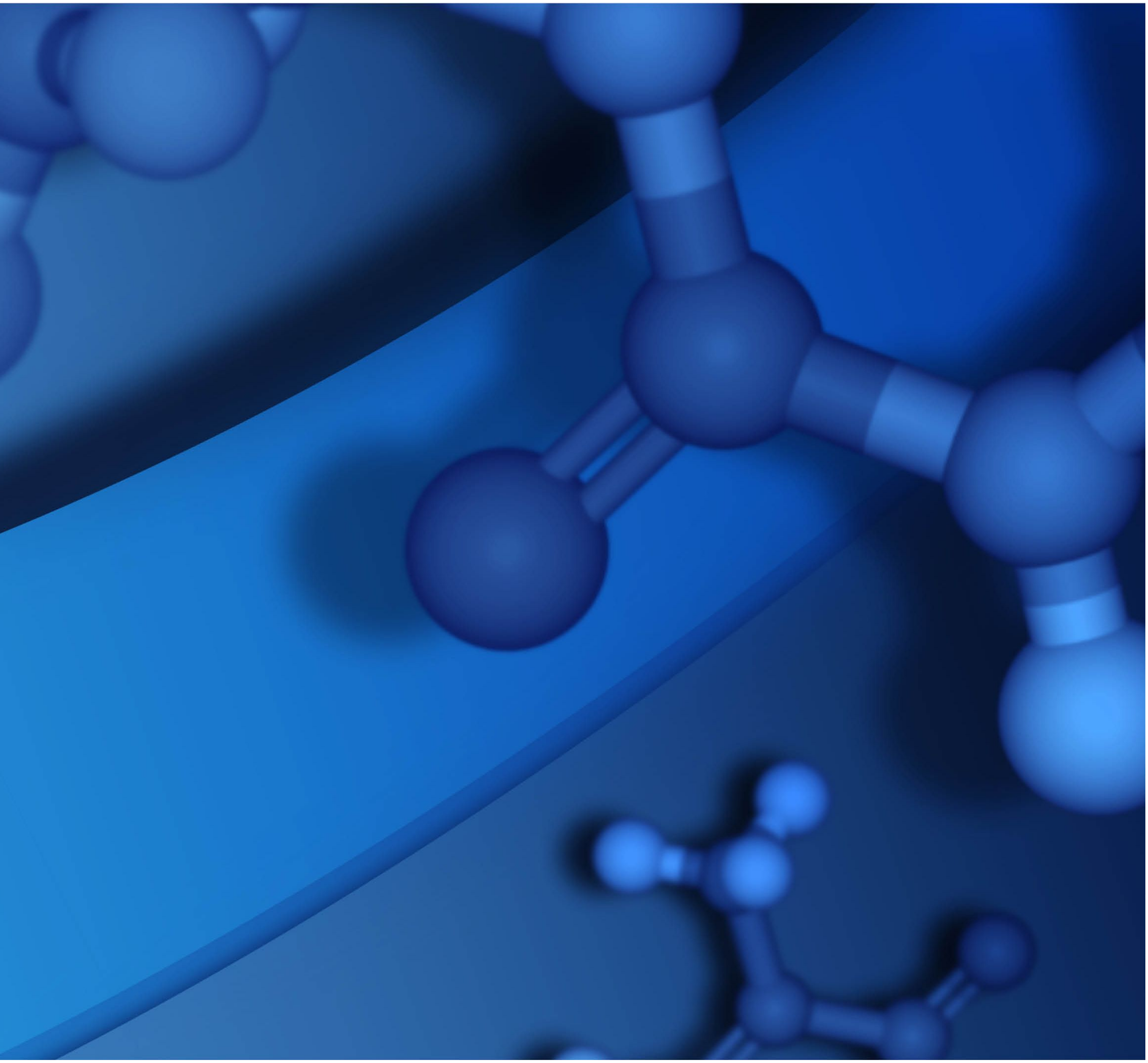
VAULT SERVER ADMINISTRATION GUIDE

BIOVIA WORKBOOK 2021



Copyright Notice

Acknowledgments and References

To print photographs or files of computational results (figures and/or data) obtained by using Dassault Systèmes software, acknowledge the source in an appropriate format. For example:

> "Computational results were obtained by using Dassault Systèmes BIOVIA software programs. BIOVIA Workbook was used to perform the calculations and to generate the graphical results."

Dassault Systèmes may grant permission to republish or reprint its copyrighted materials. Requests should be submitted to Dassault Systèmes Customer Support, either by visiting https://www.3ds.com/support/ and clicking **Call us** or **Submit a request**, or by writing to:

Dassault Systèmes Customer Support
10, Rue Marcel Dassault
78140 Vélizy-Villacoublay
FRANCE

Contents

Chapter 1: Vault Server Services and Monitoring Tools

IMPORTANT! `<INSTALL_PATH>` is the root of the Vault Server installation.
On Windows this is typically:
`C:\Program Files (x86)\BIOVIA\Vault\`

Vault Server Services

BIOVIA Vault Server uses the following Windows services and Web applications that start automatically when the Windows server boots up and stop automatically when the server is shut down:

- **Vault Tomcat Server containing RAS**

  RAS, a web service, indexes object properties, contents, structures, and the reaction STS, web application.

- **Vault Workflow Service**

  Runs objects through custom workflows. The Workflow service also sends messages to the Message Processing Service.

- **Vault Message Processing Service**

  Processes messages for indexing and keeps the workflow history current.

- **Vault Client Service**

  Runs the Vault Client.

- **Vault Hub Synchronization Service**

  Maintains synchronization with Foundation Hub.

- **World Wide Web Publishing Service**

You must use Microsoft Internet Information Services (IIS) to use the BIOVIA Vault Server. Stop IIS and the associated servers before changing any BIOVIA Vault Server configuration or executable files.

Starting the Vault Services

Use the Windows Server Manager to start the Vault Server services.

> **Note:** The Oracle instance that BIOVIA Vault Server connects to must be started before the Vault services. If the database is shut down for any reason, stop the Vault services, restart the Oracle instance, and then restart the Vault services.

To start the Vault services:

1. Open the Windows **Server Manager** and choose **Configuration** > **Services**.
2. Start the following services in this order:
   a. World Wide Web Publishing Service
   b. Vault Tomcat Server Service
   c. Vault Workflow Service
   d. Vault Message Processing Service

e. Vault Hub Synchronization Service

f. Vault Client Service

Starting Vault Services from the Command Line

You can create a batch file to reset IIS and start the Vault services, and then execute the batch file manually or by using a scheduled task.

To create a batch file that resets IIS and starts the Vault services:

1. Use any text editor to create a text file that contains the following commands:

```
iisreset /start
net start "World Wide Web Publishing Service"
net start "Vault Tomcat Server
net start "Vault Workflow Service"
net start "Vault Message Processing Service"
net start "Vault Hub Synchronization Service"
net start "Vault Client Service"
```

2. Save the file with a `.bat` or `.cmd` extension.

Stopping the Vault Services

IMPORTANT! Before you shut down the Oracle database or start an upgrade, ensure that all users are logged off, and then stop the Vault services.

To stop the Vault services:

1. Open the Windows **Server Manager** and choose **Configuration** > **Services**.
2. Stop the following services in this order:

   a. Vault Client Service

   b. Vault Hub Synchronization Service

   c. Vault Message Processing Service

   d. Vault Workflow Service

   e. Vault Tomcat Server Service

   f. World Wide Web Publishing Service

Stopping Vault Services from the Command-line

You can write the commands to stop the services to a batch file using an unformatted text file that contains one or more commands and ends with a `.bat` or `.cmd` file name extension. Execute the batch manually or use a scheduled task.

The commands are:

```
net stop "Vault Client Service"
net stop "Vault Hub Synchronization Service"
net stop "Vault Workflow Service"
net stop "Vault Message Processing Service"
net stop "Vault Tomcat Server"
net stop "World Wide Web Publishing Service"
iisreset /stop
```

Monitoring Tools

Tools you can use to monitor your Vault Server include:

- IIS Application Pool Defaults

  See http://www.iis.net/configreference/system.applicationhost/applicationpools/applicationpooldefaults

- The Vault Message Processing Service
  - Recycles message processor on win32 exceptions recorded to Vault logs.
  - Recycles message processor every 24 hours, this default value is configurable.
- Apache/Tomcat, the default Apache/tomcat behavior

Excluding Vault Server Folders from Anti-virus Scans

To avoid a negative impact on performance, exclude the following folders from anti-virus scanning:

- The Vault Server installation folders, which by default are in the following path:

  ```
  C:\Program Files (x86)\BIOVIA
  ```

- The Vault Logs folder, which by default is in the following path:

  C:\VaultLogs

> **Note:** Also exclude Workbook client system folders, as described in the *Workbook Client Installation Guide*.

Changing the Message Processor Restart Interval

You can configure the interval at which the Message Processor is restarted.

The value is specified in hours and the lowest allowed value is 0.5.

To change the message processor restart interval:

1. Stop the BIOVIA Vault Server services before modifying the server configuration file.

   For more information, see Stopping the Vault Services.

2. Navigate to the <INSTALL_PATH>`\Vault\WindowsServices` folder and open `Symyx.Vault.Message.Processing.Service.exe.config` in a text editor.
3. Back up the configuration file before you modify it.
4. Locate the `<add key="RestartInterval" value="24.0" />` element.

   For example:

   ```
   <appSettings>
   ...
   <add key="RestartInterval" value="24.0" />
   ...
   </appSettings>
   ```

5. Specify your required value for the `RestartInterval`.
6. Save and close the file.
7. Execute the `iisreset` command.

Chapter 2: Updating Connection and Configuration Parameters

You can review and update the following Vault Server connection and configuration parameters by using the Vault Configuration Tool:

- Foundation connection details (Hub URL, Name, and Password)
- SMTP connection details (SMTP Server, Port, Email To, and Email From)
- Certificate information (.pfx File path, Name, and Password)
- Service ports for RAS, Workflow, and VCS
- Log file output path
- Vault Endpoint
- Connection strings, username, and password information for connecting to the Site, File Service, RAS, and Workflow datasources

For information about these parameters, see your *Vault Server Installation Guide*.

The Vault Configuration Tool provides two interfaces:

- A graphical user interface for viewing and updating the parameter values, applying them, and saving them to a file
- A command-line interface for deploying the values specified in a file

Tip: The command-line interface facilitates updating values across multiple servers.

When you use the Vault Configuration Tool to apply changes, the tool restarts the required services automatically. After the restart, your changes take effect.

Running the Vault Configuration Tool (UI Mode)

To run the Vault Configuration Tool using its graphical user interface mode:

1. Log on to the Vault Server as a member of the Vault Global Administrators group.
2. Choose Start > Program Files > BIOVIA > Vault Configuration Tool.
3. Update the parameters on the **Server Configuration** and **Database Connections** tabs as needed.
4. (Optional) To save your entries to an XML parameters file, click **Save** and enter a filename and location.

   You can use this file for general reference and also specify it as the input file when using the command-line mode of the Vault Configuration Tool.
5. Click **Apply**.

Running the Vault Configuration Tool (cmd Mode)

To run the Vault Configuration Tool from the command-line:

1. Log on to the Vault Server as a member of the Vault Global Administrators group.
2. Update the configuration parameters file as needed. You can use the file created through the user interface mode of the tool or copy the following file and then edit its values as needed:

   `C:\Program Files (x86)\BIOVIA\$Uninstall\Install.parameters`.
3. Open a command-line prompt and navigate to the Vault `Utilities` folder, by default `C:\Program Files\BIOVIA\Vault\workbook2021\Utilities`.
4. Execute the following command:

   ```
   VaultConfigurationTool.Console.exe <parametersFilePath>
   ```

   Where `<parametersFilePath>` specifies the path to your parameters file.

   > **Tip:** You can get information about each parameter by executing the command without specifying a <parametersFilePath>.

Troubleshooting Vault on a Renamed Server

If Vault services are unresponsive on a renamed server, make sure Oracle is accessible by logging in as the site schema or RAS schema owner.

1. Verify RAS is running by browsing the status page from RAS Status shortcut.
2. Verify the Vault service page is accessible page and has no errors from Vault Web Service shortcut.
3. Check Vault service log files for errors, for example, at `C:\VaultLogs`.
4. Check `<INSTALL_PATH>\Vault\Tomcat\logs\RAScore.log` for errors.
5. Double-check the changes you made to the service configuration files and make sure there are no other occurrences of the old server name.
6. If all services appear to be running, but you cannot log in from a Workbook client:
   - Make sure you have installed the server certificate to Trusted Root Certification Authorities store on the client.
   - Browse to `https://<YourServerName>`

     If you get a certificate error: view the certificate and select the option to install it.

     You might need to use the Run As Administrator option to start the browser in order to be able to install the certificate from the browser.
   - Verify that the clock settings on the client and server are within 5 minutes of each other.
   - Remove any files or folders under:

     `%ALLUSERSPROFILE%\Symyx Technologies`

Chapter 3: Setting Logging Levels

Workbook writes all errors and alerts to log files, but the amount of detail that is logged depends on the logging level settings.

By default, the logging level for all log files is set to ERROR to minimize the impact of logging on system performance. You can, however, increase the logging levels temporarily in order to diagnose issues.

The following table lists the logging level options.

Logging Level Value	Description
ERROR	Logs all errors and alerts; creates the smallest log file. This is the default logging level.
DEBUG	Logs anything marked as DEBUG in the code and logs all warnings, errors, and alerts; creates a larger log file that contains more detailed information.
ALL	Logs all information; creates a very large file that can negatively impact performance. Use for troubleshooting a problem at the request of Dassault Systèmes Customer Support.

Log Files and Locations

Log files are written to the folder that you specified when you installed BIOVIA Vault Server. The default location is `C:\VaultLogs`.

The log files contain information about the Vault Workflow Service, Vault Message Processing Service, File Service, ADM, and the Vault Tomcat Server services.

Use the log files to analyze any problems that might occur while running the BIOVIA Vault Server. The maximum number of log files is 10. The files are reused after reaching the maximum of 10 files.

Monitor the `C:\VaultLogs` folder for growth. If disk space becomes an issue, you can archive the log files to a different location or you can delete the files from the `C:\VaultLogs` folder.

Modifying Vault IIS Service Logging Parameters

To modify logging parameters for the Vault private, public, and file services:

1. Locate the `web.config` file in the `<INSTALL_PATH>\Vault\PrivateWebService` folder on the computer where Vault Server is installed.

   Perform the same steps in the `web.config` files for the public and the file service.

   - The public service `web.config` file is located in `<INSTALL_PATH>\Vault\WebService`.
   - The file service `web.config` file is located in `<INSTALL_PATH>\Vault\FileService`.
2. Create a backup copy of the files.
3. Open each the `web.config` file in a text editor.

4. Locate the `<log4net>` child element that begins with `<logger name=`, for example:

   ```
   <logger name="Symyx">
      <level value="ERROR" />
      <appender-ref ref="Vault.Service.FileLogger.Debug" />
   </logger>
   <logger name="Accelrys">
       <level value="ERROR" />
       <appender-ref ref="Vault.Service.FileLogger.Debug" />
   <logger>
   ```

5. Change the logging level as required.
6. Change the amount of log files written to disk by changing `MaxSizeRollBackups` as required, for example:

   ```
   <param name="MaxSizeRollBackups" value="10" />
   ```

7. The log files are saved to the folder that was specified when you installed Vault Server. If you save the log files to another folder, modify the `<param name="File" ...>` element to the correct location, for example:

   ```
   <param name="File" value="C:\VaultLogs\VaultDebug.Private.log"/>
   ```

   For the public service use:

   ```
   <param name="File" value="C:\VaultLogs\FileService.log"/>
   ```

   For the file service use:

   ```
   <param name="File" value="C:\VaultLogs\VaultDebug.log"/>
   ```

8. Save and close the `web.config` files.
9. Execute the `iisreset` command.

Modifying ADM Service Logging Parameters

The `ADM-Activity.log` is the log file for the `Accelrys.Deployment.Management.Service.Rest.ActivityLogger`.

The `ADM-Diagnostic.log` is the log file for `Accelrys.Deployment.Management.Service.Rest.DiagnosticLogger`.

The `FileService.log` is the log file for `Accelrys.FileService.Logger.`

To modify logging parameters:

1. Locate the `web.config` file in the `<INSTALL_PATH>\Vault\DeploymentManager` folder on the computer where Vault Server is installed.
2. Create a backup copy of the file.
3. Open the `web.config` file in a text editor.
4. Locate the `<log4net>` child element that begins with `<logger name=`, for example:

   ```
   <logger
   name="Accelrys.Deployment.Management.Service.Rest.DiagnosticLogger">
     <level value="ERROR" />
     <appender-ref ref="Accelrys.Deployment.Manager.Diagnostics" />
   </logger>
   <logger
   name="Accelrys.Deployment.Management.Service.Rest.ActivityLogger">
   ```

```
    <level value="ERROR" />
    <appender-ref
 ref="Accelrys.Deployment.Management.Service.Rest.ActivityLogger" />
 <logger>
```

5. Change the logging level as required.
6. Change the amount of log files written to disk by changing `MaxSizeRollBackups` as required, for example:

   ```
   <param name="MaxSizeRollBackups" value="10" />
   ```

7. The log files are saved to the folder that was specified when you installed Vault Server. If you save the log files to another folder, modify the `<param name="File"...>` element to the correct location, for example:

   ```
   <param name="File" value="C:\VaultLogs\ADM-Activity.log"/>
   <param name="File" value="C:\VaultLogs\ADM-Diagnostic.log"/>
   ```

   or use:

   ```
   <param name="File" value="C:\VaultLogs\FileService.log"/>
   ```

8. Save and close the `web.config` file.
9. Execute the `iisreset` command.

Modifying Vault Message Processor Logging Parameters

To modify logging parameters:

1. On the computer on which the Vault Server is installed, navigate to `Symyx.Vault.MessageProcessor.exe.config` file in folder `<INSTALL_PATH>\Vault\WindowsServices\MessageHandlers\`VaultQueue.
2. Create a backup copy of the file and then open it in a text editor.
3. To change logging levels, find the `<log4net>` child element that begins with `<logger name=` and change the level values as needed to meet your needs, for example:

   ```
   <logger name="Symyx.Vault.Message.Processor.Service">
     <level value="ERROR" />
     <appender-ref ref="Vault.Message.Processing.Service.FileLogger.Debug"
   />
   </logger>
   <logger name=Symyx.Framework.Vault.Messaging">
      <level value="ERROR" />
      <appender-ref ref="Vault.Message.Processing.Service.FileLogger.Debug"
   />
   <logger>
   ```

4. To change the maximum size of the log files, edit the value of the following parameter:

   ```
   <param name="MaxSizeRollBackups" value="10" />
   ```

5. To change the location to which the log files are stored, edit the value of the following parameter:

   ```
   <param name="File"
    value="C:\VaultLogs\SymyxVaultMessageProcessor.log"/>
   ```

6. Save and close the file.
7. Execute the `iisreset` command.

Modifying Vault Workflow Service Logging Parameters

To modify logging parameters:

1. Locate the `Symyx.Workflow.Service.exe.config` file in the `<INSTALL_PATH>\Vault\WindowsServices` folder on the computer where Vault Server is installed.
2. Create a backup copy of the file.
3. Open the `Symyx.Workflow.Service.exe.config` file in a text editor.
4. Locate the `<log4net>` child element that begins with `<logger name=`, for example:

```
<logger name="Symyx.Workflow">
  <level value="ERROR" />
  <appender-ref ref="Symyx.Workflow.FileLogger.Debug" />
</logger>
<logger name=Symyx.Workflow">
   <level value="ALL" />
   <appender-ref ref="Symyx.Workflow.FileLogger.Debug" />
<logger>
```

5. Change the logging level as required.
6. Change the amount of log files written to disk by changing `MaxSizeRollBackups` as required, for example:

```
<param name="MaxSizeRollBackups" value="10" />
```

7. The log files are saved to the folder that was specified when you installed Vault Server. If you save the log files to another folder, modify the `<param name="File" ... >` element to the correct location, for example:

```
<param name="File" value="C:\VaultLogs\VaultWorkflow.log"/>
```

8. Save and close the `Symyx.Workflow.Service.exe.config` file.
9. Execute the `iisreset` command.

Modifying RAS Service Logging Parameters

The RAS log files are saved to two locations on the Vault server:

- `C:\vaultlogs\ras-server\*`
- `<INSTALL_PATH>\Vault\Tomcat\logs`

To modify logging parameters for the RAS service:

1. Stop the following services in this order:
   a. Vault Client Service
   b. Vault Hub Synchronization Service
   c. Vault Message Processing Service
   d. Vault Workflow Service
   e. Vault Tomcat Server Service
   f. World Wide Web Publishing Service
2. Locate the `RASConfig.xcf` file in the `<INSTALL_PATH>\Vault\Tomcat\conf` folder on the computer where BIOVIA Vault Server is installed.
3. Create a backup copy of the file.

4. Open the `RASConfig.xcf` file in a text editor.
5. Locate the `<log>` element:

   ```
   <Log>
     <PagerTo>admin@mycompany.com</PagerTo>
     <EventDebug>0</EventDebug>
     <EmailTo>admin@mycompany.com</EmailTo>
     <MapDebug>0</MapDebug>
     <SQLDebug>1</SQLDebug>
     <Level>ERROR</Level>
     <DebugXMLPath>C:\VaultLogs\ras-server\debug</DebugXMLPath>
     <CommDebugMaxMessageLength>512</CommDebugMaxMessageLength>
     <DeepDebug>0</DeepDebug>
     <PagerLevel>ERROR</PagerLevel>
     <CommDebug>1</CommDebug>
     <EmailLevel>ERROR</EmailLevel>
   </Log>
   ```

6. Change the logging level as required.
7. Change the amount of log files written to disk by changing `MaxSizeRollBackups` as required, for example:

   ```
   <param name="MaxSizeRollBackups" value="10" />
   ```

8. Save and close the `RASConfig.xcf` file.
9. Start the following services in this order:
   a. World Wide Web Publishing Service
   b. Vault Tomcat Server Service
   c. Vault Workflow Service
   d. Vault Message Processing Service
   e. Vault Hub Synchronization Service
   f. Vault Client Service

Modifying Workbook Logging Parameters

The default logging level for BIOVIA Workbook is ERROR, all errors and alerts are written to the log file. Changing the logging level value to DEBUG or ALL negatively impacts performance. You should only use DEBUG or ALL to diagnose problems.

To modify logging parameters:

1. Locate the `Symyx.Administration.MMC.dll.config` file in the `<INSTALL_PATH>\Vault Administration\MMC` folder on the computer where Vault Administration Console is installed.
2. Create a backup copy of the file.
3. Open the `Symyx.Administration.MMC.dll.config` file in a text editor.
4. Locate the `<log4net>` child element that begins with `<level value=`, for example:

   ```
   <logger name="Symyx">
     <level value="ERROR" />
     <appender-ref ref="SnapIn.Debug" />
   </logger>
   <logger name=Symyx.Workflow">
   ```

```
   <level value="ALL" />
   <appender-ref ref="Symyx.Workflow.FileLogger.Debug" />
<logger>
```

5. Change the logging level as required.
6. The log files are saved to `%ALLUSERSPORFILE%\Application Data\Symyx Technologies\LogFiles` by default. If you save the log files to another folder, modify the value in the `<param name="File" ...>` element to the location, for example:

   ```
   <param name="File"
   value="c:\AdministrationConsoleLogs\MMC.SnapIn.Debug.%property
   {LogFileName}[%random{3}].log"/>
   ```

7. Save and close the `Symyx.Administration.MMC.dll.config` file.
8. Execute the `iisreset` command.

Chapter 4: Replacing Clear-text Oracle Schema Passwords

This chapter describes how to use the Password Replacer utility to encrypt the original clear-text passwords used for the Oracle schema owner accounts. The utility is located in the `<INSTALL_PATH>/Utilities` folder and consists of two files:

- `passwordreplacer.exe`
- `passwordreplacer.exe.config`

Backing up Configuration Files

The Password Replacer modifies several `.config` files. Before you run the utility, back up the following files in the `<INSTALL_PATH>/WindowsServices` folder and subfolders:

- `/DeploymentManager/web.config`
- `/Rest/web.config`
- `/WebService/web.config`
- `/PrivateWebService/web.config`
- `Symyx.Vault.Message.Processing.Service.exe.config`
- `Symyx.Workflow.Service.exe.config`
- `/Utilities/Symyx.Workflow.Utility.exe.config`
- `/Utilities/Symyx.Vault.DatabaseUtility.exe.config`
- `/FileService/web.config`
- `/Tomcat/conf/rasconfig.xcf`

The backups of these files will contain the original password in clear text. After verifying that your system operates normally with the encrypted passwords, BIOVIA recommends that you either delete the backup files or store them in a secure area.

Running the Password Replacer Utility

The BIOVIA Vault Server configuration files that specify an Oracle connection string are modified when you run the Password Replacer utility.

To run the Password Replacer:

1. Stop the following services in this order:
   a. Vault Client Service
   b. Vault Hub Synchronization Service
   c. Vault Message Processing Service
   d. Vault Workflow Service
   e. Vault Tomcat Server Service
   f. World Wide Web Publishing Service
2. Open a command window.
3. Navigate to the `<INSTALL_PATH>/Vault/Utilities` folder.

4. At the prompt, enter the Password Replacer command with arguments, for example:

   ```
   passwordreplacer.exe –avspath
   ```

   `-avspath` tells the Password Replacer utility to automatically identify Vault file locations.

   (Optional) Follow this argument with your Vault Install path.
5. Start the following services in this order:
   a. World Wide Web Publishing Service
   b. Vault Tomcat Server Service
   c. Vault Workflow Service
   d. Vault Message Processing Service
   e. Vault Hub Synchronization Service
   f. Vault Client Service
6. Repeat steps 1-5 for each Vault server in the Vault Server installation.

There is a summary at the end of the output that shows how many files the utility modified, the number of passwords that were replaced, and the number of errors that occurred, if any.

If errors were encountered, the system might not be functional. You must investigate the log file, resolve the errors, restore the original configuration files, and rerun the Password Replacer utility.

Password Replacer Logging

Logging is configured using log4net. By default, the utility creates a `PasswordReplacer.log` in the folder from where you ran the utility. The logging level INFO messages and above are logged to the console.

Example content of the `PasswordReplacer.log` file:

```
2012-04-03 15:09:24,866 [1] INFO Accelrys.Security.PasswordReplacer.Replacer
[(null)] - Replacing passwords in 'c:\Program Files (x86)\Symyx\Symyx Vault
6.7\SymyxServer\Tomcat6\conf\RASConfig.xcf'
2012-04-03 15:09:24,866 [1] DEBUG
Accelrys.Security.PasswordReplacer.XPathReplacer [(null)] - Searching for
XPath '/RASConfig/Databases/SymyxDatabase/Password[not(Encrypted)]'
2012-04-03 15:09:24,866 [1] DEBUG
Accelrys.Security.PasswordReplacer.XPathReplacer [(null)] - Match found at
Password
2012-04-03 15:09:25,178 [1] INFO
Accelrys.Security.PasswordReplacer.RasReplacer [(null)] - Replaced password
with
'01000000D08C9DDF0115D1118C7A00C04FC297EB01000000F8CFE8795087DC4CB4A4D4BA6A0
AD00F0400000002000000
000003660000C000000010000000F5BBF653622CAB818D27D04E8AA31966000000004800000
A0000000100000009E212

537F4E583D3B3944A85CC4ABB9F280000009A710C2320012CFFAD09C8D0C175CDA8DD88C39DC
FBEB50536B331711A7201
BA6E357139894294FC140000001E30F6A8DC4289DBF1BA319558CEAB7C15304E52'
2012-04-03 15:09:25,178 [1] DEBUG
Accelrys.Security.PasswordReplacer.XPathReplacer [(null)] - Searching for
XPath '/RASConfig/Databases/SymyxDatabase/SchemaPassword[not(Encrypted)]'
2012-04-03 15:09:25,178 [1] DEBUG
```

```
Accelrys.Security.PasswordReplacer.XPathReplacer [(null)] - Match found at
SchemaPassword
2012-04-03 15:09:25,178 [1] INFO
Accelrys.Security.PasswordReplacer.RasReplacer [(null)] - Replaced password
with
'01000000D08C9DDF0115D1118C7A00C04FC297EB01000000F8CFE8795087DC4CB4A4D4BA6A0
AD00F0400000002000000
000003660000C0000000100000002A6D105ED1625D7EBA5284EE430327C90000000004800000
A00000001000000EE01B
EB06E44C033DFEF7F8F1DC9194D30000000143010CA48EABBD4D8CEE199122E4956E2588C60E
D43C3F6E3F2C9A59E89A7
BF4B63D7DF3343EA7C87CA14090728B5AB140000004735C5D7C32437968ACE24FF0689F6D262
3B9BBA'
```

Password Replacer Error Handling

If there is a problem with a configuration file, the error is logged as an `Error` in both the log file and the console output.

To recover from a file error, you must manually replace the configuration file with the backup file, if one has been generated, and then rerun the Password Replacer utility.

Modifying Encrypted Passwords

You can modify the encrypted password in one or more of the configuration files.

To modify encrypted passwords:

1. Open the configuration file in a text editor.
2. Navigate to the following element:

   ```
   <PlainTextPassword/>
   ```

3. Replace the entry with the following:

   ```
   <PlainTextPassword>clear_text_password</PlainTextPassword>
   ```

   The `clear_text_password` is the new password you want in clear text.
4. Save the configuration file.
5. Run the Password Replacer utility, for example:

   ```
   PasswordReplacer.exe -avspath
   ```

   In this example, `-avspath` refers to the default Vault Server install path.

   In the configuration file, the `<EncryptedPassword>` element has changed and the `<PlainTextPassword>` element is replaced with an empty element.
6. Check the log file for the change.

Reverting Modified Files

If you encounter errors running the Password Replacer utility you should revert the configuration files that it has generated.

Use the backup files you created before running the Password Replacer utility. Manually revert to the backups by overwriting the current corresponding `.config` files.

Chapter 5: Configuring Citrix Servers

This section provides information on configuring Vault Server to operate in a Citrix environment.

Vault Server Compatibility with Citrix

The BIOVIA Vault Administration Console and Workflow Designer are not supported in a Citrix environment.

Enabling the Citrix Clipboard to Support Chemical Structure Data

If you use BIOVIA Draw (or other chemical structure editors) installations on a Citrix server and need to cut-and-paste structures, you must update the Citrix server registry.

Otherwise, when a user copies a structure, the image of the structure is copied, but the data used to render it is not copied, and the image cannot be edited.

> **Note:** To copy chemical structures from a Workbook installation on a Citrix server to a locally installed version of a Microsoft Office application, you must have a locally installed version of BIOVIA Draw.

To add chemical structure clipboard formats to a Citrix server:

1. Log on to the Citrix server computer as an adminstrator and select **Start** > **Run**.
2. Type **regedit**, and then click **OK**.
3. In the Registry Editor, navigate to:

   ```
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\wfshell\
   Virtual Clipboard\Additional Formats
   ```

4. Add the **Additional Formats** key, and then add the following subkeys and values:

Subkey Name	Type	Value	Comment
ChemDraw Interchange Format	String	ChemDraw Interchange Format	Supports ChemDraw
Data Object	String	Data Object	
Embed Source	String	Embed Source	
Embedded Object	String	Embedded Object	
MDLCT	String	MDLCT	Supports BIOVIA Draw
MDLNative	String	MDLNative	
MDLSK	String	MDLSK	Supports BIOVIA Draw
Object Descriptor	String	ObjectDescriptor	
OLE Private Data	String	Ole private Data	

5. Navigate to:

   ```
   [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Terminal
   Server\Compatibility\Applications
   ```

6. Add the **AccelrysDraw** key, and then add the following subkeys and values:

Subkey Name	Type	Value	Comment
Flags	DWORD (32-bit) Value	dword:0000000f	Supports pasting chemical structures and reactions
OpenClipboardDelayInMilliSecs	DWORD (32-bit) Value	dword:00000050	Supports pasting chemical structures and reactions
OpenClipboardRetries	DWORD (32-bit) Value	dword:00000020	Supports pasting chemical structures and reactions

7. Save your changes and and close the Registry Editor.

Chapter 6: Administering the RAS Server

This section provides information on administration of your RAS Server.

