FlexNet Manager Suite Adapters Reference

Contents

Legal Information

Document Name: FlexNet Manager Suite 2015 R2 SP3 Adapter Reference (for on premises implementations)

Part Number: FMS-11.3.0-AR01

Product Release Date: 17 February 2016

Copyright Notice

FlexNet Manager Suite incorporates software developed by others and redistributed according to license agreements. Copyright notices and licenses for this externally-developed software are provided in the link below.

Intellectual Property

For a list of trademarks and patents that are owned by Flexera Software, see *http://www.flexerasoftware.com/intellectual-property*. All other brand and product names mentioned in Flexera Software products, product documentation, and marketing materials are the trademarks and registered trademarks of their respective owners.

Restricted Rights Legend

The Software is commercial computer software. If the user or licensee of the Software is an agency, department, or other entity of the United States Government, the use, duplication, reproduction, release, modification, disclosure, or transfer of the Software, or any related documentation of any kind, including technical data and manuals, is restricted by a license agreement or by the terms of this Agreement in accordance with Federal Acquisition Regulation 12.212 for civilian purposes and Defense Federal Acquisition Regulation Supplement 227.7202 for military purposes. The Software was developed fully at private expense. All other use is prohibited.

Preface

Adapters

FlexNet Manager Suite relies on software inventory collected from the computers in your computing estate to calculate what licenses are required. While the system includes a full inventory-gathering capacity, it also allows you to import inventory collected by other tools you may already have.

As well, the system uses a considerable amount of business-related data, including organizational structure, and records of purchases, to correctly track your existing license entitlements. This business data can also be imported from other sources in your enterprise.

The data collected by these 'third party' tools usually needs to be rationalized in different ways, and mapped into the data fields within the FlexNet Manager Suite database. The interfaces that allow this mapping are called *adapters*. Several adapters are provided by default with the system, or are available for download from the Flexera Software Product & Licensing Center. You can also build custom adapters for inventory using the Inventory Adapter Studio, or for business data using the Business Adapter Studio, supplied with the product. (These tools are documented in the online help.)

This document collects information on several of the standard adapters available for FlexNet Manager Suite.

Part I

Using the BMC ADDM Adapter

Topics:

The tool from BMC for collecting hardware and software information is called Atrium Discovery and Dependency Mapping (ADDM). This can be a useful inventory source as input to FlexNet Manager Suite to help in calculating license consumption as part of assessing your overall license compliance.

To collect inventory information from ADDM and import into the operations databases maintained by FlexNet Manager Suite requires a data adapter. The adapter is documented in the following chapters, listed at left.

Supported versions

The ADDM adapter supports inventory import from the following releases of BMC Atrium Discovery and Dependency Mapping:

- 8.3
- 9.0
- 10.0.

1

Choosing a Configuration

Topics:

- *How the Adapter Works*
- *Components Explained*

The adapter to extract data from ADDM can operate at different levels of detail, and with different overheads, that depend on what level of licensing information you need to collect. This section gives a brief overview of how the adapter works, and explains the different configurations and how to choose between them. You need to choose the configuration appropriate for your enterprise before implementing the adapter.

How the Adapter Works

Although it is downloaded as a single zipped archive, the adapter includes several components to improve your license reporting. The overview of the finished system shows:

- An optional set of patterns that can be deployed to each ADDM instance to improve the initial level of inventory detail collected by ADDM.
- A staging server, which includes a simple SQL database where data collected from ADDM can be massaged for upload to FlexNet Manager Suite. The executable that speeds the data extraction process also resides here. For on-premises solutions, this staging database may optionally reside on the central operations databases server.
- The conversion and upload component, which converts data formats and uploads the result to the central operations databases maintained by FlexNet Manager Suite.

Operation is summarized in this high-level diagram:

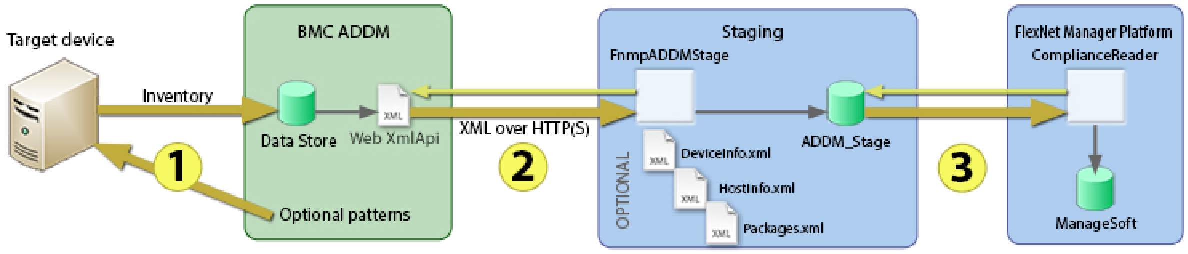


Figure 1: Process overview

The diagram shows:

1. Optionally making use of addition patterns packaged with the adapter download, inventory details are collected by BMC ADDM.
2. The executable `FnmpADDMStage.exe` uses the web API to extract the information from ADDM. This method is used because the export process is 100 times faster than using mappings, and there is no requirement to configure custom mappings for each ADDM instance. This executable optionally writes the gathered data into local XML files (available for inspection when required), and (determined by the mode of operation) also writes the data from these XML files into the staging database.
3. A standard component of FlexNet Manager Suite called a Compliance Reader, executing from the central application server, collects data from the staging database and imports it into the operations databases. This final stage occurs when you run an inventory import to process the incoming inventory.

Components Explained

This section describes each of the components in more detail, with information to help you decide which of these you will implement in your enterprise. Installation and setup follows in a separate section. The components described below are:

- Optional collection patterns to enrich the inventory detail collected by ADDM (see *Optional Patterns* on page 12)
- The FlexNet inventory agent (see *The FlexNet inventory agent* on page 14)
- The scripts for creating the staging database in Microsoft SQL Server (see *Database Table Creation* on page 15)
- The `FnmpADDMStage.exe` executable to extract inventory information from ADDM, optionally write it to XML files, and insert it into the staging database (see *The Adapter Executable* on page 15).

Optional Patterns

There are six collection patterns in all.

- `FileEvidence` allows collection of file evidence for use within FlexNet Manager Suite. This evidence may be of two sub-types:
  - Executable files that form part of the application, which are gathered only for Window platforms, and may allow application recognition (particularly to the level of editions and versions)
  - Identification files including ISO tag files, which may be gathered across any platform (including Windows and UNIX-based environments).

The next two collection patterns run both on Windows and on non-Windows devices, instructing the ADDM inventory agent to gather additional data:

- `InstallAnywhereEvidence` returns the list of package titles found in the repository maintained by Flexera Software's InstallAnywhere
- `InstallShieldMultiplatformEvidence` returns the list of packages installed by InstallShield Multiplatform, an earlier installation technology developed by Flexera Software.

The next collection pattern is for non-Windows devices only, and uses the FlexNet inventory agent to capture additional data elements and integrate these with the ADDM inventory collected in the standard way:

- `UnixHardwareData` gets accurate hardware details to allow license metrics for capacity-based license calculations (such as for Processor, Core, IBM PVU, and other license types).

  Note • *This pattern relies on the installation of two files,* `ndtrack.sh` *and* `ndtrack.ini`*, that are not included with the adapter package. These files are available with any installation of the FlexNet Beacon or application server. For more information, see The FlexNet inventory agent on page 14.*

The last collection pattern contains instructions for the ADDM inventory agent to pull additional data from the Windows Registry and WMI on Windows-based computers:

- `WindowsLastLoggedOnUser` recovers information about end-users that is required for user-based licensing (such as Named User license types).

To help you assess which of the patterns you wish to use, the following table summarizes the available collection patterns. The default state, whether the pattern is enabled or disabled, is shown in the **Pattern** column. The **Footprint** column details the additional installation impact (other than loading the pattern into ADDM) required for the collection pattern (the downside of using the pattern), and the **Impact** column shows what will *not* work if you omit this pattern (the downside of *not* using it).

Pattern	Footprint	Impact of omitting pattern
`FileEvidence` Default: Functionality is controlled in two parts: • Gathering identity (or tag) files (all platforms) is enabled by default • Gathering executable evidence (Windows only) is disabled by default.	No installation required. This pattern is configurable for paths searched (per platform), and for file name extensions used for tag files. Search times on target machines will depend on configuration and the numbers of installed applications found.	An application that • Is not already correctly recognized by ADDM (perhaps because it is installed but not currently running), and • Relies exclusively on file evidence (as distinct from installer evidence) for recognition within FlexNet Manager Suite will not be recognized for inventory collected through ADDM without this pattern. While the Application Recognition Library rarely makes use of file evidence for Windows-based applications, some publishers including IBM and Oracle make use of special identity files. Software identity tags are also in increasing use, and these are identified using this pattern. Likely impact of omitting this pattern in total is medium (through the loss of identity files). Likely impact of leaving executable gathering turned off is low. ***Note*** • *Within FlexNet Manager Suite, file evidence is also required for application usage tracking; but no application usage tracking is possible through the ADDM inventory tool.*
`InstallAnywhere Evidence` Default: Enabled	No installation required.	ADDM does not recognize installation evidence from InstallAnywhere. Unless it recognizes the application by other means, the application will be missed without this pattern.
`InstallShield MultiplatformEvidence` Default: Enabled	No installation required.	ADDM does not recognize installation evidence from InstallShield Multiplatform. Unless it recognizes the application by other means, the application will be missed without this pattern.

Pattern	Footprint	Impact of omitting pattern
`UnixHardwareData` Default: Enabled	Requires less than 13 MB installation (`ndtrack.sh` and `ndtrack.ini`), either on the hard disk of the target machines, or on a network share accessible to them all. Run-time is a second or so when triggered by ADDM.	Without this, ADDM does not capture sufficient hardware attributes from servers to support license consumption calculations on license types based on hardware capacity metrics (such as Processor, Core, IBM PVU, and other license types). Assess impact based on the license types you need to support through ADDM inventory (mandatory for capacity metrics).
`WindowsLast LoggedOnUser` Default: Enabled	No installation required.	Without this pattern, ADDM does not report any end-user identification that is needed for licenses requiring identification of the individual (such as Named User license types). Note that for general user-based licensing, in the absence of end-user identities, FlexNet Manager Suite will calculate every installation of such software as usage by an unknown end-user, so that only license types depending specifically on identity will be affected. If you wish to allocate license entitlements to specific individuals, you also require this identification. Impact: medium.

Note • *There is an additional pattern that improves the details about processors collected by ADDM. This level of detail is required only for Oracle and IBM license calculations (for other licenses, the patterns above are adequate). By contractual arrangement with BMC, this pattern can be provided only to customers approved by BMC. For further information, ask your Flexera Software consultant, who can arrange the necessary approval and provide the pattern for you.*

For more information:

- About each of these patterns, please see *Appendix A: Details of Patterns* on page 30
- About installing these patterns, see *Installing Optional Patterns* on page 24
- About enabling or disabling each of the patterns, see *Enabling Optional Patterns* on page 25.

The FlexNet inventory agent

The FlexNet inventory agent is a standard component of any installation of FlexNet Beacon, and of the application server that is included with FlexNet Manager Suite. For this reason, the necessary files are not included in the ADDM adapter zip archive, since they are already present in any standard product implementation. Installation to suit the ADDM adapter is described in *Installing the FlexNet inventory agent* on page 23.

For use on Linux or UNIX platforms, the agent has two component files:

- `ndtrack.sh`, an agent responsible for collecting inventory details (in this case, from the ADDM inventory system) and writing them in an intermediate format to a data file, ready for upload to an application server.

The script has no active elements until it is triggered by BMC ADDM through one of the enhanced collection patterns.

- `ndtrack.ini`, a text file that contains configuration variables for `ndtrack.sh`.

The combined disk space requirement of both files is under 13MB.

Database Table Creation

The staging server requires an operating version of Microsoft SQL Server 2008 or later. Any edition is suitable, including Microsoft SQL Server Express. As described in *Choosing a Staging Server* on page 20, the staging database may be located on a separate staging server, or on your central (on-premises) operations databases on your FlexNet Manager Suite server.

In the `SQL\` subdirectory of your unzipped adapter archive, the script `ADDM_staging.sql` is provided for creating the staging database, its tables, and its stored procedure. Obviously, this must be run on the SQL Server instance that is to host the staging database.

The Adapter Executable

BMC ADDM supports two ways of extracting inventory data it has collected:

- Using export mapping sets
- Using a web API.

While the former is more commonly used, the ADDM adapter for FlexNet Manager Suite uses the latter, for the following reasons:

- It avoids the need to deploy and maintain export mapping sets on every ADDM instance in your enterprise.
- Performance of data collection can be 100 times faster using the web API. For example, data extraction that can take over 24 hours using mapping sets can be completed in 20 minutes with the web API. This is because the export feature in BMC ADDM is designed for more complex export capabilities and can therefore be slow to perform simple queries. FlexNet Manager Suite requires only simple queries to be executed on ADDM, and these are performed far more efficiently using the web API.

The tool to query the web API consists of two parts:

- `FnmpADDMStage.exe` — A .NET 4.0 console program capable of querying the XML API of ADDM and writing the results into an SQL Server database, and optionally to XML files on the local file system. This program supports command line arguments, available using `FnmpADDMStage -h`
- `FnmpADDMSettings.xml` — The self-documenting configuration file for`FnmpADDMStage.exe` which contains the queries executed against ADDM (in the ADDM query language), and can include connection settings for ADDM and SQL Server.

In operation, the executable, `FnmpADDMStage.exe`, extracts the inventory data from ADDM and saves it for further processing. There are different ways that it can save the data, based on the following values of its `method` parameter:

- `Stage` — Summary**ADDM to XML**. Inventory gathered from ADDM is saved to a series of XML files on the staging server. It is not imported into the staging database. The XML files allow for review of the gathered data, but the inventory is not imported into FlexNet Manager Suite from these files.

  Tip • *The XML file option also allows for disconnected scenarios, where inventory collected from an ADDM server that is out of reach of the staging server can be written to XML, manually copied and transferred to another staging server, and the upload process resumed. See also the* `Prestaged` *method below.*

- `Staged` — Summary**ADDM to XML**/**SQL**. Inventory gathered from ADDM is first written to the XML files on disk (for example for review), and then copied into the staging database where it can be imported into FlexNet Manager Suite for use in compliance calculations.
- `Prestaged` — Summary**XML to SQL**. In this mode, inventory is not gathered from ADDM. Instead, the XML files present on the disk from a previous inventory collection (and perhaps reviewed and approved by a human agent in this format) are now copied into the staging database where it can be imported into FlexNet Manager Suite for use in compliance calculations.
- `Stream` — Summary**ADDM to SQL**. Inventory is gathered from ADDM, and loaded into the staging database where it can be imported into FlexNet Manager Suite for use in compliance calculations. In this method, inventory is not recorded in XML files on the staging server.

Default values for the `method` and all other parameters are set in the companion `FnmpADDMSettings.xml` file, and these are the values used when the executable is triggered (or run) without other command-line options. The settings file is self-documented, and the matching command-line options are available using `FnmpADDMStage -h`

When the executable writes XML files to (or reads them from) the local disk on the staging server, the files include the following (details are available in `FnmpADDMSettings.xml`):

Filename	Content
`Cluster.xml`	Details for each cluster.
`ClusterHost.xml`	Computers (host nodes) that are members of a cluster, including a key to the Cluster node to identify that cluster.
`CPUInformationDetail.xml`	Details of computer processors.
`DiscoveredPackages.xml`	Raw installer evidence gathered by ADDM from the installer technologies supported by each operating system.
`DiscoveredService.xml`	A report of particular services, used to help identify capabilities of hosts. This includes the VMMS service used to identify Windows machines with the Hyper-V role enabled.

Filename	Content
`DiscoveredVirtualMachine.xml`	Raw results from ADDM querying the list of virtual machines on a virtualization host.
`FileEvidenceDetail.xml`	File evidence produced by the `Flexera.FNMP.InventoryRawData.FileEvidence` pattern, covering software tag files and Windows executables.
`FileSystem.xml`	Name and size of all local file systems, used to approximate the total disks and storage of host.
`HardwareEvidenceDetail.xml`	Hardware details produced by the `Flexera.FNMP.InventoryRawData.UnixHardwareEvidence` pattern, using information gathered by the FlexNet inventory agent.
`Host.xml`	Details of all hosts known by ADDM including their host name, operating system details, unique identification, processor, memory, and other hardware details.
`HostInfo.xml`	Raw host details not represented in a Host node, mainly the raw LPAR information from an IBM AIX LPAR environment.
`InstallerEvidenceDetail.xml`	Installation evidence gathered using the patterns in the ADDM adaptor, including evidence from installations by Install Anywhere and InstallShield Multi-platform.
`LastLoggedOnUserDetail.xml`	Details of the last logged-on user for Windows systems.
`NetworkInterface.xml`	The IP and MAC addresses of each network interface, used to build a list of these addresses for each host. **Note •** *ADDM 9.0 introduced the `IPAddress` nodes which cover both IPv4 and IPv6. Prior to that, the IPv4 IP address was in the `NetworkInterface` node. This query checks both of these sources.*
`SoftwareInstance.xml`	Software installations identified by ADDM's pattern language. ADDM queries various properties such as processes and files of a host to determine which software is installed and its version.

Filename	Content
`SoftwareInstanceVirtual Machine.xml`	These `SoftwareInstance` nodes are used to represent virtual machines on a host. These records are typically only created when the virtual machine is running.

Conditions for Use

Use of this executable, `FnmpADDMStage.exe`, imposes the following conditions:

- Within BMC ADDM, the XML-based API must be enabled (by default, it is enabled). BMC documentation for the XML API is available at `http://discovery.bmc.com/confluence/display/90/XML+API`.
- There must be HTTP or HTTPS communication available between the staging server, on which this executable runs, and the server hosting BMC ADDM.
- The staging server requires the .NET 4.0 runtime environment.
- The staging tool must be configured with credentials that provide read access to the ADDM instance. These credentials may either be configured in `FnmpADDMSettings.xml` or supplied on the command line for `FnmpADDMStage.exe` (for details see *Account Configuration* on page 26).
- For linking upstream to the staging database, you can either
  - Use a trusted connection to the SQL server and run the executable under an account that has read/write access to the staging database; or
  - Use service account credentials for SQL through a connection string, which may either be configured in `FnmpADDMSettings.xml` or supplied on the command line for `FNMPAddmStage.exe` (see *Creating the Staging Database Tables* on page 20 for details).

