look for your namespace in the returned list:

     ```
     VaultADMStoreManager /vault <server> <domain>\<user> <password> list-
     namespaces
     ```

     For example:

     ```
     VaultADMStoreManager /vault vm-avs69 vm-avs69\vault.admin abc123
     list-namespaces
     ```

   - If the namespace is not found, add the namespace using the following command:

     ```
     VaultADMStoreManager /vault <server> <domain>\<user> <password> add-
     namespace MyOrganization MyOrganization.publickey
     ```

     For example:

     ```
     VaultADMStoreManager /vault vm-avs69 vm-avs69\vault.admin abc123 add-
     namespace MYCompany C:\Temp\My Company Name.publickey
     ```

     This adds a new package namespace to your Vault server and associates it with the public key you provided. Packages that are not signed with the corresponding key pair are rejected by the BIOVIA Vault Server. This allows you to create as many packages as you want with no risk of colliding with packages released by BIOVIA or other third-parties.

2. Publish the new Release Package from the zip file, using the following command:

   ```
   VaultADMStoreManager /vault <server> <domain>\<user> <password> import-
   package <directory\filename>.admpkg
   ```

   For example:

   ```
   VaultADMStoreManager /vault vm-avs69 vm-avs69\vault.admin abc123 import-
   package C:\Temp\MYCompany.admpkg
   ```

3. Publish the new Release Profile from the zip file:

   ```
   VaultADMStoreManager /vault <server> <domain>\<user> <password> import-
   profile <directory\profilefilename>
   ```

   For example:

   ```
   VaultADMStoreManager /vault vm-avs69 vm-avs69\vault.admin abc123 import-
   profile C:\Temp\MYCompany.admprof
   ```

4. Download the current rules by using the following command:

   ```
   VaultADMStoreManager /vault <server> <domain>\<user> <password>
   download-rules <directory\filename>
   ```

   For example:

   ```
   VaultADMstoreManager /vault vm-avs69 vm-avs69\vault.admin abc123
   download-rules C:\temp\rules.xml
   ```

5. Edit the current rules. Use the example as a guide.

   For example:

   ```
   <?xml version="1.0" encoding="utf-8"?>
     <Rules xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   ```

```
          defaultProfile="V6.9.0.818-Roaming">
   </Rules>
```

For example:

```
<?xml version="1.0" encoding="utf-8"?>
  <Rules xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns:xsd="http://www.w3.org/2001/XMLSchema"
         defaultProfile="MyPackage1.V6.9.0.818-Roaming">
  </Rules>
```

6. Upload the changed rules for users, use the following command:

   ```
   VaultADMStoreManager /vault <server> <domain>\<user> <password> upload-
   rules <directory\filename>
   ```

   For example:

   ```
   VaultADMstoreManager /vault vm-avs69 vm-avs69\vault.admin abc123 upload-
   rules C:\temp\rules.xml
   ```

   As a result, each user who is assigned to the profile automatically receives the new package download when logging into the DM client.