RAS Auditing

RAS auditing is disabled because BIOVIA Workbook performs the auditing function. BIOVIA recommends that you do not enable RAS auditing to ensure consistent BIOVIA Workbook performance.

RAS Configuration Parameters

The RAS configuration parameters use the format `tag_name.parameter_name`. You can view the complete RAS configuration parameter set in the `<INSTALL_PATH>\Vault\Tomcat\conf\RASConfig.xcf` server configuration file.

The parameter `param_name` is found inside the `tag_name` in the configuration file. Each parameter has a unique path name, for example, the path names are `Databases.Username` and `SymyxDatabase.CreatePublicSynonyms`.

Generic RAS Parameters

Name	Value	Description	Default Value
SystemUser	String	The RAS user account used to perform various system tasks.	Superuser
LocalSaveQueuePaths	0 or 1	This is applicable only when multiple application servers share the same database. 0 indicates that the application servers share a single common queue folder specified by ObjectSaveQueuePath using a network share. 1 indicates that each application server maintains its own local save queue folder specified by ObjectSaveQueuePath. The server host name is written to the RASID column of the savequeue table.	1
DisplayGlobalIDs	0 or 1	Whether to display Global IDs in client applications. 1 enables the display. 0 disables the display.	1
EnforceSingleProjectPerLibrary	0 or 1	Whether experiments in a given library can belong to different projects. 0 allows experiments in the same library to belong to more than one project. 1 requires that all experiments in a library belong to a single project, namely, the project that the Synthesis experiment belongs to.	1

RAS Security Parameters

Name	Value	Description	Default Value
LoginSecurity	0 or 1	Whether user authentication is required. 1 enables user authentication. 0 disables user authentication.	0
ProjectSecurity	0 or 1	Whether user authorization is enabled. 1 enables user authorization. 0 disables user authorization. The default is set during installation of RAS.	0
SubElementSecurity	0 or 1	Whether security is set on sub-element collections. If set to 1, then ProjectSecurity must also be set to 1. The default is set during installation of RAS.	0
RequireWindowsDomain	0 or 1	Whether to use a Windows domain with a user name. When set to 1, the domain and user name syntax is domain\user_name, where domain is a domain name specified in ValidWindowsDomains and user_name is an account user name. 1 enables the use of a Windows domain. 0 disables the use of a Windows domain.	0
ValidWindowsDomains	String	The list of valid domains. The specification format is a pipe-separated list of domain names. For example, dom1|dom2|dom3. A domain name must be used if RequireWindowsDomain is set to 1.	
BrowserMode	0 or 1	Whether to allow access to the RAS methods by a Web browser. 1 allows browser access. 0 prevents browser access.	1
SessionExpiry	Positive integer	The number of minutes that a client application can be inactive. The inactive time is the difference between the current time and the last database access time. When a client reaches session expiration time, its session ID becomes invalid. The expiration test runs every five minutes.	120
UseWindowsLogin	0 or 1	Whether a .NET client should prompt for the user name and password to be used for the connection to RAS or should pass the Windows user name with a blank password. 0 employs the .NET client prompts for the user name. 1 employs the Windows user name, this is read only and the password is disabled. In this case, the user's RAS password must not have a value or the	0

Name	Value	Description	Default Value
		authentication fails.	
AllowPasswordChange	0 or 1	Specifies whether the Change Password button is available in the RAS client control panel. 0 (or empty) does not display the Change Password button. 1 shows the Change Password button in the interface.	

RAS Client Parameters

Name	Value	Description	Default Value
IPAllow	String	The set of clients that are allowed to access the server. The server is accessible to any client with an IP address that matches the IP specification. The specification is a Java regular expression, where \d{1,3} means that the field must contain between one and three digits. For example, an IP network address is specified by the string \d{1,3}.\d{1,3}.\d{1,3}.\d{1,3} Any network address with the prefix 192.168 is specified by the string 192.168.\d {1,3}.\d{1,3}	\d{1,3}\.\d {1,3}\.\d {1,3}\.\d{1,3}
IPDeny	String	The set of clients that are denied access to the server. The server is inaccessible to any client with an IP address that matches the IP specification. The specification is a Java regular expression. See the examples in the description for IPAllow.	
HostNameAllow	String	The list of host names that a client is allowed to use. The specification format is a pipe-separated list of names. For example, the string host-A|host-B|host-C is a value that would allow a client to use the hosts named host-A, host-B, or host-C.	HostnameAllow
HostNameDeny	String	The list of host names that a client is not allowed to use. The specification format is a pipe-separated list of names. For example, the string host-A|host-B|host-C is a value that would disallow a client from using the hosts named host-A, host-B, or host-C.	

RAS Logging Parameters

Name	Value	Description	Default Value
Level	ERROR	Generate error information only.	DEBUG
	INFO	Generate a small amount of trace information.	
	DEBUG	Generate full debugging information.	
SQLDebug	0 or 1	Enables the logging of SQL statements submitted to the database. 1 enables logging, in this case set Level to DEBUG. SQL statements are logged separately to `RASSQL.log` in addition to `RASCore.log`. (Optional) SQLDebug can be set to 1 and Level to INFO. In that case `RASCore.log` contains only a subset of the SQL commands logged in `RASSQL.log`.	1
MapDebug	0 or 1		0
CommDebug	0 or 1	The logging of HTTP requests and responses sent to the server. 1 enables logging, in this case set Level to DEBUG.	1
EventDebug	0 or 1	The logging of internal events. 1 enables logging, in this case set Level to DEBUG.	0
EventDebug.CommDebugMaxMessageLength	Integer	The maximum message length written in the log file. Relevant if CommDebug is set to 1 only.	512
EmailLevel	ERROR INFO DEBUG	The type of event that triggers an e-mail notification. Typically, set this to ERROR to avoid excessive notifications. See the Level parameter.	ERROR
EmailTo	String	The e-mail address to send notifications.	

Name	Value	Description	Default Value
PagerLevel	ERROR INFO DEBUG	The type of event that triggers a pager notification. Typically, set this to ERROR to avoid excessive notifications. See the Level parameter.	ERROR
PagerTo	String	The email address to send pager notifications.	

RAS SMTP Parameters

Name	Value	Description	Default Value
DisableEmail	0 or 1	Indicates whether the server sends e-mail notifications to a client. 1 disables the sending of e-mails. 0 enable it the sending of e-mails. The default is set during installation of Vault Server.	0
Host	String	Identifies the host name of the SMTP server that RAS uses to send e-mail notifications. Relevant only if DisableEmail is set to 0. The default is set during installation of Vault Server.	
Port	Port number	Identifies the port number of the SMTP server that RAS uses to send e-mail notifications. Relevant only if DisableEmail is set to 0.	25
From	String	Identifies the originator email address that RAS uses to send e-mail notifications. Relevant only if DisableEmail is set to 0. The default is set during installation of Vault Server.	

RAS Metadata Parameters

Name	Value	Description	Default Value
RequirePublishPrivilege	0 or 1	The users who can publish a dynamic schema. 0 allows any user to publish a dynamic schema. 1 allows only users with the Admin and Publish privileges to publish a dynamic schema.	1
LoadDynamicSchemasAtStartup	0 or 1		1
AutomaticActivationEnabled	0 or 1		1

RAS Database Parameters

Name	Value	Description	Default Value
ConnectionLog	String	The path to the logs folder.	C:\tmp\logs\ testOutput
MaxRowCount	Integer	The maximum number of rows returned from a table.	25000
EnforceSingleProjectPerLibrary	0 or 1	That an experiment and the synthesis must belong in the same project. 1 forces the Experiment.project field to be equal to the field Synthesis.project. 0 allows those two fields to have different names.	1
NonAuditedObjects	String	The type of objects on which auditing is disabled.	ReplicationJob | ReplicationJobResult | ReplicationResult | ReplicationObjectState | ReplicationErrorInfo | ReplicationLibIdTranslator

RAS Database Parameters

Name	Value	Description	Default Value
UserRole	String	The database role that is granted access to the database tables.	SYMYXDB_ USER_ROLE
SchemaUsername	String	The database user name that the server uses when performing DDL operations.	symyxdb
SchemaPassword	String	The password of the SchemaUsername account. The password is saved in encrypted form. To set it to clear text, see Clear text database passwords.	symyxdb
LOBTablespace	String	The tablespace where a dynamic publication creates LOBs (large objects).	SYMYXLOB
IndexTablespace	String	The tablespace where a dynamic publication creates indexes.	SYMYXIND
URL	String	The URL that RAS uses to establish a connection with the database. For example: jdbc:oracle:thin:@127.0.0.1:1521:orcl2 The default is set during installation of RAS.	

Name	Value	Description	Default Value
MinConnections	Integer	The minimum number of connections kept in the pool of RAS-Oracle connections.	5
MaxConnections	Integer	The maximum number of connections kept in the pool of RAS-Oracle connections.	40
ConnectionPoolTimeout	Integer	The maximum time in milliseconds that a thread waits to obtain a free database connection from the connection pool if all connections in the pool are in use.	500
ConnectionInit	String	The SQL command that is executed when a connection is added to the connection pool.	ALTER SESSION SET NLS_DATE_ FORMAT ='DD-MM-YYYY HH24:MI:SS'
RowPrefetch	Integer	The number of rows pre-fetched from the database during each query. This is the JDBC connection parameter DefaultRowPrefetch on all connections in the connection pool. For more information, refer to the Oracle JDBC documentation, *Oracle Row Prefetching*.	25
TestTable	String	The connection pool periodically tests all connections in the pool to determine if they are still valid by running the query: `SELECT COUNT(*) FROM TestTable`	SYMUSER
CreatePublicSynonyms	0 or 1	The generation of public synonyms when a RED file is dynamically published. Set it to 1 to generate synonyms. Set it to 0 to suppress synonyms. This value is not relevant when a RED file is published statically.	0
Username	String	The database account name that the client uses.	symyxdbuser
Password	String	The database password. This is encrypted by default. To set it to clear text, see Clear text database passwords.	symyxdbuser

Storage Parameters

Name	Value	Description	Default Value
ObjectCachePath	String	The absolute path to the cache folder, which contains copies of the recently retrieved RAS objects. The cached object description includes the base and the derived experiments.	`C:\VaultLogs\ras-server\cache`
ObjectAuditPath	String	The absolute path to the audit folder, which contains the version history of all RAS objects.	`C:\VaultLogs\ras-server\audit`
ObjectErrorPath	String	The absolute path to the error folder, which contains any RAS errors.	`C:\VaultLogs\ras-server\errors`
ProjectResults	String	The absolute path to the project folder, which contains any RAS projects.	`C:\VaultLogs\ras-server\project`
SaveQueuePath	String	The absolute path to the queue folder, which stores temporary objects queued for asynchronous processing.	`C:\VaultLogs\ras-server\queue`

Client Application Parameters

Name	Value	Description	Default Value
LibraryStudio.DisableUserCheck	Yes or No	Yes allows any Library Studio user to modify existing documents. No allows only the creator of a document to modify the document.	No
WebAppRegistry.DataBrowser.Port	Integer	The port that Data Browser uses to communicate with the client applications.	80
WebAppRegistry.DataBrowser.Host	String	The host where Data Browser runs. If the host is single sign-on, the host name must be set as to the Fully Qualified Domain Name (FQDN), for example, *`RAS3Desktop.symyx.com`*.	127.0.0.1
WebAppRegistry.DataBrowser.Path	String	The Data Browser context path, the path in Tomcat.	databrowser
WebAppRegistry.DataBrowser.Protocol	http or https	The protocol that Data Browser uses to communicate with the client.	http

Task Parameters

RAS tasks are configured using the following parameters:

- DisplayName
- Class

- Description
- SleepInterval - specifies the task run frequency
- Enabled

The following example shows the settings for the `PurgeExpiredList` task:

```
<Tasks>
  <Enabled>1</Enabled>
  <PurgeExpiredLists>
    <DisplayName>Purge Expired Lists</DisplayName>
    <Class>com.symyx.ras.tasks.PurgeExpiredLists</Class>
    <Description>Delete expired SymyxDbElementLists</Description>
    <SleepInterval>0:10:00:00</SleepInterval>
    <Enabled>1</Enabled>
    <Parameters>
      <ListLifetime>1:00:00:00</ListLifetime>
    </Parameters>
  </PurgeExpiredLists>
</Tasks>
```

The values for the parameters are as follows:

- 1 - enabled
- 0 - disabled

Name	Description	Default Value
PurgeExpiredLists	The deletion of expired SymyxDbElementLists.	1
BackgroundSave	The processing of single background saving requests.	1
RefreshSQLCache	The refreshing of cached SQL record sets.	1
MetadataCacheRefresh	The refreshing of metadata when a cluster of servers (multiple RAS instances) point to the same database. This task causes a RED file published by one RAS instance in a cluster to be synchronized across all other RAS instances in the cluster.	1
RecompileInvalidObjects	The periodic compilation of invalid database objects.	1
MetadataRefresh	The synchronization of version 3 metadata with version 4 metadata.	1
PurgeTimedOutDotNetSessions	The deletion of expired session IDs.	0
PurgeTimedOutJavaSessions	The deletion of expired session IDs.	0
PurgeCacheFiles	The deletion of cache files.	0
PurgeLogFiles	The deletion of log files.	0
ArchiveAuditFiles	The archiving of audit files.	0

Event Listener Parameters

BIOVIA recommends removing the `XMLCaching` and `DatabaseAudit` entries from the `EventListeners` parameter.

```
<EventListeners>
  <XMLCaching>
    com.symyx.ras.events.stdimpls.DatabaseXMLCache
  </XMLCaching>
  <DatabaseAudit>
    com.symyx.ras.events.stdimpls.DatabaseAudit
  </DatabaseAudit>
</EventListeners>
```

Name	Description	Default Value
XMLCaching	Caches XML files	`com.symyx.ras.events.stdimpls.DatabaseXMLCache`
DatabaseAudit	Audits for the Database	`com.symyx.ras.events.stdimpls.DatabaseAudit`
PolicySynchronization	Synchronizes policies	`com.symyx.ras.events.stdimpls.PolicySynchronization`
AuditEventListener	Audits Events	`com.symyx.ras.events.stdimpls.AuditEventListener`

RAS Utilities

RAS utilities are located in the `<INSTALL_PATH>\Vault\Utilities\RASTools\rastools\bin` folder. The RAS utilities are as follows:

- DeleteProject
- EncryptDBPasswords
- ExportMetadata
- GenerateSubPolicies
- Jdbc-connect
- LDAPTest
- MoveObject
- OriginLinker
- RasArchive
- RasBenchmark
- Rasgetobject
- Rasgetobjectversion
- Rasload
- Rasmigrate
- Rasupgrade

- Redc
- SetClassNames
- SetEncryptionKeys
- Verifydata

Run the RAS utilities in a Command window.

Use the following command to get help with the utilities:

```
encryptDBPasswords -help
```

DeleteProject

If you use the DeleteProject utility to delete projects, the project must not have any associated experiment data. The server checks for associated data before performing the delete. If data exists, the operation fails if the project is not empty. If the project has user and group associations, associations are also deleted.

Security

The user account must have the administrator privilege to delete the projects.

Usage

```
deleteProject <connect-string> [-help] <-project projectname>
[-noUpdate] [-file filename] [-timeout integer] [-reason string]
```

To delete the project "test project 1" and save the server generated report to the file temp.xml, use a command similar to the following:

```
deleteProject adminuser/pwd@localhost -project "test project 1" -file
temp.xml
```

Parameters

The DeleteProject utility uses the parameters in the table below.

Parameter	Description
<connect-string>	String that specifies how to connect to the RAS database. This has the format: username/password@hostname:port. For example: scott/tiger@ras:2001 **Note:** If the port is left unspecified, it defaults to 2001.
[-help]	Provides the list of available options and a brief description.
<-project projectname>	Specifies the name of the project to delete.
[-noUpdate]	If this option is specified then the server rolls back the transaction instead of committing it. You can then view the generated report to view the list objects that would be modified if the transaction was committed.
[-file filename]	When the utility completes it writes an XML report of the server generated results to the file `MoveResults.xml`. The file option allows you to specify a different file name.

Parameter	Description
[-timeout integer]	Specifies the maximum number of seconds the client waits for the server response. If the server doesn't complete the request in the allotted time then it disconnects with a timeout error and doesn't generate the status report. However, the server side operation continues until it completes.
[-reason string]	If database auditing is enabled, an audit entry is generated for the project delete, and the user can specify the reason for the delete; the reason is saved in the audit entry.

EncryptDBPasswords

This utility encrypts the passwords for the symyxdb and symyxdbuser database accounts. The clear text passwords are removed from the configuration file and encrypted passwords are used.

Security

The account does not need a PDS privilege.

Usage

```
encryptDBPasswords [-help] <connect_string>
```

For example:

```
encryptDBPasswords rasuser/rasuser_password@RAS-server:2001
```

The EncryptDBPasswords utility returns output similar to the following:

```
Successfully encrypted password for Databases.SymyxDatabase.Password
Successfully encrypted password for Databases.SymyxDatabase.SchemaPassword
Successfully encrypted password for Databases.SymyxDatabase.DirectPassword
Passwords encrypted
```

Parameters

Parameter	Description
<connect_string>	A user name, a user password, a server name, a server port, and a database SID. If the port is not specified, it defaults to 2001.

ExportMetadata

ExportMetadata creates a RED file for each published RED file on a server. It extracts the object data from the MetadataObjectGroups table and saves the object XML definitions in a file.

Security

The account does not need a PDS privilege.

Usage

```
exportMetadata [-help] [-version] [-verbose] [-dir directory_name]
[-refdir directory_name] <connect_string>
```

For example:

```
exportMetadata -dir exportReds symyxdb/symyxdb@orcl_server_17:1521:orcl
```

Parameters

Parameter	Description
<connect_string>	A user name, a user password, a server name, a server port, and a database SID. If the port is not specified, it defaults to 2001.
[-help]	Displays the utility version and usage.
[-version]	Displays the version information.
[-verbose]	Displays debug information.
[-dir directory_name]	The location where the new RED files are stored. If not specified, the folder is the current folder. If the folder name contains a space character, the folder must be enclosed within double quotes.
[-refdir directory_name]	A folder containing RED files. The utility scans the RED files in the specified folder and all subfolders to determine which file names correspond to the MetadataObjectGroup names.

GenerateSubPolicies

The GenerateSubPolicies utility generates two SQL scripts that enable and disable PDS on sub-element collections.

Security

The account does not need a PDS privilege.

Usage

```
generateSubPolicies [-help] <-d directory> <connect_string>
```

For example:

```
generateSubPolicies -d pDir symyxdb/symyxdb@orcl_server_17:1521:orcl
```

Parameters

Parameter	Description
<-d directory>	The location where the new scripts are stored. If the folder name contains the space character, you must enclose the name in double quotes.
<connect_string>	A user name, a user password, a server name, a server port, and a database SID.
[-help]	Displays the utility version and usage.

Jdbc-connect

The `Jdbc-connect` utility validates the URL connection with a server using JDBC. You can use the `Jdbc-connect` utility to validate the connection that a RAS instance establishes using the `Databases.SymyxDatabase.URL` value.

Security

The account does not need a PDS privilege.

Usage

```
jdbc-connect [-help] [-count integer] <jdbc_URL> <user_name>
<user_password>
```

For example:

```
jdbc-connect jdbc:oracle:thin:@127.0.0.1:1521:orcl symyxdb
symyxdbpassword
```

Parameters

Parameter	Description
<jdbc_URL>	Specifies the URL to test.
<user_name>	Specifies the user name.
<user_password>	Specifies the user password.
[-help]	Displays the utility version and usage.
[-count integer]	Specifies the number of connection attempts.

LDAPTest

The LDAPTest utility is used to test the connection to an LDAP server. You can also use the LDAP Test to authenticate a user.

Security

The account does not need a PDS privilege.

Usage

```
LDAPTest [-help] [-help-deprecated] [-simple] [-version] <ldap_login>
```

To test a connection to an LDAP server:

```
LDAPTest ldap://srv-ldap:389
```

To test a connection and authenticate a user:

```
LDAPTest username/password@ldap://srv-ldap:389
```

If the first forward slash / and password are omitted, the utility prompts you to enter a password. The password is masked by asterisk characters (*).

Parameters

Parameter	Description
<ldap_login>	Specifies the LDAP login details.
[-help]	Displays the utility usage.
[-help-deprecated]	Displays the deprecated options.
[-simple]	Uses a simple authentication mechanism instead of the DIGEST-MD5 default.
[-version]	Displays the utility version.

MoveObject

The MoveObject utility moves selected designs and experiments between projects. If a Synthesis is moved, then any associated experiments are moved if their current project matches the synthesis. The utility skips any experiments that have the same library ID as the Synthesis but for have a different project. When moving designs, the utility does not move any synthesis or experiments associated with the design.

Move all the contents of a Project into another Project, optionally restricted to a particular object type.

Security

The account must have the rights to perform the operation on the given projects. Reassignment is disabled for objects that:

- Are In-Progress
- Belong to a Project to which the user does not have Update or greater Privilege
- The utility only allows reassignment to projects to which the user has Insert or greater Privilege.

The reassignment does not proceed if reassigning the object to the destination project results in an escalation of the user's privilege to that object. For example, you cannot reassign an object from a Project and from a user that has update privileges to a project where the user also has delete privileges.

Usage

```
moveObject <connect-string> [-help] [-help-deprecated] [-type string]
[-id string] [-all] <-currentProject string> <-newProject string>
[-allowReplicated] [-noUpdate] [-file string] [-timeout integer]
[-reason string] [-version]
```

For example:

To move Experiment 102 from the test project 1 project to test project 2, and save the server generated report to the file `temp.xml`, use the following:

```
moveObject adminuser/pwd@localhost -type Experiment -id 102
-currentProject "test project 1" -newProject "test project 2" -file
temp.xml
```

To move all objects from the project "test 1 project" to "test project 2":

```
moveObject adminuser/pwd@localhost -all -currentProject "test project 1" -
newProject "test project 2" -file temp.xml
```

To move all objects of Experiment type from the project "test project 1" to "test project ":

```
moveObject adminuser/pwd@localhost -all -type Experiment
-currentProject "test project 1" -newProject "test project 2" -file
temp.xml
```

Parameters

Parameter	Description
<connect-string>	String that specifies how to connect to the RAS database. This has the format: username/password@hostname:port. For example: scott/tiger@ras:2001 **Note:** If the port is left unspecified, it defaults to 2001.
<-currentProject projectname>	The name of the project the object(s) should be moved from.
<-newProject string>	The name of the project the object(s) should be moved to.
[-help]	Provides the list of available options and a brief description.
[-help-deprecated]	Display the deprecated options.
[-type string]	The type of the object to be moved. For example, Experiment or Synthesis.
[-id string]	The Object ID to be moved to the newProject.
[-all]	Move all of the objects in the currentProject to the newProject.
[-allowReplicated]	Allow objects to be moved even if the newProject already contains a matching object.
[-noUpdate]	If this option is specified then the server rolls back the transaction instead of committing it. You can then view the generated report to view the list of objects that would be modified if the transaction had been committed.
[-file string]	When the utility completes it writes an XML report of the server generated results to the file `MoveResults.xml`. The file option allows you to specify a different file name.
[-timeout integer]	Specifies the maximum number of seconds the client waits for the server response. If the server does not complete the request in the allotted time, then the server disconnects with a timeout error and does not generate the status report. However, the server-side operation continues until it completes.
[-reason string]	If database auditing is enabled an audit entry is generated for the project rename. The user can specify the reason for moving the object. The reason is saved in the audit entry.
[-version]	Display the utility version.

OriginLinker

The OriginLinker utility populates null OriginID, OriginDBID, OriginLibID, OriginLibDBID, and SourceLibDBID values in a target database. The utility processes all tables in the SymyxDB schema that contain the field ORIGINID and the table SYNTHESIS, which contains the field ORIGINLIBID.

For non-replicated objects, the values OriginID, OriginLibID and OriginDBID are obtained from the fields ID, LIBID, and the entry DATABASEID in the RASGLOBAL table.

For replicated objects, the locally stored value Link ID is used to locate the corresponding object in the source database, and the needed values are extracted from ID, LIBID, and DATABASEID in the source database.

The OriginLinker utility requires a RAS 4.1 target database. Replication sources do not need to be upgraded to RAS 4.1, but their database ID names must comply with the RAS 4.1 database ID syntax.

If objects have been replicated from A to B and further from B to C, in addition to A-B and B-C runs, OriginLinker must also be run with A as the source database and C as the target database to resolve the attributes on C.

The utility is not metadata-driven and does not look up information in metadata tables before processing data.

Security

The account does not need a PDS privilege.

Usage

```
originLinker [-help] [-verbose] [-spool file_name] [-rasuser user_name]
<target_connect> [source_connect] [-overwrite]
```

For example:

```
originLinker -spool cmd.sql symyxdb/symyxdb@orcl_server_17:1521:orcl
```

Parameters

Parameter	Description
<target_connect>	A user name, a user password, a server name, a server port, and a database name, all relevant to the target database. **Note:** If the port is left unspecified, it defaults to 2001.
[-help]	Display the utility version and usage.
[-verbose]	Display debug information.
[-spool file_name]	Spool SQL commands to the specified file. The SQL commands are not executed. This argument allows users to review the commands. If the file name contains the space character, the file name must be enclosed within double quotes.
[-rasuser user_name]	The RAS user name used to audit changes in the target database. If unspecified, it defaults to symyxdb.
[source_connect]	A user name, a user password, a server name, a server port, and a database name, all relevant to a source database. There can be zero or more instances of this argument.
[-overwrite]	Sets OriginID, OriginDBID, OriginLibID, OriginLibDBID, and SourceLibDBID values even if they are not null.

RasArchive

The RasArchive utility loads or archives existing audit files contained in current working folder to the RAS database.

Usage

```
rasarchive [-help] [-help-deprecated] [-all] [-count integer] [-faildir
directory] [-successdir directory] [-reason  string]
[-timeout integer] [-verbose] [-version] <connect-string> <file> [<file>
...]
```

The following example shows how to:

- Load all the audit files located the current folder to the database
- Specify an audit reason of loading archive files for each audit row
- Move any files that fail to load to the fail folder
- Connect to the RAS database using the connection string admin/pass@127.0.0.1

For example:

```
rasarchive -all -faildir fail -reason "loading archive files"
admin/pass@127.0.0.1
Archived EpochResource_21819_1.xml.gz (Id=21819)
Archived EpochResource_21820_1.xml.gz (Id=21820)
```

Parameters

Parameter	Description
<connect-string>	String that specifies how to connect to th RAS database. This has the format: username/password@hostname:port. For example, scott/tiger@ras:2001 **Note:** If the port is left unspecified, it defaults to 2001.
<file>	A specific RAS audit file to load into the RAS database.
[-all]	Archive all of the audit files in the current folder to the database.
[-count integer]	Number of objects to archive.
[-faildir directory]	Move any unsuccessfully loaded files to the specified folder.
[-successdir directory]	Move any successfully loaded files to the specified folder.
[-help]	Display detailed help.
[-help-deprecated]	Display the deprecated options.
[-reason string]	A string containing the reason for the audit entry, which is added to the database.
[-timeout integer]	An integer timeout in seconds to wait for RAS to respond; set it to 0 for no timeout limit.

Parameter	Description
[-verbose]	Display verbose output.
[-version]	Display the software version information.

RasBenchmark

The RasBenchmark utility retrieves RAS performance information.

Security

The specified account must have the privilege to update and save the settings.

Usage

```
PeformanceBenchmark [-help] [-d integer] [-n integer] [-o string]
[-t integer] [timeout integer] [-v] <connect-string>
```

Parameters

Parameter	Description
<connect-string>	String that specifies how to connect to the RAS database. This has the format: username/password@hostname:port. For example: scott/tiger@ras:2001 **Note:** If the port is left unspecified, it defaults to 2001.
[-help]	Displays the utility usage.
[-d integer]	The wait time between saves. The default time is set to 0.
[-n integer]	The number of saves per thread. The default number is set to 10.
[-o string]	The object type to be saved. The available types are: Substance, GPCExperiment, Image2, LibraryDesign, or BigSynthesis. The default is Substance.
[-t integer]	The number of RAS client threads. The default is set to 1.
[timeout integer]	An integer timeout in seconds to wait for RAS to respond, set it to 0 for no timeout limit.
[-v]	Displays verbose information.

Rasgetobject

The Rasgetobject utility retrieves the XML of a specified RAS object and writes it to a local file or to an entry in a local zip file. You can specify multiple ID values to retrieve multiple objects. You can provide an input file containing one or more object references of the form WeighExperiment#12345, separated by whitespace. The output is named after the object retrieved and located in the current folder.

Each output file is named `<object-type>_<object-ID>.xml`, for example: `WeighExperiment_12345.xml`.

Security

The specified account must have the privilege to read the object.

Usage

```
rasgetobject [-help] [-dir filename] [-file filename]
[-help-deprecated] [-version] [-zip filename] <connect-string>
<object-type> <object-ID> [<object-ID> ...]
```

For example:

```
rasgetobject symyxdb/symyxdb@orcl_server_17:1521:orcl XYDataSet 1234
```

Parameter

Parameter	Description
<connect-string>	String that specifies how to connect to the RAS database. This has the format: username/password@hostname:port. For example: scott/tiger@ras:2001 **Note:** If the port is left unspecified, it defaults to 2001.
<object-ID>	The ID of the object to retrieve.
<object-type>	The type of the data to retrieve.
[-help]	Display the utility version and usage.
[-dir filename]	Directory into which to write the XML files.
[-file filename]	Name of a file containing object references.
[-help-deprecated]	Displays the deprecated options.
[-version]	Displays the version information.
[-zip filename]	Name of an output zip file to receive all the output XML files (this overrides -dir).

Rasgetobjectversion

The Rasgetobjectversion utility extracts a specific version of an object.

Security

The specified account must have the privilege to read the object.

Usage

```
rasgetobjectversion [-help] <connect-string> <object-type> <object-ID>
<object-version>
```

The following example shows how to get the object with:

- A type of SymyxDbUser
- An ID of 42
- A version of 5

For example:

```
rasgetobjectversion admin/pass@127.0.0.1 SymyxDbUser 42 5
```

Parameters

Parameter	Description
<connect-string>	String that specifies how to connect to the RAS database. This has the format: username/password@hostname:port. For example: scott/tiger@ras:2001 **Note:** If the port is left unspecified, it defaults to 2001.
<object-type>	Type of object to get.
<object-ID>	ID of the object to get.
<object-version>	Version of the object to get.
[-help]	Displays the detailed help.

Rasload

The Rasload utility saves and publishes RED files. You can also load experiments and object XML data.

Security

The specified account must have the save or update privilege for the project. If the `RequirePublishPrivilege` RAS parameter is enabled, the specified account must have the publish privilege when the -publish option is used.

Usage

```
rasload [-all] [-asynch] [-count integer] [-delete] [-faildir directory] [-
help] [-help-deprecated] [-publish] [-replication] [-savetype integer] [-
setaudit] [-verbose] [-timeout integer] [-version] <user_and_host_
specification> <red_file_specification>
```

For example:

```
rasload -publish my_user_name\my_password@server1 *.red
```

Parameters

Parameter	Description
<red_file_specification>	The collection of RED files to publish. The character * matches any string. For example, to publish all RED files in the current folder with a name that contains the string HPG, use the specification `*HPG*.red`. To compile all RED files in the current folder, use `*.red`. If the specification contains the space character, it must be enclosed within double quotes.
<user_and_host_specification>	This is a required argument with the format user_name\user_password@host_name.

Parameter	Description
[-all]	Loads all of the files in the current folder.
[-asynch]	Performs an asynchronous save.
[-count integer]	Specifies the number of objects to save.
[-delete]	Deletes the files after they have been successfully loaded.
[-faildir directory]	Moves the files that do not load to the specified folder.
[-help]	Display the utility version and usage.
[-help-deprecated]	Display the deprecated options.
[-publish]	Publishes the specified files to the server. When this argument is used, the RED file is saved and its status is active. Otherwise, the RED file is saved and its status is inactive.
[-replication]	Sets the replication flag.
[-savetype integer]	The type of save operation: 0=normal, 1=in-progress, 2=finalize, 3=archive
[-setaudit]	Sets the setAuditFields flag.
[-timeout integer]	Sets the RAS timeout in seconds (0 for no timeout).
[-verbose]	Displays verbose output.
[-version]	Displays version information.