Important • *It is critical that the `FnmpADDMStage.exe` tool is not run simultaneously with the `ComplianceReader` tool that uploads from the staging database to the compliance server. The adapter executable commences its writing activity with a `truncate` of the staging database to clear old results, an action which (if it occurred in the middle of an upload to the compliance server) could clearly corrupt the imported dataset. While `ComplianceReader` is blocked until `FnmpADDMStage.exe` finishes, you must adjust your schedules to prevent starting `FnmpADDMStage.exe` while the `ComplianceReader` is running.*

2

Installation and Configuration

Topics:

For **on premises** implementations, files are included in your product installation archive for the application server. As well, the latest version of the adapter is available for download from the Flexera Software `flexnetoperations` website. For **cloud** implementations of FlexNet Manager Suite, you also require this download: although you do not need to make changes to your central application server, there are additional components you require in the download.

You need credentials supplied by Flexera Software to access this download. Details of the download are included in *Creating the Staging Database Tables* on page 20.

- The ADDM adapter suits FlexNet Manager Suite releases 9.2.3, and 2014 and later for on premises delivery.
- The build number for this adapter is 10.1.0.10631 (or higher). You can identify this number by right-clicking on `FNMPADDMStage.exe`, selecting **Properties** and looking at the **Details** tab.

Save the zipped archive to a suitable temporary location, and unzip it.

Full details of setting up the ADDM adapter are included in the following sections, as listed on the left.

Choosing a Staging Server

The FlexNet adapter for ADDM requires a 'staging server' that supports installation of the adapter's executable and of the staging database. Several configurations are possible. The staging server may be installed on:

- A dedicated stand-alone server (or virtual machine)
- Any other suitable machine in your enterprise, such as a print server
- An inventory beacon
- The central application server where FlexNet Manager Suite is installed (for on-premises installations).

The requirements for a suitable server include:

- A Windows-based operating system
- Access to an installation of Microsoft SQL Server 2008 or later, in any edition, where the staging database may be implemented (it may be on the staging server, or on a separate database server)
- The .NET 4.0 runtime environment installed
- Network access to the central application server (when not co-installed there)
- Efficient network access to each ADDM server in your enterprise, using the HTTP or HTTPS protocols.

Note • *Disconnected scenarios are also possible, using the intermediate XML files saved to disk to allow manual intervention. For more information, see The Adapter Executable on page 15.*

Creating the Staging Database Tables

Once you have selected your staging server, and it can access an operating implementation of Microsoft SQL Server running a database instance you intend to use for the staging database, you should use the script provided to create the staging database and set up the appropriate database tables within it. This can be done from SQL Server Management Studio, or from the command line as described in the following procedure.

1. Using the credentials supplied by Flexera Software with your order confirmation (or as renewed since), log into the Product and License Center at *`https://flexerasoftware.flexnetoperations.com`*.
2. On the first page, select `FlexNet Manager Platform`, and on the resulting second page, select the product again.
3. In the list of versions, click the product name for the version you are using (typically the most recent version).
4. In the list of components to download, select the `Tier 1 Adapter Tools.zip` archive, and save this to a convenient location (such as `C:\Temp` on a central, accessible server).
5. Right-click the zip archive, and choose **Extract All...**.
6. Navigate through the unzipped archive to `Tier 1 Adapter Tools` > `BMC Atrium Discovery and Dependency Mapping Tools` > `SQL`.

7. If necessary, copy the script ADDM_staging.sql from the SQL\ folder of your unzipped adapter archive to a temporary folder on your staging server.
8. Open a command prompt on the staging server.
9. In the command prompt window, execute the following command, as amended:

   ```
   sqlcmd -S ServerName\InstanceName -i TemporaryPath\ADDM_staging.sql
   ```

   where:

   - The database ADDM_Staging is created with all necessary tables, indices, and so on.
   - *ServerName* is the name of the database server hosting the staging database, or its IP address, or "." (dot) if you are running the staging script on the same server as the database instance
   - *InstanceName* is the name of the instance to use for the database staging tables (this parameter may be omitted if the instance is the default instance)
   - *TemporaryPath* is the location where you saved the SQL procedure.

   Example:

   ```
   sqlcmd -S 192.100.0.20\Development -i C:\temp\ADDM_staging.sql
   ```

10. Ensure that the account under which the adapter executable will run has read/write/execute permissions on this database instance. Authentication may be through Windows NT authentication or SQL Server authentication. Using Windows NT authentication, the default account is the username running the FnmpADDMStage.exe adapter. SQL connection is specified as a standard connection string, which you may supply in FnmpADDMSettings.xml, or override with the -c option on the command line.

    ***Tip** • Configuration of the account is done through SQL Server Management Studio.*

The staging database is now ready for operation.

Installing and Configuring the Staging Tool

This procedure includes many separate sub-processes to complete the set-up of the adapter executable. We start from the configuration of the inventory reader that uploads inventory gathered by the adapter.

1. Navigate to C:\ProgramData\Flexera Software\Compliance\ImportProcedures\Inventory \Reader\.

   This is the fixed location on the inventory beacon where the inventory reader (reaching out from the compliance server) must find configuration files to control its uploads. By default, the inventory import (starting with the reader) is triggered around 2am. Therefore you might consider scheduling this task for some time such as 10pm daily. The command line for the scheduled task (assuming that you have saved your preferred settings) is simply to invoke the executable. Any parameters not specified on the command line are taken from the settings file in the same folder as the executable.

2. From your unzipped adapter archive, copy the folder `BMC Atrium Discovery and Dependency Mapping` (found in the path `Adapter\Reader\`) to the location identified in the previous step. This folder includes at least ten XML files and a `reader.config` file. This completes configuration for the inventory reader.
3. Create a folder to contain the adapter executable and its configuration file. Location is not critical; a suggested path is under `C:\Program Files\Flexera Software`. In your chosen location, create a folder such as `ADDMAdapter`.
4. From the `FnmpADDMStage\` folder within your unzipped archive, copy both `FnmpADDMStage.exe` and `FnmpADDMSettings.xml` to your newly created folder (such as `C:\Program Files\Flexera Software \ADDMAdapter`).
5. Open your copy of `FnmpADDMSettings.xml` in a text editor of choice, and review the self-documenting comments within that file. Modify the following values as required (at the very minimum, correct the IP address of your ADDM server):

   - Update values in the first element describing the downstream connection to the BMC ADDM server, including the IP address, the account name and password for access. Keep a record of the account name and password for registering with ADDM (see *Account Configuration* on page 26). The default values are: `<server protocol="http" address="10.200.20.138" username="exportuser" password="Pa$$w0rd" timeout="3600"/>`

     Tip • *If you do not wish to record the password in the plain text configuration file, you can use a script to retrieve the password from an encrypted store, and supply it as a command-line option when starting the* `FnmpADDMStage.exe` *tool.*

   - Update the second element for the connection to the staging database. The default values are: `<database connection-string="Server=.;Database=ADDM_Staging;Trusted_Connection=yes;"/>`
   - Update the third element to configure whether, and where, the executable should save XML files of the inventory collected from BMC ADDM. The default value stores any XML files below the location of the executable (but turns off storage anyway): `<staging path="." method="stream"/>` You may wish to redirect the path setting for easier access for human inspection.
   - Save your modified settings file.
6. Assuming that you do not wish to trigger the adapter manually every time it needs to run, start Windows Task Scheduler, and create a basic task to run the adapter.

   Important • *It is critical that you schedule the adapter to run at times which cannot overlap with the inventory reader uploading the results to the compliance server. Check the schedule for the compliance reader on the central compliance server, and avoid this time slot.*

   By default, the inventory import (starting with the reader) is triggered around 2am. Therefore you might consider scheduling this task for some time such as 10pm daily. The command line for the scheduled task (assuming that you have saved your preferred settings) is simply to invoke the executable. Any parameters not specified on the command line are taken from the settings file in the same folder as the executable.

This completes the configuration of the adapter executable itself. Now we can turn our attention downstream, first to possible installations on target UNIX-based machines, and then to enhancements to ADDM itself.

Installing the FlexNet inventory agent

If you have chosen to install any of the UNIX-related collection patterns to enhance the inventory collection available through BMC ADDM, the FlexNet inventory agent must be copied either:

- On to each UNIX-based machine that is a target for enhanced inventory collection; or
- To a pre-configured NFS share that is accessible from each of the target UNIX-based machines.

In the following procedure, your choice of either of the above two locations is referred to as the 'target location'.

(For background information about the FlexNet inventory agent, see *The FlexNet inventory agent* on page 14. For choosing between the collection patterns, see *Optional Patterns* on page 12. Further information about deploying the FlexNet inventory agent is available in the separate PDF file *FlexNet Inventory Gathering*.)

To install the FlexNet inventory agent for UNIX collection patterns:

1. Using the installations of either FlexNet Beacon, or application server, locate the subdirectory that contains the `ndtrack` files. The default location is `C:\Program Files\Flexera Software\Inventory Beacon \RemoteExecution\Public\Inventory`
2. From this folder, copy the two files `ndtrack.sh` and `ndtrack.ini` to the target location you selected above (that is, either to a pre-configured NFS file share, or to each target UNIX-based machine). The default location pre-configured in the template file is `/opt/flexera/`, but you may modify this as required.

   Important • *The file path on all individual UNIX-based machines must be identical.*
3. Note the file path used (whether a file share, or the identical path used across all target devices) for entry into ADDM when you are enabling the optional collection patterns (see next section).

Configuring BMC ADDM

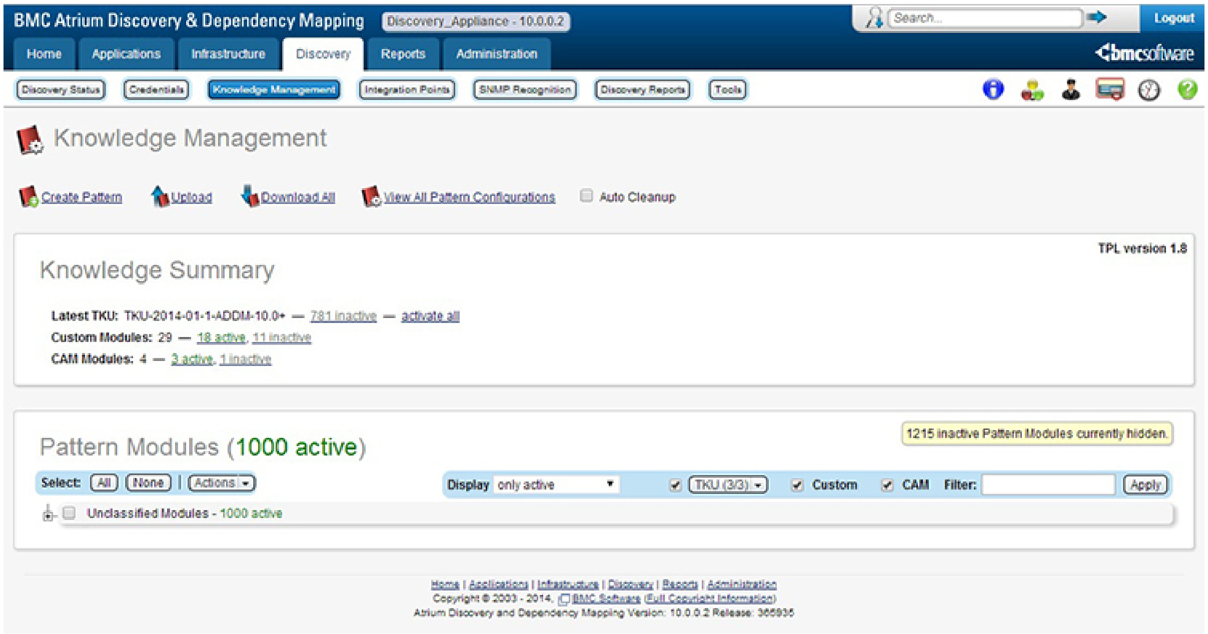
There are three customizations needed for ADDM:

- Installing any of the optional patterns needed for the adapter in your environment (see below)
- Ensuring that ADDM applies these patterns to the appropriate target computers (see *Rediscovering Affected Computers* on page 26)
- Ensuring that the account running the adapter has access to ADDM (see *Account Configuration* on page 26).

Installing Optional Patterns

For information about choosing which patterns to use, see *Optional Patterns* on page 12. For details about enabling each of the patterns, and modifying their behavior, see *Appendix A: Details of Patterns* on page 30. All these patterns are contained in a single template (`Flexera.FNMP.InventoryRawData.tpl`), which must be installed first. Thereafter, individual patterns can be enabled, disabled, and modified.

1. Log in to the ADDM interface, and select the **Discovery** tab.
2. Select the **Knowledge Management** button in ADDM version 10 (in earlier versions, select the **Pattern Management** button).

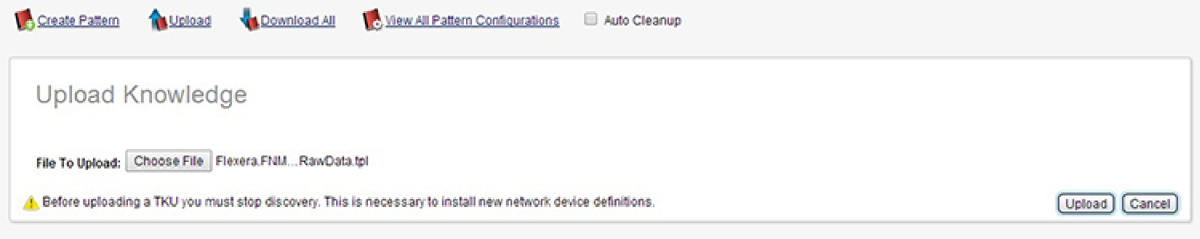


Latest TKU: TKU-2014-01-1-ADDM-10.0+ — 781 inactive — activate all
Custom Modules: 29 — 18 active, 11 inactive
CAM Modules: 4 — 3 active, 1 inactive

Copyright © 2003 - 2014, BMC Software (Full Copyright Information)
Atrium Discovery and Dependency Mapping Version: 10.0.0.2 Release: 365935

Figure 2: The workspace in ADDM 10

3. Click on **Upload** (or in earlier versions, **Upload New Package**).



File To Upload: Flexera.FNM...RawData.tpl

Before uploading a TKU you must stop discovery. This is necessary to install new network device definitions.

Figure 3: Choose the file from your unzipped archive

4. Click on **Choose File**, and in your unzipped archive of the adapter, select the `FlexNet Manager Platform\Installers\BMC Atrium Discovery and Dependency Mapping Tools\patterns\Flexera.FNMP.InventoryRawData.tpl` file.
5. If you are using ADDM version 8:

   a) Optionally modify (or accept) the default name of the package as `Flexera.FNMP.InventoryRawData`.

   b) Optionally, set the description to something you will find helpful, such as `FNMP export patterns`.

6. Click **Upload** (or in earlier versions, **Upload & Activate**).

   Ensure that the message `The Requested Changes were Successful` is displayed. The `Flexera.FNMP.InventoryRawData` pattern is now in the `Active` state. For further fine tuning, jump ahead to step 4 in the following procedure.

Tip • *If you are using data imported through the ADDM adapter to manage points-based licenses for IBM and Oracle, there are additional optional patterns that collect further details about processors available on request from Flexera Software.*

Enabling Optional Patterns

By default, after installation some patterns are enabled and others disabled (see *Optional Patterns* on page 12). The following procedure allows you to turn individual patterns on and off as required, as well as enabling (or disabling) the entire module.

1. In the user interface for BMC ADDM, select the **Discovery** tab.
2. Click **Pattern Management**, and from the list of patterns at the bottom of the page, ensure that the `Flexera.FNMP.InventoryRawData` item is enabled, and select it. (The module must be enabled before any of its member patterns can operate.)
3. From the **Pattern Package** properties, select the link for the **Pattern Module**.
4. Scroll down to the **Pattern Configuration** area.

   Several closed groups are displayed, one for each optional pattern and another for configuration of the FlexNet inventory agent on UNIX platforms.

5. For each group that you want to modify, select **Edit Configuration** to the right of the group. The group opens, and configurable controls are displayed. Select the **True** radio button to turn on a pattern, and the **False** radio button to turn off a pattern.

   More details about configuration are included in *Appendix A: Details of Patterns* on page 30.

6. Click **Apply**.

   The modified settings are displayed.

Tip • *If you later use the **Edit Module** button to modify the module, be sure to activate, validate, and then commit your changes.*

Rediscovering Affected Computers

Where (as is common) you are adding these patterns to an operational ADDM instance that has already taken inventory of computers in your estate, you must rediscover any computers that are to be targeted through these new patterns, so that ADDM applies to patterns to the rediscovered computers. To achieve this, you may either:

- Create (or wait for) a scheduled scan
- Initiate a Snapshot discovery scan
- Manually execute each pattern for suitably grouped targets, as summarized in the following procedure.

For more information about these options, see the BMC ADDM documentation available at `http://discovery.bmc.com/confluence/display/90/Documentation`.

To manually execute the new patterns:

1. In the ADDM interface, select target computers (hosts or other nodes) by adding them to a group, creating separate groups to receive distinct types of patterns (for example, one group for Windows-related patterns, and another group for UNIX-related patterns). Create your groups using either of these approaches:
    - From a view node (including host) page, select **Groups** from the **Actions** list and add the node to a group.
    - From a report or other search result, select the required target computers. Then, select **Groups** from the **Actions** list and add the target machines to a group.
2. From the **Discovery** tab, click **Pattern Management**.
3. Select the `Flexera.FNMP.InventoryRawData` pattern from the package list.
4. Click the **Pattern Modules** link.
5. Select the `Pattern Module` containing the pattern that you want to run.
6. Click the **Pattern** link in the heading table.
7. From the **Actions** list, select **Run Pattern**.
8. In the **Run against Group** list, select the group containing your target machines for the current pattern(s).
9. Set the `Expand` and `Execution Logging` preferences for the run.
10. Set `Additional Discovery` to **Get all new discovery data**. This forces a new discovery.

More details about this procedure are available from *http://discovery.bmc.com/confluence/display/90/Manual+pattern+execution*.