Rasmigrate

The Rasmigrate utility performs database steps following the upgrade of a target database with Rasupgrade. The steps are assigned names, and you can explicitly include or exclude the steps from execution. The shorter steps are always included.

You can use the procedure report_progress to monitor the progress of the execution of the utility. In a second Oracle SQL*Plus session, you run the following SQL commands:

```
SQL> SET SERVEROUTPUT ON SIZE 1000000;
SQL> CALL SYMYX.RASUTIL.REPORT_PROGRESS();
```

You can keep issuing the `CALL` to display cumulative information about the steps that Rasmigrate is `running`.

Usage

```
rasmigrate [-help] [-verbose] <connect_string> [-exclude step_name] [-
include step_name] [-steps]
```

For example:

```
rasmigrate symyxdb/symyxdb@orcl_server_17:1521:orcl
```

To display the current list of steps to include or exclude:

```
rasmigrate -steps symyxdb/symyxdb@orcl_server_17:1521:orcl
```

This returns:

```
LinkIDPrefix
NullLinkIDs
DuplicateLinkIDs
MalformedLinkIDs
NullStatus
NullFlags
NullProjects
NullAudit
MissingLibIndex GlobalIDs
```

To exclude the first step:

```
rasmigrate -exclude LinkIDPrefix symyxdb/symyxdb@orcl_server_17:1521:orcl
Target connection is jdbc:oracle:thin:@laptop-bit:1521:ray92, DatabaseID =
RY92
Executing unnamed step: call rasutil.require_symyxdb_user()
Executing unnamed step: call rasmigrate.require_valid_dbid()
Executing unnamed step: call rasutil.disable_triggers()
Executing step NullLinkIDs: call rasmigrate.repair_null_linkids()
Searching for null LINKID values...
Executing step DuplicateLinkIDs: call rasmigrate.repair_duplicate_linkids()
Searching for duplicate LINKID values... GlobalLibIDs
  ...
```

To only include the first step:

```
rasmigrate -include LinkIDPrefix symyxdb/symyxdb@orcl_server_17:1521:orcl
```

Parameters

Parameter	Description
<connect_string>	A user name, a user password, a server name, a server port, and a database SID. If the port is left unspecified, it defaults to 2001.
[-help]	Display the utility version and usage.
[-exclude step_name]	To exclude the step with the specified name. Use multiple -exclude specifications to exclude more than one step. All steps not explicitly excluded are included.
[-include step_name]	To include the step with the specified name. This option makes the utility disregard any -exclude specification. Use multiple -include specifications to exclude more than one step. All steps not explicitly included are excluded. The special specification -include none excludes all steps.
[-steps]	The display of the list of steps that can be explicitly excluded or included.
[-verbose]	Display debug information.

Rasupgrade

The Rasupgrade utility generates two scripts that are used to update a schema to a new version.

There are two ways of using the Rasupgrade utility, one uses a connect string, and the other uses the -reference and -target options.

Security

The account does not need a PDS privilege.

Usage

```
rasupgrade [-help] [-verbose] [-o file_name] <connect_string>
```

```
rasupgrade [-help] [-verbose] [-o file_name] -reference <ref_connect>
-target <tgt_connect>
```

For example:

```
rasupgrade -o upgradeScript.sql symyxdb/symyxdb@orcl_server_17:1521:orcl
```

```
rasupgrade -o upgradeRastarget.sql -reference
usrName1\usrPass1@srvName1:2001:rasRef -target
usrName2\usrPass2@srvName2:2002:rasTgt
```

Parameters

Parameter	Description
<connect_string>	A user name, a user password, a server name, a server port, and a database SID.
-reference <ref_connect>	A user name, a user password, a server name, and a server port for the reference database.
-target <tgt_connect>	A user name, a user password, a server name, and a server port for the target database.
[-o <file_name>]	The prefix for the SQL scripts. If not specified, the scripts are named `rasupgrade.sql` and `rasupgradeConstraints.sql`. If the file name contains a space character, the file name must be enclosed within double quotes.
[-verbose]	Display debug information.
[-help]	Display the utility version and usage.

Redc

The Red compiler generates proxies for the specified RED files and the corresponding server schema, SQL, and metadata components to publish the file. The Red compiler allows creating a Java proxy that is a JAR file of classes for the objects in the Experiment hierarchy as specified in a RED file. By default, redc writes its output to the following folders:

- Metadata

  Stores XML files
- SQL

  Stores two SQL scripts
- Schema

  Stores a static schema with two subfolders:

- classes
- src/Symyx subfolders

Security

The account does not need a PDS privilege.

Usage

```
redc [-help] [-version] [-verbose] [-d <directory>]
[-classpath <class_path>] [-proxies] [-proxies-only]
[-proxies-jar <jar_name>] [-public-synonyms] [-redpath <path_list>]
[-schema-name <name>] [-sql-alter <connect_string>] [-sql-drop]
<red_file_spec>
```

Examples

To generate a JAR file from a RED file:

```
redc -proxies redFile.red -proxies-jar myJARFile
```

To generate a JAR file from a RED file that is dependent on another RED file:

```
redc -redpath ..\export -proxies RamanMicroscopy2.red -proxies-jar
RamanMicroscopy2.jar -schema-name RamanMicroscopy2
```

The `RamanMicroscopy2.red` depends on `SampleMapping.red`.

To use custom directories and schema names:

Write all output to the folder outDir, name the generated schema file replicat.schema, and parse every RED file in the current folder that has a file with the prefix replic:

```
redc -d outDir -schema-name replicat replic*.red
```

To generate several files and the folders Metadata:

This will generate Schema\classes, Schema\src\Symyx, and the SQL in the current folder:

```
redc myREDFile.red
```

To generate an extended RED file:

This will generate an extended file whose base RED file is located in the folder baseReds that is a sibling of the current folder:

```
redc extendedRED.red -redpath ..\baseReds
```

Generate the SQL statements to publish the `alterRedFileName.sql` file:

```
redc -sql-alter symyxdb/symyxdb@orcl_server_17:1521:orcl myREDFile.red
```

Parameters

Parameter	Description
<red_file_spec>	Specifies the collection of RED files to compile. The * character matches any string. For example, to compile all RED files in the current folder with a name that contains the string HPG, use the specification `*HPG*.red`. To compile all RED files in the current folder, use `*.red`.
[-d <directory>]	Specifies the name of the folder where all output is written. The default

Parameter	Description
	folder is the current folder. If the output folder exists, any existing data is overwritten. If the folder name contains a space character, the name must be enclosed within double quotes.
[-classpath <class_path>]	Specifies the class path used to compile Java classes. If the path name contains a space character, the name must be enclosed within double quotes.
[-proxies]	Generates client proxies for the specified RED files.
[-proxies-only]	Generates the client proxies for this object.
[-proxies-jar <jar_name>]	Specifies the name of the JAR file to be created.
[-public-synonyms]	Generates SQL statements that create public synonyms for tables, views, and sequences in the symyxdb schema. The name of synonym created for the view experiment_obj is also experiment_obj. A public synonym allows an Oracle developer to refer to an object using its name without the need to prefix it with symyxdb.
[-redpath <path_list>]	Specifies a semi-colon separated string of folder paths. This list is used to search for super classes of objects referenced by the BaseObject property.
[-schema-name <name>]	Specifies the name of the generated schema file. The name should not contain the suffix.schema: it is automatically appended to the name.
[-sql-alter <connect_string>]	Generates a SQL script to alter the specified database. The SQL statements are written to the `alterRedFileName.sql` file.
[-sql-drop]	Generates a SQL script to drop the specified database.
[-verbose]	Displays debug information.
[-version]	Displays the utility version.
[-help]	Displays the utility version and usage.

SetClassNames

The SetClassNames utility sets the attribute ClassName in every experiment object in a database. SetClassNames looks up information in metadata tables before processing data.

Usage

```
setClassNames [-help] [-verbose] [-overwrite] [-noUpdates]
<connect_string>
```

For example:

```
setClassNames -overwrite symyxdb/symyxdb@orcl_server_17:1521:orcl
```

Parameters

Parameter	Description
<connect_string>	Specifies a user name, a user password, a server name, a server port, and a database SID.
[-help]	Displays the utility version and usage.
[-verbose]	Displays debug information.
[-overwrite]	Overwrite the existing class names.
[-noUpdates]	Displays the objects to be written. No actual writing of class names is carried out.

SetEncryptionKeys

The SetEncryptionKeys utility resets the public and private key pairs that RAS and the client APIs use to encrypt the communication of user credentials between the clients and the server. The public key is stored in the database table rasglobal, and the corresponding private key is stored in the database table rasprivate.

Usage

```
setEncryptionKeys [-help] [-help-deprecated] [-version]
<connect_string>
```

For example:

```
setEncryptionKeys symyxdb/symyxdb_password@ora:1521:ras
```

Parameters

Parameter	Description
<connect_string>	Specifies a user name, a user password, a server name, a server port, and a database SID.
[-help]	Displays the utility version and usage.
[-help-deprecated]	Displays the deprecated options.
[-version]	Displays the utility version.

Verifydata

The Verifydata utility verifies that the data in two databases are consistent with each other. Use to verify that data in a source database has been correctly replicated to a target database.

Security

The specified account must have the read privilege for the projects whose objects to verify.

Usage

```
verifydata [-help] [-verbose] [-output file_name] <job_ID> <run_ID> <source_
connect> <target_connect>
```

For example:

```
verifydata -output C:\diffs.txt 2 34 s/sp@srv1:2001 t/tp@srv2:2001
```

Parameters

Parameter	Description
<job_ID>	Specifies the ID of the Data Replicator job that replicated the data.
<run_ID>	Specifies the ID of the Data Replicator run that replicated the data.
<source_connect>	Specifies a user name, a user password, a server name, and a server port for the database where the data has been replicated.
<target_connect>	Specifies a user name, a user password, a server name, and a server port for the database where the data has been replicated.
[-output file_name]	Specifies the file that contains the differences between the source and target data. If the file name contains a space character, the name must be enclosed within double quotes.
[-help]	Displays the utility version and usage.
[-verbose]	Displays debug information.

Changing the Case of Encrypted Database Passwords

The following example shows how encrypted database passwords are stored in the `RASConfig.xcf` file located in the `<INSTALL_PATH>\Vault\Tomcat\conf` folder.

```
<Databases>
...
  <SymyxDatabase>
...
  <SchemaUsername>symyxdb</SchemaUsername>
  <SchemaPassword>
    <Encrypted>01000000D08C9DDF</Encrypted>
  </SchemaPassword>
...
  <Username>symyxdbuser</Username>
  <Password>
    <Encrypted>0245600D08C9AABCAD445F</Encrypted>
  </Password>
 ...
  </SymyxDatabase>
</Databases>
```

If you need to change the case of the passwords for compatibility with BIOVIA Vault Server, the process should look like this:

Set the passwords to the appropriate case in Oracle:

```
SQL> Alter user symyxdb identified by SYMYXDBPASSWORD;
SQL> Alter user symyxdbuser identified by SYMYXDBUSERPASSWORD;
```

Edit the `RASConfig.xcf` file and set the passwords in plain text with a syntax similar to the following:

```
<Password>symyxdbuserpassword</Password>
<SchemaPassword>symyxdbpassword</SchemaPassword>
<DirectPassword>symyxdbpassword</DirectPassword>
```

Restart RAS services and initialize RAS by browsing to `https://RAS_servername:1521` substitute your server name for RAS_servername.

Re-encrypt the passwords with the EncryptDBPasswords utility by using `encryptDBPasswords RASuser/RASUserPassword@localhost:2001` or by saving any configuration change from the RASAdmin configuration page.

The `EncryptDBPasswords` utility should return the output like the following:

```
Successfully encrypted password for Databases.SymyxDatabase.Password
Successfully encrypted password for Databases.SymyxDatabase.SchemaPassword
Successfully encrypted password for Databases.SymyxDatabase.DirectPassword
Passwords encrypted
```

Storing Clear Text Database Passwords

If you need to store clear text passwords in the RAS configuration file, you must edit the configuration file and replace the encrypted password sections with clear text. For example:

```
<Databases>
...
   <SymyxDatabase>
...
     <SchemaUsername>symyxdb</SchemaUsername>
     <SchemaPassword>symyxdb_password</SchemaPassword>
...
     <Username>symyxdbuser</Username>
     <Password>symyxdbuser_password</Password>
...
  </SymyxDatabase>
</Databases>
```

Chapter 7: Monitoring Message Handler Activities

BIOVIA Vault Server provides features you can use to track and manage the indexing state of your Vault deployment. You can determine the overall indexing health of your deployment, as well as the indexing status of a single item. You can also easily re-queue items for indexing based on their status.

The Vault Service and the Vault Message Processing Service both record processing results in the `MessageHandlerStatus` table. This table provides visibility into the message handler activities in a central, easy-to-manage way.

The RequeueVaultObjects utility simplifies common tasks such as re-indexing items that failed to index; for details, see Using RequeueVaultObjects Utility on page 49.

Message Handler Success and Failure

Message handlers report status for each item that is processed. In general, items are considered successfully processed unless an error is encountered. In this case, handlers report a failure, and the item status is set to Failed in the MessageHandlerStatus table. In the absence of an error, the item status is set to Processed.

The Processed state does not mean that data exists in a particular index or that the item has been enrolled in workflow (WorkflowActivator message handler). It is possible that the item has no data for, or is not subject to, that particular type of message handler. If so, the item was processed correctly even though no action was taken. For more information, see the table in Message Handler States.

Message Handlers

Several message handlers process items as they are checked in to Vault. The following table lists the message handlers and their functions.

Message Handler	Description
`FullTextIndexer`	Adds any text provided by the item to the full-text search.
`IndexerObjectHandler`	Indexes form data for Advanced Search.
`PropertyExtractionMessageHandler`	Indexes Vault Objects so that data displays in the grid.
`PropertyIndexMessageHandler`	Indexes VaultObject core properties to support Standard Search.
`stereogroupnumberreducermessagehandler`	Indexes and reduces the number of stereo groups in a search.
`VaultObjectIndexMessageHandler`	Indexes any Property Set Definition (PSD) data including structures and reactions; used by Vault for internal querying.

Message Handler	Description
`RecipesSectionLockedWorkflowStagesHandler`	Locks and unlocks the **Recipes** section content as the user goes through workflow stages listed in the `WorkflowStagesForLocking` property.
`SynchronizeTaskPlanLockStateHandler`	Locks and unlocks the **Task Plan** section content as the user goes through workflow stages listed in the `WorkflowStagesForLocking` property.
`WorkflowActivator`	Enrolls items in workflows based on the active workflow association; used by Vault for internal querying.
`WorkflowTransitionMessageHandler`	Indexes workflow stage value after transitions.

Custom Message Handlers

Systems might have additional message handlers that were developed by BIOVIA Services or deployed by other applications. Custom message handlers created prior to Accelrys Vault Server 6.9 SP1 might not get updated. For assistance, contact Dassault Systèmes Customer Support.

Message Handler States

The message handling status can be one of six states:

- Queued
- Processed
- Failed
- Requeued
- Ignore
- Unknown

The sixth state, Unknown, is implied for items that do not have records in the MessageHandlerStatus table. The following table describes the message handling states.

State	Meaning	Description
Queued	A change occurred that was not processed.	The system sets the state to Queued whenever the item is saved or its properties are updated.
Processed	The item was processed by the message handler.	The system sets the state to Processed when the message handler reports that it processed the item successfully.

State	Meaning	Description
Failed	The Message Handler could not process the item.	The system sets the state to Failed when the message handler reports an error processing the item. This state means that the system was not able to process the item and does not try again. The administrator should review the `VaultMessageProcessor.log` file to find the error and either correct it or contact Dassault Systèmes Customer Support for assistance.
Requeued	The item has been submitted to the handler by the RequeueVaultObjects utility.	Items are set to this state when they are re-queued using the RequeueVaultObjects utility. This utility is used to submit items for re-processing. For more information, see RequeueVaultObjects Utility.
Ignore	The item is not subject to status tracking.	Customers can manually set items to Ignore using SQL statements. Use the Ignore state to prevent an item from being re-queued when you know that the item does not process successfully.
Unknown	The item has not been processed by the index status tracking feature.	Items without records in the MessageHandlerStatus are in the Unknown state. **Note:** All existing data is in the Unknown state at the time the indexing update is installed. Items move to one of the other states as they are saved, updated, or re-processed using the re-queuing utility.

Successful processing does not mean that the item has data in the index maintained by the handler. The following examples illustrate items in the processed state without data in the index:

Example 1: No data of that type, an empty document is checked in to Vault and processing completes. The document is in the processed state. Attempts to find the document using a full text search fail because it has no text.

Example 2: The item does not implement that type of handling, a workflow definition is added to Vault and processing completes. The workflow definition is in the processed state. The `FullTextIndexer` reports success handling the workflow definition because it understands that workflow definitions do not implement full-text indexing.

Example 3: A subsystem fails but does not report the failure, if Oracle Full Text Search is not configured on the database, the `FullTextIndexer` reports processed, but text searching does not work.

MessageHandlerStatus Table Definitions

The following table describes the format of the status tracking table.

Column Name	Data Type	Description
VaultID	VarChar (255)	The VaultId of the item being tracked.

Column Name	Data Type	Description
HandlerName	VarChar (255)	The name of the Message Handler being tracked.
Action	VarChar (255)	INTERNAL USE ONLY: An Internal Vault action code that is used by the message handlers to determine what processing is needed.
Status	VarChar (255)	The Message Handling Status of the item being tracked. Valid values are Queued, Processed, Failed, Ignore, and Requeued. For more information, see Message Handler states.
LastModifiedTime	TimeStamp	The last time the entry in the table was modified.
LastProcessedVersion	Integer	The version number of the last version that was successfully processed. This column is empty if the item has never been successfully processed.
LastProcessedTime	TimeStamp	The time, in UTC, that the item was last processed successfully. This column is empty if the item has never been successfully processed. **Note:** This time could be significantly later than the time that the version was saved and should only be used to indicate the time at which it was processed.

Using RequeueVaultObjects Utility

RequeueVaultObjects Overview

The `RequeueVaultObjects` utility is used to submit items for processing based on their message handling status.

Use `RequeueVaultObjects` to:

- Requeue items that failed processing.
- Requeue items for a specific message handler to recover from an external error.
- Requeue items that were purged from a message queue.
- Requeue items so their message handling state can become known.
- Requeue items when a new indexing feature is added.
- Requeue items after changing viewable columns.
- Requeue a repository after renaming it in order to re-index its contents for a standard search.

RequeueVaultObjects Utility Command Line Arguments

`RequeueVaultObjects` utility places messages on the queues at a lower priority than messages from user check-in or transition operations to ensures that re-indexing occurs as a background task and does not take precedence over user initiated actions.

RequeueVaultObjects is a command line utility that accepts several arguments. The arguments are listed in the following table. The arguments and values are not case sensitive in use.

Argument	Description	Required	Default
`-Repository <repository name>`	The repository that is requeued. The Home repository is not indexed and is therefore not valid for re-queueing. You could use the `VaultID` of the repository instead of the repository name.	Yes	N/A
`-MessageLimit <positive integer>`	Sets the number of messages to submit to the queues. The value must be between 1 and 100,000. If omitted, the utility defaults to its upper limit. The value entered here, `RequeueVaultObjects` creates a maximum of 100,000 messages in a single run to avoid overloading a message queue. The utility can be run as needed to complete loading of messages into queues.	No	100000
`-Status <message handling state>`	The message handling state to be queued. Only items matching this state are queued. Valid values are Queued, Processed, Failed, and Unknown.	No	Failed
`-Queues <list of queues>`	One or more queue names to which messages should be re-queued. For more information, see Message Handler States. Multiple queue names must be entered in a comma separated list. If omitted, all queues are processed.	No	All

IMPORTANT! If you have renamed a repository, you must use the `-Repository <repository name>` command to re-index the repository and enable the search process to locate the renamed repository.

Examples

To run the `RequeueVaultObjects` utility, open a command window, and change the directory to the `<INSTALL_PATH>\Vault\Utilities` directory containing the `RequeueVaultObjects.exe` file.

- Requeue all items in the site repository that failed to process

  ```
  RequeueVaultObjects -repository Site
  ```

  The `-status` and `-queues` arguments are not necessary in this case because they default to Failed and All, respectively.

- Requeue up to 5000 items that do not have their status tracked in the Site repository

  ```
  RequeueVaultObjects -repository Site -status Unknown -messagelimit 5000
  ```

- Requeue all items in the R&D repository that are pending for the full text and property handlers.

  ```
  RequeueVaultObjects -repository "R&D" -status Queued -queues
  "FullTextIndexer, PropertyIndexMessageHandler"
  ```

  You must enclose the queue list in double quotes and separate the queues using commas.

- Requeue all items in the R&D repository that are processed for the full-text handler.

  ```
  RequeueVaultObjects -repository "R&D" -status Processed -queues
  FullTextIndexer
  ```

  Enclose the *R&D* repository name in double quotes because it contains a special character, the ampersand (&). You must also enclose repository names with spaces in double quotes.

  After a command is executed, you should see *job completed successfully*. If the message *job failed* displays, verify that the correct parameter values were passed in the argument.

A log file is created containing the actions that were taken using the `RequeueVaultObjects` utility. The log file is located in the `<INSTALL_PATH>\Vault\Utilities` folder.

Running RequeueVaultObjects

After you add viewable columns in Notebook Explorer, run the `Requeuevaultobjects` utility to enable existing experiments to display the added columns. For more information, see "Customize Viewable Columns for Data" in the *BIOVIA Workbook Administration Guide*.

To run the RequeueVaultObjects utility:

1. Open a command window.
2. Navigate to the `RequeueVaultObjects.exe` utility, which resides in a folder similar to the following:

   ```
   C:\Program Files (x86)\BIOVIA\Vault\Utilities
   ```

3. Execute the following command once for each versioned repository and each relevant status:

   ```
   Requeuevaultobjects.exe -repository <RepositoryName> -queues
   propertyExtractionMessageHandler -status <Status>
   ```

   where:

   `<RepositoryName>` is the versioned repository name, enclosed in quotation marks if it contains spaces. Example: "Research and Development".

   <Status> is one of the following:

   - Processed. Always run the command for this status.
   - Unknown. Always run the command for this status.
   - Failed. Run the command if the repository contains objects that have this status and those objects contain data for the viewable columns that were added.

4. Open an experiment in Notebook Explorer and verify that the columns are displayed.

Automating the Management of Message Handling Status

You can use the combination of status tracking and the requeuing utility to automate the management of message handling within a Vault deployment. The requeuing utility is designed for use in automated jobs:

- You can specify all options on a single command line.
- No passwords are required.
- No desktop access or user interaction is required.

Using Windows scheduled tasks to automatically perform different requeuing tasks can enable proactively managing the state of the Vault server.

You can automating the resubmission of failed items by creating a batch file that calls RequeueVaultObjects for each repository, for example:

```
RequeueVaultObjects -repository Site RequeueVaultObjects -repository General
RequeueVaultObjects -repository MyRep
```

Create a task scheduler job to run the batch file at your specified interval.

Managing Large Requeuing Jobs

Note: The RequeueVaultObjects utility will not act on more than 100,000 records.

Large requeuing jobs present a couple of challenges to administer. They create a variable load on the middle-tier servers and they create a fairly sustained load against the database. Historically, they have been difficult to distribute across multiple servers and tool failures have been difficult to recover from.

Administrators can now manage a large re-queue job over time and distributed among multiple servers, so that they can minimize load created by the additional work, and thus, the impact of the work on end-users. Since the RequeueVaultObjects utility keeps track of its progress in the MessageHandlerStatus table, if it has a failure, the utility can start from where it left off the next time.

There are two strategies an administrator can choose from to accomplish large jobs.

The first is to schedule re-queuing jobs against a large set of messages during long periods of system downtime such ad nights and weekends.

The second is to “trickle” relatively small numbers of messages at regular intervals around the clock. These two strategies can be combined to best fit your particular Vault Server usage patterns.

The first large re-queue job an administrator is likely to tackle is re-queuing to get all their objects tracked in the MessageHandlerStatus table. Below is an example of how to do that across multiple Vault Servers.

On server 1, the administrator issues the following command to re-queue 25,000 items:

```
RequeueVaultObjects -repository "R&D" –status Unknown –messagelimit 25000
RequeueVaultObjects
```

This adds 25,000 messages to each queue on server 1 and sets the items' status to Requeued.

Next, on server 2, the administrator issues the following comment to re-queue the remaining 25,000 items:

```
RequeueVaultObjects -repository "R&D" –status Unknown –messagelimit 25000
RequeueVaultObjects
```

This adds 25,000 messages to each queue on server 2 and sets the items' status to Requeued.

After both of those tasks have run, 50,000 distinct objects are being reprocessed with their status tracked.

Message Limits on Automated Requeuing Jobs

Deciding on message limits for automated jobs is an inexact science. Since different handlers process different objects at different rates, there can be quite a bit of variability of the load caused between different batches of requeued messages, even when the batches are the same size. For this reason, BIOVIA recommends the constant trickle approach to managing very large requeue jobs. Also, it is much easier for administrators to monitor the system and adjust parameters on jobs that occur during the workday.

For example, consider a repository with 1 million objects in it, rows in the VaultObjects table, with three Vault Servers load balanced in front of it. Scheduling re-queue jobs for 5000 messages per hour on each of the three servers, scheduled at a slight offset from one another to prevent duplicate processing requests, brings the repository to a completely known state while managing queue backlogs and server load.

Administrators can monitor load on the database and servers closely for the first day or two, and make adjustments. For example, you might find that during the last two hours of the workday you do not want the re-queue jobs to run at all because the load on the database is greater due to people checking in experiments before they go home. You also might decide to run the jobs three times an hour during the night when very few users are on the system. You might decide to adjust the number of messages to 3000 or 7000.

Command line exit codes

`RequeueVaultObjects` has three different exit codes that you can use when automating the tool.

- 1 indicates an error understanding the arguments provided.
- 2 indicates that the arguments were understood, but there was a problem performing the requeuing task.
- 0 indicated successful completion.

Querying the Message Handler Status Table

Use the data in the Message Handler Status table to report on the indexing state of the Vault deployment.

The following query reports the overall indexing state within a repository:

```
select status, count(*) from messagehandlerstatus group by status order by
status;
```

The following query reports the status broken down by message handler:

```
select handlername, status, count(*) from messagehandlerstatus group by
handlername, status order by handlername;
```

Repository Items in a Unknown State

The following query reports the number of objects in a completely unknown state:

```
select count(*) from vaultobject
where guid not in (select vaultid from messagehandlerstatus);
```

The following query reports the list of objects in a completely unknown state:

```
select guid from vaultobject
where guid not in (select vaultid from messagehandlerstatus);
```

To get the count or list of unknown objects for a given handler, modify the above queries like this:

```
select guid from vaultobject
where guid not in (select vaultid from messagehandlerstatus where
handlername='FullTextIndexer');
```

FullTextIndexer is an example value. Replace it with the name of whichever handler you are interested in checking.

Items in a Queued State

An item might get sent to the Message Processing Service, but never gets processed, for example, if a queue is deleted with items in it. The following SQL statement identifies items in a Queued or Requeued state for more than a day.

```
select vaultid, handlername, status from messagehandlerstatus
where status in ('Queued','Requeued')
and
lastmodifiedtime < cast(sys_extract_utc(systimestamp) -1 as timestamp with
time zone);
```

To find items that have been queued more than a week, adjust the “-1” value in the cast statement to "-7".

You can run the query slowly depending on the amount of data in your system.

Experiments in the Failed State

Correlate a failure entry to the experiment that created the failure. A section in an experiment could cause the failure, while other sections and the experiment might have succeeded without the failed section. To get a list of all failures by the document, run the following query:

```
select name as "Name", vaultpath as "Vault Path", guid as "Vault ID" from
vaultobject
where guid in (select nvl(containerid, guid)
from vaultobject
where guid in (select vaultid from messagehandlerstatus
               where status='Failed'))
order by name;
```

If you want a similar report for another state such as Queued, replace the state name in the query with the state to use in the report.

To find items that failed to index in the past day:

```
select vaultid from messagehandlerstatus
where (status='Failed') and lastmodifiedtime > cast(sys_extract_utc
(systimestamp) -1 as timestamp with time zone);
```

Document State

To determine the state of a document and its sections, perform the following query. In the query, substitute the `Vaultid` for the document you want to find.

```
select vaultid, status, handlername from messagehandlerstatus
where vaultid in (select guid from vaultobject where
guid='Document.53435730-99e1-4f35-a803-90f096f2d5cd' or
containerid='Document.53435730-99e1-4f35-a803-90f096f2d5cd');
```

Query Service Configuration

The Query Service is a web service that enables searching and retrieving BIOVIA Vault Server data. The query service leverages Isentris Integrating Data Source (IDS) technology to provide hierarchical views of the Vault Server data models for data analysis and searching.

Improving Query Performance

You can modify the state queries to use a Parallel Hint to improve performance, for example:

```
select /*+ parallel */ vaultid, status, handlername from
messagehandlerstatus
where vaultid in (select guid from vaultobject where
guid='Document.53435730-99e1-4f35-a803-90f096f2d5cd' or
containerid='Document.53435730-99e1-4f35-a803-90f096f2d5cd');
```

Configuring the Query Service Partition Size

You can change the default partition size for the Query Service by modifying the public Vault Service `web.config` file.

To modify the file:

1. Using a text editor, open the `web.config` file in the following folder: `<INSTALL_PATH>\Vault\WebService`.
2. Edit the value setting of the following element (*`partition.size`*), which is 10,000 by default, and then save the file:

```
<appSettings>
...
<add key="SecurityFilteringService.PartitionSize" value="partition size"
/>
...
</appSettings>
```

Checking the State of Core Vault Services

From your Vault server, you can view the following services. Under normal circumstances if the following services are running, then your Vault installation is operational. There are rare instances where these services are running but Vault is not functional.

- Vault Workflow Service
- Vault Tomcat Service
- Vault Message Processing Service
- World Wide Web Publishing Service
- Vault Hub Synchronization Service
- Vault Client Service

To check the health of core Vault Services:

1. Click the **Server Manager**, in the **Services** tab, verify that the Vault services are running.
2. If all the services are running, then click the **Process** tab, and verify that the `Symyx.Vault.Messageprocessor.exe*32` process is running.
3. If the Vault services and message processing service processor application are not running, do any of the following actions:
   - Restart Vault servers every 30 days.
   - Automate by using windows scheduled tasks.
   - Archive Vault and SVS logs every 30 days.
   - Automate by using windows scheduled tasks.
   - Restart all Vault services weekly.
   - Automate by using windows scheduled tasks.
   - Check disk fragmentation every 3 months.
   - Review windows application and system logs every 60 days for errors, correct errors as needed.
   - Integrate with existing network management environment for advanced service monitoring.

Chapter 8:
Database Maintenance

The BIOVIA Vault Server database contains a set of repositories, associated data, and datawarehouse data. Configuring the database components includes creating multiple tablespaces and multiple accounts in an Oracle database. For more information, see the *BIOVIA Vault Server Installation Guide*.

You need to monitor and maintain the Oracle database components and processes.

Maintaining Oracle Text Indexes

BIOVIA Vault Server supports full-text searching of Vault documents in the RAS data warehouse schema through the use of Oracle Text. You must install Oracle Text component to make full-text searching available.

When implementing Oracle Text for a Vault implementation, there are two issues to consider:

- Access to the AUTO filter that indexes most document formats
- Regular updates of the indexes

AUTO Filter

One Oracle Text CONTEXT index, VAULTTEXT_TEXT on table VAULTTEXT, is created by the RAS schema SQL scripts. This indexes data contained in the Vault Server documents that have been loaded to the RAS schema.

VAULTTEXT_TEXT relies on the Oracle filtering technology to index binary documents. This allows full-text searching of Adobe Acrobat, Microsoft Office, and other binary document types.

Index Updates

Oracle Text indexes are not automatically updated when indexed content changes.

Automatic Index Synchronization

The index creation script utilizes the SYNC parameter and sets an interval of 3 minutes. The interval can be modified by issuing the following command logged in to SQL*Plus as the RAS schema owner:

```
SQL> ALTER INDEX VAULTTEXT_TEXT REBUILD PARAMETERS ('REPLACE METADATA SYNC
(EVERY "sysdate+(n/1440)")');
```

Where: n is the number of minutes. For example, `"sysdate+(2/1440)"` schedules a SYNC job every two minutes.

Index Maintenance

Oracle Text indexes must be maintained to remove old data and minimize index fragmentation. Index maintenance can be accomplished by either dropping and rebuilding the indexes or scheduling OPTIMIZE jobs for the index to improve query response time.

Optimizing Indexes

Oracle recommends using OPTIMIZE INDEX in REBUILD mode. An OPTIMIZE job can be submitted to the job scheduler. This job should be scheduled during off hours when Vault is inactive or system utilization is low. Refer to the Oracle Text Reference manual for details. A sample script, `Schedule_Optimize_Job.sql`, is provided and is available in the following location:

`\workbook2021\DatabaseScripts\Schemas\RAS\v1`

The content of the `Schedule_Optimize_Job.sql` is:

```
REM Schedule_Optimize_Job.sql
REM Example of job to optimize the text index VAULTTEXT_TEXT owned by
SYMYXDB submitted to the job scheduler
REM May be submitted as SYSTEM or as SYMYXDB
REM This example schedules the optimize job to run every Saturday night at 9
PM (Pacific time)
REM   starting on July 16, 2017.
REM Modify the date and timezone based on your physical location.  Pacific
time is GMT -7 hours
REM Modify other parameters as necessary
set echo on
BEGIN
sys.dbms_scheduler.create_job(
job_name => '"SYMYXDB"."OPTIMIZE_VAULTTEXT_INDEX"',
job_type => 'PLSQL_BLOCK',
job_action => 'ctx_ddl.optimize_index(''VAULTTEXT_TEXT'',CTX_DDL.OPTLEVEL_
FULL);',
repeat_interval => 'FREQ=WEEKLY;BYDAY=SAT;BYHOUR=21;BYMINUTE=0;BYSECOND=0',
start_date => to_timestamp_tz('2008-07-19 -7:00', 'YYYY-MM-DD TZH:TZM'),
job_class => 'DEFAULT_JOB_CLASS',
comments => 'optimize job',
auto_drop => TRUE,
enabled => FALSE);
sys.dbms_scheduler.set_attribute
       (name => '"SYMYXDB"."OPTIMIZE_VAULTTEXT_INDEX"', attribute => 'job_weight',
value => 1);
sys.dbms_scheduler.enable('"SYMYXDB"."OPTIMIZE_VAULTTEXT_INDEX"');
END;
```

Rebuilding Indexes

Instead of running the OPTIMIZE jobs, drop and rebuild the Text index during a maintenance window when users are not accessing the system. Ensure adequate space has been allocated to the TEMP tablespace prior to rebuilding these indexes. Verify that adequate time is available for the index rebuild to complete.

The syntax to drop and rebuild the index, logged in to SQL*Plus as the RAS schema owner is:

```
Drop index VAULTTEXT_TEXT force;
Create index VAULTTEXT_TEXT on VAULTTEXT (TEXT) indextype is ctxsys.context
parameters ('FILTER CTXSYS.AUTO_FILTER FORMAT COLUMN format CHARSET COLUMN
encoding SYNC (EVERY "sysdate+(5/1440)")');
```

> **Note:** Dropping the Text index might invalidate one or more views. Query for invalid objects and recompile as necessary.

Monitoring TEXT Index Operations, CTX Logging

In order to troubleshoot index problems, logging can be enabled for text indexes. However, logging only works if the user runs the OPTIMIZE job directly, rather than as a scheduled job in the job queue. Optionally, a TEXT index can be rebuilt rather than optimized. You can use logging to monitor index rebuild jobs, if optimization is not utilized.

Prior to executing the CTX.Logging script, any scheduled jobs should be removed from the job queue to avoid conflicts. Additionally, CTXSYS must have granted EXECUTE privilege on two packages: CTX_DDL and CTX_OUTPUT.

The log file is written to the default `LOG_DIRECTORY of ORACLE_HOME\ctx\log` or to a directory explicitly named by the CTXSYS user using the `CTX_ADM.SET_PARAMETER` package. Refer to the Oracle Text Reference manual for details.

Querying Tablespace Usage

The Database Administrator can run the `spaceusage.sql` script, as SYS or SYSTEM, to determine space utilization. For example:

```
/* script spaceusage.sql        */
/* run as SYS or user with DBA role    */ set pagesize 300
  set lines 150 set echo off
    column Allocated format 999,999,999,999,999
    column Free format 999,999,999,999,999
    column Pct_Free format 999,999.99
PROMPT SPACE AVAILABLE IN TABLESPACES
 select a.tablespace_name,sum(a.tots) Allocated, sum(a.sumb) Free, sum
(a.sumb)*100/sum(a.tots) Pct_Free from(select tablespace_name,
0 tots, sum(bytes) sumb from dba_free_space a group by tablespace_name union
 select tablespace_name,sum(bytes) tots,0 from dba_data_files group by
tablespace_name) a group by a.tablespace_name order by a.tablespace_name;
```

TABLESPACE_NAME	ALLOCATED	FREE	PCT_FREE
MDL	104,857,600	102,760,448	98.00
SYMYXAUDIT	104,857,600	103,809,024	99.00
SYMYXAUDITINDEX	209,715,200	208,666,624	99.50
SYMYXAUDITLOB	209,715,200	208,666,624	99.50
SYMYXDB	10,013,900,800	4,142,071,808	41.36
SYMYXIND	5,242,880,000	4,521,984,000	86.25
SYMYXLOB	5,242,880,000	5,079,760,896	96.89
SYMYXUSER	104,857,600	103,809,024	99.00
SYSAUX	859,832,320	57,606,144	6.70
SYSTEM	1,426,063,360	1,048,576	.07
UNDOTBS1	1,599,078,400	1,501,954,048	93.93

TABLESPACE_NAME	ALLOCATED	FREE	PCT_FREE
USERS	5,242,880	4,194,304	80.00
VAULT	1,048,576,000	711,589,888	67.86
VAULTIDX	1,048,576,000	862,126,080	82.22
VAULTLOB	4,367,319,040	440,401,920	10.08

15 rows selected.

Updating Pending Workbook Structure and Section Indexes

You need to update the reactions and structures that are entered into indexes from BIOVIA Workbook experiments. BIOVIA Direct creates index entries in a separate table in the RAS schema but does not add the transactions to the index until Direct UPDATE commands are run. Use the following commands to count and update these pending transactions.

Performance is affected if the number of pending transactions exceeds 3000.

To check for and execute pending updates:

1. For inverted molecule and fastsearch index transactions, enter:

   ```
   SELECT MDLAUX.PENDINGINVERSIONS('ALLCHEMSTRUCS') FROM DUAL; SELECT
   MDLAUX.PENDINGFASTSEARCH('ALLCHEMSTRUCS') FROM DUAL;
   ```

2. For inverted reaction fastsearch and inverted key index transactions, enter:

   ```
   SELECT MDLAUX.PENDINGINVERSIONS('ALLCHEMRXNS') FROM DUAL; SELECT
   MDLAUX.PENDINGFASTSEARCH('ALLCHEMRXNS') FROM DUAL;
   ```

3. Back up your database.
4. Ensure that the RAS schema tablespace has extra space available.
5. Log in to the RAS schema.
6. Open SQL*Plus.
7. Enter the following command to execute pending updates.

   Run this command on each index or table.

   ```
   SQL> EXECUTE MDLAUX.UPDATEPENDINGINVERSIONS(‘ALLCHEMRXNS’); SQL> EXECUTE
   MDLAUX.UPDATEPENDINGINVERSIONS(‘ALLCHEMSTRUCS’);
   ```

8. Enter the following command to update the indexes.

   Run this command on each index or table.

   ```
   SQL> EXECUTE MDLAUX.UPDATEPENDINGFASTSEARCH(‘ALLCHEMRXNS’); SQL> EXECUTE
   MDLAUX.UPDATEPENDINGFASTSEARCH(‘ALLCHEMSTRUCS’);
   ```

9. Exit SQL*Plus.
10. Log off the RAS schema.

For more information on these commands, see the *BIOVIA Direct Administration Guide* for the version you are running.

Collecting Database Statistics

By default, Oracle runs a nightly job that collects database statistics. If statistics about the Vault repository schemas need to be gathered more frequently, the DBA should set up and schedule a job for this purpose.

Monitoring Database Performance

The BIOVIA Vault Server security system performance might vary depending on the execution plan used by the Oracle optimizer. Your Oracle database administrator should monitor system performance on a daily basis, and select the Oracle-recommended baselines and profiles for security-related SQL.

- Use the Oracle Enterprise Manager Advisor Central/SQL Advisors /SQL Tuning Advisor page to schedule a SQL Tuning Advisor job. Select the Top Activity, Historical SQL (AWR), or SQL Tuning Sets to find the security SQL.
- Review the tuning recommendations to select a baseline and suggested profile to ensure the best performance.
- Select long running SQL statements to tune.
- Select Schedule SQL Tuning Advisor.

Upgrading the Oracle Database Client Post-Installation

If you upgrade Oracle Database Client after the Vault Server installation is complete and in production use and you have subsequent issues with the Vault Message Processing Service, Vault File Service, or Vault Workflow Service, re-enable them as follows:

1. Stop the World Wide Web Publishing Service and all Vault services.
2. Locate the following files in the Vault installation directory (by default, `C:\Program Files (x86)\BIOVIA\Vault`):
   - `DeploymentManager\web.config`
   - `FileService\Web.config`
   - `PrivateWebService\web.config`
   - `Rest\web.config`
   - `Utilities\PasswordReplacer.exe.config`
   - `Utilities\RequeueVaultObjects.exe.config`
   - `Utilities\Symyx.Vault.DatabaseUtility.exe.config`
   - `Utilities\Symyx.Workflow.Utility.exe.config`
   - `Utilities\VaultADMStoreManager.exe.config`
   - `Utilities\VaultToHubBootstrapper.exe.config`
   - `WebService\web.config`
   - `WindowsServices\Accelrys.Vault.IsolationChamber.exe.config`
   - `WindowsServices\Symyx.Vault.Message.Processing.Service.exe.config`
   - `WindowsServices\Symyx.Vault.MessageProcessor.exe.config`
   - `WindowsServices\Symyx.Workflow.Service.exe.config`
   - `WindowsServices\VaultHubSynchronizationService.exe.config`
3. Open each file in a text editor and update each file as follows:

a. Search for **Oracle.DataAccess**.

   Example: `<add assembly="Oracle.DataAccess, Version=2.112.4.0>`

b. Update the Version setting to reflect your updated Oracle Client version:

   - For Oracle 12.1.0.2, use **`4.121.2.0`**
   - For Oracle 12.1.0.1, use **`2.121.1.0`**
   - For Oracle 11.2.0.4, use **`2.112.4.0`**

c. Search for **assemblyBinding**.

d. If **`assemblyBinding`** does not yet exist, create it inside the <runtime> element of <configuration>:

```
<assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
</assemblyBinding>
```

e. Inside the assemblyBinding element, add a dependentAssembly for your new Oracle version:

```
<dependentAssembly>
  <assemblyIdentity name="Oracle.DataAccess" culture="neutral"
   publicKeyToken="89b483f429c47342" />
  <publisherPolicy apply="no" />
  <bindingRedirect oldVersion="0.0.0.0-255.255.65535.65535"
   newVersion="4.121.2.0" />  (use same version from step 3b)
</dependentAssembly>
```

f. Save and close the file.

4. Restart the World Wide Web Publishing Service and all Vault services.

Vault Database Backup Types and Frequencies

You can use the Oracle Recovery Manager (RMAN) or other tools and user-managed methods to perform database backups and restores. Database backups are necessary to protect the database against data loss.

Backup Types

Several different backup types are useful when working with Vault and RAS databases:

- **Hot backups**: recommended for production Vault and RAS databases

  Hot backups copy data files while a database is up and available for use.

  To support hot backups, the database must be running in archived log mode.

  Hot backups support both complete and point-in-time data recovery methods.

- **Cold backups**: recommended after initial database configuration and immediately prior to moving or cloning Vault and RAS databases

  Cold backups copy data files while the database is shut down. The Vault Server services must also be shut down before a cold database backup is taken, and then restarted after the backup completes. For more information, see Starting the Vault Services and Stopping the Vault Services.

  The database does not need to be running in archived log mode to support cold backups.

  Cold backups support recovery only up to the point in time that the cold backup was taken.

- **Exports**: recommended for copying data between databases

  Schema definitions and data are copied to an Oracle binary-format file while the database is up. This type of backup is generally used to transport data between databases. Data can only be recovered to the time the export was taken. This type of backup is not recommended for recovery purposes.

Backup Frequency

Frequency of backups depends on database usage and the amount of changes that occur. BIOVIA recommends the following backup frequency.

Usage	Type of Backup	Frequency
Heavy	Hot	Once a day during low usage hours (night)
Medium	Hot	2 - 3 times per week during low usage hours (night)
Low	Hot	Once per week during low usage hours (night)
All	Export - full	Once per week during low or no usage (weekend)

When you perform hot backups, you should also back up relevant archived redo log files.

Chapter 9: Repository Creation, Monitoring, and Maintenance

Most installations use separate versioned Vault Server repositories for separate purposes. The Workbook Workbook Installer enables you to create up to 20 separate repositories during installation, and you can add new repositories as needed by copying and modifying an existing vaultvariables.nb#.config file. For more information, see the *Vault Server Installation Guide*.

The following topics provide guidance for repository creation and management.

Determining What Repositories You Need

To determine what repositories you need to create for a Workbook installation, consider your current user community, how it might change in the future, and how your BIOVIA Workbook might change to meet the needs of the community.

Always create a separate repository for *shared* Vault objects such as forms, templates, section templates, and operations and subscribe all users to that repository. A separate repository for shared objects enables users seeking such an object to search only that repository, instead of having to search large folder hierarchies in which templates are mixed in with experiments. In most environments, different users require access to different intellectual property and sets of experiments. Consequently, a single repository for the shared objects and a separate repository for each different set of intellectual property and experiments is usually created.

The following is a list of things to consider when determining your repository requirements:

- Security
- Repository subscriptions
- Data organization that facilitates browsing and access
- Discipline, for example discovery, analytical, and process
- Geography
- User type, such as internal or external
- Environment: GXP or NonGXP
- Workflow associations that can be passed down through folder structures

For each individual repository, you can meet business and user needs by carefully building workflow associations, security models, and folder structures.

Implementing Folder Hierarchies that Avoid Folder Restrictions

For performance reasons, Workbook prevents users from adding more than **99,999** objects to any folder in a repository. Each of the following is counted as a separate object:

- A new document or template
- A new section in a document or template

  > **Note:** Individual files in a File section are not counted as separate objects.

- A new immediate-child subfolder (but not the child subfolder's objects, such as its own documents, templates, sections, and subfolders)

Workbook increments a counter (`orderSeq` on the `VaultCollection` table) each time an object is added, but it does **not** decrement this counter when an object is removed. After 99,999 objects have been added, users cannot commit any new objects. Attempts to do so result in an error message, and removing objects from the folder does not enable new objects to take their place.

Consequently, it is important to avoid any possibility of reaching this limit. One recommended approach to keeping within the limit is to create a new folder every month and require users who create experiments during that month to store them under the folder created for that month.

> **Tip:** In addition to keeping object counts in mind, also be aware that the number of folder *levels* in a repository can impact performance and the user experience. For best results, avoid creating more than five levels in your folder hierarchies.

Queries for Obtaining the Current Object Counter Values

You can execute queries to count the current actual number of objects in a folder and to return the value of the `orderSeq` counter, which will be *higher* than the actual number of objects, if objects have been removed from a folder.

1. To see the `GUID` of each folder, the count (actual number) of items currently in each folder, and the value of each folder's `orderSeq` counter, sorted from highest to lowest `orderSeq` value, execute the following query:

   ```
   select guid, count(*), max(orderseq) from vaultcollection group by guid
   order by 3 desc;
   ```

2. To determine which folder is represented by a specific `GUID` that was returned from your Step 1 query, copy-and-paste that `GUID` from your Step 1 results into the following query:

   ```
   select vaultpath from vaultobject where guid=<guid>;
   ```

3. To get a list of each object (Experiment, Procedure, Results, Conclusion, and so on) added to each folder, sorted from most recent to oldest, execute the following query:

   ```
   select vo.vaultpath, count(*), max(orderseq)
   from vaultcollection vc, vaultobject vo
   where vo.guid=vc.VAULTOBJECTGUID
   group by vo.vaultpath
   order by 3 desc;
   ```

Adding a Versioned Repository

After you install the Vault database and BIOVIA Vault Server, you can add versioned repositories to the Vault database. BIOVIA does not support the installation of repositories in a separate database from the Site repository.

Before you start this procedure, verify that the Windows User Account Control (UAC) is set to Never Notify on the BIOVIA Vault Server and any other computer on which the upgrade is performed.

To add a versioned repository:

1. Navigate to the temporary folder to which you extracted BIOVIA_Workbook_2021.zip.
2. Locate the `AddRepository.bat` file in the temporary folder's `DatabaseScripts` folder.
3. Copy the `vaultvariables.nb<x>.config` file that has the highest value for <x> and rename it to `vaultvariables.nb<x+1>.config`.

For example, if you already have 5 versioned repositories, copy `vaultvariables.nb5.config` to `vaultvariables.nb6.config`.

4. Edit the DEFINE statements in the `vaultvariables.nb<x+1>.config` file to reflect the repository that you are adding.
5. Run `CreateNewRepositoryUser.bat` to create the Oracle user account, and specify the following values:
   - Systemuser
   - Systempass
   - ConfigFileName

   For example:

   ```
   .\CreateNewRepositoryUser.bat systemuser systempass
   vaultvariables.nb<x+1>.config
   ```
6. Run the `AddRepository.bat` file with the `configFileName` as the parameter.

   For example:

   ```
   .\AddRepository.bat vaultvariables.nb<x+1>.config
   ```
7. Run `Symyx.Vault.DatabaseUtility.exe` with the `/mode:a` flag.

   `Symyx.Vault.DatabaseUtility.exe` is located in the `<INSTALL_PATH>\Utilities` folder.
8. Stop IIS and restart it by entering **iisreset** on a command-line prompt.

   Note: If Vault Server is installed in a farmed environment, you must execute `iisreset` on all the servers in the farm.
9. Stop the Vault services and then restart them. You can stop and start Vault services by using the Windows Services tool or by entering commands on the command-line. For details, see Vault Server Services on page 1.
10. Log in to the Workbook client and verify access to the new repository.

Adding and Managing External Repositories

BIOVIA Workbook enables creating new documents by importing (converting) of specific types of documents from external repositories into BIOVIA Workbook complete with all the existing data and associated structures, reactions, and text and image files.

These external repositories are hosted by BIOVIA Notebook Browser. The repositories are read-only. They are intended to allow access to external data. Users cannot modify the data in the external repositories. However, users can convert the data to a new Workbook experiment that is like the previous one and edit the converted experiment as needed.

If the imported (converted) documents contain ChemSeek structures with query features, these structures are not interpreted correctly. For more information, see the topic "Guidelines for handling ChemSeek structures with query features" in the BIOVIA Workbook online help.

The types of documents that can be imported from external sources include:

- Lab Journal (Symyx Discovery Notebook) files (`*.lj-axp`)
- Matrix (Symyx) Process Notebook Version 5.6 files (`*.mx-asp`)
- Matrix (Symyx) Formulations Notebook Version 5.6 files (`*.mx-asf`)

- XML files (`*.xml`)

  Only those XML files that have the format of the files in this list.

Adding External Repositories for Notebook Browser

You must add permissions to groups rather than directly to users. Log in to BIOVIA with global Foundation Hub Administrator permissions.

You need the following information to run the command in the following procedure:

- The host name of the Notebook Browser server.
- The TCP port number that Notebook Browser uses.

To add external repositories to use with BIOVIA Notebook Browser:

1. Install Notebook Browser using the instructions in the *BIOVIA Notebook Browser Installation Guide*.
2. Navigate to the BIOVIA Vault Server `<INSTALL_PATH>\Utilities`.
3. Open a command window and run the following:

   ```
   Symyx.Vault.DatabaseUtility.exe /mode:er /url:<Notebook Browser server
   name>:<port>
   ```

4. If the error message, *Could not establish a connection to the Notebook Browser using the <server>:<port> parameter provided,* displays in the command window, verify that the Notebook Browser service is running, and that the server name and port number are correct.

   a. Open a browser window to test the server name and port number with the following URL:

   ```
   http://Notebook Browser server
   name:<port>/NotebookBrowser/services/Repositories
   ```

   b. If you receive a 500 error message with message text similar to the following, *Access Restricted, REST API feature disabled. See documentation for how to enable REST API.* modify the nbb-`config.xml` file of the Notebook Browser service to enable the REST API.

   - In the `nbb-config.xml` file, update the enabled attribute in the services element to *true*, for example:

     ```
     <services enabled="true">
         <ip-address></ip-address>
     </services>
     ```

5. Restart the World Wide Web Publishing service from the `Services.msc` MMC snap-in, or by executing the `iisreset` command.
6. In the Vault Administration Console, grant repository subscription permissions to the external repositories to make the repositories available to users and groups.

   Use the Workbook client to import the `ExternalDocumentConversionTemplates.voexp` file.

Setting up the Conversion Service

After the repositories are added, only members of the Global Administrators group can access the repositories.

To enable access for other BIOVIA Workbook users:

- Create an *external repositories* group to access the repositories.

  For more information, see "Vault Users" in the *BIOVIA Vault Server Administration Tools Guide*.

- (Optional) Configure the ExternalDataConversionService application service properties.

BIOVIA provides a default document template for use with the converted data. The template is defined in ExternalDataConversionService permission properties. If you do not want to use the default template, you must modify the application permission to include a path to the specific document template to use.

Assign the application permissions to the new group.

To enable access to the external data, you need to grant the ExternalDataConversionService application permission.

For more information, see "Application Permissions" in the *BIOVIA Vault Server Administration Tools Guide*.

Configuring Example Forms and Templates

Workbook users might need to configure the default example forms and templates that are provided with the External Data Conversion service feature.

The information about configuring the default example forms and templates for the converted documents is available in the Workbook online Help section "Using external repositories".

Removing External Repositories

Remove external repositories c from the BIOVIA Vault Server by running the batch file DelExternalReps.bat from a command window.

The file is located in the <Extraction folder>\BIOVIA Workbook_<version_number>\DatabaseScripts folder. The <Extraction folder> is the location you selected to extracted the distribution ZIP file.

Updating External Repositories

Running the file DelExternalReps.bat removes all external repositories from the BIOVIA Vault Server.

If BIOVIA Notebook Browser is installed on a computer that has been replaced and has a new name, the external repositories in BIOVIA Vault Server cannot function properly. You must remove the external repositories, re-add the repositories, and resetting subscription permissions.

If the name of a repository changes in BIOVIA Notebook Browser or in the original repository, for example, in Discovery Notebook, the name is not updated in BIOVIA Vault Server. To update the name, you can change the name in BIOVIA Vault Server manually using the BIOVIA Workbook Administration or by removing and adding the repositories again.

Chapter 10: Cloning the Vault Database

This chapter explains how to clone the Vault database to a new server or to a new Oracle instance.

> **Note:** If you are also cloning the Foundation Hub database, see "Cloning the Foundation Hub Environment" in the *Foundation HUB Installation Guide*. You can clone the Foundation Hub database before or after you clone the Vault database.

Summary of Vault database cloning steps:

1. Inform Vault Server and BIOVIA Workbook users of the estimated downtime.
2. Identify an Oracle directory object.
3. Make a **cold** back up of the data that will be moved.

   > **IMPORTANT!** Creating a **cold** backup is necessary for data consistency. You can use the Oracle datapump export utility or whatever backup software you like. Oracle privileges are required. For more information, see Vault Database Backup Types and Frequencies on page 61.

4. Export the Vault and RAS schemas.
5. Prepare the target database by creating tablespaces, accounts, roles, and profiles and installing the appropriate version of BIOVIA Direct.
6. Import the Vault schemas to the target database.
7. Import the RAS schemas to the target database.
8. If the Vault database connection information (hostname, database service name, port number) has changed:
   - Enter the new connection information in the Vault Configuration Tool as described in Updating Connection and Configuration Parameters on page 4.
   - Update the `G_REPOSITORY` table in the `SITE` schema as described in Update Connection Details in the G_REPOSITORY Table on page 72.
   - Update the `VAULTCONFIG` table in the `SITE` schema as described in Update Connection Details in the VAULTCONFIG Site Table on page 73

   > **Tip:** If you need to change connection information without cloning the database, Step 8 is the only required step.

9. Restart the Vault services and IIS on the Vault Server computer.
10. Verify the cloned database.

Identify an Oracle Directory Object

The Data Pump utility writes export files to an Oracle directory object. When a new Oracle database is created, Oracle creates a default data pump directory, DATA_PUMP_DIR.

If you do not want to use the default data pump directory, you can create a new one.

To create a new Oracle directory object:

1. Log in to the database using SQL*Plus as a user with DBA permissions.
2. Ensure that the server you want to use for the new directory object is available on the network.
3. In SQL*Plus, issue the following command:

   ```
   CREATE DIRECTORY Oracle_datapump_dir as 'c:\folder_name\Oracle_datapump_
   dir;'
   ```

4. Use a command similar to the following:

   ```
   CREATE DIRECTORY BIOVIA_DATA_PUMP_DIR as 'c:\biovia_data\biovia_data_
   pump_dir';
   ```

5. Grant read and write access to the Vault Oracle users who need to access the object:

   ```
   GRANT read, write on DIRECTORY BIOVIA_DATA_PUMP_DIR to SYSTEM;
   ```

6. Log out of SQL*Plus.

Exporting the Vault Repository and RAS Schemas

You must export the existing Vault repository and RAS schemas from your source database before you can import them into the target database.

When you export the schemas, be sure that the dump files for each schema use either a different name or a different path.

Vault Repository Schemas to Export

Export the following Vault repository schemas as described in see Export the Vault Repository Schemas on page 69

- Site
- VaultUser (Home)
- Versioned (all Versioned repositories)
- WFTools
- Fileservice

RAS Schemas to Export

Export the RAS user and dbuser schemas as described in Export the RAS Schemas on page 70.

The RAS schemas contain the tables and indexes used to store and search reaction and structure data in Workbook. These tables and indexes need special handling during export and import.

Before you export RAS schemas, BIOVIA recommends that you review the *BIOVIA Direct 2021 Administration Guide* for more information about exporting schema tables, particularly the information in the "Command Reference" chapter.

Export the Vault Repository Schemas

The Oracle account used for this task must be a dba account with read/write permissions for the referenced directory object.

To export Vault Repository schemas:

1. Log on to the database computer as a member of the Administrators group and open a command window.
2. Execute the Data Pump export command (expdp) for the Vault Site schema:

   ```
   expdp dbaaccount/password@Source_tnsname SCHEMAS=(schema_name)
   DUMPFILE=siteuser.dmp DIRECTORY=Oracle_datapump_dir
   LOGFILE=expsiteuser.log
   ```

   Where:

   - `dbaaccount` is an Oracle account that has DBA privileges
   - `password` is the password for that account, for example, `MANAGER`
   - `Source_tnsname` is the tnsname of the source database, for example `DBw2k3`
   - `schema_name` is the name of the schema, for example `vaultsite`
   - `Oracle_datapump_dir` is the name of the directory object to which to export the schema, for example `DATA_PUMP_DIR`
3. Repeat the export command for each of the following additional Vault schemas:
   - VaultUser (Home)
   - VaultVersion1
   - VaultWFTools
   - Fileservice
   - All other versioned repository schemas

Export the RAS Schemas

The Oracle account used for this task be a dba account with read/write permissions for the referenced directory object.

Before you export the RAS schemas, prepare Direct indexes for exporting:

Prepare the Direct indexes by logging in to SQL*Plus as the SYMYXDB user and typing the following commands.

1. For structure index tables:

   ```
   select mdlaux.prepareindexexport ('allchemstrucs') from dual;
   ```
2. For reaction index tables:

   ```
   select mdlaux.prepareindexexport ('allchemrxns') from dual;
   ```

To export the RAS schemas:

1. Log on to the RAS database computer as a member of the Administrators group and open a command window.
2. Execute the Data Pump export command (expdp) for the RAS SYMYXDB schema:

   ```
   expdp dbaaccount/password@Source_tnsname SCHEMAS=symyxdb
   DUMPFILE=symyxdbuser.dmp DIRECTORY=Oracle_datapump_dir
   LOGFILE=expsymyxdbuser.log
   ```

   Example:

   ```
   expdp system/MANAGER@DBw2k3 SCHEMA=symyxdb DUMPFILE=symyxdb.dmp
   ```

```
DIRECTORY=data_pump_dir LOGFILE=expsymyxdb.log
```

3. Repeat the export command for the SYMYXDBUSER schema.

Prepare the Target Database

1. Add the BIOVIA Vault Server tablespaces to the target database, using the same names as the source database.

   The target tablespace names must be the same as the source tablespace names to prevent import errors.
2. Add the VAULT_USER_ROLE and VAULT_PROFILE to the target database if the role or profile does not exist.
3. Add the Vault Oracle user IDs to the target database, Site, Home, Versioned, and WFTools if present. Implement roles and permissions that are the same as those in the source database.

   If the Oracle accounts on the target database are the same as the accounts on the source database but are used for an existing Vault implementation, then create different Oracle accounts for the new Vault schemas.
4. If you plan to also move the RAS schemas:
   - Add the RAS tablespaces to the target database using the same names as the source database.
   - Add the RAS_USER_ROLE and RAS_PROFILE to the target database if these objects do not yet exist.
   - Add the RAS Oracle user ids and role to the target database. The roles and permissions should be the same as those in the source database.
5. Verify that the same version of BIOVIA Direct exists on the target database.

Import the Vault Repository Schemas

After you export the Vault repository schemas, you can use the Data Pump Import utility to import them into the Vault and RAS schemas.

To import the Vault repository schemas:

1. Log on to the target database computer as a member of the Administrators group.

   > **Note:** If you prefer to perform the import while logged in to the source database computer, first be sure that it can connect with the target database computer.
2. Open a command window.
3. Execute the Data Pump import command for the Vault Site schema:

   ```
   impdp dbaaccount/password@Target_tnsname SCHEMAS=schema_name
   DUMPFILE=siteuser.dmp DIRECTORY=Oracle_datapump_dir
   LOGFILE=impsiteuser.log
   ```

   Where:
   - `dbaaccount` is the Oracle account with DBA privileges.
   - `password` is the password for that user, for example MANAGER.
   - `Target_tnsname` is the tnsname of the target database, for example, DBw2k8.
   - `schema_name` is the name of the schema that you plan to import, for example vaultsite.

   - `Oracle_datapump_dir` is the name of the directory object into which the export files were copied or written, for example symyx_data_pump_dir.

4. Repeat the command for the following additional Vault repository schemas:
   - VaultUser (Home)
   - VaultVersion1
   - VaultWFTools
   - FILESERVICE
   - All other Versioned repository schemas
5. Recompile any invalid objects.
6. Gather statistics on the Vault repository schemas.

Import the RAS Schemas

If you are moving the RAS schemas as well as the Vault repository schemas, you must import them the same way you imported the Vault repository schemas.

To import data extracted using the Data Pump Export Utility:

1. Log on to the target database computer as a member of the Administrators group.

   > **Note:** If you prefer to perform the import while logged in to the source database computer, first be sure that it can connect with the target database computer.

2. Open a command window.
3. Execute the Datapump Import command for the RAS schema:

   ```
   impdp dbaaccount/password@Target_tnsname SCHEMAS=schema_name
   DUMPFILE=symyxdbuser.dmp SCHEMA=symyxdb DIRECTORY=Oracle_datapump_dir
   LOGFILE=impsymyxdbuser.log
   ```

4. Connect as SYS and grant the following permissions:

   ```
   CONNECT sys/sys_password AS sysdba;
   GRANT EXECUTE ON DBMS_SESSION TO SYMYXDB;
   GRANT EXECUTE ON DBMS_RLS TO SYMYXDB;
   GRANT EXECUTE ON DBMS_LOB TO SYMYXDB WITH GRANT OPTION;
   ```

5. Connect as symyxdb and set the RAS security context:

   ```
   CONNECT symyxdb/symyxdb_password; CREATE OR REPLACE CONTEXT RAS_SECURITY
   USING RASPROJECTSECURITY;
   ```

6. Repeat Step 3 for the SYMYXDBUSER schema.
7. Recompile any invalid objects.
8. Gather statistics on the RAS schemas.

Update Connection Details in the G_REPOSITORY Table

Permissions required: On the Vault database computer, log in to SQL*Plus as the Site schema owner (for example, vaultsite/vaultsite_password).