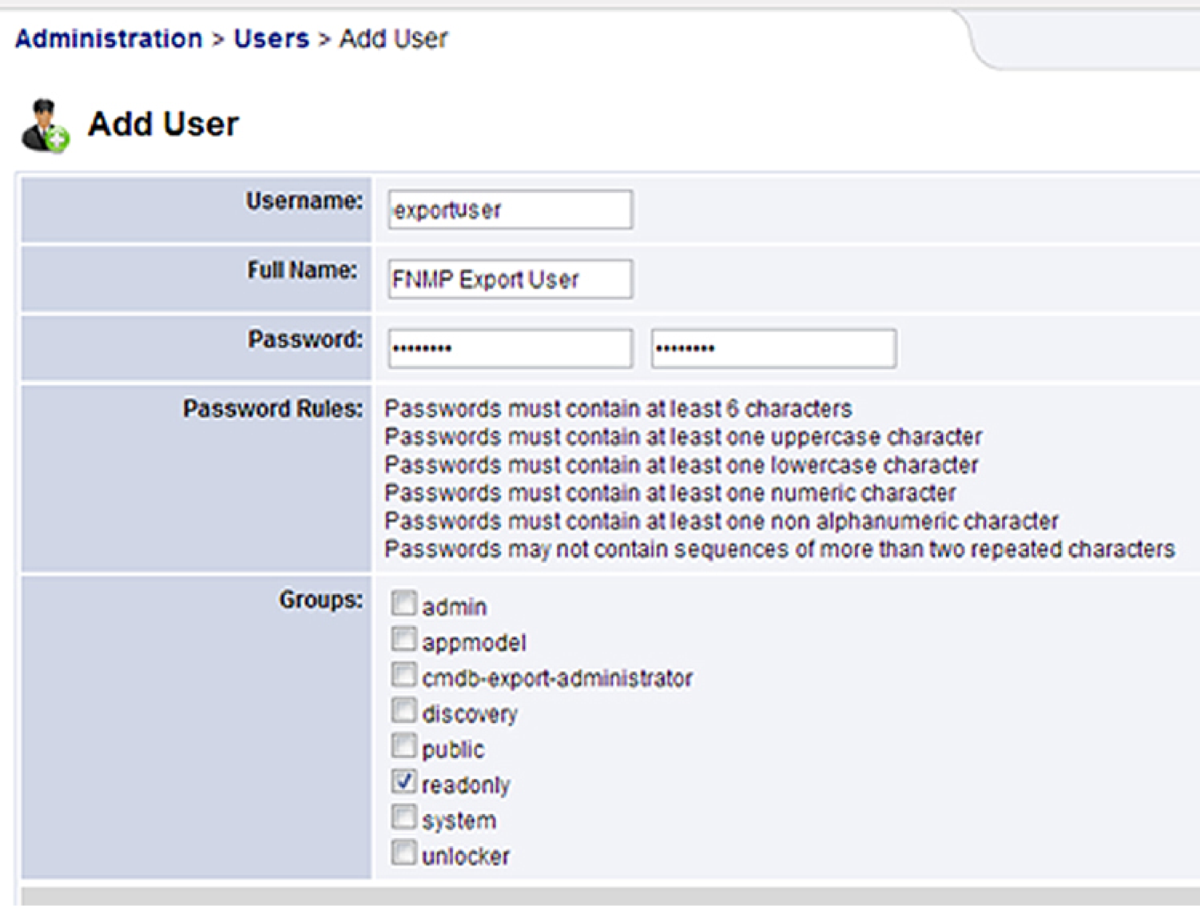
Account Configuration

The user name and password that the adapter needs to access the ADDM API is defined in `FnmpADDMSettings.xml` (see *Installing and Configuring the Staging Tool* on page 21). Here we grant that account adequate rights.

1. In the ADDM interface, select the **Administration** tab.
2. In the Security section of the Administration page, click the **Users** icon.
3. From the Users page, click **Add** at the bottom of the page.

Tip • If the account has already been registered in ADDM, you can select it from the list of users and review its settings.

4. Configure the adapter account, which can be a member of the **readonly** group. Ensure that the user name and password are exactly same values that you configured in `FnmpADDMSettings.xml`.



Administration > Users > Add User

Add User

Username:	exportuser
Full Name:	FNMP Export User
Password:	•••••••• ••••••••
Password Rules:	Passwords must contain at least 6 characters Passwords must contain at least one uppercase character Passwords must contain at least one lowercase character Passwords must contain at least one numeric character Passwords must contain at least one non alphanumeric character Passwords may not contain sequences of more than two repeated characters
Groups:	□ admin □ appmodel □ cmdb-export-administrator □ discovery □ public ☑ readonly □ system □ unlocker

Figure 4: Sample settings for a adapter account to access ADDM

5. Save your settings.

BMC ADDM is now configured for use of the adapter.

Validation and Operation

Normal operation relies on the following sequence of events:

1. BMC ADDM gathers inventory using the enhanced patterns you have enabled (details: *Installing Optional Patterns* on page 24, and *Rediscovering Affected Computers* on page 26).

2. At the time you scheduled (*Installing and Configuring the Staging Tool* on page 21), the adapter reads the current content of the ADDM database and stages the new data in the staging database (previous data is first removed with a single `truncate` statement). The resulting status is flagged within the database.

3. Following the schedule on the central compliance server, and provided that the staging status is `Success`, the import reader uploads this content to the inventory database.

4. The next inventory import brings the final data set into the compliance database, where it is automatically taken into account for compliance calculations. As always, the inventory records must be recognized by the Application Recognition Library, and you must have the resulting application records linked to the appropriate license, for compliance calculations to proceed.

To validate operation of the adapter:

1. Wait until ADDM collects new inventory, so that the enhanced collection patterns are exercised. For further details, see the BMC ADDM documentation.
2. Manually trigger the adapter executable.

   By specifying a different method parameter, you have a one-time override of the default you set in the settings XML file. For example:

   ```
   C:\Program Files\Flexera Software\ADDMAdapter\FnmpADDMStage.exe
                         -f "C:\temp" -m staged
   ```

   This will write XML files under your `C:\temp` directory for review. As well, it writes data into the staging database.
3. Inspect the saved XML files to validate the inventory gathered.
4. Use SQL Server Management Studio to validate that the data is written to the staging database. Also review the `StagingState` property in the `ADDMStagingDatabaseConfiguration` table in the staging database. Possible values are `Running`, `Failed`, or `Success`. This value must be `Success` before the ADDM data can be uploaded from the staging database to the central inventory database.
5. Wait (for example, overnight) until the next inventory import and calculations have run.
6. Use FlexNet Manager Suite to validate that new evidence has been recovered. Identify which evidence has been recognized by the ARL and which new rules are required. Link the applications to appropriate licenses.

3

Known Issues

The following issue has been identified:

- UNIX-based devices that have the same host name and no detected serial number in ADDM are merged into a single computer record in FlexNet Manager Suite.

4

Appendix A: Details of Patterns

Topics:

While the BMC ADDM discovery tool is a state-of-the-art tool to support ITAM, the out-of-the-box collection of inventory does not capture all of the data elements required by FlexNet Manager Suite to perform an accurate software license calculation. The optional patterns in this appendix extend those data capture capacities.

Overview of Patterns

The BMC ADDM Adapter available for FlexNet Manager Suite includes six additional patterns that can be incorporated into ADDM to capture these additional data attributes.

Pattern name	FlexNet inventory agent dependency	Default
`FileEvidence`	No dependency.	Tag files (all platforms): Enabled Executables (Windows only): Disabled
`InstallAnywhereEvidence`	None	Enabled
`InstallShieldMultiplatformEvidence`	None	Enabled
`UnixHardwareData`	Required	Disabled
`WindowsLastLoggedOnUser`	None	Enabled

The patterns are written in the ADDM Pattern Language (TPL), for which documentation is available at `http://discovery.bmc.com/confluence/display/90/The+Pattern+Language+TPL`.

For a summary of the patterns to help decide which ones to enable or disable for your enterprise, see *Optional Patterns* on page 12. The following sections go into more detail about each of the patterns.

***Note •** The optional patterns listed above do not provide enough detail about processors to allow management of points-based licenses for IBM and Oracle based on data collected through ADDM. If you need this capability through ADDM, ask your Flexera Software consultant to request additional optional patterns for you.*

FileEvidence

Some applications cannot be recognized by installer package information alone. It is sometimes necessary to examine files that form part of the software installation, for either of two reasons:

- Some files are intended to provide identification details about an application, sometimes in human-readable formats
- Examining executable files installed with the application may help with identification, even though this is not their primary purpose.

Of the first class, identification files may take various forms. For example, many IBM applications are correctly identified by specific files that IBM installs for this purpose. Oracle and Adobe are among other publishers using specific files to identify some applications. Thus the Application Recognition Library (ARL) requires this file information to correctly identify such applications. ISO/IEC 19770-2 SWID tags are also increasingly available, and these ID tags are another useful form of identification file. The ADDM Inventory Agent by default does not capture the complete set of identity files.

Secondly, in addition to gathering those specific files that identify an application (and have no other function in the application), it is sometimes necessary to identify installed executable files that are part of the application. These may be the only way to identify an application, such as a particular edition or version of a product. A standard implementation of BMC ADDM does not track executable files.

With this pattern, the functionality of ADDM is extended to gather identification (tag) files on all platforms, and executable file evidence on Windows platforms. Note that the pattern does not search network shares or NFS mounts; nor does it follow symlinks (because of the risk of self-referencing loops). It also skips any files or folders that are inaccessible. Within these constraints, it provides details of files matching the specification and found within the defined paths on the local file system.

***Important* •** *Enabling and configuring the tracking of executables within this pattern should be handled with skill and care, for two reasons:*

- *Tracking Windows executables using ADDM is slow. It may be unacceptably slow to use for a wide range of directories, and you may require very targeted inventory gathering using this facility.*
- *Tracking executable files can produce a very large data set. Across large Windows server farms, the number of installation records can quickly run even into the millions, which may comprise a stress test for ADDM implementations and concentrators, and (to a slightly lesser extent) for your FlexNet Manager Suite implementation.*

***Note* •** *In FlexNet Manager Suite, you can use file evidence to track the usage of an application (perhaps with a view to reclaiming under-used licenses). While that functionality requires you to identify a watch-list of at least one executable file for each tracked application, file evidence alone is not sufficient for usage tracking: it also requires additional FlexNet agents on the managed device, and an upload path through inventory beacons that preserve the additional usage data. File evidence gathered through the ADDM adaptor is not sufficient for usage tracking.*

Results

The file evidence gathered on Windows servers is somewhat richer than on UNIX-based servers.

- On both platforms, the pattern first collects the target directories from the pattern configuration file (under **Flexera.FNMP.InventoryRawData.FileEvidenceConfigs**).
- In the specified folders and their subfolders, the pattern retrieves:
  - All files with extensions listed under **File extensions to report as tag files** (assuming that **Report software tag files** has its default value of true). For each matching file found, ADDM creates a `Detail` node linked to the host record. This happens for both Windows and UNIX-based systems.
  - On Windows only, and only when the setting for **Report executable files on Windows platforms** has been changed to true, files with a `.exe` extension from the same paths. For each executable file found, by default a WMI query is used to retrieve the file's name, version, and manufacturer. A

DiscoveredWMI node in ADDM is created for each WMI query (essentially for each file). However, because DiscoveredWMI nodes are ephemeral (may not survive future inventory gathering), the information is duplicated into a Detail node under the host server for each file discovered there.

By the appropriate means, then, a Detail node is created for each file evidence record, with the following properties dependent on the collection method:

Property	Value	Notes
name	File path and name	
type	FNMP_FileEvidence	
size	File size	
key	A unique key for this file, combining the values of *name/type/host.key*	
version	The release number of the file	Available only on Windows servers when executables are tracked.
company	The publisher of the application of which this file forms a part	Available only on Windows servers when executables are tracked.

Configuration

In the FileEvidence pattern, you may:

- Separately enable or disable collection of:
  - Identity tag files
  - Executable files (remember to consider the potential volume of data for this option)
- Identify the file name extensions for tag files (but there is no need to identify executable file name extensions)
- Separately for Windows and UNIX-like hosts, specify the starting point(s) in the file systems for inventory scanning to begin (searches recurse through local subdirectories).

Important • *The ADDM inventory agent stops scanning at partition boundaries, even those on the local file system. This has important implications on systems, such as IBM AIX, that typically mount key paths like /opt on separate partitions. You must specify a search starting point within each target partition on the file system. For example, the default values of /opt, /var, and /usr are well suited for inventory gathering with ADDM.*

All configuration items are covered in the following procedure.

To configure the FileEvidence pattern

1. In the user interface for BMC ADDM, select the **Discovery** tab.
2. Click **Pattern Management**, and from the list of patterns at the bottom of the page, ensure that your current Flexera.FNMP.InventoryRawData item is enabled, and select it. (The module must be enabled before any of its member patterns can operate.)

3. From the **Pattern Package** properties, select the link for the **Pattern Module**.
4. Scroll down to the **Pattern Configuration** area.
5. In the **FileEvidence** group, click **Edit Configuration**.
6. Select the appropriate true/false radio button for **Report software tag files**. If you are turning off the collection of tag files, skip the next step.
7. Adjust the list of **File extensions to report as tag files**, if necessary deleting values or adding new ones that you have identified. Each extension must stand alone on its own line.
8. Select the appropriate true/false radio button for **Report executable files**. Review the discussion before this procedure while considering the data quantities implied by enabling this option.

   If you have now turned off both options, so that both options now have **False** selected, you have completed the process, and may skip the next step. If either option has **True** selected, continue to define the paths for the inventory gathering.

   The configuration changes are saved.
9. For both types of operating system, customize the ...**scan these file paths for evidence** list.
   - Each file path must stand alone on its own line, and must be absolute (starting from root).
   - For Windows, the path must include the drive letter with its colon delimiter.
   - These same paths are scanned for identity (tag) files and for executable files.
10. Click **Apply** (at the bottom of this group).

    The configuration changes are saved.

InstallAnywhereEvidence

InstallAnywhere is a package installation solution developed by Flexera Software. Packages track their installation status in an XML file, which is interrogated by this pattern.

After collection of the inventory data, a `Detail` node is linked to the host computer for each package installed on the computer:

Property	**Value**
`name`	The name of the application as identified by InstallAnywhere.
`type`	`FNMP_InstallerEvidence`
`vendor`	The publisher of the application
`version`	The release number of the application.
`install_date`	The date that the application was installed on this host computer.
`evidence`	`IA` (fixed string literal).
`key`	A unique key for this installation record, combining (with the different literal text separators shown) the values of *`name`*`:`*`version`*`/`*`type`*`:`*`evidence`*`/`*`host.key`*

To configure the InstallAnywhereEvidence pattern:

1. In the user interface for BMC ADDM, select the **Discovery** tab.
2. Click **Pattern Management**, and from the list of patterns at the bottom of the page, ensure that your current `Flexera.FNMP.InventoryRawData` item is enabled, and select it. (The module must be enabled before any of its member patterns can operate.)

   Tip • The `InstallAnywhereEvidence` pattern is enabled by default when the pattern collection is initially enabled. If it has previously been disabled, you can re-enable it with the remainder of this procedure.

3. From the **Pattern Package** properties, select the link for the **Pattern Module**.
4. Scroll down to the **Pattern Configuration** area.
5. In the **InstallAnywhereEvidence** group, click **Edit Configuration**.
6. Select the **True** radio button for **Report installations by InstallAnywhere**.
7. Click **Apply** (at the bottom of this group).

InstallShieldMultiplatformEvidence

InstallShield Multiplatform is an older packaging solution from Flexera Software, in use by many software publishers. InstallShield stores software information in “vital product data” (VPD) collections, which were originally stored in flat files and subsequently in SQL scripts.

This pattern locates either format of VPD storage (which may exist in a range of locations across different platforms), extracts the data, and creates a `Detail` node for the software installation linked to the host computer:

Property	Value
`name`	The name of the application as identified by InstallShield.
`type`	`FNMP_InstallerEvidence`
`vendor`	The publisher of the application
`version`	The release number of the application.
`install_date`	The date that the application was installed on this host computer.
`evidence`	`ISMP` (fixed string literal).
`product_code`	The product identification code recorded (usually) by the publisher for the particular application. This is not standardized and may be used as the publisher desires.
`key`	A unique key for this installation record, combining (with the different literal

Property	Value
	text separators shown) the values of *`name:version/type:evidence/host.key`*

To configure the `InstallShieldMultiplatformEvidence` pattern:

1. In the user interface for BMC ADDM, select the **Discovery** tab.
2. Click **Pattern Management**, and from the list of patterns at the bottom of the page, ensure that your current `Flexera.FNMP.InventoryRawData` item is enabled, and select it. (The module must be enabled before any of its member patterns can operate.)

   Tip • *The `InstallShieldMutiplatformEvidence` pattern is enabled by default when the pattern collection is initially enabled. If it has previously been disabled, you can re-enable it with the remainder of this procedure.*

3. From the **Pattern Package** properties, select the link for the **Pattern Module**.
4. Scroll down to the **Pattern Configuration** area.
5. In the **InstallShieldMultiplatformEvidence** group, click **Edit Configuration**.
6. Select the **True** radio button for **Report installations by InstallShield Multiplatform**.
7. Click **Apply** (at the bottom of this group).

UnixHardwareData

Especially for managing the corporate Data Centre, it is critical for FlexNet Manager Suite to have accurate hardware inventory for capacity-based license metrics (Processor, Core, IBM PVU, and so on). While ADDM can capture this information for a Windows server, it may not consistently capture it for servers running UNIX or Linux. For example, ADDM does not currently report:

- CPUs and cores on Linux, nor cores on virtual machines
- LPARs on IBM AIX (correctly)
- Solaris resource pools.

This pattern retrieves hardware data on UNIX-based systems using the FlexNet inventory agent.

Note • *The pattern must be configured with the installed location of the agent. The agent may be installed either in the same location on the file system of each target UNIX server, or on a file share accessible to all target devices. This is included in the configuration process described below.*

It executes the inventory agent, which writes the collected data to a file in the `/var/tmp/flexera/addm/` folder on the host server. ADDM then reads this output, and for each installed file, creates a `Detail` node linked to the host record:

Property	Value
Name	FNMP hardware evidence for %host.name%
Type	FNMP_HardwareEvidence
Key	%type%/%host.key%
(Additional properties)	Each records one of the following hardware properties: • Disk size • IP Address • MAC Address • Model • Number of cores • Number of disks • Number of logical processors • Number of processors • OS • Processor speed • Processor type • RAM (total physical memory) • Vendor. If the machine is found to be a virtual machine, the following additional properties are collected: • Node capacity • Node capacity in cores • Node capacity in threads • Physical shared pool capacity • Physical shared pool capacity in cores • Physical shared pool ID • Shared pool capacity • Shared pool capacity in cores • Shared pool ID • VM capacity • VM capacity in cores

Property	Value
	- VM entitlement - VM ID - VM is capped - VM is shared type - VM name - VM type

To configure the `UnixHardware` pattern:

1. In the user interface for BMC ADDM, select the **Discovery** tab.
2. Click **Pattern Management**, and from the list of patterns at the bottom of the page, ensure that your current `Flexera.FNMP.InventoryRawData` item is enabled, and select it. (The module must be enabled before any of its member patterns can operate.)
3. From the **Pattern Package** properties, select the link for the **Pattern Module**.
4. Scroll down to the **Pattern Configuration** area.
5. In the **Flexera.FNMP.InventoryRawData.CommonFlexeraInventoryAgentConfigs** group, click **Edit Configuration**.
6. Enter the path (only, not including the file name) to the inventory agent executable. (For details about installing the FlexNet inventory agent, see *Installing the FlexNet inventory agent* on page 23.)
7. Click **Apply** (at the bottom of this group).
8. In the **UnixHardware** group, click **Edit Configuration**.
9. Select the **True** radio button for **Report UNIX hardware properties**.
10. Click **Apply** (at the bottom of this group).