Update the `databaseconnection` column of the `G_REPOSITORY` table in the `Site` schema to reflect the new database server, listener port, and service name. If Oracle accounts changed, you must also update the User ID.

To update the G_REPOSITORY Site table:

1. Determine the old database host name:

   ```
   select databaseconnection from g_repository;
   ```

2. Update the database HOST name:

   ```
   Update g_repository set databaseconnection = replace
   (databaseconnection, 'HOST=Source_name', 'HOST=’Target_name');
   ```

   Where:

   - `Source_name` is the computer name of the source database
   - `Target_name` is the computer name of the target database.

   The host names are case sensitive.

   Example:

   ```
   Update g_repository set databaseconnection = replace
   (databaseconnection, 'HOST=DBw2k3', 'HOST=DBw2k8');
   ```

3. Update the database port number if necessary:

   ```
   Update G_REPOSITORY set databaseconnection = replace
   (databaseconnection, 'PORT=Source_###', 'PORT=Target_###') ;
   ```

4. Update the database service name:

   ```
   Update G_REPOSITORY set databaseconnection = replace
   (databaseconnection, 'SERVICE_NAME=Source_name',
   'SERVICE_NAME=Target_name');
   ```

5. Update the database userid name if necessary.

   If the Oracle accounts are the same, skip this step.

   ```
   Update G_REPOSITORY set databaseconnection = replace
   (databaseconnection,'User ID=Source_ID','User ID=Target_ID') ;
   ```

6. Commit the changes:

   ```
   Commit;
   ```

7. Exit SQL*Plus.

Update Connection Details in the VAULTCONFIG Site Table

To update the `VAULTCONFIG` table in the Site schema:

1. Log in to the Vault database computer as the Site schema owner (for example, vaultsite/valutsite_password)
2. To move Vault Server and continue using the same Foundation Hub database, set the `Hub.AppReg.Vault.VaultEndpoint` in the `VAULTCONFIG` table by executing the following command:

   ```
   Update VAULTCONFIG set entry = 'newvaulthostname' where name =
   'Hub.AppReg.Vault.VaultEndpoint';
   ```

Verify the Cloned Vault Database

To verify the cloned Vault database:

1. Log in to the Vault Administration Console and verify that it functions.
2. Log on to the Workbook client system and verify that it works.
3. Review the log file for any errors.

Appendix A: Workbook Data Model

BIOVIA Workbook provides document creation and editing capabilities and an interface to interact with BIOVIA Vault Server repositories.

The BIOVIA Workbook data model contains domain-specific objects for experiment design and data management tasks.

In BIOVIA Workbook, scientific information is stored and managed as a document. A document consists of a collection of sections defined to hold a specific type of information such as Text, File, Form, Table, Materials, Results, and Reaction sections.

The document section types provide different presentation formats for managing data that feed into business processes such as compliance and audit management, operations on domain content models, and experiment management.

Core Content Management Data Model

Document data is stored and managed as Vault objects using the Framework Core Content Management Data Model.

A Vault object consists of core properties, content, preview, and hierarchical member objects. Vault objects are stored in relational Vault database tables.

Vault objects are also system objects that:

- Independently identify objects using URL references
- Maintain object status indicators
- Maintain audit history
- Manage security
- Manage associations
- Administer signature policies
- Facilitate workflows

The core content management data model uses the following:

- Dublin core metadata
- Vault data tables
- System objects
- Domain data model

Dublin Core Metadata

A Vault object metadata consists of core properties that describe the basic properties of a Vault object. These core properties are based on standards set by the Dublin Core Metadata Initiative.

The Symyx Framework, the Vault repository, a client application, or an end-user can assign core properties. Some core properties are read-only, while other core properties are editable.

The table lists and defines the core properties of a Vault object:

Display Name	Property Key	Value Data Type	Description
Associations	Associations	Associations	Defines the relationships assigned to the object.
Association Target	Association Target	Boolean	Indicates if the object can act as the target of a user-defined association. When set to True the object can act as the target of a user-defined association.
Auto Name	Autoname	String	Specifies the autoname string mask.
Check-in Signature Policy	CheckInSignaturePolicy	CheckInSignaturePolicies	Specifies the signature policies that apply when the object is checked into a Vault repository.
Availability	CheckoutState	CheckoutState	Specifies the check out status of the object at when the object was retrieved from the server. Read-only.
Class	Class	String	Specifies the assembly-qualified typename of the .NET class that implements the containing object.
Creation Date	ClientCreationDate	DateTimeOffset	Specifies a point or period of time associated with an event in the lifecycle of the resource. Read-only.
Size	ContentSize	Integer	Specifies the size of the object data. Read-only.
Contributors	Contributor	String	Specifies an entity responsible for making additions to the resource.
Scope	ContentSize	Integer	Specifies the spatial or temporal topic of the resource, the spatial applicability of the resource, or the jurisdiction under which the resource is relevant.

Display Name	Property Key	Value Data Type	Description
Created by	Creator	VaultID	Specifies an entity responsible for making contributions to the resource.
Description	Description	String	Provides information about the resource.
Flags	Flags	Integer	Describes the system level flags that might apply to a Vaultobject.
Format	Format	String	Specifies the file format, physical medium, or dimensions of a resource. Read-only.
ID	Identifier	VaultID	Specifies an unique reference to the resource within a given context. Read-only.
IsLocked	IsLocked	Boolean	Indicates if the Vault resource is locked by another user.
Language	Language	String	Specifies the language used with the resource.
ObjectData	ObjectData	Byte[]	Contains hidden object properties.
PendingContentHistory	PendingContentHistory	MutableContentHistory	Specifies a list of pending content history entries.
Permissions	Permissions	ObjectPermissions	Specifies the current permissions to the object. Read-only.
Preview	Preview	Image	Specifies an implementation-defined preview image of the object.
PropertySets	PropertySets	List<PropertySetIdentifier>	Specifies a list of property set identifiers.
Published By	Publisher	String	Specifies an entity responsible for making the resource available. Read-only.
Related To	Relation	String	Identifies a related resource.
RepositoryVersion	RepositoryVersion	String	Specifies the version of a repository.
Rights	Rights	String	Contains information about rights held in and over the resource.

Display Name	Property Key	Value Data Type	Description
Initial Check-In Date	ServerCreationDate	DateTimeOffset	Specifies the creation date and time of the object from the server.
Source	Source	String	Specifies the resource from which the described resource is derived.
Subject	Subject	String	Specifies the topic of the resource.
Name	Title	String	Specifies a name given to the resource.
Total Content Size	TotalContentSize	Long	Specifies the size of the object data and all of its contained children.
Type	Type	String	Specifies the type of resource.
Vault Path	VaultPath	String	Specifies the location of the object in the Vault repository.
Current Version	Version	Integer	Specifies the version number of the object. Read-only.
Version Comment	VersionComment	String	Provides a description of the current version for the object count.
Version Creation Date	VersionCreationDate	DateTimeOffset	Indicates the creation date and time of the object from the server. Read-only.
Version Created By	VersionCreator	VaultID	Specifies the User ID responsible for the current version of the object content. Read-only.
VersionIdentifier	VersionIdentifier	VaultID	Specifies the System Identifier for the specific version of the object.
Workflow Stages	WorkflowSummary	String	Specifies a deliminated list of the current stage for each of the workflows in which the object is currently participating.

System Objects

Actor

Inherits properties from VaultObject.

ApplicationPermissions

Properties	Type
Application	String
Permission	String

Assembly

Properties	Type
Assembly	Assembly

Document or DocumentTemplate

Properties	Type
ActiveLayoutForEdit	DocumentLayoutType
ActiveLayoutForReadOnly	DocumentLayoutType
AllowBroseForInsertableSEctions	Boolean
Application	IDocumentEditor
DefaultInsertableSections	Dictionary<DataScope, List<DocumentSection>>
DefaultInsertableSectionsVaultUris	List<VaultUri>
DocumentProperties	PropertySet<DocumentPropertyKey>
DynamicInsertableSectionLocations	Dictionary<DataScope, List<Folder>>
EnforceSectionOrder	Boolean
ForceInitialCheckIn	Boolean
InsertableSectionLocations	List<VaultUri>
IsTemplate	Boolean
Item	DocumentSection
PreviewSection	VaultID
SectionLayout	String
Sections	List<DocumentSection>

Properties	Type
SettingsControl	UserControl
SettingsProviders	List<ISettingsProviders>
SettingsTitle	String
TemplateCheckInSignaturePolicies	List<VaultID>
TemplateUri	VaultUri

DocumentReportTemplate

Properties	Type
ContextObject	Object
DynamicSEctionToReportableMap	IDictionary<string, IReportable>
ID	String
IsAuditTrailReport	Boolean
PersistentSettings	XmlReader
ReportSections	IList<ReportSection>
Settings	XmlReader
StaticReportSections	IList<ReportSection>

DocumentSection

Properties	Type
AllowCloneWithoutData	Boolean
AllowSectionDelete	Boolean
AllowSectionRename	Boolean
Application	IDocumentEditor
Document	Document
Icon	Image
IsDynamic	Boolean
IsLocked	Boolean
IsReferenceable	Boolean
IsTemplate	Boolean
LockSection	SignatureAndUsers
SectionCanUnlock	Boolean

Properties	Type
SectionIcon	Image
SectionNoClone	Boolean
SectionProperties	PropertySet<DocumentSectionPropertyKey>
SectionControl	UserControl
SettingsProviders	List<ISettingsProviders>
SettingsTitle	String
TemplateUri	VaultUri
UnlockSection	SignatureAndUsers

Favorites

Inherits properties from VaultObject.

Folder

Properties	Type
AutonameFormat	String
BaseType	BaseVaultObjectType
Count	Integer
Item	VaultObject
List	VaultObjectInnerList
MaxCount	Long

Form

Properties	Type
AssemblyBits	Byte[]
AssemblyName	String
Definition	String
Metadata	String
Namespace	String
Preview	Image
PublishedWidgets	Dictionary<string, int>
QueryConfigurationManager	QueryServiceConfiguration

Group

Properties	Type
Name	String
EmailAddress	String
IsOpen	Boolean

Property

Properties	Type
AssignedBy	AssignedBy
Comment	String
DisplaySpecifier	PropertyDisplaySpecifier
DisplayValue	String
IsNull	Boolean
IsValid	Boolean
Key	String
PropertyClass	PropertyClass
PropertySet	PropertySet
Type	Type
TypeName	String
ValidationResults	ValidationResult[]
Value	Object
ValueIsNull	Boolean
ValueTypeConverter	TyperConverter

Query

Properties	Type
ObtainParentContainers	Boolean
SQL	String
QueryClause	QueryClause
SQLGenerator	QuerySQLGenerator

Recipe

Properties	Type
ConcentrationUnits	UnitKey
MassUnits	UnitKey
MaterialRoles	IEnumerable<string>
UnitProcedures	IEnumerable<UnitProcedure>
VolumeUnits	UnitKey

ReportDefinition

Properties	Type
FixedSettings	Boolean

Repository

Properties	Type
AllowArchiving	Boolean
AllowAuditing	Boolean
AllowDeletes	Boolean
AllowSynchronization	Boolean
AllowWorkflow	Boolean
AutonameFormats	List<AutonameFormat>
Behaviors	RespositoryBehaviors
IsManaged	Boolean
StoreAllVersions	Boolean

Resources

Properties	Type
AllOf	String
AnyOf	String
Culture	CultureInfo
Material	Bitmap
OneOf	String
Operation	Bitmap

Properties	Type
ResourceManager	ResourceManager
UnitProcedure	Bitmap

SearchResults

Inherits properties from VaultObject.

SignaturePolicies

Properties	Type
SignatureProvider	String
AuthenticationProvider	String
CredentialInputControl	String
DefaultMeaning	String
DeafultReason	String
ReasonChoices	List<String>
MeaingChoices	List<String>
RequireSingleObjectSigning	Boolean

Vocabulary

Properties	Type
Phrases	IEnumerable<string>

User

Properties	Type
UserName	String
FullName	String
EmailAddress	String
HomeFolder	Folder
Profile	UserProfile
SiteAffinity	SiteAffinity
IsActive	Boolean
ApplicationPermissions	ApplicationPermissionsCollection

WorkflowAssociation

Properties	Type
Code	String
Enabled	Boolean

WorkflowDefinition

Properties	Type
CodeBehind	String
DefinitionType	String
Layout	String
MaxNumberActivations	Integer
Rules	String
Xoml	String

Domain Data Model

The BIOVIA Workbook domain-specific data models include property set definitions, you can deploy using the Vault Deployment Utility. A property set definition contains the metadata of a domain object, and directs behaviors and user interface for structured data.

The table describes the property set definitions available with the domain-specific data models.

Key	Name	Type	Description
Property Set: ActualAmount			
Amount	ActAmount	Measurement	
CalcEquiv	Act Calc Equiv	Value	
CalcMass	Act Calc Mass	Quantity	
CalcMoles	Act Calc Moles	Quantity	
CalcVol	Act Calc Vol	Quantity	
TheoryAct	Theory (Act.)	Quantity	
Yield	% Yield	Quantity	
Property Set: AnalyticalMaterial			
CorrectionFactor	Correction Factor	Quantity	Specifies the correction factor for the material (mass fraction).
ExpiryDate	Expiry Date	Date/Time	Specifies the expiry date for the material.

Key	Name	Type	Description
Grade	Grade	String	Specifies the material's grade.
LotNumber	Lot #	String	Specifies the lot number of this substance.
Manufacturer	Manufacturer	String	Specifies the manufacturer of the material.
Property Set: AnalyticalPreparation			
FinalpH	Final pH	Measurement	The measured, final pH of the preparation. A dimensionless quantity.
Notes	Notes	String	
NotifyChanged	Notify Changed	Boolean	
PlannedAmount	Planned Amt	Quanity	The planned amount of preparation. A quantity in mass or volume units.
PreparationDate	Preparation Date	Date/Time	
TotalAmount	Total Amt	Meaurement	The total volume of mass of preparation. A measurement in volume or mass units.
Property Set: AnalyticalPreparationStep			
ActualAmount	Actual Amt	Measurement	The measured amount of material in mass or volume units.
ActualConcentration	Actual Con	Quantity	The concentration of the material based on the actual amount.
AdditionalActualAmount	Additional Actual Amount	Measurement	An additional measured amount of material in mass or volume units. This can be used if the material was added from two lots.
AdditionalAmountLot	Addition Lot #	String	The Log number for the additional amount of material.
Adjust	Adjust To	String	Mark this component as the Adjust to Total Volume or Adjust to Final pH component.

Key	Name	Type	Description
Description	Description	String	
DilutionFactor	Dilution Factor	Value	A calculated dilution factor.
DrugConcentration	Corrected Conc	Quantity	The corrected concentration of the material, different from the actual amount if the correction factor is less than one.
Material	Material	|material	The material added in this step.
Plan Amount	Planned Amt	Quantity	The planned amount of material. A quantity in mass, volume, or concentration.
PlanConcentration	PlanConc	Quantity	The calculated corrected concentration, based on the value of the correction factor.
PlanDrugConcentration	Plan Corr Conc	Quantity	The calculated mass of material based on the plan amount.
PlanMass	Planned Mass	Quantity	The calculated volume of material based on the plan amount.
PlanVolume	Planned Vol	Quantity	The calculated corrected concentration, based on the value of the correction factor.
Property Set: Container			
Amount	Amount	Quantity	
Barcode	Barcode	String	
Capacity	Capacity	Quantity	
Comments	Comments	String	
Label	Label	String	
Location	Container Location	Location	
Material	Material	Material	
TareWeight	Tare Weight	Quantity	
Type	Type	String	
Property Set: Equipment			
Category	Category	String	The category of this equipment

Key	Name	Type	Description
Classification	Classification	String	The classification of the equipment
Department	Department	String	The department of the equipment
EquipmentID	EquipmentID	String	The ID of this equipment
EquipmentName	Name	String	The name of this equipment
EquipmentType	Type	String	The type of this equipment
IntendedUse	Intended Use	String	The intended use of this equipment
LastCalibrationDate	Last Calibration Date	Date/Time	The date of this equipment's last calibration
LastServiceDate	Last Service Date	Date/Time	The date of this equipment's last service
Location	Location	String	The location of this equipment
Manufacturer	Manufacturer	String	The manufacturer of this equipment
Model	Model	String	The model of this equipment
NextCalibrationDate	Next Calibration Date	Date/Time	The date of this equipment's next calibration
NextServiceDate	Next Service Date	Date/Time	The date of this equipment's next service
ReferenceNumber	Reference Number	String	The reference number of this equipment
Role	Role	String	
Property Set: Material			
AbstractMaterial	Abstract Material	AbstractMaterial	The Abstract Material
Comments	Comments	String	
Density	Density	Quantity	
FormattedMolFormula	MF	Molecular Formula	Formatted Molecular Formula
InitialAmount	InitialAmount	Quantity	
Location	Material	Location	

Key	Name	Type	Description
	Location		
MF	MF (Unformatted)	String	Molecular Formula
MW	MW	Value	
Name	Name	String	The name of the material
Parent	Material	Material	
Preparation	Preparation	Preparation	
Role	Role	String	
Structure	Structure	Structure	
Property Set: MaterialGroup			
Group	Group	Group	A description of the material group
Origin	Source	String	The source of this material
Property Set: MaterialRegistration			
BatchID NormalizedStructure Registered Response SubstanceId	Batch ID Normalized Structure Registered Registration Response Substance ID	String Structure Boolean String String	The batch identity of the registered structure The normalized form of the registered substance held in the registration system Indicates whether this substance has been registered A message generated by the registration system in response to this registration request The unique identify of the registered structure
Property Set: Operation			
Alerts	Alerts	String	The alerts collected during operation execution
Category	Category	String	Operation category for ordering purposes.
ContentFields	Content Fields	List	List of content fields to wrap and order property class.
ExecutionState	Execution State	String	The current state of the operation. This is only valid during operation.

Key	Name	Type	Description
Icon	Icon	Image	An icon displayed for this operation.
IsActive	Is Active	Boolean	Whether the operation is available for building unit procedures and recipes.
Name	Name	String	The name of the operation, for example, heat, or stir.
Notes	Notes	String	The description of the operation provided by the recipe designer.
Number	Number	String	The position of the operation within the unit procedure.
Observations	Execution Observations	String	The observations of the operation provided by the user running the recipe. The description of the operation
Phrase	Phrase	String	
SelectedPropertySetDefinitions	Selected	List	The currently selected property sets that is associated with the operation.
TimeStamp	Time Stamp	Date/Time	The time at which the operation was executed.
UsedPropertySets	Used Display	List	The actual property sets being used by the operation.
Property Set: PlannedAmount			
Amount	Plan Amount	Quantity	
CalcEquiv	Plan Calc Equiv	Value	
CalcMass	Plan Calc Mass	Quantity	
CalcMoles	Plan Calc Moles	Quantity	
CalcVolume	Plan Calc Vol	Quantity	
Property Set: Plate			
PlateId	Plate ID	String	Plate ID for identifying the plate.
WellId	Well ID	String	Well ID for the position of the well in the plate.

Key	Name	Type	Description
Property Set: ReactionMaterial			
FractionalSaltData	Salt/Hydrate	FractionalSaltData	Information about the fractional salt or hydrate composition of this material.
Label	Label	String	Label for the material within the reaction.
LimitingReagent	LR	Boolean	Limiting Reagent for this reaction
NEMAKey	NEMAKey	String	NEMA key
PurityConcentration	Purity/Conc	Quantity	Purity or Concentration of material.
Step	Step	String	Material Step
StepId	Step Id	String	
StoichiometricCoefficient	SC	Integer	Stoichiometric Coefficient
Property Set: ReactionStep			
Description	Description	String	The description of the reaction step.
Name	Name	String	The name of the reaction step.
Path	Path	String	The path of the reaction step.
Reaction Step	Reaction Step	Reaction String	The step name of the reaction step.
Property Set: TargetReference			
TargetReference	Sample Reference	TargetReference	Refers to SampleID property in the Sample Identification PropertySet Collection of Sample References.
Property Set: TargetReferences			
TargetReferences	Sample Reference	List	Represents a collection of Sample References; refers to the SampleID property in the Sample Identification PropertySet.
Property Set: UnitProcedure			
Alerts	Alerts	String	Unit procedure alerts collected during execution.

Key	Name	Type	Description
Category	Category	String	
Description	Description	String	
EnforceOrder	Enforce Order	Boolean	
ExecutionState	Execution State	String	The execution state of the unit procedure.
Icon	Icon	Bitmap	An icon displayed in the User Interface to identify this Unit procedure.
Location	Location	Point	The location of the unit procedure in the flowchart display.
Name	Name	String	
Notes	Notes	String	
Number	Number	String	The position of the unit procedure within the recipe.
Observations	Observations	String	Unit procedure observations collected during execution.
Predecessors	Predecessors	String	The Predecessors of the unit procedure.
Repeatable	Repeatable	Boolean	If true, allow the unit procedure to be performed multiple times, false to only one time.
TimeStamp	Time Stamp	Date/Time	The time at which the unit procedure was completed.

Appendix B:
Setting the IP Address and Domain Restrictions in IIS

To enable the IP and domain Restrictions role service:

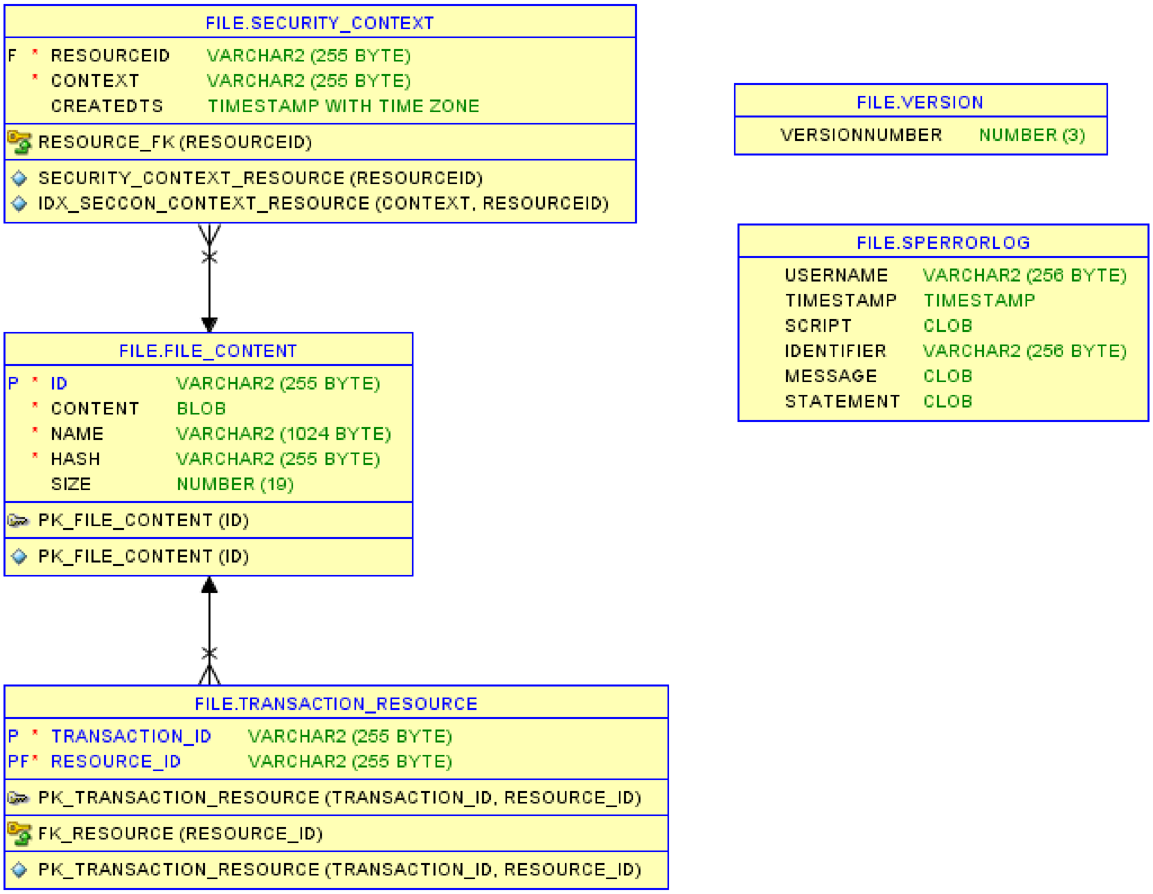
1. On the Vault server computer, click **Start** > **Administrative Tools** > **Service Manager** > **Roles**.
2. Click **Add Role Services**, select **IP and Domain Restrictions**, and then **Next**.
3. In the **Confirm Installation Selections** dialog, review the messages, and then click **Install**.
4. Click **Roles** > **Web Server (IIS)** > **Internet Information Services (IIS) Manager**.
5. Double-click the **IP Address and Domain Restrictions** icon.
6. In the **IP Address and Domain Restrictions** dialog, in **Actions**, click **Add Allow Entry**.
7. Add the IP address or range of address, and then click **OK**.
8. In Actions, click **Edit Feature Settings**, in **Access for unspecified clients**, select **Allow**, and click **OK**.

Appendix C: Entity Relationship Diagrams

The following topics provide Entity Relationship Diagrams for Vault Server entities:

- File Entities
- Site Entities
- User Entities
- Versioned Entities
- Workflow Entities

File Entities



FILE.SECURITY_CONTEXT			
F	*	RESOURCEID	VARCHAR2 (255 BYTE)
	*	CONTEXT	VARCHAR2 (255 BYTE)
		CREATEDTS	TIMESTAMP WITH TIME ZONE

RESOURCE_FK (RESOURCEID)

SECURITY_CONTEXT_RESOURCE (RESOURCEID)
IDX_SECCON_CONTEXT_RESOURCE (CONTEXT, RESOURCEID)

FILE.VERSION	
VERSIONNUMBER	NUMBER (3)

FILE.SPERRORLOG	
USERNAME	VARCHAR2 (256 BYTE)
TIMESTAMP	TIMESTAMP
SCRIPT	CLOB
IDENTIFIER	VARCHAR2 (256 BYTE)
MESSAGE	CLOB
STATEMENT	CLOB

FILE.FILE_CONTENT			
P	*	ID	VARCHAR2 (255 BYTE)
	*	CONTENT	BLOB
	*	NAME	VARCHAR2 (1024 BYTE)
	*	HASH	VARCHAR2 (255 BYTE)
		SIZE	NUMBER (19)

PK_FILE_CONTENT (ID)

PK_FILE_CONTENT (ID)

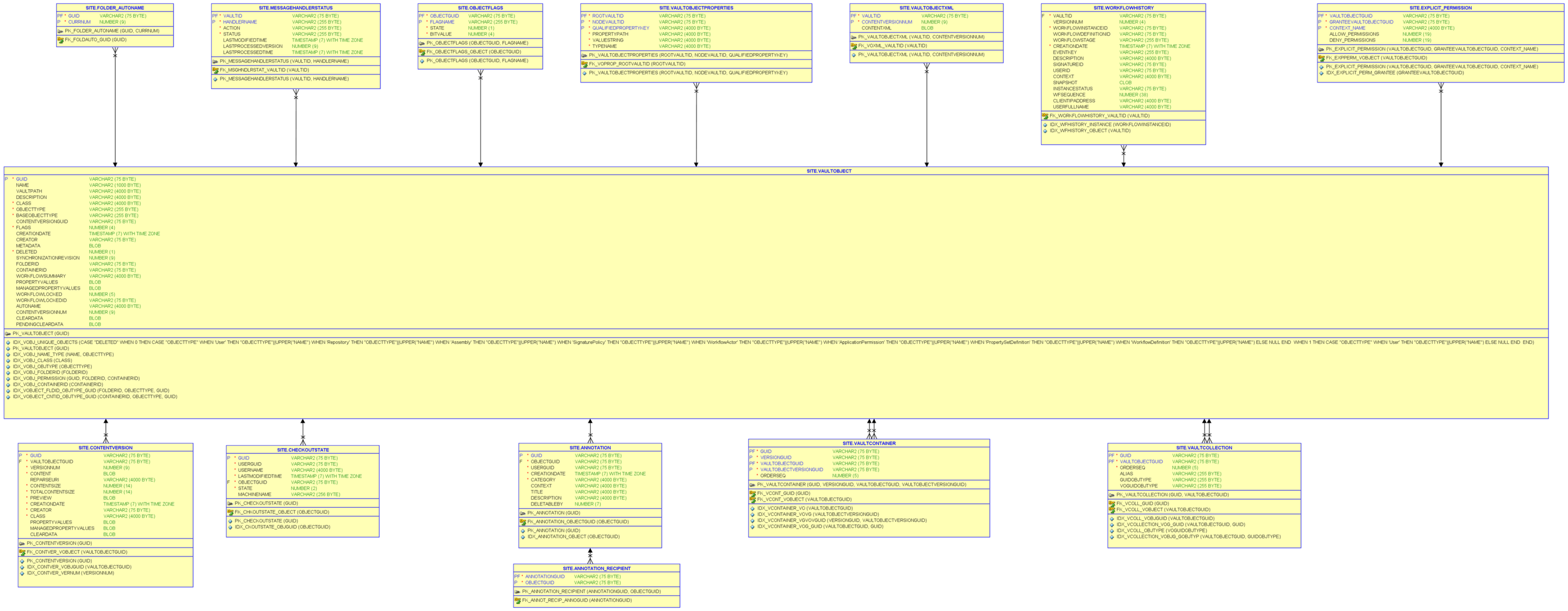
FILE.TRANSACTION_RESOURCE			
P	*	TRANSACTION_ID	VARCHAR2 (255 BYTE)
PF	*	RESOURCE_ID	VARCHAR2 (255 BYTE)

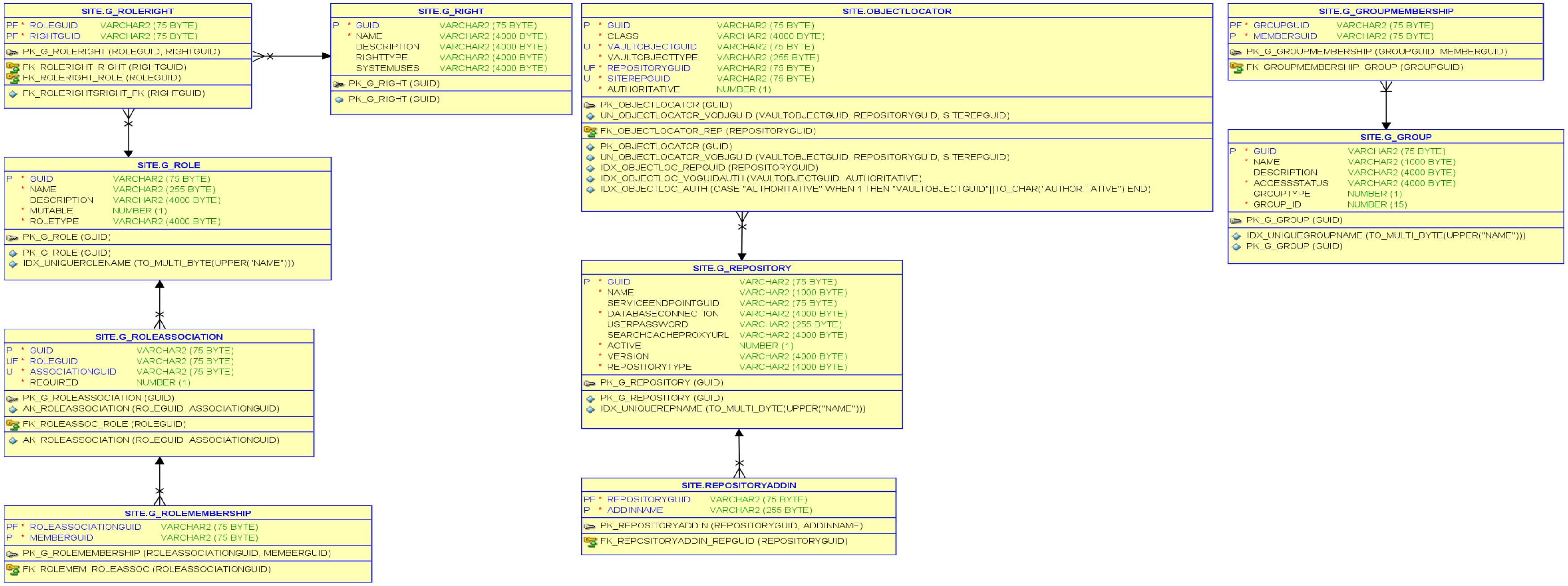
PK_TRANSACTION_RESOURCE (TRANSACTION_ID, RESOURCE_ID)