WindowsLastLoggedOnUser

For a Windows-based computer, standard ADDM collection does not capture any inventory related to the Windows Logon. User identification is important for FlexNet Manager Suite to accurately calculate license consumption for user-based licensed, such as a Named User license.

This pattern queries the registry at `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\LogonUI\LastLoggedOnUser` to find the last logged-on user (for legacy Windows platforms before Vista, it queries WMI for the value of `UserName` from `Win32_ComputerSystem`). If the query is successful, a `DiscoveredRegistryValue` node is created for the host computer:

Property	Value
`Name`	The user name.
`Type`	`FNMP_LastLoggedOnUser`

Property	Value
`Key`	`%type%/%host.key%`

To configure the `WindowsLastLoggedOnUser` pattern:

1. In the user interface for BMC ADDM, select the **Discovery** tab.
2. Click **Pattern Management**, and from the list of patterns at the bottom of the page, ensure that your current `Flexera.FNMP.InventoryRawData` item is enabled, and select it. (The module must be enabled before any of its member patterns can operate.)

   Tip • *The `WindowsLastLoggedOnUser` pattern is enabled by default when the pattern collection is initially enabled. If it has previously been disabled, you can re-enable it with the remainder of this procedure.*

3. From the **Pattern Package** properties, select the link for the **Pattern Module**.
4. Scroll down to the **Pattern Configuration** area.
5. In the **WindowsLastLoggedOnUser** group, click **Edit Configuration**.
6. Select the **True** radio button for **Report Windows last logged-on user**.
7. Click **Apply** (at the bottom of this group).

Part II

App-V Server Adapter

Topics:

- *Architecture, Components, and Prerequisites*
- *Set Up and Operations*
- *Issues and Limitations*
- *Data mapping*

Microsoft App-V (full name Microsoft Application Virtualization) is an application virtualization and application streaming solution. It allows access to applications in three different ways:

- Users may stream applications directly from a central App-V Management Server to their client computers, executing the code locally in light virtual machines that provide a protective 'bubble' around the executing software.
- The applications may be deployed using Microsoft System Center Configuration Manager (SCCM).
- Applications may be deployed in 'stand alone' mode, such as manual delivery through file shares or on a USB stick, without the use of any server infrastructure.

Applications delivered in any of these ways require licensing, and you can manage the appropriate licenses using FlexNet Manager Suite:

- Applications streamed from the App-V Publishing Server(s) are monitored using the App-V server adapter supplied as a standard part of FlexNet Manager Suite, and (for App-V release 5.0 and later) the `AppVMgmtSvr.ps1` PowerShell script installed on the App-V Management Server. Both the App-V server adapter and the `AppVMgmtSvr.ps1` PowerShell script are documented in this chapter.
- Applications deployed using Microsoft SCCM are recorded automatically as part of the standard inventory import from SCCM.
- Applications deployed manually can be recorded manually in FlexNet Manager Suite. Manual deployment is not a recommended best practice because of the inherent difficulties in management and demonstrating compliance, and there is no automation possible through FlexNet direct inventory gathering to cover manual deployment.

Supported versions

The App-V server adapter supports the current version of FlexNet Manager Suite, and releases 4.6, 5.0, and 5.1 of Microsoft Application Virtualization.

Because of significant architectural change between release 4.6 and release 5.0 of App-V, there are matching significant differences in the App-V server adapter for the different versions. It is important to read this document carefully, noting the differences that apply to "release 4.6" and "release 5.0 and later".

1

Architecture, Components, and Prerequisites

Topics:

Following the changes that Microsoft made in the architecture of App-V between release 4.6 and release 5.0, the App-V server adapter is also significantly different when interfacing to the different App-V releases. There are separate chapters for each different architecture. Identify the release of Microsoft App-V in use in your enterprise, and focus on the architecture that is appropriate to that release.

Architecture and Operation for App-V 4.6

This discussion applies to use of Microsoft App-V server infrastructure, streaming applications to App-V clients on end-point devices. (Where applications are instead installed by Microsoft SCCM, use the inventory import from SCCM instead of this adapter.)

In its streaming implementation, Microsoft App-V release 4.6 has three main kinds of components:

- A database (referred to here as the App-V Management Server database), which may be on a separate server
- One or more Management Servers that access the App-V Management Server database and provide a user interface for system control
- One or more streaming servers that may directly deliver application packages.

Of these, only the App-V Management Server database is relevant to the App-V server adapter for FlexNet Manager Suite.

Prerequisites

Operation requires that you have:

- A supported version of Microsoft App-V (see *App-V Server Adapter* on page 40).
- An operational App-V Management Server database.
- A FlexNet inventory beacon that has network access to your App-V Management Server database, and is also able to upload gathered inventory to the central FlexNet Manager Suite server (either directly or through a hierarchy of inventory beacons).
- An inventory beacon importing Active Directory data from the same domain where the App-V server resides. (This may be the *same* inventory beacon that runs the App-V server adapter, but this is not a requirement.)

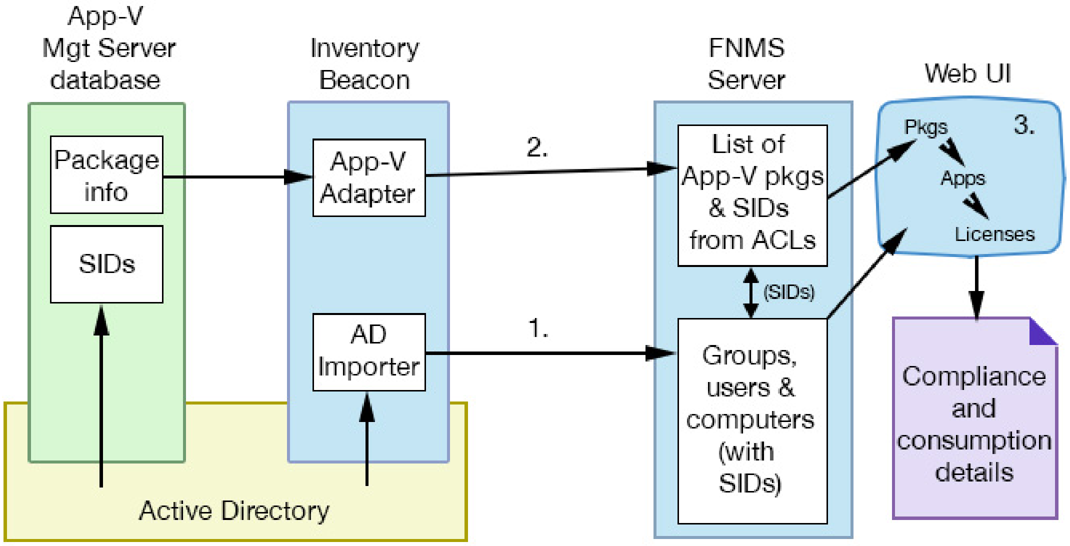
  Tip • *If you have multiple Active Directory domains, ensure that the Active Directory import runs against all the domains in your environment. It is possible for users from other domains to be granted access to App-V packages (and the applications they contain).*

- Operators who can identify the applications represented by the App-V packages, and link those applications to the appropriate licenses.

Tip • *You may have multiple App-V Management Servers, and multiple streaming servers, that link to a single App-V Management Server database. This requires only one connection from the FlexNet Manager Suite App-V server adapter, because this connects only to the database. However, if you have multiple App-V Management Server databases in your estate, configure a separate connection to each of them on appropriate inventory beacons. Where helpful, you may configure multiple such connections (each separately scheduled as you choose) on one inventory beacon.*

In operation

The following diagram shows the operational architecture for the App-V server adapter for App-V release 4.6.


App-V Mgt Server database
Package info
SIDs
Inventory Beacon
App-V Adapter
AD Importer
Active Directory
2.
1.
FNMS Server
List of App-V pkgs & SIDs from ACLs
(SIDs)
Groups, users & computers (with SIDs)
Web UI
Pkgs
3.
Apps
Licenses
Compliance and consumption details

The numbers here refer to the numbers shown in the diagram above:

1. The inventory beacon imports data from Active Directory, including groups (and their members), users, and computers, and the security identifiers for each item within Active Directory. (These security identifiers, or SIDs, are the same identifiers that App-V reports for usage of the applications delivered through App-V packages.)
   - These are immediately uploaded to the central application server for FlexNet Manager Suite.
   - As soon as the upload is completed, the data is imported into the compliance database.
2. On the schedule you specify on the inventory beacon, the App-V adapter:
   - Connects to the App-V Management Server database
   - Imports a list of the App-V packages from the database, and the access control lists (ACLs) that determine which Active Directory groups and users have access to the applications inside the packages. The latter are identified by their security identifiers (SIDs).
   - Immediately uploads the data to the central application server for FlexNet Manager Suite. (If the upload fails for some reason, there is a catch-up upload task that by default is scheduled overnight.)
   - The data waits in the staging area on the central application server for the next scheduled inventory import and compliance calculation (by default, scheduled overnight).
3. When information about a new App-V package is first imported, an operator must identify the package and link it (like installer evidence) to an application record. This work must be done manually because (in release 4.6) App-V packages are opaque about the applications they contain. As well, for any meaningful calculations of consumption, the application must be linked to a suitable license. This linking effort is required only for the first import of each new package.

Once the links are established, each subsequent compliance calculation assigns consumption by the correct users and computers to the appropriate (linked) license. This consumption information is then available both in the management views and in reports.

Architecture and Operation for App-V 5 and Later

This discussion applies to use of Microsoft App-V server infrastructure, streaming applications to App-V clients on end-point devices. (Where applications are instead installed by Microsoft SCCM, use the inventory import from SCCM instead of this adapter.)

In its streaming implementation, Microsoft App-V release 5.0 (and later) has the following components, apart from the App-V clients:

- A database (referred to here as the App-V Management Server database), which may be on a separate server
- A separate reporting database (referred to here as the App-V reporting database), which may also be on a separate server (importantly, this database stores application usage information)
- One or more Management Servers that access the App-V Management Server database and provide a user interface for system control
- One or more Reporting Servers that access the App-V reporting database and provide operational reports to help manage the App-V infrastructure
- One or more streaming servers (called App-V Publishing Servers) that may directly deliver application packages.

Of these, for App-V 5.0 and later, only the App-V reporting database and an App-V Management Server are relevant to the App-V server adapter for FlexNet Manager Suite. (If you are familiar with the adapter for release 4.6 of App-V, notice that we have switched databases, and added the Management Server — the architecture is completely different.)

Prerequisites

Operation requires that you have:

- A supported version of Microsoft App-V (see *App-V Server Adapter* on page 40).
- An operational App-V reporting database.
- An operational AppV Management Server.
- The `AppVMgmtSvr.ps1` PowerShell script installed, configured and scheduled on your AppV Management Server (see *Obtaining (and Deploying) the Adapter Components* on page 50 for details). This is one of the significant changes since the previous adapter.
- A FlexNet inventory beacon that has network access to your App-V reporting database, and is also able to upload gathered inventory to the central FlexNet Manager Suite server (either directly or through a hierarchy of inventory beacons).

- An inventory beacon importing Active Directory data from the same domain where the App-V server resides. (This may be the *same* inventory beacon that runs the App-V server adapter, but this is not a requirement.)

  Tip • If you have multiple Active Directory domains, ensure that the Active Directory import runs against all the domains in your environment. It is possible for users from other domains to be granted access to App-V packages (and the applications they contain).

- Operators who can link the applications identified in the App-V packages to the appropriate licenses.

Tip • You need only one connection from the FlexNet Manager Suite App-V server adapter (on an inventory beacon) to the App-V reporting database. This single App-V reporting database may support multiple App-V Management Servers, and multiple Publishing Servers; but only a single connection to the database is required.

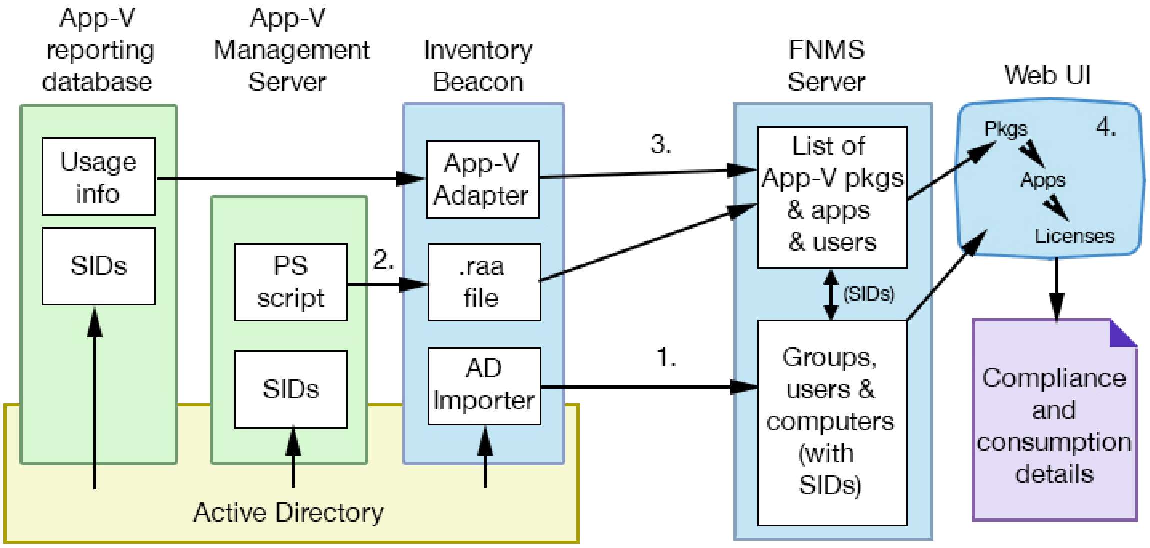
Limitation

For App-V release 5.0 and later, the system supports installation of the `AppVMgmtSvr.ps1` PowerShell script on only one App-V Management Server. Different App-V Management Servers do not self-identify in the `.raa` inventory file, and the App-V reporting database does not identify which application usage information is associated with which App-V Management Server. For these reasons, only a single App-V Management Server (for release 5.0 and later) is supported.

If your App-V (release 5.0 or later) environment has multiple Management Servers, choose one as the data source for App-V packages and the applications they contain. For example, if you have Production, Dev, and Test servers, place the `AppVMgmtSvr.ps1` PowerShell script on the Production App-V Management Server. Also ensure that the App-V server adapter (on an inventory beacon) connects to the matching Production App-V reporting database.

In operation

The following diagram shows the operational architecture for the App-V server adapter for release 5.0 and later.



The numbers here refer to the numbers shown in the diagram above:

1. The inventory beacon imports data from Active Directory, including groups (and their members), users, and computers, and the security identifiers for each item within Active Directory. (These security identifiers, or SIDs, are the same identifiers that App-V reports for usage of the applications delivered through App-V packages.)
    - These are immediately uploaded to the central application server for FlexNet Manager Suite.
    - As soon as the upload is completed, the data is imported into the compliance database.
2. On the schedule you specify on the App-V Management Server, the `AppVMgmtSvr.ps1` PowerShell script:
    - Uses the API to gather a list of the available App-V packages
    - Imports from the database, and the access control lists (ACLs) that determine which Active Directory groups and users have access to the applications inside the packages. The latter are identified by their security identifiers (SIDs)
    - Uploads the collected data in a remote application access (`.raa`) file to its configured inventory beacon, which in turn uploads the file to the central application server for FlexNet Manager Suite.
    - The data waits in the staging area on the central application server for the next scheduled inventory import and compliance calculation (by default, scheduled overnight).
3. On the schedule you specify on the inventory beacon, the App-V adapter:
    - Connects to the App-V reporting database
    - Imports App-V package usage by users and computers. These are all identified by their security identifiers (SIDs).
    - Immediately uploads the data to the central application server for FlexNet Manager Suite. (If the upload fails for some reason, there is a catch-up upload task that by default is scheduled overnight.)

- The data waits in the staging area on the central application server for the next scheduled inventory import and compliance calculation (by default, scheduled overnight).

4. When the compliance calculation is run, FlexNet Manager Suite uses the uploaded SIDs to correlate the various data elements:

    - App-V packages are shown as installer evidence (based on the MSI information uploaded by the `AppVMgmtSvr.ps1` PowerShell script).
    - If an appropriate application record exists (either in the Application Recognition Library or as a locally-created record) with a suitable installer evidence rule, the installed evidence (package) is automatically matched with the application.
    - All users with access to an App-V package are shown as having an installation of the related application on every computer for which the user is either the assigned or calculated user.
    - All computers with access to an App-V package are shown as having an installation of the related application.
    - If the application is linked to a license, consumption is shown for the correct users and computers on that license (or, if it is linked to multiple licenses, on the highest priority license still having unconsumed entitlements). This consumption information is then available both in the management views and in reports. (If this is the first import to reveal an application in an App-V package, an operator needs to link the application record to an appropriate license.)

2

Set Up and Operations

Topics:

This chapter covers the configuration of the adapter, and the work needed to enable application recognition from the imported inventory.

Keep clearly in mind the distinctions required for App-V release 4.6, and release 5.0 and later. For example, *Obtaining (and Deploying) the Adapter Components* on page 50 documents processes needed only for release 5.0 and later; and *Configuring the Adapter* on page 56 requires a distinct database connection in each of the cases.

Obtaining (and Deploying) the Adapter Components

For App-V release 4.6, the App-V server adapter is a standard part of the FlexNet Manager Suite implementation, available on installed inventory beacons for both for cloud and on premises implementations. No action is required to install the adapter.

For App-V release 5.0 and later, the core of the App-V server adapter is also a standard part of the FlexNet Manager Suite implementation, available on installed inventory beacons for both for cloud and on premises implementations. No action is required to install the core App-V server adapter on an inventory beacon.

However, for release 5.0 and later you also need the `AppVMgmtSvr.ps1` PowerShell script for installation on your App-V Management Server. This is available within the Tier 1 adapters archive.

The Tier 1 adapter archive includes content for many adapters, and is updated on the Flexera Software website from time to time.

Start this procedure using a web browser on a computer that has good network accessibility from all the machines needing installations for your adapter.

1. Using the credentials supplied by Flexera Software with your order confirmation (or as renewed since), log into the Product and License Center at *`https://flexerasoftware.flexnetoperations.com`*.
2. On the first page, select `FlexNet Manager Platform`, and on the resulting second page, select the product again.
3. In the list of versions, click the product name for the version you are using (typically the most recent version).
4. In the list of components to download, select the `Adapter Tools.zip` archive, and save this to a convenient location (such as `C:\Temp` on a central, accessible server).
5. Right-click the zip archive, and choose **Extract All...**.
6. In the extracted archive, navigate to `Adapter Tools\App-V Management Server Agent \AppVManagementServer5`.
7. Use your preferred method to deploy the `AppVMgmtSvr.ps1` PowerShell script to your App-V Management Server.