FK_RESOURCE (RESOURCE_ID)

PK_TRANSACTION_RESOURCE (TRANSACTION_ID, RESOURCE_ID)

Site Entities


SITE.FOLDER_AUTONAME
PF * GUID VARCHAR2 (75 BYTE)
P * CURRNUM NUMBER (9)
PK_FOLDER_AUTONAME (GUID, CURRNUM)
FK_FOLDAUTO_GUID (GUID)
SITE.MESSAGEHANDLERSTATUS
PF * VAULTID VARCHAR2 (75 BYTE)
P * HANDLERNAME VARCHAR2 (255 BYTE)
* ACTION VARCHAR2 (255 BYTE)
* STATUS VARCHAR2 (255 BYTE)
LASTMODIFIEDTIME TIMESTAMP (7) WITH TIME ZONE
LASTPROCESSEDVERSION NUMBER (9)
LASTPROCESSEDTIME TIMESTAMP (7) WITH TIME ZONE
PK_MESSAGEHANDLERSTATUS (VAULTID, HANDLERNAME)
FK_MSGHNDLRSTAT_VAULTID (VAULTID)
PK_MESSAGEHANDLERSTATUS (VAULTID, HANDLERNAME)
SITE.OBJECTFLAGS
PF * OBJECTGUID VARCHAR2 (75 BYTE)
P * FLAGNAME VARCHAR2 (255 BYTE)
* STATE NUMBER (1)
* BITVALUE NUMBER (4)
PK_OBJECTFLAGS (OBJECTGUID, FLAGNAME)
FK_OBJECTFLAGS_OBJECT (OBJECTGUID)
PK_OBJECTFLAGS (OBJECTGUID, FLAGNAME)
SITE.VAULTOBJECTPROPERTIES
PF * ROOTVAULTID VARCHAR2 (75 BYTE)
P * NODEVAULTID VARCHAR2 (75 BYTE)
P * QUALIFIEDPROPERTYKEY VARCHAR2 (4000 BYTE)
* PROPERTYPATH VARCHAR2 (4000 BYTE)
* VALUESTRING VARCHAR2 (4000 BYTE)
* TYPENAME VARCHAR2 (4000 BYTE)
PK_VAULTOBJECTPROPERTIES (ROOTVAULTID, NODEVAULTID, QUALIFIEDPROPERTYKEY)
FK_VOPROP_ROOTVAULTID (ROOTVAULTID)
PK_VAULTOBJECTPROPERTIES (ROOTVAULTID, NODEVAULTID, QUALIFIEDPROPERTYKEY)
SITE.VAULTOBJECTXML
PF * VAULTID VARCHAR2 (75 BYTE)
P * CONTENTVERSIONNUM NUMBER (9)
CONTENTXML BLOB
PK_VAULTOBJECTXML (VAULTID, CONTENTVERSIONNUM)
FK_VOXML_VAULTID (VAULTID)
PK_VAULTOBJECTXML (VAULTID, CONTENTVERSIONNUM)
SITE.WORKFLOWHISTORY
F * VAULTID VARCHAR2 (75 BYTE)
VERSIONNUM NUMBER (4)
* WORKFLOWINSTANCEID VARCHAR2 (75 BYTE)
WORKFLOWDEFINITIONID VARCHAR2 (75 BYTE)
WORKFLOWSTAGE VARCHAR2 (255 BYTE)
* CREATIONDATE TIMESTAMP (7) WITH TIME ZONE
EVENTKEY VARCHAR2 (255 BYTE)
DESCRIPTION VARCHAR2 (4000 BYTE)
SIGNATUREID VARCHAR2 (75 BYTE)
USERID VARCHAR2 (75 BYTE)
CONTEXT VARCHAR2 (4000 BYTE)
SNAPSHOT CLOB
INSTANCESTATUS VARCHAR2 (75 BYTE)
WFSEQUENCE NUMBER (38)
CLIENTIPADDRESS VARCHAR2 (4000 BYTE)
USERFULLNAME VARCHAR2 (4000 BYTE)
FK_WORKFLOWHISTORY_VAULTID (VAULTID)
IDX_WFHISTORY_INSTANCE (WORKFLOWINSTANCEID)
IDX_WFHISTORY_OBJECT (VAULTID)
SITE.EXPLICIT_PERMISSION
PF * VAULTOBJECTGUID VARCHAR2 (75 BYTE)
P * GRANTEEVAULTOBJECTGUID VARCHAR2 (75 BYTE)
P * CONTEXT_NAME VARCHAR2 (4000 BYTE)
ALLOW_PERMISSIONS NUMBER (19)
DENY_PERMISSIONS NUMBER (19)
PK_EXPLICIT_PERMISSION (VAULTOBJECTGUID, GRANTEEVAULTOBJECTGUID, CONTEXT_NAME)
FK_EXPPERM_VOBJECT (VAULTOBJECTGUID)
PK_EXPLICIT_PERMISSION (VAULTOBJECTGUID, GRANTEEVAULTOBJECTGUID, CONTEXT_NAME)
IDX_EXPLICIT_PERM_GRANTEE (GRANTEEVAULTOBJECTGUID)
SITE.VAULTOBJECT
P * GUID VARCHAR2 (75 BYTE)
NAME VARCHAR2 (1000 BYTE)
VAULTPATH VARCHAR2 (4000 BYTE)
DESCRIPTION VARCHAR2 (4000 BYTE)
* CLASS VARCHAR2 (4000 BYTE)
* OBJECTTYPE VARCHAR2 (255 BYTE)
* BASEOBJECTTYPE VARCHAR2 (255 BYTE)
CONTENTVERSIONGUID VARCHAR2 (75 BYTE)
* FLAGS NUMBER (4)
CREATIONDATE TIMESTAMP (7) WITH TIME ZONE
CREATOR VARCHAR2 (75 BYTE)
METADATA BLOB
* DELETED NUMBER (1)
SYNCHRONIZATIONREVISION NUMBER (9)
FOLDERID VARCHAR2 (75 BYTE)
CONTAINERID VARCHAR2 (75 BYTE)
WORKFLOWSUMMARY VARCHAR2 (4000 BYTE)
PROPERTYVALUES BLOB
MANAGEDPROPERTYVALUES BLOB
WORKFLOWLOCKED NUMBER (5)
WORKFLOWLOCKEDID VARCHAR2 (75 BYTE)
AUTONAME VARCHAR2 (4000 BYTE)
CONTENTVERSIONNUM NUMBER (9)
CLEARDATA BLOB
PENDINGCLEARDATA BLOB
PK_VAULTOBJECT (GUID)
IDX_VOBJ_UNIQUE_OBJECTS (CASE "DELETED" WHEN 0 THEN CASE "OBJECTTYPE" WHEN 'User' THEN "OBJECTTYPE"||UPPER("NAME") WHEN 'Repository' THEN "OBJECTTYPE"||UPPER("NAME") WHEN 'Assembly' THEN "OBJECTTYPE"||UPPER("NAME") WHEN 'SignaturePolicy' THEN "OBJECTTYPE"||UPPER("NAME") WHEN 'WorkflowActor' THEN "OBJECTTYPE"||UPPER("NAME") WHEN 'ApplicationPermission' THEN "OBJECTTYPE"||UPPER("NAME") WHEN 'PropertySetDefinition' THEN "OBJECTTYPE"||UPPER("NAME") WHEN 'WorkflowDefinition' THEN "OBJECTTYPE"||UPPER("NAME") ELSE NULL END WHEN 1 THEN CASE "OBJECTTYPE" WHEN 'User' THEN "OBJECTTYPE"||UPPER("NAME") ELSE NULL END END)
PK_VAULTOBJECT (GUID)
IDX_VOBJ_NAME_TYPE (NAME, OBJECTTYPE)
IDX_VOBJ_CLASS (CLASS)
IDX_VOBJ_OBJTYPE (OBJECTTYPE)
IDX_VOBJ_FOLDERID (FOLDERID)
IDX_VOBJ_PERMISSION (GUID, FOLDERID, CONTAINERID)
IDX_VOBJ_CONTAINERID (CONTAINERID)
IDX_VOBJECT_FLDID_OBJTYPE_GUID (FOLDERID, OBJECTTYPE, GUID)
IDX_VOBJECT_CNTID_OBJTYPE_GUID (CONTAINERID, OBJECTTYPE, GUID)
SITE.CONTENTVERSION
P * GUID VARCHAR2 (75 BYTE)
F * VAULTOBJECTGUID VARCHAR2 (75 BYTE)
* VERSIONNUM NUMBER (9)
* CONTENT BLOB
REPARSEURI VARCHAR2 (4000 BYTE)
* CONTENTSIZE NUMBER (14)
* TOTALCONTENTSIZE NUMBER (14)
* PREVIEW BLOB
* CREATIONDATE TIMESTAMP (7) WITH TIME ZONE
* CREATOR VARCHAR2 (75 BYTE)
* CLASS VARCHAR2 (4000 BYTE)
PROPERTYVALUES BLOB
MANAGEDPROPERTYVALUES BLOB
CLEARDATA BLOB
PK_CONTENTVERSION (GUID)
FK_CONTVER_VOBJECT (VAULTOBJECTGUID)
PK_CONTENTVERSION (GUID)
IDX_CONTVER_VOBJGUID (VAULTOBJECTGUID)
IDX_CONTVER_VERNUM (VERSIONNUM)
SITE.CHECKOUTSTATE
P * GUID VARCHAR2 (75 BYTE)
* USERGUID VARCHAR2 (75 BYTE)
* USERNAME VARCHAR2 (4000 BYTE)
* LASTMODIFIEDTIME TIMESTAMP (7) WITH TIME ZONE
F * OBJECTGUID VARCHAR2 (75 BYTE)
* STATE NUMBER (2)
MACHINENAME VARCHAR2 (256 BYTE)
PK_CHECKOUTSTATE (GUID)
FK_CHKOUTSTATE_OBJECT (OBJECTGUID)
PK_CHECKOUTSTATE (GUID)
IDX_CKOUTSTATE_OBJGUID (OBJECTGUID)
SITE.ANNOTATION
P * GUID VARCHAR2 (75 BYTE)
F * OBJECTGUID VARCHAR2 (75 BYTE)
* USERGUID VARCHAR2 (75 BYTE)
* CREATIONDATE TIMESTAMP (7) WITH TIME ZONE
* CATEGORY VARCHAR2 (4000 BYTE)
CONTEXT VARCHAR2 (4000 BYTE)
TITLE VARCHAR2 (4000 BYTE)
DESCRIPTION VARCHAR2 (4000 BYTE)
DELETABLEBY NUMBER (7)
PK_ANNOTATION (GUID)
FK_ANNOTATION_OBJECTGUID (OBJECTGUID)
PK_ANNOTATION (GUID)
IDX_ANNOTATION_OBJECT (OBJECTGUID)
SITE.ANNOTATION_RECIPIENT
PF * ANNOTATIONGUID VARCHAR2 (75 BYTE)
P * OBJECTGUID VARCHAR2 (75 BYTE)
PK_ANNOTATION_RECIPIENT (ANNOTATIONGUID, OBJECTGUID)
FK_ANNOT_RECIP_ANNOGUID (ANNOTATIONGUID)
SITE.VAULTCONTAINER
PF * GUID VARCHAR2 (75 BYTE)
P * VERSIONGUID VARCHAR2 (75 BYTE)
PF * VAULTOBJECTGUID VARCHAR2 (75 BYTE)
P * VAULTOBJECTVERSIONGUID VARCHAR2 (75 BYTE)
* ORDERSEQ NUMBER (5)
PK_VAULTCONTAINER (GUID, VERSIONGUID, VAULTOBJECTGUID, VAULTOBJECTVERSIONGUID)
FK_VCONT_GUID (GUID)
FK_VCONT_VOBJECT (VAULTOBJECTGUID)
IDX_VCONTAINER_VO (VAULTOBJECTGUID)
IDX_VCONTAINER_VOVG (VAULTOBJECTVERSIONGUID)
IDX_VCONTAINER_VGVOVGUID (VERSIONGUID, VAULTOBJECTVERSIONGUID)
IDX_VCONTAINER_VOG_GUID (VAULTOBJECTGUID, GUID)
SITE.VAULTCOLLECTION
PF * GUID VARCHAR2 (75 BYTE)
PF * VAULTOBJECTGUID VARCHAR2 (75 BYTE)
* ORDERSEQ NUMBER (5)
ALIAS VARCHAR2 (255 BYTE)
GUIDOBJTYPE VARCHAR2 (255 BYTE)
VOGUIDOBJTYPE VARCHAR2 (255 BYTE)
PK_VAULTCOLLECTION (GUID, VAULTOBJECTGUID)
FK_VCOLL_GUID (GUID)
FK_VCOLL_VOBJECT (VAULTOBJECTGUID)
IDX_VCOLL_VOBJGUID (VAULTOBJECTGUID)
IDX_VCOLLECTION_VOG_GUID (VAULTOBJECTGUID, GUID)
IDX_VCOLL_OBJTYPE (VOGUIDOBJTYPE)
IDX_VCOLLECTION_VOBJG_GOBJTYP (VAULTOBJECTGUID, GUIDOBJTYPE)



```
SITE.G_ROLERIGHT
PF * ROLEGUID        VARCHAR2 (75 BYTE)
PF * RIGHTGUID       VARCHAR2 (75 BYTE)
PK_G_ROLERIGHT (ROLEGUID, RIGHTGUID)
FK_ROLERIGHT_RIGHT (RIGHTGUID)
FK_ROLERIGHT_ROLE (ROLEGUID)
FK_ROLERIGHTSRIGHT_FK (RIGHTGUID)
```

```
SITE.G_RIGHT
P  * GUID            VARCHAR2 (75 BYTE)
   * NAME            VARCHAR2 (4000 BYTE)
     DESCRIPTION     VARCHAR2 (4000 BYTE)
     RIGHTTYPE       VARCHAR2 (4000 BYTE)
     SYSTEMUSES      VARCHAR2 (4000 BYTE)
PK_G_RIGHT (GUID)
PK_G_RIGHT (GUID)
```

```
SITE.OBJECTLOCATOR
P  * GUID              VARCHAR2 (75 BYTE)
   * CLASS             VARCHAR2 (4000 BYTE)
U  * VAULTOBJECTGUID   VARCHAR2 (75 BYTE)
   * VAULTOBJECTTYPE   VARCHAR2 (255 BYTE)
UF * REPOSITORYGUID    VARCHAR2 (75 BYTE)
U  * SITEREPGUID       VARCHAR2 (75 BYTE)
   * AUTHORITATIVE     NUMBER (1)
PK_OBJECTLOCATOR (GUID)
UN_OBJECTLOCATOR_VOBJGUID (VAULTOBJECTGUID, REPOSITORYGUID, SITEREPGUID)
FK_OBJECTLOCATOR_REP (REPOSITORYGUID)
PK_OBJECTLOCATOR (GUID)
UN_OBJECTLOCATOR_VOBJGUID (VAULTOBJECTGUID, REPOSITORYGUID, SITEREPGUID)
IDX_OBJECTLOC_REPGUID (REPOSITORYGUID)
IDX_OBJECTLOC_VOGUIDAUTH (VAULTOBJECTGUID, AUTHORITATIVE)
IDX_OBJECTLOC_AUTH (CASE "AUTHORITATIVE" WHEN 1 THEN "VAULTOBJECTGUID"||TO_CHAR("AUTHORITATIVE") END)
```

```
SITE.G_GROUPMEMBERSHIP
PF * GROUPGUID       VARCHAR2 (75 BYTE)
P  * MEMBERGUID      VARCHAR2 (75 BYTE)
PK_G_GROUPMEMBERSHIP (GROUPGUID, MEMBERGUID)
FK_GROUPMEMBERSHIP_GROUP (GROUPGUID)
```

```
SITE.G_GROUP
P  * GUID            VARCHAR2 (75 BYTE)
   * NAME            VARCHAR2 (1000 BYTE)
     DESCRIPTION     VARCHAR2 (4000 BYTE)
   * ACCESSSTATUS    VARCHAR2 (4000 BYTE)
     GROUPTYPE       NUMBER (1)
   * GROUP_ID        NUMBER (15)
PK_G_GROUP (GUID)
IDX_UNIQUEGROUPNAME (TO_MULTI_BYTE(UPPER("NAME")))
PK_G_GROUP (GUID)
```

```
SITE.G_ROLE
P  * GUID            VARCHAR2 (75 BYTE)
   * NAME            VARCHAR2 (255 BYTE)
     DESCRIPTION     VARCHAR2 (4000 BYTE)
   * MUTABLE         NUMBER (1)
   * ROLETYPE        VARCHAR2 (4000 BYTE)
PK_G_ROLE (GUID)
PK_G_ROLE (GUID)
IDX_UNIQUEROLENAME (TO_MULTI_BYTE(UPPER("NAME")))
```

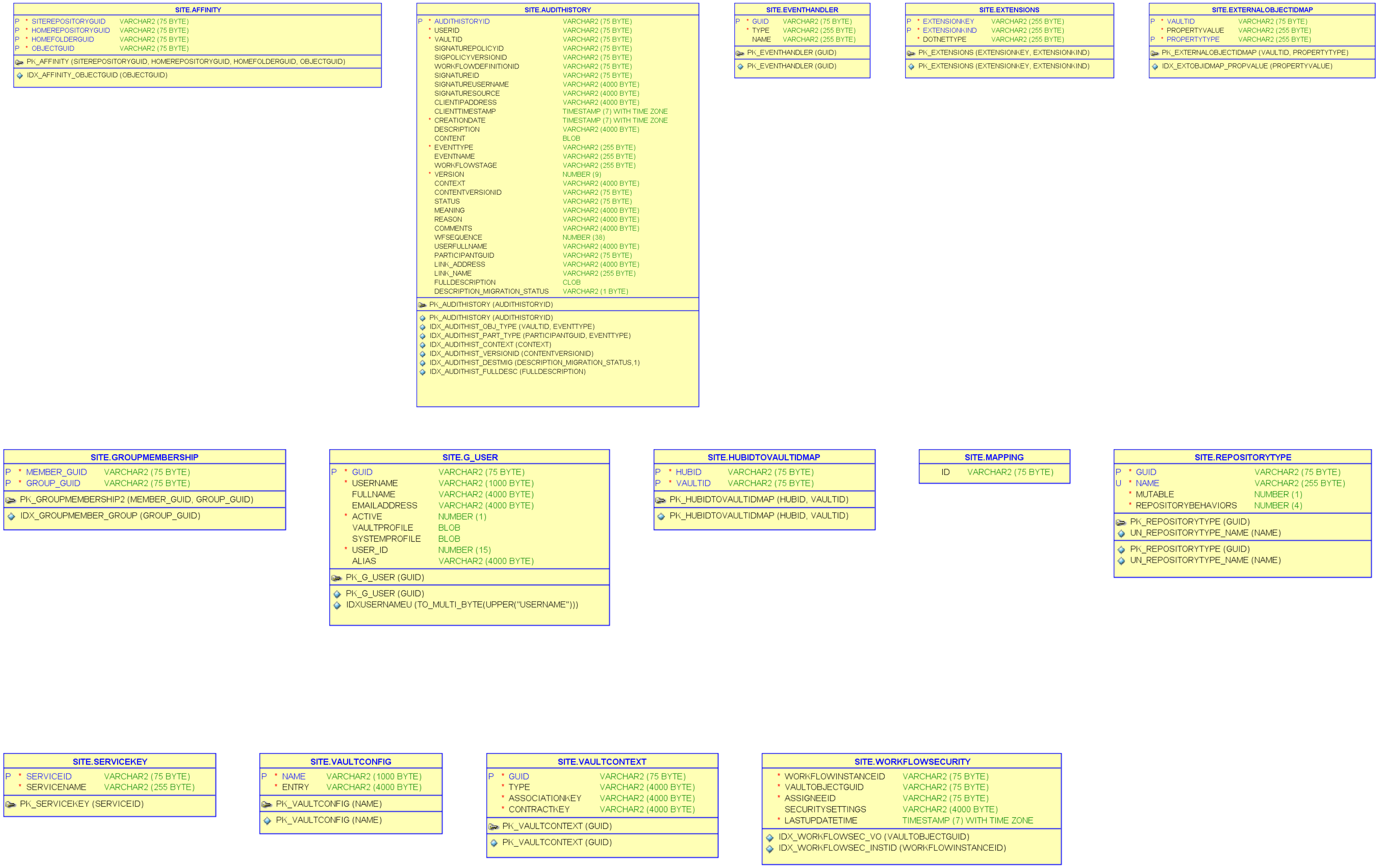
```
SITE.G_REPOSITORY
P  * GUID                  VARCHAR2 (75 BYTE)
   * NAME                  VARCHAR2 (1000 BYTE)
     SERVICEENDPOINTGUID   VARCHAR2 (75 BYTE)
   * DATABASECONNECTION    VARCHAR2 (4000 BYTE)
     USERPASSWORD          VARCHAR2 (255 BYTE)
     SEARCHCACHEPROXYURL   VARCHAR2 (4000 BYTE)
   * ACTIVE                NUMBER (1)
   * VERSION               VARCHAR2 (4000 BYTE)
   * REPOSITORYTYPE        VARCHAR2 (4000 BYTE)
PK_G_REPOSITORY (GUID)
PK_G_REPOSITORY (GUID)
IDX_UNIQUEREPNAME (TO_MULTI_BYTE(UPPER("NAME")))
```

```
SITE.G_ROLEASSOCIATION
P  * GUID              VARCHAR2 (75 BYTE)
UF * ROLEGUID          VARCHAR2 (75 BYTE)
U  * ASSOCIATIONGUID   VARCHAR2 (75 BYTE)
   * REQUIRED          NUMBER (1)
PK_G_ROLEASSOCIATION (GUID)
AK_ROLEASSOCIATION (ROLEGUID, ASSOCIATIONGUID)
FK_ROLEASSOC_ROLE (ROLEGUID)
AK_ROLEASSOCIATION (ROLEGUID, ASSOCIATIONGUID)
```

```
SITE.REPOSITORYADDIN
PF * REPOSITORYGUID    VARCHAR2 (75 BYTE)
P  * ADDINNAME         VARCHAR2 (255 BYTE)
PK_REPOSITORYADDIN (REPOSITORYGUID, ADDINNAME)
FK_REPOSITORYADDIN_REPGUID (REPOSITORYGUID)
```

```
SITE.G_ROLEMEMBERSHIP
PF * ROLEASSOCIATIONGUID   VARCHAR2 (75 BYTE)
P  * MEMBERGUID            VARCHAR2 (75 BYTE)
PK_G_ROLEMEMBERSHIP (ROLEASSOCIATIONGUID, MEMBERGUID)
FK_ROLEMEM_ROLEASSOC (ROLEASSOCIATIONGUID)
```

Appendix C: Entity Relationship Diagrams



```
SITE.AFFINITY
P  * SITEREPOSITORYGUID     VARCHAR2 (75 BYTE)
P  * HOMEREPOSITORYGUID     VARCHAR2 (75 BYTE)
P  * HOMEFOLDERGUID         VARCHAR2 (75 BYTE)
P  * OBJECTGUID             VARCHAR2 (75 BYTE)
PK_AFFINITY (SITEREPOSITORYGUID, HOMEREPOSITORYGUID, HOMEFOLDERGUID, OBJECTGUID)
IDX_AFFINITY_OBJECTGUID (OBJECTGUID)
```

```
SITE.AUDITHISTORY
P  * AUDITHISTORYID                VARCHAR2 (75 BYTE)
   * USERID                        VARCHAR2 (75 BYTE)
   * VAULTID                       VARCHAR2 (75 BYTE)
     SIGNATUREPOLICYID             VARCHAR2 (75 BYTE)
     SIGPOLICYVERSIONID            VARCHAR2 (75 BYTE)
     WORKFLOWDEFINITIONID          VARCHAR2 (75 BYTE)
     SIGNATUREID                   VARCHAR2 (75 BYTE)
     SIGNATUREUSERNAME             VARCHAR2 (4000 BYTE)
     SIGNATURESOURCE               VARCHAR2 (4000 BYTE)
     CLIENTIPADDRESS               VARCHAR2 (4000 BYTE)
     CLIENTTIMESTAMP               TIMESTAMP (7) WITH TIME ZONE
   * CREATIONDATE                  TIMESTAMP (7) WITH TIME ZONE
     DESCRIPTION                   VARCHAR2 (4000 BYTE)
     CONTENT                       BLOB
   * EVENTTYPE                     VARCHAR2 (255 BYTE)
     EVENTNAME                     VARCHAR2 (255 BYTE)
     WORKFLOWSTAGE                 VARCHAR2 (255 BYTE)
   * VERSION                       NUMBER (9)
     CONTEXT                       VARCHAR2 (4000 BYTE)
     CONTENTVERSIONID              VARCHAR2 (75 BYTE)
     STATUS                        VARCHAR2 (75 BYTE)
     MEANING                       VARCHAR2 (4000 BYTE)
     REASON                        VARCHAR2 (4000 BYTE)
     COMMENTS                      VARCHAR2 (4000 BYTE)
     WFSEQUENCE                    NUMBER (38)
     USERFULLNAME                  VARCHAR2 (4000 BYTE)
     PARTICIPANTGUID               VARCHAR2 (75 BYTE)
     LINK_ADDRESS                  VARCHAR2 (4000 BYTE)
     LINK_NAME                     VARCHAR2 (255 BYTE)
     FULLDESCRIPTION               CLOB
     DESCRIPTION_MIGRATION_STATUS  VARCHAR2 (1 BYTE)
PK_AUDITHISTORY (AUDITHISTORYID)
PK_AUDITHISTORY (AUDITHISTORYID)
IDX_AUDITHIST_OBJ_TYPE (VAULTID, EVENTTYPE)
IDX_AUDITHIST_PART_TYPE (PARTICIPANTGUID, EVENTTYPE)
IDX_AUDITHIST_CONTEXT (CONTEXT)
IDX_AUDITHIST_VERSIONID (CONTENTVERSIONID)
IDX_AUDITHIST_DESTMIG (DESCRIPTION_MIGRATION_STATUS,1)
IDX_AUDITHIST_FULLDESC (FULLDESCRIPTION)
```

```
SITE.EVENTHANDLER
P  * GUID   VARCHAR2 (75 BYTE)
   * TYPE   VARCHAR2 (255 BYTE)
     NAME   VARCHAR2 (255 BYTE)
PK_EVENTHANDLER (GUID)
PK_EVENTHANDLER (GUID)
```

```
SITE.EXTENSIONS
P  * EXTENSIONKEY    VARCHAR2 (255 BYTE)
P  * EXTENSIONKIND   VARCHAR2 (255 BYTE)
   * DOTNETTYPE      VARCHAR2 (255 BYTE)
PK_EXTENSIONS (EXTENSIONKEY, EXTENSIONKIND)
PK_EXTENSIONS (EXTENSIONKEY, EXTENSIONKIND)
```

```
SITE.EXTERNALOBJECTIDMAP
P  * VAULTID          VARCHAR2 (75 BYTE)
   * PROPERTYVALUE    VARCHAR2 (255 BYTE)
P  * PROPERTYTYPE     VARCHAR2 (255 BYTE)
PK_EXTERNALOBJECTIDMAP (VAULTID, PROPERTYTYPE)
IDX_EXTOBJIDMAP_PROPVALUE (PROPERTYVALUE)
```

```
SITE.GROUPMEMBERSHIP
P  * MEMBER_GUID   VARCHAR2 (75 BYTE)
P  * GROUP_GUID    VARCHAR2 (75 BYTE)
PK_GROUPMEMBERSHIP2 (MEMBER_GUID, GROUP_GUID)
IDX_GROUPMEMBER_GROUP (GROUP_GUID)
```

```
SITE.G_USER
P  * GUID            VARCHAR2 (75 BYTE)
   * USERNAME        VARCHAR2 (1000 BYTE)
     FULLNAME        VARCHAR2 (4000 BYTE)
     EMAILADDRESS    VARCHAR2 (4000 BYTE)
   * ACTIVE          NUMBER (1)
     VAULTPROFILE    BLOB
     SYSTEMPROFILE   BLOB
   * USER_ID         NUMBER (15)
     ALIAS           VARCHAR2 (4000 BYTE)
PK_G_USER (GUID)
PK_G_USER (GUID)
IDXUSERNAMEU (TO_MULTI_BYTE(UPPER("USERNAME")))
```

```
SITE.HUBIDTOVAULTIDMAP
P  * HUBID     VARCHAR2 (75 BYTE)
P  * VAULTID   VARCHAR2 (75 BYTE)
PK_HUBIDTOVAULTIDMAP (HUBID, VAULTID)
PK_HUBIDTOVAULTIDMAP (HUBID, VAULTID)
```

```
SITE.MAPPING
     ID   VARCHAR2 (75 BYTE)
```

```
SITE.REPOSITORYTYPE
P  * GUID                   VARCHAR2 (75 BYTE)
U  * NAME                   VARCHAR2 (255 BYTE)
   * MUTABLE                NUMBER (1)
     REPOSITORYBEHAVIORS    NUMBER (4)
PK_REPOSITORYTYPE (GUID)
UN_REPOSITORYTYPE_NAME (NAME)
PK_REPOSITORYTYPE (GUID)
UN_REPOSITORYTYPE_NAME (NAME)
```