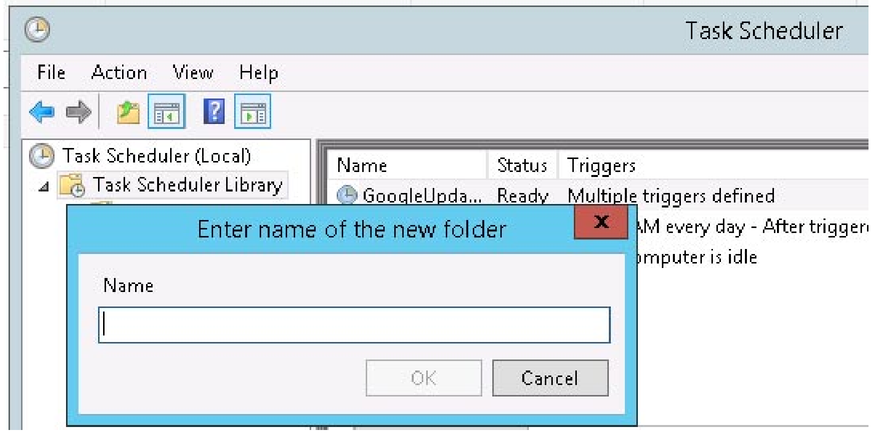
   You may install the script in your preferred folder.
8. Use your preferred task scheduling technology to schedule data collection by the PowerShell script.

   Typically you want the `.raa` file uploaded to the central FlexNet Manager Suite operations databases before the system import and license calculations take place. By default, this occurs daily at 2am central server time. As a two-hour upload buffer should be more than adequate, this suggests (within a single time zone) a data collection trigger at around midnight.

   These example steps are for Windows Server 2012. Adjust as necessary for your server operating system, or your chosen scheduling tool.

   a) In Windows Explorer, navigate to **Control Panel** > **System and Security** > **Administrative Tools**, and double-click **Task Scheduler**.
   The **Task Scheduler** window appears.

b) In the navigation tree on the left, select **Task Scheduler Library**, and then in the **Actions** list on the right, click **New Folder...**.
A dialog appears for entering the folder name.



A suggested value is `FlexNet Manager Suite`.

c) Click **OK**, and select the new folder in the navigation tree.

d) Select **Action** > **Create Task...**.
The **Create Task** dialog appears.

e) Enter an appropriate **Name**, such as `FlexNet Manager Agent for App-V 5.x`, and add any **Description** to help future maintenance of this task.

Your description may be something like `Collects App-V data from the Management Server and uploads to an inventory beacon`.

f) Click **Change User or Group...**.
The **Select User, Service Account, or Group** dialog appears.

g) Enter the account name that is to run the scheduled task, and click **OK**.

An appropriate account:

- Can run a Windows scheduled task on the App-V Management Server
- Can execute the PowerShell script
- Is an App-V Management Server administrator
- May conveniently be a domain account that can upload the results (using HTTP PUT) to the inventory beacon, although a separate account and password can be configured in the command line (required only where that inventory beacon is using Basic Authentication — if the inventory beacon uses anonymous authentication, ignore this requirement).

h) Further down in the **Security options** group, select **Run whether user is logged in or not**.

i) Switch to the **Triggers** tab, and click **New...**.
The **New Trigger** dialog appears.

j) Ensure that the default setting **Begin the task** `On a schedule` is selected, set the parameters for the schedule, and from the **Advanced settings** group, be sure that **Enabled** is selected.

   The suggested schedule is daily at or before midnight local time, but be sure that this suits the upload procedures for your enterprise.

k) Switch to the **Action** tab, and click **New...**.
   The **New Action** dialog appears.

l) Ensure that the default **Action**, `Start a program`, is selected, and browse to your local copy of `AppVMgmtSvr.ps1`.

m) In the **Add arguments (optional)** field, specify all the command-line arguments you need for the agent.

   All command line arguments are documented in *Command Line for PowerShell Script* on page 53. For common implementations, you need to define only the URL to the inventory beacon.

   This example uploads the `.raa` file to the *flexnetbeacon* inventory beacon, using the credentials of the account running the scheduled task.

```
.\AppVMgmtSvr.ps1 -beaconUrl http://flexnetbeacon
```

n) Click **OK**.

o) Optionally, make any preferred adjustments to the **Conditions** or **Settings** tabs (normally the defaults are acceptable).

p) Click **OK** to close the **Create Task** dialog.
   The new task appears in the list of scheduled tasks for this server.

q) Right-click the new task, and click **Run** in the context menu.

   This checks that the scheduled task completes successfully.

r) Validate operations in the following ways:

   - Review the log file (by default, `AppVMgmtSvr.log` in the same folder as the PowerShell script) for any errors or warning messages.
   - Review the content of the output file (by default, `FNMS_AppV.raa` in the same folder as the PowerShell script). This file contains the results of the most recent execution of the PowerShell script, and is replaced at each invocation of the script. For more details of the file format, see *File Format for .raa* on page 55.
   - If uploaded to the inventory beacon, check for the presence of the `FNMS_AppV.raa` output file in `%CommonAppData%\Flexera Software\Incoming\RemoteApplications` on the inventory beacon. Remember that you have only a brief time window to check this before it is uploaded to the central application server and removed (by default, around 10 minutes).

The `AppVMgmtSvr.ps1` PowerShell script is now configured on your App-V release 5.0 or later Management Server. You can now configure the adapter itself that runs on the inventory beacon (see *Configuring the Adapter* on page 56).

Command Line for PowerShell Script

The `AppVMgmtSvr.ps1` PowerShell script is required only when importing inventory from Microsoft App-V release 5.0 or later. (It is not required if you are using App-V release 4.6.)

For applicable releases, `AppVMgmtSvr.ps1` is installed on the App-V Management Server, where, on a schedule that you determine, it collects details of the available App-V packages, and the users and computers that have access to the packages. (Separately, usage information is collected by the App-V server adapter, with its separate configuration described in *Configuring the Adapter* on page 56.)

The following options are supported, both for running `AppVMgmtSvr.ps1` manually and for executing it from a scheduled task or other scheduling tool. None of the options is mandatory, although some are required for normal operation.

Syntax

`AppVMgmtSvr.ps1` [*options*...]

Options

```
-beaconUrl validURL
-logFilePath log_file
-outputFilePath output_file
-password password
-upload $true | $false
-username account
```

where

`-beaconUrl` *`validURL`*	The URL to the appropriate inventory beacon to which the script should upload the generated `.raa` file. Include the protocol (`HTTP` or `HTTPS`), and if your inventory beacon uses a non-default port, include the port number in the standard way.
	Note • *Include only the basic URL of the server. No internal paths are needed, as these details are added automatically by `AppVMgmtSvr.ps1`.*
	There is no default value for `beaconUrl`, so that in production use (when `-upload $true`), a value must be supplied. It can be omitted for local testing on the server when uploading is not required. If `-upload $true` and `beaconUrl` is not set, obviously the upload must fail.
	Example values:
	`http://flexnetbeacon.example.com`
	`https://flexnetbeacon.example.com:499`

`-logFilePath` *`log_file`*	A file path (either absolute, or relative to the folder in which the script is executing) and file name for the log file generated by `AppVMgmtSvr.ps1`. Enclose the path in double quotation marks. A file with consistent name and path over time is replaced

	at each execution, containing only the results of the last execution, thus preventing unmanaged storage requirements. When this option is not specified, the default value is `AppVMgmtSvr.log`, saved in the folder where the script is executing.
`-outputFilePath` *`output_file`*	A file path (either absolute, or relative to the folder in which the script is executing) and file name for the output remote application access (`.raa`) file generated by `AppVMgmtSvr.ps1`. Enclose the path in double quotation marks. If the same name and path is used, the file is overwritten at each run of the script. When the option is not specified, the default value is `FNMS_AppV.raa`, saved in the folder where the script is executing.
`-password` "*`password`*"	The password (in plain text) for the account specified in `username`. Omitted when that option is not required. If specified, enclose the value in double quotation marks. When required and not specified, the script uses details of the account running the process.
`-upload $true | $false`	A Boolean that determines whether to attempt uploading the output file to an inventory beacon identified in `beaconUrl`. When not specified, the default is true, requiring that `beaconUrl` is specified so that the upload can succeed.
`-username` "*`account`*"	The account name used to upload the generated `.raa` file to the inventory beacon identified in `beaconUrl`. This is not required for inventory beacons using anonymous authentication. It may be specified for inventory beacons using Windows Basic Authentication. If specified, enclose the value in double quotation marks. When it is specified, the matching password must be provided in the `–password` option. When required and not specified, the script uses details of the account running the process.

Examples

(Examples here may be line-wrapped for convenient presentation; but should be entered on a single command line.)

Collect the inventory from the App-V Management Server, saving the file locally for inspection:

```
.\AppVMgmtSvr.ps1 -upload $false
```

Similarly, collect the inventory, saving the output and logs to temporary locations to avoid overwriting the normal output:

```
.\AppVMgmtSvr.ps1 -outputFilePath "C:\temp\FNMS_AppV.raa"
                  -logFilePath "C:\temp\AppVMgmtSvr.log"
                  -upload $false
```

Collect the inventory, and upload it to the specified inventory beacon, using the username and password for the account currently running the script:

```
.\AppVMgmtSvr.ps1 -beaconUrl http://flexnetbeacon.example.com
```

Upload the collected inventory to the specified inventory beacon, using the testdomain\administrator account:

```
.\AppVMgmtSvr.ps1 -beaconUrl http://flexnetbeacon.example.com
                  -username "testdomain\administrator"
                  -password "somepassword"
```

File Format for .raa

The AppVMgmtSvr.ps1 PowerShell script interrogates the App-V Management Server, and saves the resulting data in a remote application access file (filename extension .raa), by default called FNMS_AppV.raa. This is an XML file that encapsulates data about each application's App-V package, and the MSI installer information known to App-V.

Here is an excerpt from an FNMS_AppV.raa file (here line-wrapped for presentation):

```
<remoteApplications accessModeID="2">
  <app farmName=""
       appID="9b09bc0d-9634-4838-b0ba-8c256ef4710d"
       appName="Blender"
       appFileName=""
       appFileVersion=""
       appFilePublisher=""
       appFileDesc=""
       userSid=""
       serverName=""
       serverDomainName=""
       isStreamingProfile="1" />
  <msiData farmName=""
           appID="9b09bc0d-9634-4838-b0ba-8c256ef4710d"
           msiDisplayName="Blender"
           msiPublisher="Blender"
           msiVersion="1.0"
           msiProductCode="{DFFFE0C6-3E9A-44E9-9EC1-B5C92DCEE4AF}" />
  <app farmName=""
       appID="5d80bef6-de4a-44f6-b4f8-9aa6a657880e"
       appName="FileZilla_3.2.4.1_win32-setup"
       appFileName=""
       appFileVersion=""
       appFilePublisher=""
       appFileDesc=""
       userSid="S-1-5-21-1336908958-3350896562-3141117955-1690"
       serverName=""
       serverDomainName=""
       isStreamingProfile="1" />
  <msiData farmName=""
           appID="5d80bef6-de4a-44f6-b4f8-9aa6a657880e"
           msiDisplayName="FileZilla Client 3.2.4.1"
           msiPublisher=""
           msiVersion="3.2.4.1"
           msiProductCode="filezilla client" />
 <app farmName=""
       appID="02787044-d434-4e7d-8770-cda46c988de8"
       appName="AdobeReader9"
       appFileName=""
       appFileVersion=""
       appFilePublisher=""
       appFileDesc=""
       userSid="S-1-5-21-1336908958-3350896562-3141117955-1690"
       serverName=""
       serverDomainName=""
       isStreamingProfile="1" />
  <msiData farmName=""
           appID="02787044-d434-4e7d-8770-cda46c988de8"
           msiDisplayName="Adobe Reader 9.1"
           msiPublisher="Adobe Systems Incorporated"
           msiVersion="9.1.0"
```

```
            msiProductCode="{ac76ba86-7ad7-1033-7b44-a91000000001}" />
  ...
</remoteApplications>
```

Some points of interest to note:

- The complete set of MSI attributes needed for FlexNet Manager Suite makes use of Asset Intelligence on the App-V release 5.0+ system, which saves a series of `AssetIntelligenceProperties` in the manifest file. From these properties, `AppVMgmtSvr.ps1` extracts the four `msi...` properties shown for the applications above.
- Not all applications deployed through App-V use MSI, as some applications use third-party installers. App-V packages using third-party installers cannot include the Asset Intelligence properties available through MSI. For such cases, the `AppVMgmtSvr.ps1` PowerShell script interrogates the App-V application registry to get complete installer evidence, populating the same attributes in the `.raa` file. (Thus the presence of `AssetIntelligenceProperties` in the manifest file does not necessarily mean that Asset Intelligence was available; but does mean that equivalent data has been obtained.)
- The `msiDisplayName` becomes the **Name** property of the installer evidence record, the `msiPublisher` maps to **Publisher**, `msiVersion` maps to **Version**. As with all installer evidence, the resulting record is matched against installer evidence "rules" (previously-recorded installer evidence records, most often generalized with judicious use of wild card `%` characters), and when matched, adds an installation count to the application linked to the rule. This means that in listings of installer evidence, the visible entry remains the generalized rule that matched our individual piece of installer evidence. However, the individual installer evidence record is visible by drilling down into the properties of the inventory device on which the inventory evidence was found.
- For a worked example of how the installer evidence for the FileZilla application in the extract above is matched against an installer evidence rule, see *Investigating Issues* on page 64.

Configuring the Adapter

The (core) App-V server adapter is set up on an inventory beacon.

Only two tasks are required for configuring this built-in adapter:

- Specifying the connection to the appropriate database. There are distinct databases in the two versions:
  - For App-V release 4.6, the App-V server adapter connects to the Microsoft App-V Management Server database
  - For App-V release 5.0 and later, the App-V server adapter connects to the Microsoft App-V reporting database.
- Scheduling the imports.

Both tasks are summarized here. Further details are available in the inventory beacon help.

1. On the appropriate inventory beacon, start the inventory beacon interface.

   An appropriate beacon has network access to the appropriate database as described above; and it can upload to the central FlexNet Manager Suite server, either directly or through a hierarchy of inventory beacons.

Tip • Remember that logging into an inventory beacon requires an account with administrator privileges.

2. Select the **Inventory Systems** page in the inventory beacon interface.
3. At the bottom of the page, click **New...**.
   The **Create SQL Source Connection** dialog opens.
4. Complete the values required in this dialog:

Option	Description
Connection name	A descriptive name for this connection that you will recognize later in lists.
Source Type	Select **App-V Standalone**. (Use this same value whether you are connecting to App-V release 4.6, or release 5.0 or later.) *Tip • 'Standalone' means that you are using the adapter to connect directly to the appropriate App-V database, rather than collecting inventory through another source such as Microsoft SCCM.*
Server	Type the server name or IP address. If the database instance you need is not the default one on the server you identify, add the instance name, separated with a backslash character.
Authentication	Select one of: • **Windows Authentication** — Select this option to use standard Windows authentication to access the database server. The credentials of the account (on the inventory beacon) running the scheduled task for importing inventory are used to access the SQL Server database. This account must be added to a security group that has access to the database. • **Windows (specific account)** — Specify an account on the inventory beacon that can make a connection to the SQL database. • **SQL Authentication** — If you select this option, you must then specify an account and password already known to SQL Server on the target database. This account is used to access the database, regardless of the local account running the scheduled task on the beacon server.
Username	The account name used for SQL authentication, or Windows (specific account). (Not required for Windows Authentication.)
Password	The password for the account name required for SQL authentication, or Windows (specific account). (Not required for Windows Authentication.)
Database	Enter the name of the database, or use the pull-down list to select from database names automatically detected on your specified server.

Option	Description
Connection is in test mode (do not import results)	Ensure that this check box is clear for production use. (For more details, see *Managing Microsoft SQL Server Database Connections* on the online help for FlexNet Manager Suite, in the section covering the inventory beacon.) 💡 ***Tip* •** *When using App-V release 4.6, you cannot complete configuration for operation of this adapter (specifically, you cannot map the App-V packages to real applications) until you have run an import in production mode, with this check box clear.*
Overlapping Inventory Filter	If you use more than one inventory source, it is possible to get overlapping inventory (records about the same endpoint device in multiple inventory tools). In the compliance browser (in FlexNet Manager Suite), you may nominate one inventory source as primary. The choices here give more fine-grained control, even when this connection is defined as your primary inventory source: • **Ignore the device's inventory from this data source**: When you have inventory from another source for the same device, the record from this source will be completely ignored. This setting is valuable, for instance, when a device has migrated from one inventory source to another (perhaps by moving offices), but has not yet been obsoleted from this first source. • **Ignore this device's inventory if older than *nn* days**: If you select this option, overlapping inventory collected by this source more than the set number of days before the import is ignored. Fresher overlapping data is still imported and considered for data merging. • **Import the inventory from this source for possible merging**: Choose this option (the default) to declare that overlapping inventory collected from this connection is never considered stale.

5. Click **Test Connection**.

   - If the inventory beacon can successfully connect to the nominated database using the details supplied, a `Database connection succeeded` message displays. Click **OK** to close the message. Click **Save** to complete the addition. The connection is added to (or updated in) the list.

   - If the inventory beacon cannot connect, a `Database connection failed` message is displayed, with information about why that connection could not be made. Click **OK** to close the message. Edit the connection details and retest the connection.

   You cannot save the connection details if the connection test fails. If you cannot get the connection test to succeed, click **Cancel** to cancel the addition of these connection details.

6. If you do not already have a schedule specified that can be used to run the adapter for this connection, create one now (see *Creating a Data Gathering Schedule* in the online help).

7. With the connection for this adapter selected in the **Inventory systems** page, click **Schedule...**.
   The **Select Schedule** dialog opens.

8. From the option list, select the schedule you wish to apply.

   *Tip • As you select each schedule from the list, the area below displays a summary of the schedule settings and the expected **Next run time** for this schedule.*

9. Click **OK** to apply the selected schedule.
10. Click **Save** to store these details.

    The list of connections is updated, and the **Next run** column for your selected connection shows the projected run time from the schedule you just attached.
11. With the connection for this adapter still selected in the **Inventory systems** page, click **Execute now**.
    The adapter collects information from the appropriate database (App-V Management Server database for App-V release 4.6, and App-V reporting database for App-V release 5.0 and later), packages it, and uploads it to the central FlexNet Manager Suite. Allow time for this process to complete before continuing with the next setup procedure.

Import Evidence and Recognize Applications

Because your App-V package naming may be unique to your enterprise, you may need to identify the applications they contain.

If you have many App-V packages, setting up application recognition may be a significant effort on the first import from your App-V server adapter. Once the initial work is done, it is a simpler task to maintain recognition as new App-V packages are put into production.

This procedure continues from the work just completed at the inventory beacon where the adapter runs. Move now to the web interface for FlexNet Manager Suite.

1. Ensure that the upload of data from the App-V server adapter is complete:

   a) In the **Management** view, open the System menu (⚙▼ in the top right corner) and select **Data Inputs**. The **Data Inputs** page appears.

   b) Select the **Inventory Data** tab, and select the **Show details** check box near to top.

   c) Find the App-V server adapter listed for the appropriate inventory beacon, and check its **Last import** date, **Duration**, **Validation issues**, and **Status**.

      When these are appropriate (in particular, **Status** is `Successful` on the appropriate **Last import** date), continue this procedure. Until then, you need either to remediate any upload problems, or simply wait until the upload is completed.

      Tip • This page does not dynamically update results, but shows the status when the page was opened. Therefore, if you are waiting for an upload to finish, refresh the page (`F5`) from time to time to see updated information.

For App-V 5.0 and later, the upload of the `.raa` file saves the contents into staging tables in the compliance database, awaiting the arrival of usage information. For both App-V 4.6 and 5.0 (and later), the upload of the datafile from the App-V server adapter (on an inventory beacon) queues a job with the batch scheduler. When this job can next be processed, the adapter data (which, for App-V 5.0 and later, is combined in this process with the installer evidence from the `.raa` file) is imported into the compliance database. Thereafter, either of two results applies:

- Recognized installer evidence is linked to the application, and shows an installation count for each device on which the evidence was found (and can be further examined on the properties for each of those devices).

  Tip • *The next step, showing consumption against an appropriate license, requires both that the application is linked to the license, and that a reconcile has been run, either automatically on schedule or manually.*

- The installer evidence that was successfully imported but *not* matched to an application record needs your attention to map it to the correct application. (This is more common with the adapter for release 4.6.) To do that, continue with this process.

2. Identify newly-imported evidence (the App-V packages) that requires a link to application records:

   a) Navigate to **License Compliance** >**Discovered Evidence** (in the **Evidence** column).
   The **Discovered Evidence** page appears. Ensure that the default **Installer evidence** tab is selected.

   b) Near the top of the tab, click **Add filter**, and from the pull-down option list select `Type`.
   A second option list appears, listing the possible values of the evidence type.

   c) Select `App-V`.

   d) Click **Add filter** again, and from the pull-down option list select `Assigned`, then choose the value `No`. With both filters defined, click the blue check mark (tick) to apply this filter.
   The list is redrawn to show only evidence of type App-V (the list of App-V packages discovered through the adapter) that have not been matched (manually or automatically) to an application.

   Tip • *You may have no results when these filters are applied. This is a healthy state, meaning that all your App-V evidence is successfully matched to applications. When you have no records here, and want further validation of success, you can drill down into the properties of devices that have a record of installation for the appropriate application.*

3. Use your special knowledge of the App-V packages to link each piece of unassigned evidence to an application record. You may do this either by:

   - Clicking the **Name** of the evidence, which opens the property sheet for this evidence where you can work on the **Applications** tab
   - Following these guidelines to edit locally, still on the **Discovered Evidence** page:

   a) Click anywhere else in a row (other than on the **Name**) to select that App-V package from the list.

The action buttons above the list become active.

b) Click **Assign**.
A blue editor **Assign evidence to an application** appears above the tabs.

c) Click in the search field, optionally enter a few characters from the application name, and click **Search**.
The editing area expands to include a list of application results matching your search. These applications include any previously created in your enterprise, together with all matching applications from the Application Recognition Library regularly updated by Flexera Software.

d) Select your chosen application from the list, and (above the list) click **Add**.
The search field is updated to show the name of the application you selected.

*Tip • If you cannot find the correct application in the search results, you may create a new application record by clicking **Create an application**, and completing the details in the application properties (the App-V package is automatically listed in the **Evidence** tab of the application properties).*

e) Click **Assign**.

- The blue editing area closes.
- The App-V package is linked to the application you chose (the App-V package now functions like installer evidence to show consumption against any license linked to the application).
- In the list of evidence, the **Assigned** column is updated to `Yes`, showing that this package has been linked (or assigned) to an application. (If you currently have a filter on the **Assigned** column to show only rows with a `No` value, the evidence you just assigned must disappear from the list.)

  *Tip • This links the evidence to the application as an exact match across publisher, name, and version records. To protect against future upgrades of the App-V package, you may wish to generalize the version number with a `%` wild-card. For example, if the original version was `1.0`, manually editing the **Version** property of the installer evidence to `1.%` means the link to the application remains valid through all the minor upgrades of this package.*

4. If the selected application is not yet linked to a license, you can:

a) Double-click the App-V package in the evidence list (or, while it is selected, click **Open**).
The evidence property sheet opens.

b) Select the **Applications** tab.

c) In the list of applications, click the name of the one you have just assigned to the App-V package (or double-click elsewhere in the row; or select the row and click **Open**).
The application property sheet opens.

d) Select the **Licenses** tab in the application properties.

e) Optionally enter a few characters of a license name (where you know it); click **Search**.
The search area expands to show the list of available licenses (matching any characters you entered).

f) Select the appropriate license from those offered, and click **Add license**.

Where a suitable license does not already exist, you may instead create one (for example, navigate to **License Compliance** >**Create a license** (in the **Licenses** column).

Tip • *Don't forget to link some purchase records to the license to provide your entitlement count.*

3

Issues and Limitations

Topics:

Diagnosis is covered in this chapter, along with limitations and known issues.

Limitations

The following limitations apply to the current releases of the App-V server adapter:

- In the unlikely event that App-V packages have been shared to non-persistent XenDesktop VDI devices (instead of users), and FlexNet Manager Suite is linked to a XenDesktop broker, no license consumption occurs for those non-persistent devices (because non-persistent VDI devices are not modeled within FlexNet Manager Suite when XenDesktop broker information is available).
- License consumption calculations depend on the integration of data both from Active Directory, and from the appropriate App-V database (App-V Management Server database for release 4.6, or reporting database for release 5.0 and later) and the `AppVMgmtSvr.ps1` PowerShell script. As linking of this data occurs when the App-V server adapter data is importedS, current users, computers, and groups must be imported from Active Directory *first*, before the latest App-V server adapter import occurs. From FlexNet Manager Suite 2014 R2, this ordering is automatic, since Active Directory data is imported to the central database as soon as it is uploaded from an inventory beacon.

  Tip • *If you have multiple Active Directory domains, ensure that the Active Directory import runs against all the domains in your environment. It is possible for users from other domains to be granted access to App-V packages (and the applications they contain).*

- The quality of installer evidence recovered from App-V release 4.6 is not high. You should expect to do remedial work both to generalize the evidence found (for example, using the `%` wild card to generalize version numbering), and to link the evidence to appropriate applications.
- For App-V release 5.0 and later, the system supports installation of the `AppVMgmtSvr.ps1` PowerShell script on only one App-V Management Server. Different App-V Management Servers do not self-identify in the `.raa` inventory file, and the App-V reporting database does not identify which application usage information is associated with which App-V Management Server. For these reasons, only a single App-V Management Server (for release 5.0 and later) is supported.

Investigating Issues

Proof that the App-V server adapter is operational, and the data upload is also successful, can be seen in either or both of two ways:

- Installation records for the appropriate applications against expected inventory devices (possibly including `Remote` devices for which no hardware inventory is available)
- The presence of any newly-discovered inventory of type `App-V` in the **Discovered Inventory** list, typically with an **Assigned** value of `No`.

If neither of these is the case, the issues could be with:

- Imports from the App-V server adapter on an inventory beacon (for both App-V release 4.6, and App-V release 5.0 or later)

- Imports of the `.raa` file from your App-V Management Server (only for App-V release 5.0 or later)
- Missing links between the installer evidence (representing App-V packages) and the applications in the packages
- Missing consumption on an appropriate license.

Each of these is covered in turn below.

No data imports from adapter on inventory beacon

Check the following to identify the problem with imports from the App-V server adapter:

1. Are you sure that there should be new records? Have new packages been brought into production since the last time inventory was collected and fully processed?
2. Check the **Status** of the latest upload (⚙▼ > **Data Inputs** > **Inventory Data** tab > select **Show details**). Also validate that the **Last import** date is as expected, so that the upload occurred *after* new App-V packages were brought into production.
3. If uploads are not happening, move to the inventory beacon where the adapter runs, and check the status of the App-V connection. Use the **Test connection** button to ensure it can connect to the appropriate database (App-V Management Server database for App-V release 4.6, and the App-V reporting database for release 5.0 and later). Details about setting the connection are in *Configuring the Adapter* on page 56.
4. On the same inventory beacon, test its connection to the central application server for FlexNet Manager Suite, using the **Parent connection** page and the **Test connection** button there. (If this inventory beacon is part of a hierarchy, check the connections all the way up the hierarchy to prove that uploads can reach the central server.)
5. Check for stalled uploads by looking for an App-V inventory file in the `%CommonAppData%\Flexera Software\Beacon\IntermediateData` directory on the inventory beacon (or, in a hierarchy, in the chain of inventory beacons). (Notice that the folder for files from the App-V server adapter is separate from the folder for `.raa` files from the PowerShell scripts used with release 5.0 and later.) App-V data files are named in part for the connection you established (see *Configuring the Adapter* on page 56). Once an inventory file of this type is successfully uploaded, it is removed from this intermediate data location on the inventory beacon; so any file in this folder on any server in the hierarchy has not yet been uploaded to the parent server. Upload failures may occur for temporary reasons, such as a network timeout; but there is a catch-up task run overnight to re-attempt uploads of any stalled files.
6. Has a reconciliation calculation occurred since the upload? Until this occurs (normally overnight), new App-V inventory cannot be displayed in the web interface for FlexNet Manager Suite. Check the date and time on the **System Health** page (⚙▼ > **System Health**), in the **License reconciliation** card. The last import and reconciliation must be after the latest upload from the inventory beacon.
7. If you are using App-V release 5.0 or later, a failure to upload and save `.raa` files may also prevent presentation of results, even when the application usage information is successfully imported from the App-V server adapter. Issues with `.raa` files are covered in the next section.

No data imports from the PowerShell script for App-V release 5.0 and later

If uploads of `.raa` files do not appear to be working:

1. Ensure that a new upload is required: that is, that the `AppVMgmtSvr.ps1` script has executed successfully since the last inventory import and license consumption calculation (the most recent file is always saved on the App-V Management Server for checking, and is only replaced the next time that the script executes):
   - Check your scheduling for the script's execution, in particular that the command line options are correct (see *Obtaining (and Deploying) the Adapter Components* on page 50).
   - Check the log file for the last run (you may have renamed or relocated the log file, as described in *Command Line for PowerShell Script* on page 53). In particular, ensure that there were no problems with the file upload to the inventory beacon.
2. Move to that inventory beacon, and check for stalled uploads by looking for an `.raa` file in `%CommonAppData%\Flexera Software\Incoming\RemoteApplications` (this file path is different from the one used for files uploaded by the App-V server adapter). If you have a hierarchy of inventory beacons, check each in turn.
3. If file uploads are happening successfully, has there been a reconcile since the last `.raa` file was uploaded?

   Check the date and time on the **System Health** page (⚙▼ > **System Health**), in the **License reconciliation** card. The last import and reconciliation must be after the latest upload from the inventory beacon. If not, you may (as an administrator) manually trigger a reconcile (see the *FlexNet Manager Suite Help* > *What Is an Inventory Beacon?* > *What Is an Inventory Beacon?*).

Missing application recognition

For App-V release 4.6, data imported from the App-V server adapter includes only the App-V package name and the Active Directory groups (or individuals) that may access the package, according to the Access Control Lists (ACL). Specifically, this imported information cannot recognize what application is hidden within the App-V package. Application recognition requires a separate step. Even for App-V release 5.0 and later, where much better installer evidence allows automatic matching of the evidence rules for the appropriate application, there may be cases where the installer evidence needs manual attention. This will be the case when:

- There is no matching application available within FlexNet Manager Suite, either in application records that you have created locally, or in the Application Recognition Library
- There is an appropriate application, but the values returned from App-V do not match with the existing inventory rules for the application record.

In cases where installer evidence from App-V is unmatched, you can check as follows:

1. First be sure that data uploads and imports are happening, as validated in the previous sections.
2. Navigate to **License Compliance** > **Evidence** column > **Discovered Evidence** > **Installer evidence** tab, filter for **Type**=`App-V`, and check the **Assigned** column. If it displays `No`, you need to link this package to an application, as described in *Import Evidence and Recognize Applications* on page 59.) For App-V release 4.6, newly imported evidence is always unassigned, and requires you to manually associate the evidence (or App-V package) with an application.
3. For App-V release 5.0 (and later), when installer evidence appears in the above listing, it may be worth checking why the data from the App-V installer evidence did not match existing evidence rules for the appropriate application (assuming the application is already present locally or in the Application Recognition Library). To do this, navigate to the **Evidence** tab of the application's properties, where all existing evidence "rules" are listed (making sure that **Installer** is the selected subtab). Compare the data displayed there with

content in your `.raa` file (see sample `.raa` file in *File Format for .raa* on page 55). Here is a worked example of successful matching using the FileZilla 3 application:

App-V element's attribute	Application's installer evidence property
`msiDisplayName="FileZilla Client 3.2.4.1"`	**Name** `FileZilla Client 3.2.%`
`msiPublisher=""`	**Publisher** (blank)
`msiVersion="3.2.4.1"`	**Version** `3.2.%`
`accessModeID="2"` (produces the evidence type `App-V`)	**Type** `Any`

Thus the `.raa` entry produces installer evidence that is immediately matched by the existing installer evidence rule for the application, and produces an installation count against that application. But looking ahead (in imagination) to the day when the `.raa` entry covers a version of `4.2.3.1`, the App-V package data in the `.raa` file would no longer match this evidence rule. At that time, if a new rule was yet to be published in the Application Recognition Library, the installer evidence created from the `.raa` file would appear in the **Discovered Evidence** listing, and you could link it to the application, preferably generalizing it (similarly to the example above) to create a rule that would match several minor releases.

Missing consumption

Showing consumption of license entitlements for App-V packages (and the applications they contain) requires:

- The adapter gathers data from the App-V server and uploads and imports it into FlexNet Manager Suite (see first section above for more details).
- For App-V release 5.0 or later, the `.raa` file is uploaded from your App-V Management Server
- The application is recognized, either automatically, or because you have linked the App-V package to an application record (see previous section)
- The application is linked to a license (and in turn the license should be linked to purchase records to show your legal entitlements) — here, this is left as an exercise for the reader.
- Active Directory imports are current, allowing mapping of the groups and users from the ACLs in the App-V Management Server database to user records in FlexNet Manager Suite.

These notes address the last stage, enabling Active Directory to map from the ACL lists to user records. This process is automatic, provided that all the necessary data is available.

1. On the appropriate inventory beacon, open the **Active Directory** page (from the **Connections** group), and validate that a connection is both established and scheduled. (For details, see *Importing from Active Directory* in the online help.) Review the **Last run** time to see when data was last collected, uploaded, and imported. You may also choose **Execute Now** if imports have been disrupted.
2. Once sufficient time has passed for Active Directory data collection, upload, and import (normally, 30 minutes should be more than adequate), review the list of **All Users** to establish that the expected user names are all available (navigate to **Enterprise** > **Users** group > **All Users**).

Tip • *If you have multiple Active Directory domains, ensure that the Active Directory import runs against all the domains in your environment. It is possible for users from other domains to be granted access to App-V packages (and the applications they contain).*

Note • *You cannot review Active Directory group memberships within FlexNet Manager Suite. Only the resulting list of users is available (along with computers, sites, and subnets).*

Known Issues

The following issues relate primarily to the stability of data available for collection from the App-V server.

Renaming packages

App-V allows you to rename packages (without necessarily changing their contents). When a package is renamed, the App-V application record no longer links to usage records within the appropriate App-V database. This means that the App-V server adapter (and, for App-V release 5.0 and later, the `AppVMgmtSvr.ps1` PowerShell script) cannot import meaningful data for license consumption calculations. As well, the installer evidence generated from the App-V import now has a different name, which may not match evidence rules for the appropriate application. If that is the case, the consumption calculations for any license linked to an application that was linked to the previous installer evidence from App-V drops to zero (from this import connection).

For this reason, it is *strongly recommended* that you do not change the name of existing App-V packages in the App-V Management Server interface. Where renaming is unavoidable, remember that you may need to link the new package (represented as installer evidence in FlexNet Manager Suite) to the application to restore consumption calculations (especially for App-V release 4.6).

Updating package versions

Typical "rules" for applying installer evidence to identify applications use the publisher, the application, and the version recorded in the installer evidence. If you use a version number for your App-V packages (such as 1.0), and link exactly this App-V 'installer' evidence to the application record in FlexNet Manager Suite, then updating the version number of the package on your App-V Server (say, to 1.1) may break the link to the application, and the linked license loses its consumption as a result.

You can avoid this problem (at least for minor version changes) by using wild-cards in the linking of evidence to applications. To do this, after you have linked the App-V evidence to the application:

1. Navigate to the **Installer Evidence Properties** page.
2. Select the **General** tab.
3. Edit the **Version** property, using a wild-card percentage sign (`%`) to generalize the match across multiple versions. For example:
    - The original (say `9.2`) is an exact match for only one version number.

- A value of `9.%` matches all the minor releases of version 9.
- A value of `%` matches all versions of the App-V package with the same name and publisher.

4

Data mapping

Topics:

- *App-V Release 4.6 Data Transfers*
- *App-V Release 5.0 (and Later) Data Transfers*

This chapter covers the relationships between the data fields in the source App-V data and the final locations of the data inside the FlexNet Manager Suite database.

App-V Release 4.6 Data Transfers

This table lists the data extracted from App-V release 4.6, and where it is stored the compliance database in FlexNet Manager Suite.

Imported data is stored in temporary staging tables for data manipulation, and then moved into their final destination tables as noted below. These columns in the compliance database are available for use in custom reports, and the like.