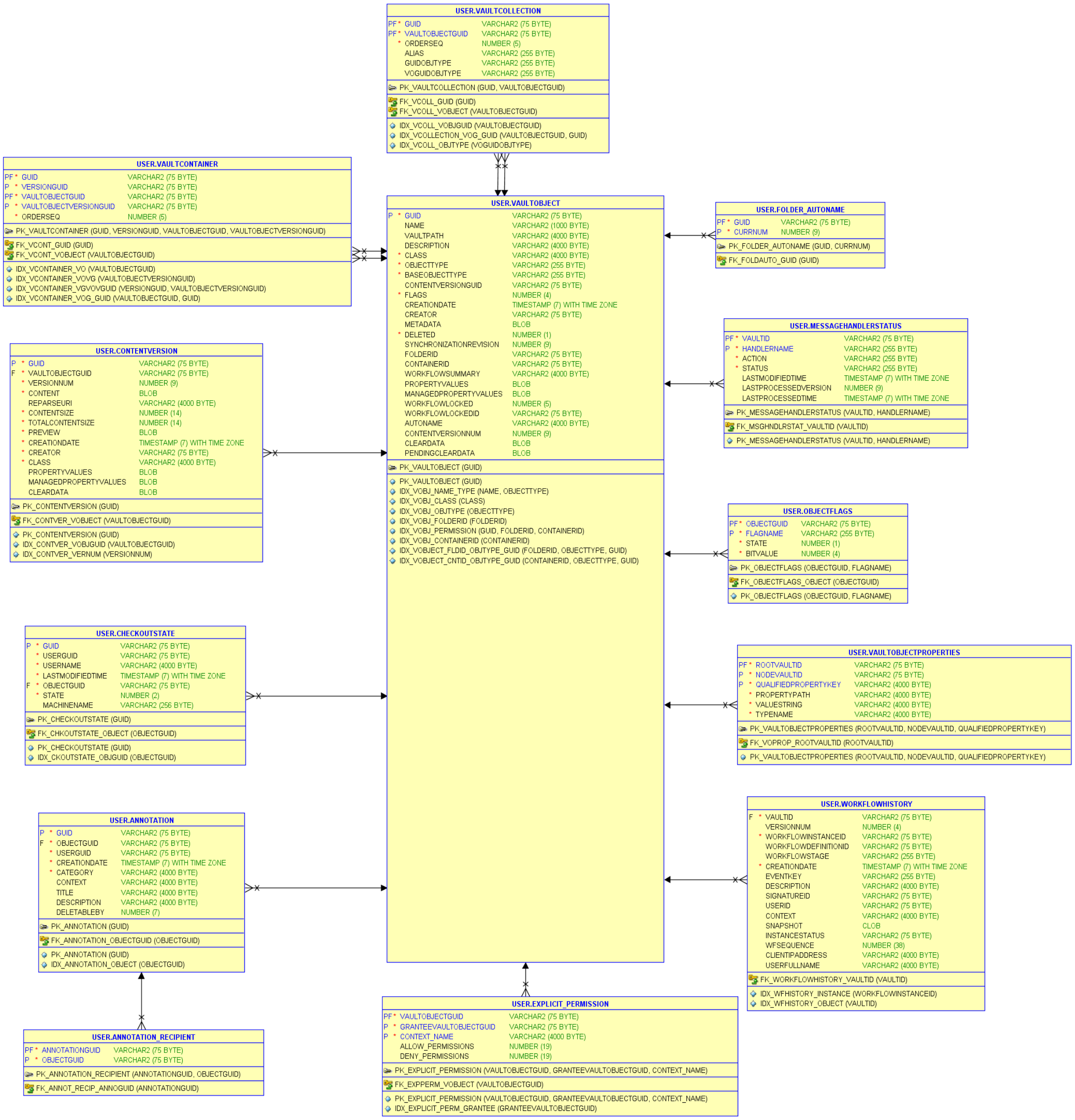
```
SITE.SERVICEKEY
P  * SERVICEID     VARCHAR2 (75 BYTE)
   * SERVICENAME   VARCHAR2 (255 BYTE)
PK_SERVICEKEY (SERVICEID)
```

```
SITE.VAULTCONFIG
P  * NAME    VARCHAR2 (1000 BYTE)
   * ENTRY   VARCHAR2 (4000 BYTE)
PK_VAULTCONFIG (NAME)
PK_VAULTCONFIG (NAME)
```

```
SITE.VAULTCONTEXT
P  * GUID             VARCHAR2 (75 BYTE)
   * TYPE             VARCHAR2 (4000 BYTE)
   * ASSOCIATIONKEY   VARCHAR2 (4000 BYTE)
   * CONTRACTKEY      VARCHAR2 (4000 BYTE)
PK_VAULTCONTEXT (GUID)
PK_VAULTCONTEXT (GUID)
```

```
SITE.WORKFLOWSECURITY
   * WORKFLOWINSTANCEID   VARCHAR2 (75 BYTE)
   * VAULTOBJECTGUID      VARCHAR2 (75 BYTE)
   * ASSIGNEEID           VARCHAR2 (75 BYTE)
     SECURITYSETTINGS     VARCHAR2 (4000 BYTE)
   * LASTUPDATETIME       TIMESTAMP (7) WITH TIME ZONE
IDX_WORKFLOWSEC_VO (VAULTOBJECTGUID)
IDX_WORKFLOWSEC_INSTID (WORKFLOWINSTANCEID)
```

User Entities



USER.VAULTCOLLECTION

PF *	GUID	VARCHAR2 (75 BYTE)
PF *	VAULTOBJECTGUID	VARCHAR2 (75 BYTE)
*	ORDERSEQ	NUMBER (5)
	ALIAS	VARCHAR2 (255 BYTE)
	GUIDOBJTYPE	VARCHAR2 (255 BYTE)
	VOGUIDOBJTYPE	VARCHAR2 (255 BYTE)

PK_VAULTCOLLECTION (GUID, VAULTOBJECTGUID)

FK_VCOLL_GUID (GUID)
FK_VCOLL_VOBJECT (VAULTOBJECTGUID)

IDX_VCOLL_VOBJGUID (VAULTOBJECTGUID)
IDX_VCOLLECTION_VOG_GUID (VAULTOBJECTGUID, GUID)
IDX_VCOLL_OBJTYPE (VOGUIDOBJTYPE)

USER.VAULTCONTAINER

PF *	GUID	VARCHAR2 (75 BYTE)
P *	VERSIONGUID	VARCHAR2 (75 BYTE)
PF *	VAULTOBJECTGUID	VARCHAR2 (75 BYTE)
P *	VAULTOBJECTVERSIONGUID	VARCHAR2 (75 BYTE)
*	ORDERSEQ	NUMBER (5)

PK_VAULTCONTAINER (GUID, VERSIONGUID, VAULTOBJECTGUID, VAULTOBJECTVERSIONGUID)

FK_VCONT_GUID (GUID)
FK_VCONT_VOBJECT (VAULTOBJECTGUID)

IDX_VCONTAINER_VO (VAULTOBJECTGUID)
IDX_VCONTAINER_VOVG (VAULTOBJECTVERSIONGUID)
IDX_VCONTAINER_VGVOVGUID (VERSIONGUID, VAULTOBJECTVERSIONGUID)
IDX_VCONTAINER_VOG_GUID (VAULTOBJECTGUID, GUID)

USER.VAULTOBJECT

P *	GUID	VARCHAR2 (75 BYTE)
	NAME	VARCHAR2 (1000 BYTE)
	VAULTPATH	VARCHAR2 (4000 BYTE)
	DESCRIPTION	VARCHAR2 (4000 BYTE)
*	CLASS	VARCHAR2 (4000 BYTE)
*	OBJECTTYPE	VARCHAR2 (255 BYTE)
*	BASEOBJECTTYPE	VARCHAR2 (255 BYTE)
	CONTENTVERSIONGUID	VARCHAR2 (75 BYTE)
*	FLAGS	NUMBER (4)
	CREATIONDATE	TIMESTAMP (7) WITH TIME ZONE
	CREATOR	VARCHAR2 (75 BYTE)
	METADATA	BLOB
*	DELETED	NUMBER (1)
	SYNCHRONIZATIONREVISION	NUMBER (9)
	FOLDERID	VARCHAR2 (75 BYTE)
	CONTAINERID	VARCHAR2 (75 BYTE)
	WORKFLOWSUMMARY	VARCHAR2 (4000 BYTE)
	PROPERTYVALUES	BLOB
	MANAGEDPROPERTYVALUES	BLOB
	WORKFLOWLOCKED	NUMBER (5)
	WORKFLOWLOCKEDID	VARCHAR2 (75 BYTE)
	AUTONAME	VARCHAR2 (4000 BYTE)
	CONTENTVERSIONNUM	NUMBER (9)
	CLEARDATA	BLOB
	PENDINGCLEARDATA	BLOB

PK_VAULTOBJECT (GUID)

PK_VAULTOBJECT (GUID)
IDX_VOBJ_NAME_TYPE (NAME, OBJECTTYPE)
IDX_VOBJ_CLASS (CLASS)
IDX_VOBJ_OBJTYPE (OBJECTTYPE)
IDX_VOBJ_FOLDERID (FOLDERID)
IDX_VOBJ_PERMISSION (GUID, FOLDERID, CONTAINERID)
IDX_VOBJ_CONTAINERID (CONTAINERID)
IDX_VOBJECT_FLDID_OBJTYPE_GUID (FOLDERID, OBJECTTYPE, GUID)
IDX_VOBJECT_CNTID_OBJTYPE_GUID (CONTAINERID, OBJECTTYPE, GUID)

USER.FOLDER_AUTONAME

PF *	GUID	VARCHAR2 (75 BYTE)
P *	CURRNUM	NUMBER (9)

PK_FOLDER_AUTONAME (GUID, CURRNUM)

FK_FOLDAUTO_GUID (GUID)

USER.CONTENTVERSION

P *	GUID	VARCHAR2 (75 BYTE)
F *	VAULTOBJECTGUID	VARCHAR2 (75 BYTE)
*	VERSIONNUM	NUMBER (9)
*	CONTENT	BLOB
	REPARSEURI	VARCHAR2 (4000 BYTE)
*	CONTENTSIZE	NUMBER (14)
*	TOTALCONTENTSIZE	NUMBER (14)
*	PREVIEW	BLOB
*	CREATIONDATE	TIMESTAMP (7) WITH TIME ZONE
*	CREATOR	VARCHAR2 (75 BYTE)
*	CLASS	VARCHAR2 (4000 BYTE)
	PROPERTYVALUES	BLOB
	MANAGEDPROPERTYVALUES	BLOB
	CLEARDATA	BLOB

PK_CONTENTVERSION (GUID)

FK_CONTVER_VOBJECT (VAULTOBJECTGUID)

PK_CONTENTVERSION (GUID)
IDX_CONTVER_VOBJGUID (VAULTOBJECTGUID)
IDX_CONTVER_VERNUM (VERSIONNUM)

USER.MESSAGEHANDLERSTATUS

PF *	VAULTID	VARCHAR2 (75 BYTE)
P *	HANDLERNAME	VARCHAR2 (255 BYTE)
*	ACTION	VARCHAR2 (255 BYTE)
*	STATUS	VARCHAR2 (255 BYTE)
	LASTMODIFIEDTIME	TIMESTAMP (7) WITH TIME ZONE
	LASTPROCESSEDVERSION	NUMBER (9)
	LASTPROCESSEDTIME	TIMESTAMP (7) WITH TIME ZONE

PK_MESSAGEHANDLERSTATUS (VAULTID, HANDLERNAME)

FK_MSGHNDLRSTAT_VAULTID (VAULTID)

PK_MESSAGEHANDLERSTATUS (VAULTID, HANDLERNAME)

USER.OBJECTFLAGS

PF *	OBJECTGUID	VARCHAR2 (75 BYTE)
P *	FLAGNAME	VARCHAR2 (255 BYTE)
*	STATE	NUMBER (1)
*	BITVALUE	NUMBER (4)

PK_OBJECTFLAGS (OBJECTGUID, FLAGNAME)

FK_OBJECTFLAGS_OBJECT (OBJECTGUID)

PK_OBJECTFLAGS (OBJECTGUID, FLAGNAME)

USER.CHECKOUTSTATE

P *	GUID	VARCHAR2 (75 BYTE)
*	USERGUID	VARCHAR2 (75 BYTE)
*	USERNAME	VARCHAR2 (4000 BYTE)
*	LASTMODIFIEDTIME	TIMESTAMP (7) WITH TIME ZONE
F *	OBJECTGUID	VARCHAR2 (75 BYTE)
*	STATE	NUMBER (2)
	MACHINENAME	VARCHAR2 (256 BYTE)

PK_CHECKOUTSTATE (GUID)

FK_CHKOUTSTATE_OBJECT (OBJECTGUID)

PK_CHECKOUTSTATE (GUID)
IDX_CKOUTSTATE_OBJGUID (OBJECTGUID)

USER.VAULTOBJECTPROPERTIES

PF *	ROOTVAULTID	VARCHAR2 (75 BYTE)
P *	NODEVAULTID	VARCHAR2 (75 BYTE)
P *	QUALIFIEDPROPERTYKEY	VARCHAR2 (4000 BYTE)
*	PROPERTYPATH	VARCHAR2 (4000 BYTE)
*	VALUESTRING	VARCHAR2 (4000 BYTE)
*	TYPENAME	VARCHAR2 (4000 BYTE)

PK_VAULTOBJECTPROPERTIES (ROOTVAULTID, NODEVAULTID, QUALIFIEDPROPERTYKEY)

FK_VOPROP_ROOTVAULTID (ROOTVAULTID)

PK_VAULTOBJECTPROPERTIES (ROOTVAULTID, NODEVAULTID, QUALIFIEDPROPERTYKEY)

USER.ANNOTATION

P *	GUID	VARCHAR2 (75 BYTE)
F *	OBJECTGUID	VARCHAR2 (75 BYTE)
*	USERGUID	VARCHAR2 (75 BYTE)
*	CREATIONDATE	TIMESTAMP (7) WITH TIME ZONE
*	CATEGORY	VARCHAR2 (4000 BYTE)
	CONTEXT	VARCHAR2 (4000 BYTE)
	TITLE	VARCHAR2 (4000 BYTE)
	DESCRIPTION	VARCHAR2 (4000 BYTE)
	DELETABLEBY	NUMBER (7)

PK_ANNOTATION (GUID)

FK_ANNOTATION_OBJECTGUID (OBJECTGUID)

PK_ANNOTATION (GUID)
IDX_ANNOTATION_OBJECT (OBJECTGUID)

USER.WORKFLOWHISTORY

F *	VAULTID	VARCHAR2 (75 BYTE)
	VERSIONNUM	NUMBER (4)
*	WORKFLOWINSTANCEID	VARCHAR2 (75 BYTE)
	WORKFLOWDEFINITIONID	VARCHAR2 (75 BYTE)
	WORKFLOWSTAGE	VARCHAR2 (255 BYTE)
*	CREATIONDATE	TIMESTAMP (7) WITH TIME ZONE
	EVENTKEY	VARCHAR2 (255 BYTE)
	DESCRIPTION	VARCHAR2 (4000 BYTE)
	SIGNATUREID	VARCHAR2 (75 BYTE)
	USERID	VARCHAR2 (75 BYTE)
	CONTEXT	VARCHAR2 (4000 BYTE)
	SNAPSHOT	CLOB
	INSTANCESTATUS	VARCHAR2 (75 BYTE)
	WFSEQUENCE	NUMBER (38)
	CLIENTIPADDRESS	VARCHAR2 (4000 BYTE)
	USERFULLNAME	VARCHAR2 (4000 BYTE)

FK_WORKFLOWHISTORY_VAULTID (VAULTID)

IDX_WFHISTORY_INSTANCE (WORKFLOWINSTANCEID)
IDX_WFHISTORY_OBJECT (VAULTID)

USER.ANNOTATION_RECIPIENT

PF *	ANNOTATIONGUID	VARCHAR2 (75 BYTE)
P *	OBJECTGUID	VARCHAR2 (75 BYTE)

PK_ANNOTATION_RECIPIENT (ANNOTATIONGUID, OBJECTGUID)

FK_ANNOT_RECIP_ANNOGUID (ANNOTATIONGUID)

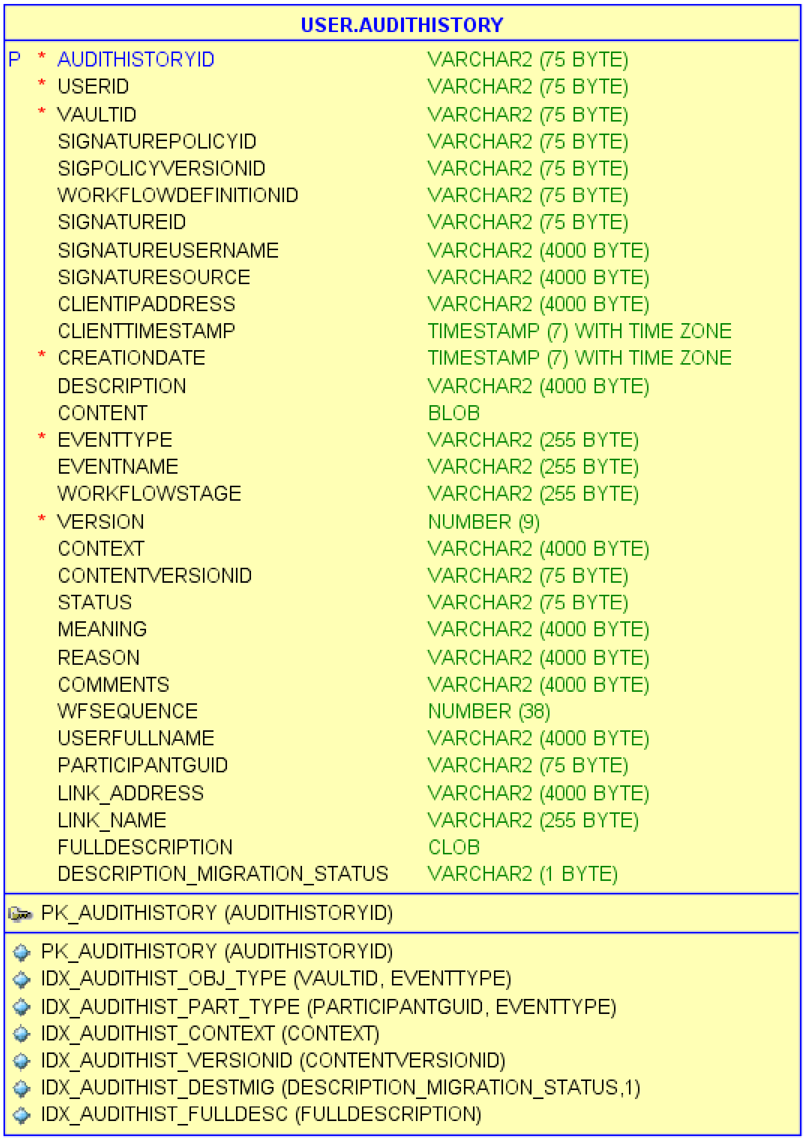
USER.EXPLICIT_PERMISSION

PF *	VAULTOBJECTGUID	VARCHAR2 (75 BYTE)
P *	GRANTEEVAULTOBJECTGUID	VARCHAR2 (75 BYTE)
P *	CONTEXT_NAME	VARCHAR2 (4000 BYTE)
	ALLOW_PERMISSIONS	NUMBER (19)
	DENY_PERMISSIONS	NUMBER (19)

PK_EXPLICIT_PERMISSION (VAULTOBJECTGUID, GRANTEEVAULTOBJECTGUID, CONTEXT_NAME)

FK_EXPPERM_VOBJECT (VAULTOBJECTGUID)

PK_EXPLICIT_PERMISSION (VAULTOBJECTGUID, GRANTEEVAULTOBJECTGUID, CONTEXT_NAME)
IDX_EXPLICIT_PERM_GRANTEE (GRANTEEVAULTOBJECTGUID)



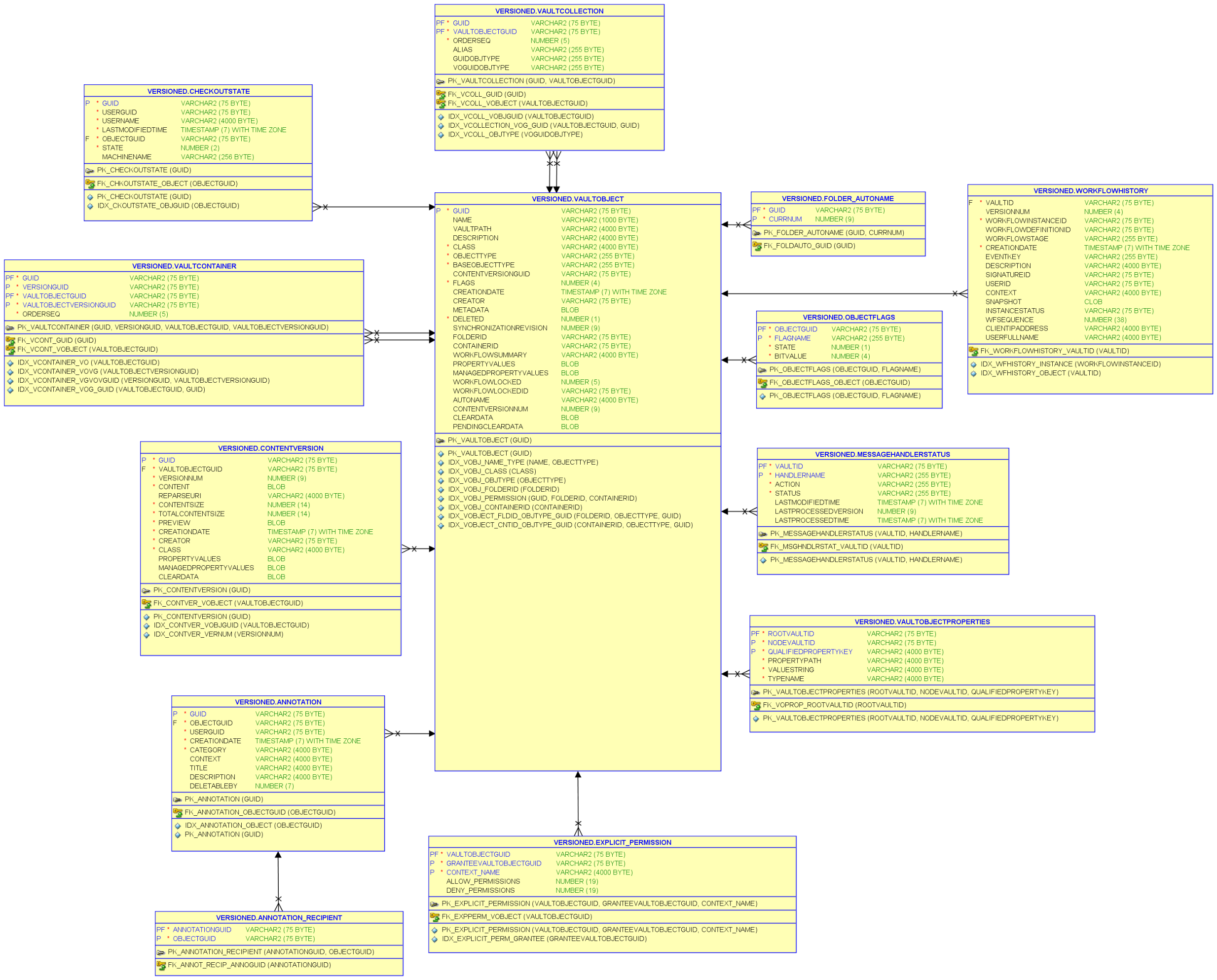
USER.AUDITHISTORY

P	*	AUDITHISTORYID	VARCHAR2 (75 BYTE)
	*	USERID	VARCHAR2 (75 BYTE)
	*	VAULTID	VARCHAR2 (75 BYTE)
		SIGNATUREPOLICYID	VARCHAR2 (75 BYTE)
		SIGPOLICYVERSIONID	VARCHAR2 (75 BYTE)
		WORKFLOWDEFINITIONID	VARCHAR2 (75 BYTE)
		SIGNATUREID	VARCHAR2 (75 BYTE)
		SIGNATUREUSERNAME	VARCHAR2 (4000 BYTE)
		SIGNATURESOURCE	VARCHAR2 (4000 BYTE)
		CLIENTIPADDRESS	VARCHAR2 (4000 BYTE)
		CLIENTTIMESTAMP	TIMESTAMP (7) WITH TIME ZONE
	*	CREATIONDATE	TIMESTAMP (7) WITH TIME ZONE
		DESCRIPTION	VARCHAR2 (4000 BYTE)
		CONTENT	BLOB
	*	EVENTTYPE	VARCHAR2 (255 BYTE)
		EVENTNAME	VARCHAR2 (255 BYTE)
		WORKFLOWSTAGE	VARCHAR2 (255 BYTE)
	*	VERSION	NUMBER (9)
		CONTEXT	VARCHAR2 (4000 BYTE)
		CONTENTVERSIONID	VARCHAR2 (75 BYTE)
		STATUS	VARCHAR2 (75 BYTE)
		MEANING	VARCHAR2 (4000 BYTE)
		REASON	VARCHAR2 (4000 BYTE)
		COMMENTS	VARCHAR2 (4000 BYTE)
		WFSEQUENCE	NUMBER (38)
		USERFULLNAME	VARCHAR2 (4000 BYTE)
		PARTICIPANTGUID	VARCHAR2 (75 BYTE)
		LINK_ADDRESS	VARCHAR2 (4000 BYTE)
		LINK_NAME	VARCHAR2 (255 BYTE)
		FULLDESCRIPTION	CLOB
		DESCRIPTION_MIGRATION_STATUS	VARCHAR2 (1 BYTE)

PK_AUDITHISTORY (AUDITHISTORYID)

PK_AUDITHISTORY (AUDITHISTORYID)
IDX_AUDITHIST_OBJ_TYPE (VAULTID, EVENTTYPE)
IDX_AUDITHIST_PART_TYPE (PARTICIPANTGUID, EVENTTYPE)
IDX_AUDITHIST_CONTEXT (CONTEXT)
IDX_AUDITHIST_VERSIONID (CONTENTVERSIONID)
IDX_AUDITHIST_DESTMIG (DESCRIPTION_MIGRATION_STATUS,1)
IDX_AUDITHIST_FULLDESC (FULLDESCRIPTION)

Versioned Entities



VERSIONED.VAULTCOLLECTION

PF	*	GUID	VARCHAR2 (75 BYTE)
PF	*	VAULTOBJECTGUID	VARCHAR2 (75 BYTE)
	*	ORDERSEQ	NUMBER (5)
		ALIAS	VARCHAR2 (255 BYTE)
		GUIDOBJTYPE	VARCHAR2 (255 BYTE)
		VOGUIDOBJTYPE	VARCHAR2 (255 BYTE)

PK_VAULTCOLLECTION (GUID, VAULTOBJECTGUID)

FK_VCOLL_GUID (GUID)
FK_VCOLL_VOBJECT (VAULTOBJECTGUID)

IDX_VCOLL_VOBJGUID (VAULTOBJECTGUID)
IDX_VCOLLECTION_VOG_GUID (VAULTOBJECTGUID, GUID)
IDX_VCOLL_OBJTYPE (VOGUIDOBJTYPE)

VERSIONED.CHECKOUTSTATE

P	*	GUID	VARCHAR2 (75 BYTE)
	*	USERGUID	VARCHAR2 (75 BYTE)
	*	USERNAME	VARCHAR2 (4000 BYTE)
	*	LASTMODIFIEDTIME	TIMESTAMP (7) WITH TIME ZONE
F	*	OBJECTGUID	VARCHAR2 (75 BYTE)
	*	STATE	NUMBER (2)
		MACHINENAME	VARCHAR2 (256 BYTE)

PK_CHECKOUTSTATE (GUID)

FK_CHKOUTSTATE_OBJECT (OBJECTGUID)

PK_CHECKOUTSTATE (GUID)
IDX_CKOUTSTATE_OBJGUID (OBJECTGUID)

VERSIONED.VAULTOBJECT

P	*	GUID	VARCHAR2 (75 BYTE)
		NAME	VARCHAR2 (1000 BYTE)
		VAULTPATH	VARCHAR2 (4000 BYTE)
		DESCRIPTION	VARCHAR2 (4000 BYTE)
	*	CLASS	VARCHAR2 (4000 BYTE)
	*	OBJECTTYPE	VARCHAR2 (255 BYTE)
	*	BASEOBJECTTYPE	VARCHAR2 (255 BYTE)
		CONTENTVERSIONGUID	VARCHAR2 (75 BYTE)
	*	FLAGS	NUMBER (4)
		CREATIONDATE	TIMESTAMP (7) WITH TIME ZONE
		CREATOR	VARCHAR2 (75 BYTE)
		METADATA	BLOB
	*	DELETED	NUMBER (1)
		SYNCHRONIZATIONREVISION	NUMBER (9)
		FOLDERID	VARCHAR2 (75 BYTE)
		CONTAINERID	VARCHAR2 (75 BYTE)
		WORKFLOWSUMMARY	VARCHAR2 (4000 BYTE)
		PROPERTYVALUES	BLOB
		MANAGEDPROPERTYVALUES	BLOB
		WORKFLOWLOCKED	NUMBER (5)
		WORKFLOWLOCKEDID	VARCHAR2 (75 BYTE)
		AUTONAME	VARCHAR2 (4000 BYTE)
		CONTENTVERSIONNUM	NUMBER (9)
		CLEARDATA	BLOB
		PENDINGCLEARDATA	BLOB

PK_VAULTOBJECT (GUID)

PK_VAULTOBJECT (GUID)
IDX_VOBJ_NAME_TYPE (NAME, OBJECTTYPE)
IDX_VOBJ_CLASS (CLASS)
IDX_VOBJ_OBJTYPE (OBJECTTYPE)
IDX_VOBJ_FOLDERID (FOLDERID)
IDX_VOBJ_PERMISSION (GUID, FOLDERID, CONTAINERID)
IDX_VOBJ_CONTAINERID (CONTAINERID)
IDX_VOBJECT_FLDID_OBJTYPE_GUID (FOLDERID, OBJECTTYPE, GUID)
IDX_VOBJECT_CNTID_OBJTYPE_GUID (CONTAINERID, OBJECTTYPE, GUID)

VERSIONED.FOLDER_AUTONAME

PF	*	GUID	VARCHAR2 (75 BYTE)
P	*	CURRNUM	NUMBER (9)

PK_FOLDER_AUTONAME (GUID, CURRNUM)

FK_FOLDAUTO_GUID (GUID)

VERSIONED.WORKFLOWHISTORY

F	*	VAULTID	VARCHAR2 (75 BYTE)
		VERSIONNUM	NUMBER (4)
	*	WORKFLOWINSTANCEID	VARCHAR2 (75 BYTE)
		WORKFLOWDEFINITIONID	VARCHAR2 (75 BYTE)
		WORKFLOWSTAGE	VARCHAR2 (255 BYTE)
	*	CREATIONDATE	TIMESTAMP (7) WITH TIME ZONE
		EVENTKEY	VARCHAR2 (255 BYTE)
		DESCRIPTION	VARCHAR2 (4000 BYTE)
		SIGNATUREID	VARCHAR2 (75 BYTE)
		USERID	VARCHAR2 (75 BYTE)
		CONTEXT	VARCHAR2 (4000 BYTE)
		SNAPSHOT	CLOB
		INSTANCESTATUS	VARCHAR2 (75 BYTE)
		WFSEQUENCE	NUMBER (38)
		CLIENTIPADDRESS	VARCHAR2 (4000 BYTE)
		USERFULLNAME	VARCHAR2 (4000 BYTE)

FK_WORKFLOWHISTORY_VAULTID (VAULTID)

IDX_WFHISTORY_INSTANCE (WORKFLOWINSTANCEID)
IDX_WFHISTORY_OBJECT (VAULTID)

VERSIONED.VAULTCONTAINER

PF	*	GUID	VARCHAR2 (75 BYTE)
P	*	VERSIONGUID	VARCHAR2 (75 BYTE)
PF	*	VAULTOBJECTGUID	VARCHAR2 (75 BYTE)
P	*	VAULTOBJECTVERSIONGUID	VARCHAR2 (75 BYTE)
	*	ORDERSEQ	NUMBER (5)

PK_VAULTCONTAINER (GUID, VERSIONGUID, VAULTOBJECTGUID, VAULTOBJECTVERSIONGUID)

FK_VCONT_GUID (GUID)
FK_VCONT_VOBJECT (VAULTOBJECTGUID)

IDX_VCONTAINER_VO (VAULTOBJECTGUID)
IDX_VCONTAINER_VOVG (VAULTOBJECTVERSIONGUID)
IDX_VCONTAINER_VGVOVGUID (VERSIONGUID, VAULTOBJECTVERSIONGUID)
IDX_VCONTAINER_VOG_GUID (VAULTOBJECTGUID, GUID)

VERSIONED.OBJECTFLAGS

PF	*	OBJECTGUID	VARCHAR2 (75 BYTE)
P	*	FLAGNAME	VARCHAR2 (255 BYTE)
	*	STATE	NUMBER (1)
	*	BITVALUE	NUMBER (4)

PK_OBJECTFLAGS (OBJECTGUID, FLAGNAME)

FK_OBJECTFLAGS_OBJECT (OBJECTGUID)

PK_OBJECTFLAGS (OBJECTGUID, FLAGNAME)

VERSIONED.CONTENTVERSION

P	*	GUID	VARCHAR2 (75 BYTE)
F	*	VAULTOBJECTGUID	VARCHAR2 (75 BYTE)
	*	VERSIONNUM	NUMBER (9)
	*	CONTENT	BLOB
		REPARSEURI	VARCHAR2 (4000 BYTE)
	*	CONTENTSIZE	NUMBER (14)
	*	TOTALCONTENTSIZE	NUMBER (14)
	*	PREVIEW	BLOB
	*	CREATIONDATE	TIMESTAMP (7) WITH TIME ZONE
	*	CREATOR	VARCHAR2 (75 BYTE)
	*	CLASS	VARCHAR2 (4000 BYTE)
		PROPERTYVALUES	BLOB
		MANAGEDPROPERTYVALUES	BLOB
		CLEARDATA	BLOB

PK_CONTENTVERSION (GUID)

FK_CONTVER_VOBJECT (VAULTOBJECTGUID)

PK_CONTENTVERSION (GUID)
IDX_CONTVER_VOBJGUID (VAULTOBJECTGUID)
IDX_CONTVER_VERNUM (VERSIONNUM)