App-V (Table)/Column	FNMS (Table)/Column	Notes
(`REPORTING_CLIENT_INFORMATION`) `host_name`	(`ComplianceComputer`) `ComputerName`	If the computer name already exists in the `ComplianceComputer` table, this device is identified as the consuming device. If it does not already exist, the device is added to the `ComplianceComputer` table with a `ComplianceComputerTypeID` of 4, which (through the `ComplianceComputerType` table) indicates a remote device from which inventory cannot be collected. All such created records display their connection name (as the Inventory Agent), are marked as incomplete records, and are given a fictitious serial number of `1`.
(`REPORTING_CLIENT_INFORMATION`) `host_name`	(`ComplianceComputer`) `ComplianceDomainID`	The domain name is extracted from the host record on the App-V server. If the domain does not already exist in the compliance database, it is added to the `ComplianceDomain` table, and as required the `ComplianceDomainID` may be updated in the `ComplianceComputer` table.
(`APPLICATION_USAGE`) `username`	(`ComplianceComputer`) `ComplianceUserID`	The last logged on user for this computer.
(`VW_APPLICATIONS`) `app_name`	(`InstallerEvidence`) `DisplayName`	The name of the App-V package (which may bear no resemblance to the application inside) is used to identify the evidence record. ***Note** • It is possible for an App-V package to contain more than one application. This makes it impossible to link the App-V evidence to a single application record.*

App-V (Table)/Column	FNMS (Table)/Column	Notes
		Best practice is to make each package contain exactly one licensable application.
(`VW_APPLICATIONS`) `version`	(`InstallerEvidence`) `Version`	The version of the App-V package. Again, this bears no necessary relationship to the version of the application inside the package, and is at the whim of the person preparing the package (for example, it may represent the number of times the application package was modified).
String literal `App-V`	(`InstallerEvidence`) `Publisher`	All App-V evidence is given the publisher name of `App-V`.

App-V Release 5.0 (and Later) Data Transfers

This data is imported from Microsoft App-V release 5.0 and later. It is imported by the combination of the PowerShell script installed on your App-v Management Server, and the App-V server adapter on an inventory beacon that can access the App-V reporting database.

Several staging tables in the compliance database store information uploaded from the App-V reporting database and the `.raa` files. These include:

- `ImportedComputer`
- `ImportedInstalledInstallerEvidenceUsage`
- `ImportedInstallerEvidence`
- `ImportedRemoteApplication`
- `ImportedRemoteApplicationAccess`
- `ImportedRemoteApplicationServer`
- `ImportedRemoteApplicationInstallerData`

Data is held in these tables until the import from the App-V server adapter is complete, and then the resolvers are triggered to combine the data from:

- The most recent Active Directory import(s) across all relevant domains
- The most recent imports of the `.raa` file from all App-V Management Server
- The App-V reporting database.

For this reason, it is important that the import from the App-V reporting database happens last.

The following table lists the elements and their attributes from the `.raa` file, and their final table and column in the compliance database within FlexNet Manager Suite.

XML (Element) / Attribute	FNMS (Table) / Column	Notes
(app) appID	n.a.	Used as a key as required across the temporary tables listed above.
(app) userSid	n.a.	The group SID (from Active Directory) identifying the users and computers with access to the package. This drives the linking of inventory device and user records to the application.
(msiData) msiDisplayName	(InstallerEvidence) DisplayName	Visible as the **Name** in Installer Evidence properties.
(msiData) msiPublisher	(InstallerEvidence) Publisher	Visible as the **Publisher** in Installer Evidence properties.
(msiData) msiVersion	(InstallerEvidence) Version	Visible as the **Version** in Installer Evidence properties.
(msiData) msiProductCode	(InstallerEvidence) ProductCode	Not visible in the web interface.

The following are the views used for source data from the App-V reporting database. It is not possible to give a simple mapping of this source data to columns in particular database tables within FlexNet Manager Suite, because the data is normalized and otherwise processed at each stage. For example, the `username`column collected from `view_ApplicationUsage` in the App-V reporting database is already subject to considerable validation and processing before it is stored in the `lastloggedonuser` column in the staging tables (staging tables are listed above). Thereafter, the user name is correlated with other inventory sources, resulting in further processing before it is used to determine a link between the application and the user. For that reason, this table shows only the source content in the App-V reporting database.

App-V View	Columns
dbo.view_ApplicationUsage	• app_name • app_version • end_time • host_id • start_time • username • version_guid
dbo.view_ClientInformation	• host_id

App-V View	Columns
	• `host_name`
`dbo.view_PackageInformation`	• `host_id` • `package_id` • `version_guid`

Part III

BMC Atrium and Remedy Integration

Topics:

The BMC Atrium adapter is a mechanism for two-way information synchronization between an on-premises implementation of FlexNet Manager Suite, and BMC Software's Atrium configuration management database (CMDB) and Remedy IT Services Management product. Data flows in two directions, which in this document are described from the viewpoint of FlexNet Manager Suite as export and import:

- The adapter exports computer system and recognized product information from FlexNet Manager Suite to the CMDB. (Details of fields exported are available in *Appendix 1: Export of Computers* on page 112 and *Appendix 2: Export of Applications/Products* on page 116.)
- The adapter also imports Asset role, status and owner information from the CMDB and Remedy into FlexNet Manager Suite. (Details of the imported data are available in *Appendix 3: Import of Assets* on page 120.)

Note • *This adapter is only available for on-premises implementations of FlexNet Manager Suite. Specifically, it cannot be used with the cloud implementation.*

This documentation covers the following tested products:

- FlexNet Manager Suite version 2015 R2
- BMC Atrium CMDB version 8.1 (together with Atrium Integrator version 8.1)
- BMC Remedy ITSM Applications version 7.6.04 SP 4.

1

Architecture, Operation, and Prerequisites

Topics:

This chapter provides a framework for your understanding of later, more detailed information.

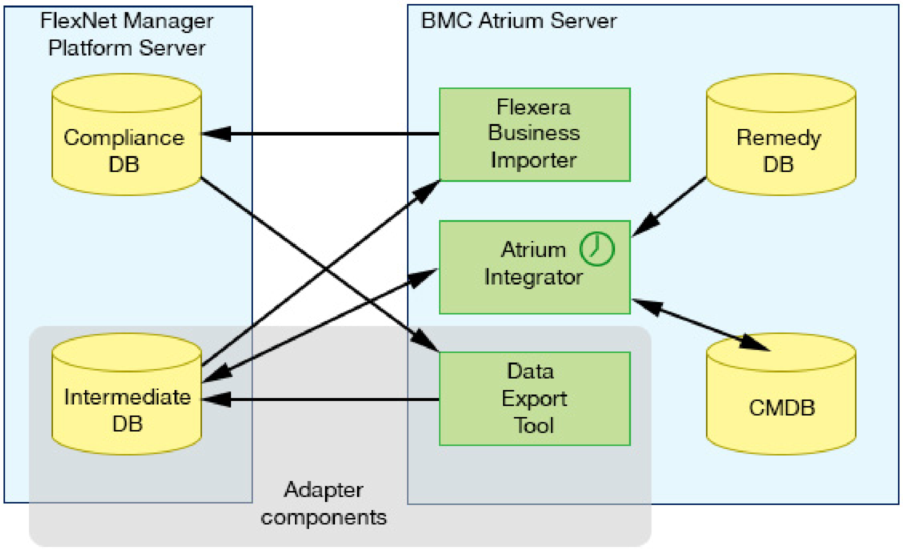
Architecture and Operations

To facilitate the two-way exchange of data between FlexNet Manager Suite and BMC Atrium, components already supplied with both systems are used, and additional elements are supplied with the downloaded adapter.

The following diagram combines the positioning of these elements on different servers, with the data flows for both export and import (as defined in *BMC Atrium and Remedy Integration* on page 75). There are two determining factors in this structural arrangement:

- The schedule for triggering both exports and imports is configured within Atrium, and the Atrium Integrator needs to invoke the data export utility and the FlexNet Business Importer. For this reason, all three components (Atrium Integrator, export utility, and Business Importer) are located on the same server.
- For performance, it is preferable that the intermediate database be on the same database server as your compliance database for FlexNet Manager Suite. (Where you choose a different location, ensure there is a high-speed network connection between the two servers, and a trust relationship for the accounts used during operations.)

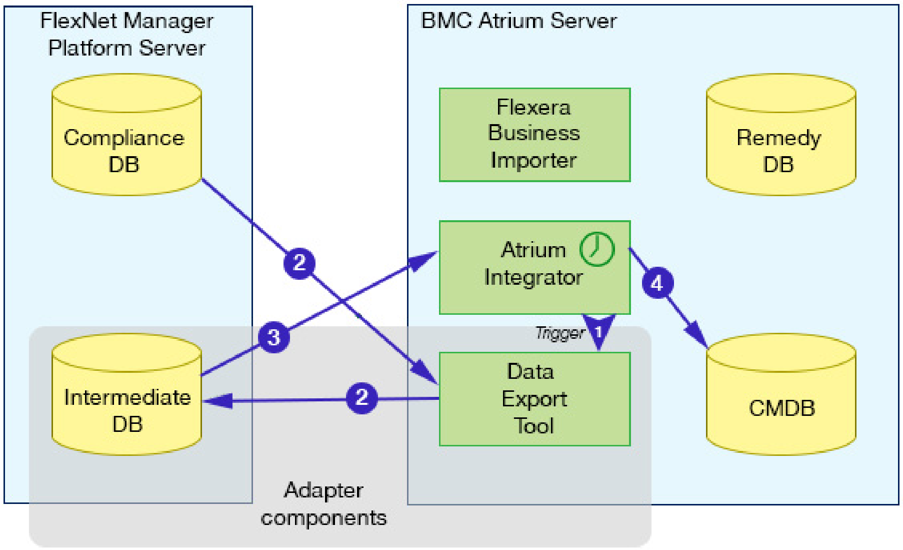
The combination of these two determinants produces the following physical architecture. Net data flows are shown here, with each process also shown separately below.



Once the adapter components are installed and all systems are configured (as described in the chapter *Installing the Adapter* on page 82), operation is fully automatic, triggered by schedules you set during configuration. The two directions of data exchange function independently, and are summarized below.

Export of Computers and Products

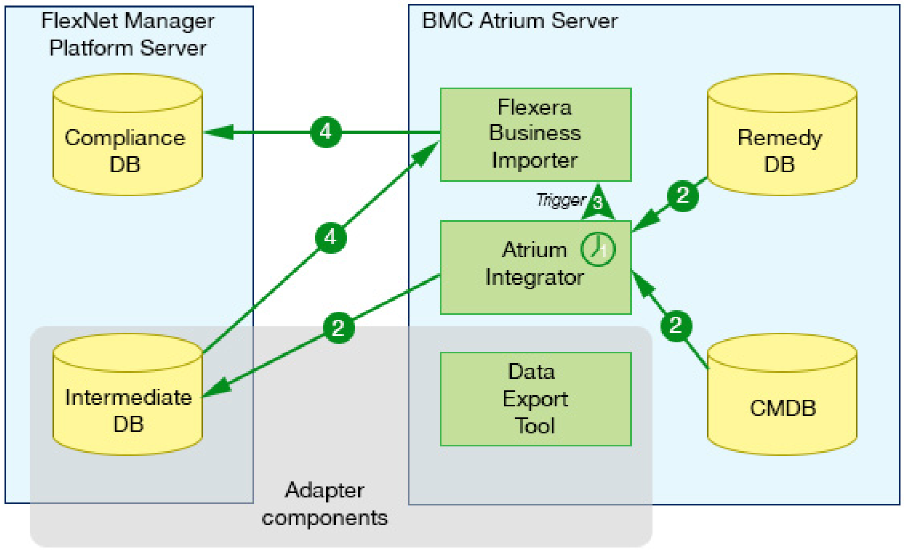
In summary, the process for exporting computer and product information from FlexNet Manager Suite to BMC Atrium (shown below in blue arrows) is as follows:



1. The schedule for export fires in Atrium, and the Atrium Integrator calls the data export tool (installed as part of the adapter).
2. The data export tool reads content from the compliance database for FlexNet Manager Suite, and writes the data into the intermediate database. (It also creates entries in the log file, and sends alert emails if there any errors.)
3. On success, the Atrium Integrator reads the data from the intermediate database, maps it using the defined data transforms to the fields required in Atrium
4. The Atrium Integrator writes the results into the CMDB.

Import of Assets

In summary, the process for importing assets (and their ownership) from BMC Atrium into FlexNet Manager Suite (shown below in green arrows) is as follows:



1. The schedule for import fires in Atrium.
2. The Atrium Integrator reads data from both the Remedy database and the CMDB, and writes the results into the Intermediate database. (It also sends alert emails if there are any errors.)
3. On success, the Atrium Integrator calls the FlexNet Business Importer.
4. The Business Importer reads the data from the intermediate database, transforms it as required, and writes the results into the compliance database of FlexNet Manager Suite.

Prerequisites

The prerequisites for the BMC Atrium adapter include the following:

- On the server where the data utility is to be installed, .NET Framework 4.5 is required. (To validate, open Windows **Programs and Features**, and search for `.NET`.)
- FlexNet Business Importer must be installed on the same server where the Atrium Integrator is installed. This is installed as a part of the FlexNet Business Adapter Studio. Collecting the installer files is covered in *Obtaining the Adapter Components* on page 80, and completing the installation and configuration is detailed in *Install the Business Importer and Link with Remedy* on page 91.
- The adapter requires an intermediate database to be installed, conceptually between the compliance database for FlexNet Manager Suite and the CMDB for BMC Remedy. In practice, this intermediate database may reside on the same database server as the compliance database; or any other convenient server running at least Microsoft SQL Server 2008 that is accessible from both your BMC Atrium server and your FlexNet Manager Suite database server. Scripts to set up the database are included with the adapter, and the process is covered in *Preparing the Databases* on page 84.
- FlexNet Manager Suite version 2015 R2

- BMC Atrium CMDB version 8.1 (together with Atrium Integrator version 8.1)
- BMC Remedy ITSM Applications version 7.6.04 SP 4.

Obtaining the Adapter Components

Here's where to collect the adapter, and the main elements it contains.

It is most convenient to complete this procedure on the computer where you will install the adapter (see *Architecture and Operations* on page 77).

1. Use your account supplied with your original order confirmation to log into the Flexera Software Product and Licensing Center.

   The URL for logging in is *https://flexerasoftware.subscribenet.com*.
2. From the **Product List**, select `FlexNet Manager Platform`, and in the next **Product Information** page, also select `FlexNet Manager Platform` for release 9.2.3.
3. Select `BMCAtriumAdapter (version 2).zip`, and save to a convenient location (such as `C:\temp`) on the server where Atrium Integrator Spoon is installed.
4. Similarly, download the `FlexNet Manager Platform Business Adapter Studio` (`fnmpbas923.exe`) and save on the same server.

   Installation of the Business Adapter Studio is covered in *Install the Business Importer and Link with Remedy* on page 91.
5. Unzip the `BMCAtriumAdapter.zip` archive into `C:\Program Files\Flexera\`.

   ***Important* •** *This path is referenced in various steps in the Atrium jobs and transformations (see file list below). If you do not use the same path, you must edit the Atrium Integrator Spoon jobs and transformations manually to point the shell scripts to different location.*

The archive includes the following key components:

- SQL scripts to:
  - Create the intermediate database (`*archive*\BMCAtriumAdapter\scripts \CreateIntermediateDatabase.sql`)
  - Modify the database for FlexNet Manager Suite (`*archive*\BMCAtriumAdapter\scripts \FNMPDatabaseChanges.sql`)
- Atrium Integrator jobs (`.kjb`) and transformations (`.ktr`):
  - `*archive*\BMCAtriumAdapter\BMCAtriumAdapter.kjb`
  - `*archive*\BMCAtriumAdapter\FNMPExports\Execute FNMP Export Tool.kjb`
  - `*archive*\BMCAtriumAdapter\FNMPExports\FNMPExports.kjb`
  - `*archive*\BMCAtriumAdapter\FNMPExports \ExportComputerProductRelationshipsTransform.ktr`

    - `*archive*\BMCAtriumAdapter\FNMPExports\ExportComputersTransform.ktr`
    - `*archive*\BMCAtriumAdapter\FNMPExports\ExportProductsTransform.ktr`
    - `*archive*\BMCAtriumAdapter\FNMPExports\ExportUpdateIntermediateDBTransform.ktr`
    - `*archive*\BMCAtriumAdapter\FNMPImports\Asset.ktr`
    - `*archive*\BMCAtriumAdapter\FNMPImports\ImportUpdateIntermediateDBTransform.ktr`
- The adapter's main executable (*archive*\BMCAtriumAdapter\FNMPExports\Utility\FNMPDataExportUtility.exe) and its many dependencies, including:
    - Query files (*archive*\BMCAtriumAdapter\FNMPExports\Utility\Queries\Computers.txt, ProductIDs.txt, Products.txt)
    - Adapter configuration file (*archive*\BMCAtriumAdapter\FNMPExports\Utility\FNMPDataExportUtility.exe.config)
- Business Importer adapter definition file (*archive*\BMCAtriumAdapter\FNMPImports\FNMPImportDefinition\Asset.xml).

2

Installing the Adapter

Topics:

Integrating two sophisticated systems likes these involves a number of procedures, documented in the topics shown:

1. Installing the new, intermediate DB used for data manipulation and transfer (see *Preparing the Databases* on page 84).
2. Modify the compliance database in FlexNet Manager Suite with a custom property for additional values (also in *Preparing the Databases* on page 84).
3. Installing and configuring the export utility that brings data out of FlexNet Manager Suite and into the intermediate database (still in *Preparing the Databases* on page 84).
4. Declaring a dataset within Atrium for data exchange, and authorizing an account to use that dataset (see *Reserving an Atrium Dataset* on page 86).
5. Importing the supplied Atrium jobs and data transforms into your Atrium system (see *Importing Atrium Jobs and Transforms* on page 87).
6. Configuring the supplied Atrium jobs with appropriate settings for your environment (see *Configure Atrium Jobs* on page 87).
7. For the import of asset data from Atrium to FlexNet Manager Suite, installing the FlexNet Business Adapter Studio (which brings with it the Business Importer), and integrating the Business Importer with BMC Remedy Mid-Tier (see *Install the Business Importer and Link with Remedy* on page 91).
8. Initially populating the CMBD with configuration items from FlexNet Manager Suite, and then adjusting settings to allow for reuse for future imports (see *Tuning Creation of Configuration Items* on page 98).
9. Schedule regular operations of the adapter (see *Scheduling the Adapter* on page 99).

This chapter also includes details about *Verifying Data Export to BMC Atrium* on page 101.

If you are transferring large datasets, you should also prepare an environment variable as described in *Preparing for Large Datasets* on page 105. (Other configuration options, including relocating or renaming the log file, are also documented in the chapter *Additional Customization of the Adapter* on page 104.)

Preparing the Databases

You have a clear understanding of your physical architecture for the installation (see *Architecture and Operations* on page 77), and have downloaded and unzipped the archive containing the adapter (see *Obtaining the Adapter Components* on page 80).

1. Determine where the intermediate database used by the adapter will reside, and log into that database server using an account that has DB Owner privileges.

   For details, refer back to *Architecture and Operations* on page 77.

2. In Microsoft SQL Server Studio, navigate through the unzipped archive for the adapter to `C:\Program Files\Flexera\BMCAtriumAdapter\scripts`, and run `CreateIntermediateDatabase.sql`.

   Tip • *If the archive cannot be accessed from this database server, first copy the script (or both scripts for a common DB server) from this folder to a location accessible from SQL Server Studio.*

   This SQL procedure creates the intermediate database with the required tables, and pre-populates some fixed values.