VERSIONED.MESSAGEHANDLERSTATUS

PF	*	VAULTID	VARCHAR2 (75 BYTE)
P	*	HANDLERNAME	VARCHAR2 (255 BYTE)
	*	ACTION	VARCHAR2 (255 BYTE)
	*	STATUS	VARCHAR2 (255 BYTE)
		LASTMODIFIEDTIME	TIMESTAMP (7) WITH TIME ZONE
		LASTPROCESSEDVERSION	NUMBER (9)
		LASTPROCESSEDTIME	TIMESTAMP (7) WITH TIME ZONE

PK_MESSAGEHANDLERSTATUS (VAULTID, HANDLERNAME)

FK_MSGHNDLRSTAT_VAULTID (VAULTID)

PK_MESSAGEHANDLERSTATUS (VAULTID, HANDLERNAME)

VERSIONED.VAULTOBJECTPROPERTIES

PF	*	ROOTVAULTID	VARCHAR2 (75 BYTE)
P	*	NODEVAULTID	VARCHAR2 (75 BYTE)
P	*	QUALIFIEDPROPERTYKEY	VARCHAR2 (4000 BYTE)
	*	PROPERTYPATH	VARCHAR2 (4000 BYTE)
	*	VALUESTRING	VARCHAR2 (4000 BYTE)
	*	TYPENAME	VARCHAR2 (4000 BYTE)

PK_VAULTOBJECTPROPERTIES (ROOTVAULTID, NODEVAULTID, QUALIFIEDPROPERTYKEY)

FK_VOPROP_ROOTVAULTID (ROOTVAULTID)

PK_VAULTOBJECTPROPERTIES (ROOTVAULTID, NODEVAULTID, QUALIFIEDPROPERTYKEY)

VERSIONED.ANNOTATION

P	*	GUID	VARCHAR2 (75 BYTE)
F	*	OBJECTGUID	VARCHAR2 (75 BYTE)
	*	USERGUID	VARCHAR2 (75 BYTE)
	*	CREATIONDATE	TIMESTAMP (7) WITH TIME ZONE
	*	CATEGORY	VARCHAR2 (4000 BYTE)
		CONTEXT	VARCHAR2 (4000 BYTE)
		TITLE	VARCHAR2 (4000 BYTE)
		DESCRIPTION	VARCHAR2 (4000 BYTE)
		DELETABLEBY	NUMBER (7)

PK_ANNOTATION (GUID)

FK_ANNOTATION_OBJECTGUID (OBJECTGUID)

IDX_ANNOTATION_OBJECT (OBJECTGUID)
PK_ANNOTATION (GUID)

VERSIONED.EXPLICIT_PERMISSION

PF	*	VAULTOBJECTGUID	VARCHAR2 (75 BYTE)
P	*	GRANTEEVAULTOBJECTGUID	VARCHAR2 (75 BYTE)
P	*	CONTEXT_NAME	VARCHAR2 (4000 BYTE)
		ALLOW_PERMISSIONS	NUMBER (19)
		DENY_PERMISSIONS	NUMBER (19)

PK_EXPLICIT_PERMISSION (VAULTOBJECTGUID, GRANTEEVAULTOBJECTGUID, CONTEXT_NAME)

FK_EXPPERM_VOBJECT (VAULTOBJECTGUID)

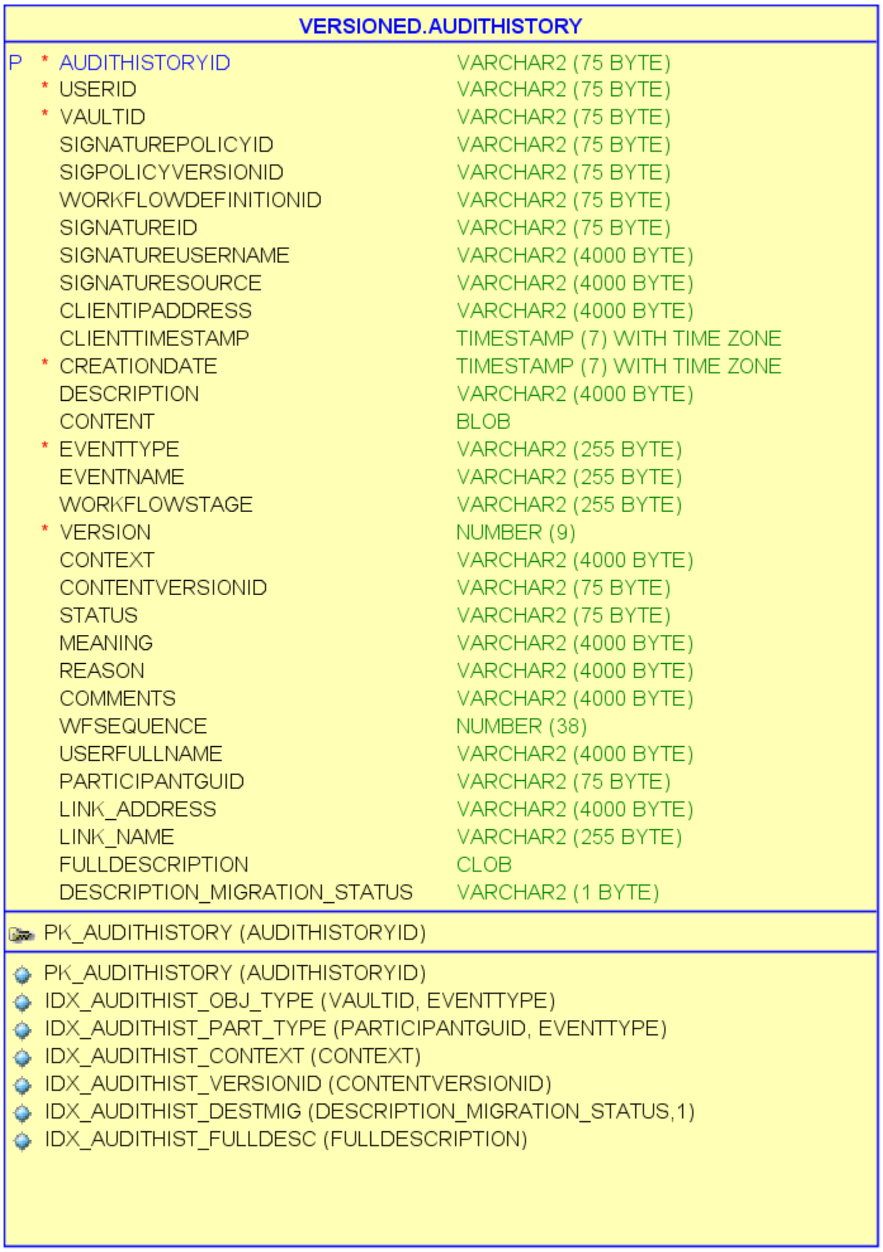
PK_EXPLICIT_PERMISSION (VAULTOBJECTGUID, GRANTEEVAULTOBJECTGUID, CONTEXT_NAME)
IDX_EXPLICIT_PERM_GRANTEE (GRANTEEVAULTOBJECTGUID)

VERSIONED.ANNOTATION_RECIPIENT

PF	*	ANNOTATIONGUID	VARCHAR2 (75 BYTE)
P	*	OBJECTGUID	VARCHAR2 (75 BYTE)

PK_ANNOTATION_RECIPIENT (ANNOTATIONGUID, OBJECTGUID)

FK_ANNOT_RECIP_ANNOGUID (ANNOTATIONGUID)



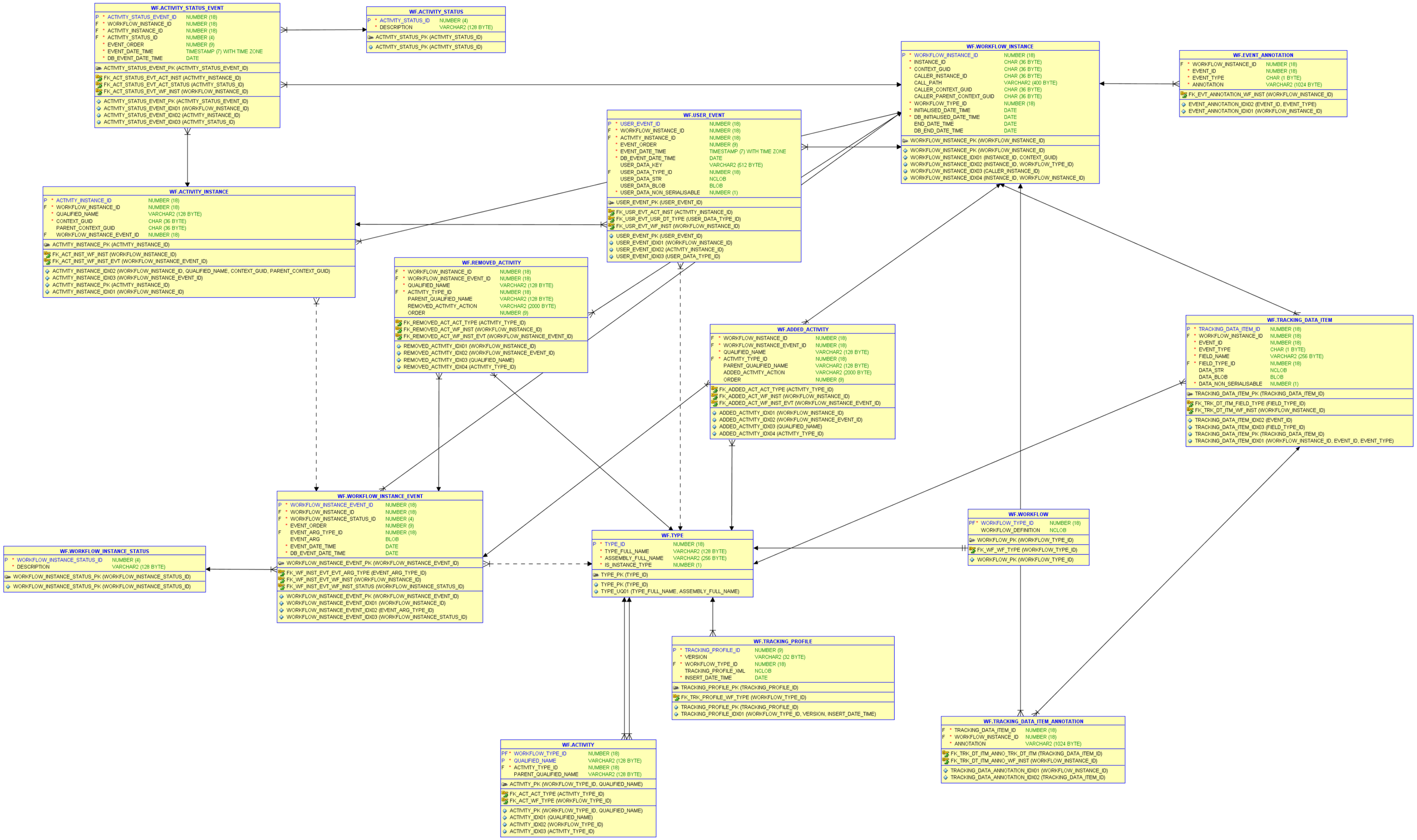
VERSIONED.AUDITHISTORY

P	*	AUDITHISTORYID	VARCHAR2 (75 BYTE)
	*	USERID	VARCHAR2 (75 BYTE)
	*	VAULTID	VARCHAR2 (75 BYTE)
		SIGNATUREPOLICYID	VARCHAR2 (75 BYTE)
		SIGPOLICYVERSIONID	VARCHAR2 (75 BYTE)
		WORKFLOWDEFINITIONID	VARCHAR2 (75 BYTE)
		SIGNATUREID	VARCHAR2 (75 BYTE)
		SIGNATUREUSERNAME	VARCHAR2 (4000 BYTE)
		SIGNATURESOURCE	VARCHAR2 (4000 BYTE)
		CLIENTIPADDRESS	VARCHAR2 (4000 BYTE)
		CLIENTTIMESTAMP	TIMESTAMP (7) WITH TIME ZONE
	*	CREATIONDATE	TIMESTAMP (7) WITH TIME ZONE
		DESCRIPTION	VARCHAR2 (4000 BYTE)
		CONTENT	BLOB
	*	EVENTTYPE	VARCHAR2 (255 BYTE)
		EVENTNAME	VARCHAR2 (255 BYTE)
		WORKFLOWSTAGE	VARCHAR2 (255 BYTE)
	*	VERSION	NUMBER (9)
		CONTEXT	VARCHAR2 (4000 BYTE)
		CONTENTVERSIONID	VARCHAR2 (75 BYTE)
		STATUS	VARCHAR2 (75 BYTE)
		MEANING	VARCHAR2 (4000 BYTE)
		REASON	VARCHAR2 (4000 BYTE)
		COMMENTS	VARCHAR2 (4000 BYTE)
		WFSEQUENCE	NUMBER (38)
		USERFULLNAME	VARCHAR2 (4000 BYTE)
		PARTICIPANTGUID	VARCHAR2 (75 BYTE)
		LINK_ADDRESS	VARCHAR2 (4000 BYTE)
		LINK_NAME	VARCHAR2 (255 BYTE)
		FULLDESCRIPTION	CLOB
		DESCRIPTION_MIGRATION_STATUS	VARCHAR2 (1 BYTE)

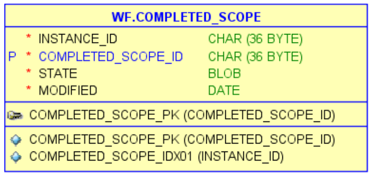
PK_AUDITHISTORY (AUDITHISTORYID)

- PK_AUDITHISTORY (AUDITHISTORYID)
- IDX_AUDITHIST_OBJ_TYPE (VAULTID, EVENTTYPE)
- IDX_AUDITHIST_PART_TYPE (PARTICIPANTGUID, EVENTTYPE)
- IDX_AUDITHIST_CONTEXT (CONTEXT)
- IDX_AUDITHIST_VERSIONID (CONTENTVERSIONID)
- IDX_AUDITHIST_DESTMIG (DESCRIPTION_MIGRATION_STATUS,1)
- IDX_AUDITHIST_FULLDESC (FULLDESCRIPTION)

Worfklow Entities


WF.ACTIVITY_STATUS_EVENT
P * ACTIVITY_STATUS_EVENT_ID NUMBER (18)
F * WORKFLOW_INSTANCE_ID NUMBER (18)
F * ACTIVITY_INSTANCE_ID NUMBER (18)
F * ACTIVITY_STATUS_ID NUMBER (4)
* EVENT_ORDER NUMBER (9)
* EVENT_DATE_TIME TIMESTAMP (7) WITH TIME ZONE
* DB_EVENT_DATE_TIME DATE
ACTIVITY_STATUS_EVENT_PK (ACTIVITY_STATUS_EVENT_ID)
FK_ACT_STATUS_EVT_ACT_INST (ACTIVITY_INSTANCE_ID)
FK_ACT_STATUS_EVT_ACT_STATUS (ACTIVITY_STATUS_ID)
FK_ACT_STATUS_EVT_WF_INST (WORKFLOW_INSTANCE_ID)
ACTIVITY_STATUS_EVENT_PK (ACTIVITY_STATUS_EVENT_ID)
ACTIVITY_STATUS_EVENT_IDX01 (WORKFLOW_INSTANCE_ID)
ACTIVITY_STATUS_EVENT_IDX02 (ACTIVITY_INSTANCE_ID)
ACTIVITY_STATUS_EVENT_IDX03 (ACTIVITY_STATUS_ID)
WF.ACTIVITY_STATUS
P * ACTIVITY_STATUS_ID NUMBER (4)
* DESCRIPTION VARCHAR2 (128 BYTE)
ACTIVITY_STATUS_PK (ACTIVITY_STATUS_ID)
ACTIVITY_STATUS_PK (ACTIVITY_STATUS_ID)
WF.WORKFLOW_INSTANCE
P * WORKFLOW_INSTANCE_ID NUMBER (18)
* INSTANCE_ID CHAR (36 BYTE)
* CONTEXT_GUID CHAR (36 BYTE)
CALLER_INSTANCE_ID CHAR (36 BYTE)
CALL_PATH VARCHAR2 (400 BYTE)
CALLER_CONTEXT_GUID CHAR (36 BYTE)
CALLER_PARENT_CONTEXT_GUID CHAR (36 BYTE)
* WORKFLOW_TYPE_ID NUMBER (18)
* INITIALISED_DATE_TIME DATE
* DB_INITIALISED_DATE_TIME DATE
END_DATE_TIME DATE
DB_END_DATE_TIME DATE
WORKFLOW_INSTANCE_PK (WORKFLOW_INSTANCE_ID)
WORKFLOW_INSTANCE_PK (WORKFLOW_INSTANCE_ID)
WORKFLOW_INSTANCE_IDX01 (INSTANCE_ID, CONTEXT_GUID)
WORKFLOW_INSTANCE_IDX02 (INSTANCE_ID, WORKFLOW_TYPE_ID)
WORKFLOW_INSTANCE_IDX03 (CALLER_INSTANCE_ID)
WORKFLOW_INSTANCE_IDX04 (INSTANCE_ID, WORKFLOW_INSTANCE_ID)
WF.EVENT_ANNOTATION
F * WORKFLOW_INSTANCE_ID NUMBER (18)
* EVENT_ID NUMBER (18)
* EVENT_TYPE CHAR (1 BYTE)
* ANNOTATION VARCHAR2 (1024 BYTE)
FK_EVT_ANNOTATION_WF_INST (WORKFLOW_INSTANCE_ID)
EVENT_ANNOTATION_IDX02 (EVENT_ID, EVENT_TYPE)
EVENT_ANNOTATION_IDX01 (WORKFLOW_INSTANCE_ID)
WF.USER_EVENT
P * USER_EVENT_ID NUMBER (18)
F * WORKFLOW_INSTANCE_ID NUMBER (18)
F * ACTIVITY_INSTANCE_ID NUMBER (18)
* EVENT_ORDER NUMBER (9)
* EVENT_DATE_TIME TIMESTAMP (7) WITH TIME ZONE
* DB_EVENT_DATE_TIME DATE
USER_DATA_KEY VARCHAR2 (512 BYTE)
F USER_DATA_TYPE_ID NUMBER (18)
USER_DATA_STR NCLOB
USER_DATA_BLOB BLOB
* USER_DATA_NON_SERIALISABLE NUMBER (1)
USER_EVENT_PK (USER_EVENT_ID)
FK_USR_EVT_ACT_INST (ACTIVITY_INSTANCE_ID)
FK_USR_EVT_USR_DT_TYPE (USER_DATA_TYPE_ID)
FK_USR_EVT_WF_INST (WORKFLOW_INSTANCE_ID)
USER_EVENT_PK (USER_EVENT_ID)
USER_EVENT_IDX01 (WORKFLOW_INSTANCE_ID)
USER_EVENT_IDX02 (ACTIVITY_INSTANCE_ID)
USER_EVENT_IDX03 (USER_DATA_TYPE_ID)
WF.ACTIVITY_INSTANCE
P * ACTIVITY_INSTANCE_ID NUMBER (18)
F * WORKFLOW_INSTANCE_ID NUMBER (18)
* QUALIFIED_NAME VARCHAR2 (128 BYTE)
* CONTEXT_GUID CHAR (36 BYTE)
PARENT_CONTEXT_GUID CHAR (36 BYTE)
F WORKFLOW_INSTANCE_EVENT_ID NUMBER (18)
ACTIVITY_INSTANCE_PK (ACTIVITY_INSTANCE_ID)
FK_ACT_INST_WF_INST (WORKFLOW_INSTANCE_ID)
FK_ACT_INST_WF_INST_EVT (WORKFLOW_INSTANCE_EVENT_ID)
ACTIVITY_INSTANCE_IDX02 (WORKFLOW_INSTANCE_ID, QUALIFIED_NAME, CONTEXT_GUID, PARENT_CONTEXT_GUID)
ACTIVITY_INSTANCE_IDX03 (WORKFLOW_INSTANCE_EVENT_ID)
ACTIVITY_INSTANCE_PK (ACTIVITY_INSTANCE_ID)
ACTIVITY_INSTANCE_IDX01 (WORKFLOW_INSTANCE_ID)
WF.REMOVED_ACTIVITY
F * WORKFLOW_INSTANCE_ID NUMBER (18)
F * WORKFLOW_INSTANCE_EVENT_ID NUMBER (18)
* QUALIFIED_NAME VARCHAR2 (128 BYTE)
F * ACTIVITY_TYPE_ID NUMBER (18)
PARENT_QUALIFIED_NAME VARCHAR2 (128 BYTE)
REMOVED_ACTIVITY_ACTION VARCHAR2 (2000 BYTE)
ORDER NUMBER (9)
FK_REMOVED_ACT_ACT_TYPE (ACTIVITY_TYPE_ID)
FK_REMOVED_ACT_WF_INST (WORKFLOW_INSTANCE_ID)
FK_REMOVED_ACT_WF_INST_EVT (WORKFLOW_INSTANCE_EVENT_ID)
REMOVED_ACTIVITY_IDX01 (WORKFLOW_INSTANCE_ID)
REMOVED_ACTIVITY_IDX02 (WORKFLOW_INSTANCE_EVENT_ID)
REMOVED_ACTIVITY_IDX03 (QUALIFIED_NAME)
REMOVED_ACTIVITY_IDX04 (ACTIVITY_TYPE_ID)
WF.ADDED_ACTIVITY
F * WORKFLOW_INSTANCE_ID NUMBER (18)
F * WORKFLOW_INSTANCE_EVENT_ID NUMBER (18)
* QUALIFIED_NAME VARCHAR2 (128 BYTE)
F * ACTIVITY_TYPE_ID NUMBER (18)
PARENT_QUALIFIED_NAME VARCHAR2 (128 BYTE)
ADDED_ACTIVITY_ACTION VARCHAR2 (2000 BYTE)
ORDER NUMBER (9)
FK_ADDED_ACT_ACT_TYPE (ACTIVITY_TYPE_ID)
FK_ADDED_ACT_WF_INST (WORKFLOW_INSTANCE_ID)
FK_ADDED_ACT_WF_INST_EVT (WORKFLOW_INSTANCE_EVENT_ID)
ADDED_ACTIVITY_IDX01 (WORKFLOW_INSTANCE_ID)
ADDED_ACTIVITY_IDX02 (WORKFLOW_INSTANCE_EVENT_ID)
ADDED_ACTIVITY_IDX03 (QUALIFIED_NAME)
ADDED_ACTIVITY_IDX04 (ACTIVITY_TYPE_ID)
WF.TRACKING_DATA_ITEM
P * TRACKING_DATA_ITEM_ID NUMBER (18)
F * WORKFLOW_INSTANCE_ID NUMBER (18)
* EVENT_ID NUMBER (18)
* EVENT_TYPE CHAR (1 BYTE)
* FIELD_NAME VARCHAR2 (256 BYTE)
F * FIELD_TYPE_ID NUMBER (18)
DATA_STR NCLOB
DATA_BLOB BLOB
* DATA_NON_SERIALISABLE NUMBER (1)
TRACKING_DATA_ITEM_PK (TRACKING_DATA_ITEM_ID)
FK_TRK_DT_ITM_FIELD_TYPE (FIELD_TYPE_ID)
FK_TRK_DT_ITM_WF_INST (WORKFLOW_INSTANCE_ID)
TRACKING_DATA_ITEM_IDX02 (EVENT_ID)
TRACKING_DATA_ITEM_IDX03 (FIELD_TYPE_ID)
TRACKING_DATA_ITEM_PK (TRACKING_DATA_ITEM_ID)
TRACKING_DATA_ITEM_IDX01 (WORKFLOW_INSTANCE_ID, EVENT_ID, EVENT_TYPE)
WF.WORKFLOW_INSTANCE_EVENT
P * WORKFLOW_INSTANCE_EVENT_ID NUMBER (18)
F * WORKFLOW_INSTANCE_ID NUMBER (18)
F * WORKFLOW_INSTANCE_STATUS_ID NUMBER (4)
* EVENT_ORDER NUMBER (9)
F EVENT_ARG_TYPE_ID NUMBER (18)
EVENT_ARG BLOB
* EVENT_DATE_TIME DATE
* DB_EVENT_DATE_TIME DATE
WORKFLOW_INSTANCE_EVENT_PK (WORKFLOW_INSTANCE_EVENT_ID)
FK_WF_INST_EVT_EVT_ARG_TYPE (EVENT_ARG_TYPE_ID)
FK_WF_INST_EVT_WF_INST (WORKFLOW_INSTANCE_ID)
FK_WF_INST_EVT_WF_INST_STATUS (WORKFLOW_INSTANCE_STATUS_ID)
WORKFLOW_INSTANCE_EVENT_PK (WORKFLOW_INSTANCE_EVENT_ID)
WORKFLOW_INSTANCE_EVENT_IDX01 (WORKFLOW_INSTANCE_ID)
WORKFLOW_INSTANCE_EVENT_IDX02 (EVENT_ARG_TYPE_ID)
WORKFLOW_INSTANCE_EVENT_IDX03 (WORKFLOW_INSTANCE_STATUS_ID)
WF.WORKFLOW_INSTANCE_STATUS
P * WORKFLOW_INSTANCE_STATUS_ID NUMBER (4)
* DESCRIPTION VARCHAR2 (128 BYTE)
WORKFLOW_INSTANCE_STATUS_PK (WORKFLOW_INSTANCE_STATUS_ID)
WORKFLOW_INSTANCE_STATUS_PK (WORKFLOW_INSTANCE_STATUS_ID)
WF.WORKFLOW
PF * WORKFLOW_TYPE_ID NUMBER (18)
WORKFLOW_DEFINITION NCLOB
WORKFLOW_PK (WORKFLOW_TYPE_ID)
FK_WF_WF_TYPE (WORKFLOW_TYPE_ID)
WORKFLOW_PK (WORKFLOW_TYPE_ID)
WF.TYPE
P * TYPE_ID NUMBER (18)
* TYPE_FULL_NAME VARCHAR2 (128 BYTE)
* ASSEMBLY_FULL_NAME VARCHAR2 (256 BYTE)
* IS_INSTANCE_TYPE NUMBER (1)
TYPE_PK (TYPE_ID)
TYPE_PK (TYPE_ID)
TYPE_UQ01 (TYPE_FULL_NAME, ASSEMBLY_FULL_NAME)
WF.TRACKING_PROFILE
P * TRACKING_PROFILE_ID NUMBER (9)
* VERSION VARCHAR2 (32 BYTE)
F * WORKFLOW_TYPE_ID NUMBER (18)
TRACKING_PROFILE_XML NCLOB
* INSERT_DATE_TIME DATE
TRACKING_PROFILE_PK (TRACKING_PROFILE_ID)
FK_TRK_PROFILE_WF_TYPE (WORKFLOW_TYPE_ID)
TRACKING_PROFILE_PK (TRACKING_PROFILE_ID)
TRACKING_PROFILE_IDX01 (WORKFLOW_TYPE_ID, VERSION, INSERT_DATE_TIME)
WF.TRACKING_DATA_ITEM_ANNOTATION
F * TRACKING_DATA_ITEM_ID NUMBER (18)
F * WORKFLOW_INSTANCE_ID NUMBER (18)
* ANNOTATION VARCHAR2 (1024 BYTE)
FK_TRK_DT_ITM_ANNO_TRK_DT_ITM (TRACKING_DATA_ITEM_ID)
FK_TRK_DT_ITM_ANNO_WF_INST (WORKFLOW_INSTANCE_ID)
TRACKING_DATA_ANNOTATION_IDX01 (WORKFLOW_INSTANCE_ID)
TRACKING_DATA_ANNOTATION_IDX02 (TRACKING_DATA_ITEM_ID)
WF.ACTIVITY
PF * WORKFLOW_TYPE_ID NUMBER (18)
P * QUALIFIED_NAME VARCHAR2 (128 BYTE)
F * ACTIVITY_TYPE_ID NUMBER (18)
PARENT_QUALIFIED_NAME VARCHAR2 (128 BYTE)
ACTIVITY_PK (WORKFLOW_TYPE_ID, QUALIFIED_NAME)
FK_ACT_ACT_TYPE (ACTIVITY_TYPE_ID)
FK_ACT_WF_TYPE (WORKFLOW_TYPE_ID)
ACTIVITY_PK (WORKFLOW_TYPE_ID, QUALIFIED_NAME)
ACTIVITY_IDX01 (QUALIFIED_NAME)
ACTIVITY_IDX02 (WORKFLOW_TYPE_ID)
ACTIVITY_IDX03 (ACTIVITY_TYPE_ID)

Appendix C: Entity Relationship Diagrams

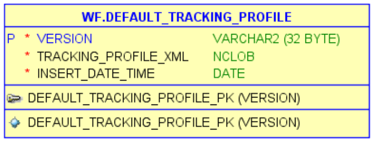


WF.COMPLETED_SCOPE

	*	INSTANCE_ID	CHAR (36 BYTE)
P	*	COMPLETED_SCOPE_ID	CHAR (36 BYTE)
	*	STATE	BLOB
	*	MODIFIED	DATE

COMPLETED_SCOPE_PK (COMPLETED_SCOPE_ID)

COMPLETED_SCOPE_PK (COMPLETED_SCOPE_ID)
COMPLETED_SCOPE_IDX01 (INSTANCE_ID)

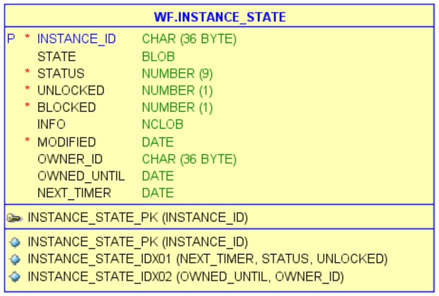


WF.DEFAULT_TRACKING_PROFILE

P	*	VERSION	VARCHAR2 (32 BYTE)
	*	TRACKING_PROFILE_XML	NCLOB
	*	INSERT_DATE_TIME	DATE

DEFAULT_TRACKING_PROFILE_PK (VERSION)

DEFAULT_TRACKING_PROFILE_PK (VERSION)

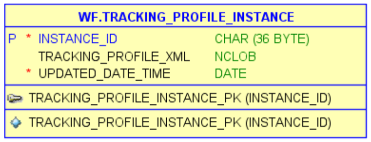


WF.INSTANCE_STATE

P	*	INSTANCE_ID	CHAR (36 BYTE)
		STATE	BLOB
	*	STATUS	NUMBER (9)
	*	UNLOCKED	NUMBER (1)
	*	BLOCKED	NUMBER (1)
		INFO	NCLOB
	*	MODIFIED	DATE
		OWNER_ID	CHAR (36 BYTE)
		OWNED_UNTIL	DATE
		NEXT_TIMER	DATE

INSTANCE_STATE_PK (INSTANCE_ID)

INSTANCE_STATE_PK (INSTANCE_ID)
INSTANCE_STATE_IDX01 (NEXT_TIMER, STATUS, UNLOCKED)
INSTANCE_STATE_IDX02 (OWNED_UNTIL, OWNER_ID)

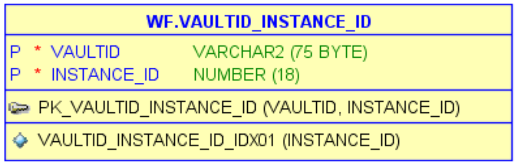


WF.TRACKING_PROFILE_INSTANCE

P	*	INSTANCE_ID	CHAR (36 BYTE)
		TRACKING_PROFILE_XML	NCLOB
	*	UPDATED_DATE_TIME	DATE

TRACKING_PROFILE_INSTANCE_PK (INSTANCE_ID)

TRACKING_PROFILE_INSTANCE_PK (INSTANCE_ID)

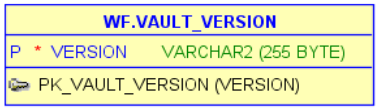


WF.VAULTID_INSTANCE_ID

P	*	VAULTID	VARCHAR2 (75 BYTE)
P	*	INSTANCE_ID	NUMBER (18)

PK_VAULTID_INSTANCE_ID (VAULTID, INSTANCE_ID)

VAULTID_INSTANCE_ID_IDX01 (INSTANCE_ID)



WF.VAULT_VERSION

P	*	VERSION	VARCHAR2 (255 BYTE)

PK_VAULT_VERSION (VERSION)