3. If the intermediate database is on another server separate from your compliance database for FlexNet Manager Suite, switch over and log into the FlexNet Manager Suite database server, again using an account that has DB Owner privileges.

   You need access to the second SQL script here, so if necessary, copy the second script to this server before continuing.

4. In Microsoft SQL Server Studio, navigate to and execute `C:\Program Files\Flexera\BMCAtriumAdapter\scripts\FNMPDatabaseChanges.sql`.

   This SQL script adds a custom property to the **Details** tab of computer properties in FlexNet Manager Suite where the machine's **Role** can be imported from BMC Atrium.

5. On the server where the data extraction executable will run (likely, the Atrium server), open a command prompt, and navigate to `C:\ProgramFiles\Flexera\BMCAtriumAdapter\FNMPExports\Utility`.

   This executable requires .NET Framework 4.5 installed, a high-speed network access to the database(s), and an account trusted to access both the compliance database and the intermediate database. It must be callable by the Atrium Integrator. If need be, first copy the *entire contents* of the folder shown above (including subdirectories) to a convenient location on the appropriate machine.

   Tip • *To validate that the host in correctly configured with .NET Framework 4.5 or later, execute*

   ```
   FNMPDataExportUtility.exe /help
   ```

   A list of command-line options is displayed. If not, investigate the version of .NET installed.

6. Execute the following (on a single line):

   ```
   FNMPDataExportUtility.exe
                                 /Source "FNMP-DBConnectionString"
                                 /Target "IntermediateDBConnectionString"
   ```

```
                          /Encrypt
```

where:

Parameter	Notes

FNMP-DBConnectionString	The connection string (enclosed in double quotation marks) for the export utility to connection with the compliance database for FlexNet Manager Suite.
	The `Data Source` value must be the fully-qualified domain name of the database server. Where the compliance database is not in the default instance on this server, add the instance name (separated by a slash).
	An example connection string when the default name for the compliance database is `FNMSCompliance`, is (all on one line):
	`Data Source=DBServer.example.com/myInstance;`
	`Initial Catalog=FNMSCompliance;Integrated Security=SSPI;`
	`Connection Timeout=60;Min Pool Size=2;Max Pool Size=20;`

IntermediateDB ConnectionString	The connection string (enclosed in double quotation marks) for the export utility to connection with the intermediate database you recently created. An example connection string when this is on the same server and instance as the compliance database, and the default name for the intermediate database is `IntermediateDB`, is (all on one line):
	`Data Source=DBServer.example.com/myInstance;`
	`Initial Catalog=IntermediateDB;Integrated Security=SSPI;`
	`Connection Timeout=60;Min Pool Size=2;Max Pool Size=20;`

`/Encrypt`	This option encrypts the connection strings when they are saved into the configuration file for the export utility. (The configuration file lives in the same folder as the executable, and is called `FNMPDataExportUtility.exe.config`.)

7. Configure where the export utility sends email notifications about any export errors.

   a) From the same folder, open `FNMPDataExportUtility.exe.config` in a flat text (or XML) editor.

   b) Locate the following section and replace the values shown:

```
<appender name="SmtpAppender" type="log4net.Appender.SmtpAppender">
 <to value="toaddress@somedomain.com"></to>
 <from value="fromaddress@somedomain.com"></from>
 <subject value=" Atrium Integration Adapter Error"></subject>
 <smtpHost value="smtp.somedomain.com"></smtpHost>
```

   where:

Value	Notes
toaddress @somedomain.com	Valid email address for the person who is to receive alerts when errors occur during export.

Value	Notes
fromaddress @somedomain.com	Valid email address for an account known on the specified SMTP host that is identified as the sender of the email when an error occurs.
smtp.somedomain.com	The fully qualified domain name of the SMTP email server.

a) While here, you may want to customize the location of the log file for the export utility:

```
<log4net>
  <appender name="RollingFileAppender" type="log4net.Appender.RollingFileAppender">
    <file value="C:\Log.txt" />
```

a) Save the edited file, and exit the editor.

Reserving an Atrium Dataset

You must specify a dataset for information exchange, and ensure you have an account with write access to that dataset.

This procedure sets up a data area within Atrium reserved for data exchange with the FlexNet Manager Suite adapter. Your Atrium Administrator may assist with an account that had permissions to access this dataset. You will reference this dataset later in the installation of the adapter.

1. In the BMC Remedy User application, open the CMDB console.
2. Select **Reconciliation Manager** > **Create** > **Dataset**.
   A **Dataset Information** form appears.
3. Complete the following parameters:

Dataset name	A friendly name for the dataset that you will recognize later in other listings.
Dataset ID	The system identifier for the dataset. *Tip • This value is needed when you configure your Atrium jobs.*
Accessibility	Use the option list to set this to `Writable`.
DatasetType	Use the option list to set this to `Regular`.

4. Click **Save**.
   The new dataset is created.
5. Consult with your Atrium Administrator about an account with write access to the created dataset. This account must be used on the Atrium side to run the adapter.

For more information about creating the dataset, see *http://discovery.bmc.com/confluence/display/81/Configuring+the+BMC+Atrium+Adapter*.

Importing Atrium Jobs and Transforms

You need to get the supplied files into the Atrium repository, configured for your local connections.

For this procedure, you must be logged into your Atrium server with administrator privileges. This server needs access to the unzipped archive for the adapter (in this procedure, shown as on the same server).

1. Open the Atrium Integrator Spoon tool.
2. Select **File** > **Open URL...**, navigate to and open `C:\Program Files\Flexera\BMCAtriumAdapter\FNMPExports\ExportComputersTransform.ktr`.
3. Modify the connection strings for each of:
    - Atrium_CMDB
    - AR Server
    - IntermediateDB.
4. Click **File** > **Save**.

Repeat this procedure for each of the following files *in the order listed:*

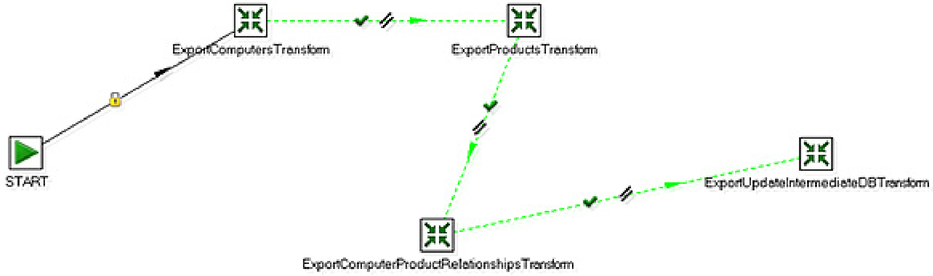
1. C:\Program Files\Flexera\BMCAtriumAdapter\FNMPExports\ExportProductsTransform.ktr
2. C:\Program Files\Flexera\BMCAtriumAdapter\FNMPExports\ExportComputerProductRelationshipsTransform.ktr
3. C:\Program Files\Flexera\BMCAtriumAdapter\FNMPExports\ExportUpdateIntermediateDBTransform.ktr
4. C:\Program Files\Flexera\BMCAtriumAdapter\FNMPImports\ Asset.ktr
5. C:\Program Files\Flexera\BMCAtriumAdapter\FNMPImports\ ImportUpdateIntermediateDBTransform.ktr
6. C:\Program Files\Flexera\BMCAtriumAdapter\FNMPExports\FNMPExports.kjb
7. C:\Program Files\Flexera\BMCAtriumAdapter\FNMPExports\ Execute FNMP Export Tool.kjb
8. C:\Program Files\Flexera\BMCAtriumAdapter\BMCAtriumAdapter.kjb

Configure Atrium Jobs

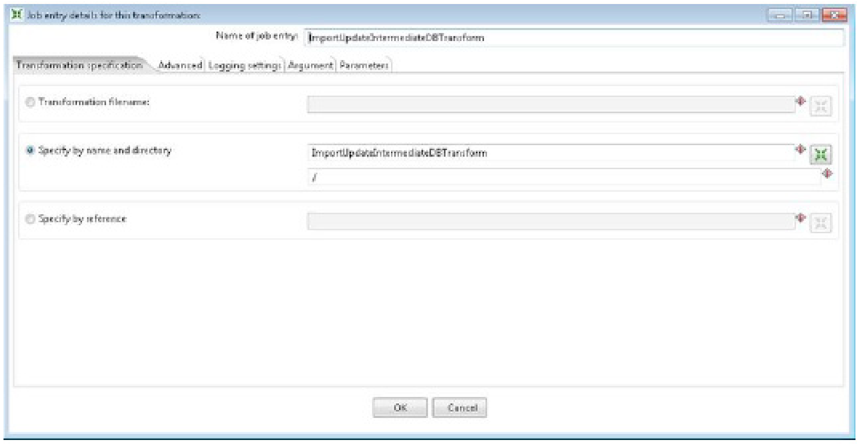
In Atrium, jobs must be linked to transforms.

You have completed the import of both tasks and jobs into the Atrium repository (as in *Importing Atrium Jobs and Transforms* on page 87). Now configure the jobs, still logged into Atrium Spoon.

1. Select **File** > **Open...**, and select `FNMPExports.kjb` from the repository.
   A map of the transformations required for the job is displayed.


ExportComputersTransform
ExportProductsTransform
START
ExportUpdateIntermediateDBTransform
ExportComputerProductRelationshipsTransform

2. Right-click `ExportComputersTransform`, and from the context menu select **Edit job entry**.
   The **Job entry details for this transformation** dialog appears. The transform name as recorded in the job entry appears in the first field, **Name of job entry**.



Job entry details for this transformation

Name of job entry: ImportUpdateIntermediateDBTransform

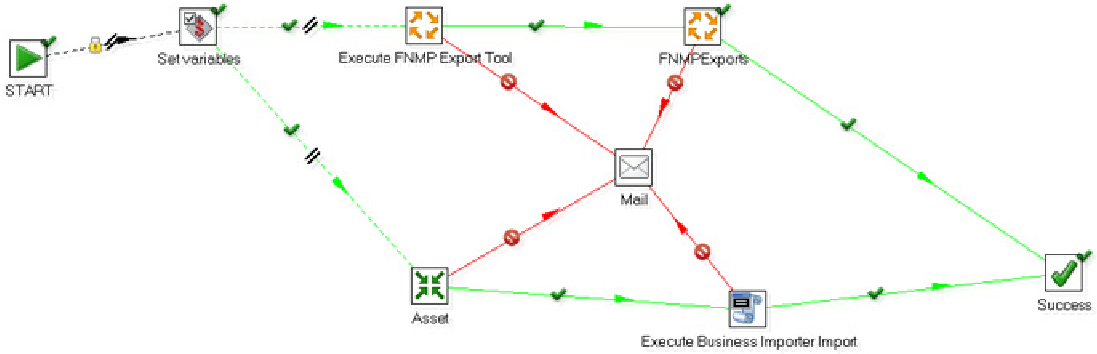
○ Transformation filename:

● Specify by name and directory: ImportUpdateIntermediateDBTransform
/

○ Specify by reference

3. Select the **Specify by name and directory** option.
4. To provide a value in the associated field, browse the repository and select the transform name matching the first field.

   For example, when processing the job entry for `ExportComputersTransform`, select the transform of the same name from the repository.
5. Click **OK**.
6. Repeat steps 2-5 for each of the remaining job entries in this chart:
   - `ExportProductsTransform.ktr`
   - `ExportComputerProductRelationshipsTransform.ktr`
   - `ExportUpdateIntermediateDBTransform.ktr`.
7. Select **File** > **Save**.
8. Select **File** > **Open...**, and select `BMCAtriumAdapter.kjb` from the repository.
   A map of the job is displayed.

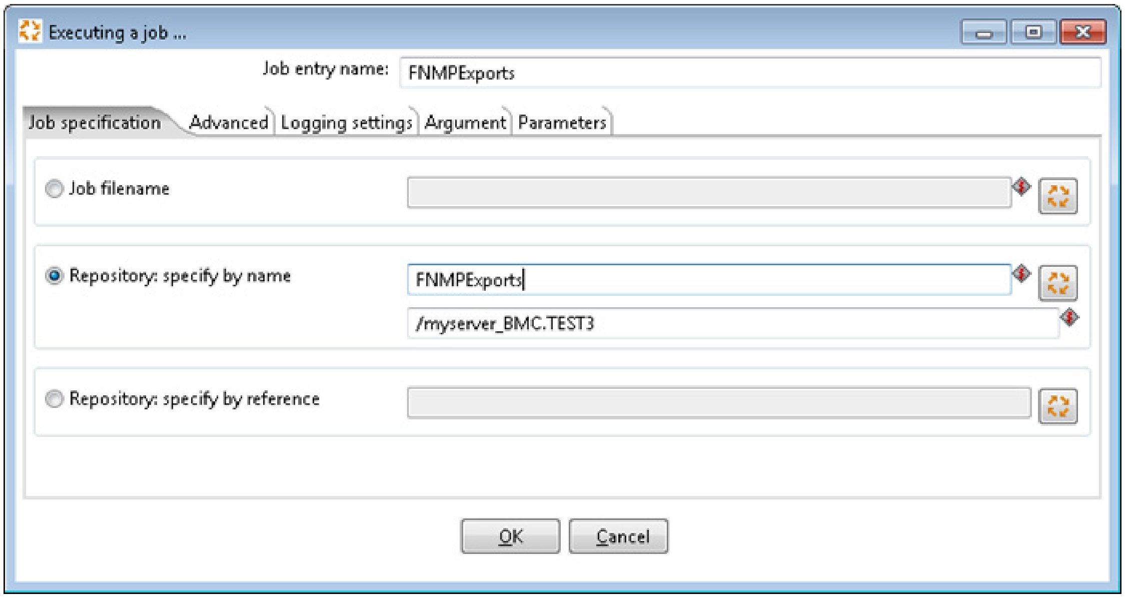


9. Repeat steps 2-5 for the transform in this chart:

    - Asset.ktr.

10. Double-click the `FNMPExports` job.

    The **Executing a job** dialog appears. The **Job entry name** appears as the first field.



Executing a job ...

Job entry name: FNMPExports

○ Job filename

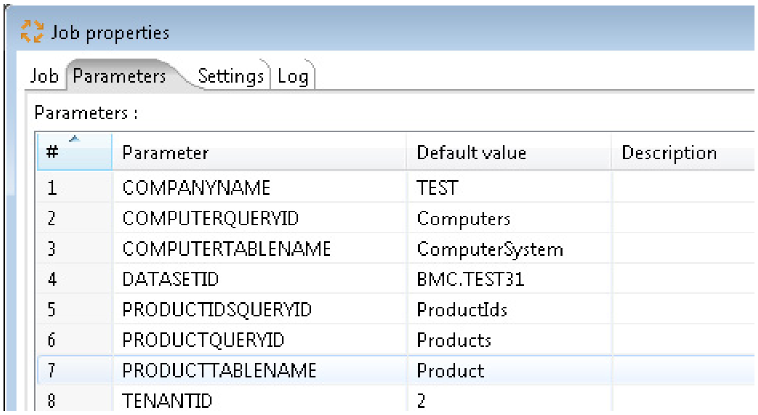
● Repository: specify by name: FNMPExports
/myserver_BMC.TEST3

○ Repository: specify by reference

11. Select the **Respository: specify by name** option.
12. To provide a value in the associated field, browse the repository and select the job name matching the first field.
13. Click **OK**.
14. Repeat steps 10-13 for the remaining job in this chart:

    - `Execute FNMP Export Tool.kjb.`

15. Select **File** > **Save**.
16. Either press `Ctrl-J`; or right-click the `BMCAtriumAdapter` diagram background, and from the context menu select **Job settings**.
    The **Job properties** dialog opens.



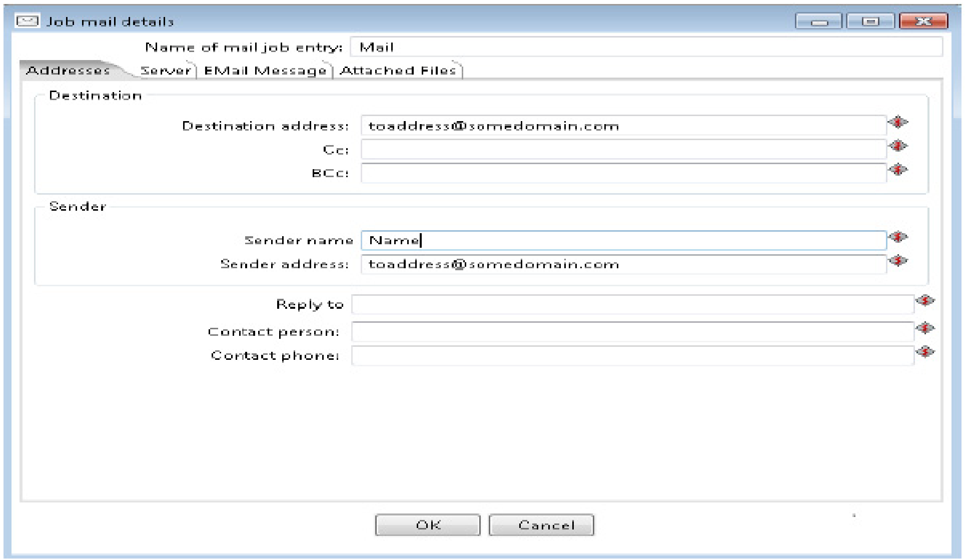
Parameters :

#	Parameter	Default value	Description
1	COMPANYNAME	TEST	
2	COMPUTERQUERYID	Computers	
3	COMPUTERTABLENAME	ComputerSystem	
4	DATASETID	BMC.TEST31	
5	PRODUCTIDSQUERYID	ProductIds	
6	PRODUCTQUERYID	Products	
7	PRODUCTTABLENAME	Product	
8	TENANTID	2	

17. Edit the following values:

Option	Description
`COMPANYNAME`	Enter your company name as it is recorded in the CMDB. (A match is required.)
`DATASETID`	Enter the ID for the dataset that you created for the adapter (see *Reserving an Atrium Dataset* on page 86).

18. Select **File** > **Save**.
19. To configure email alerts for errors occurring on the Atrium side of the adapter:
    a) In the `BMCAtriumAdapter` job, right-click on the `Mail` step in the center.
    b) From the option menu, select **Edit job entry**.
    The **Job mail details** dialog appears.



Name of mail job entry: Mail

Destination address: toaddress@somedomain.com

Sender name Name

Sender address: toaddress@somedomain.com

a) In the **Addresses** tab, insert the following:

- **Destination address** - the person who is to receive the email alert for each error.
- **Sender name** - the value to appear on the emails as the sender.
- **Sender address** - The email address from which the email will appear to come. This must be a valid email account known to the mail server (specified next).

b) Switch to the **Server** tab, and complete the **SMTP Server** details.

c) Click **OK**.

20. Select **File** > **Save**.

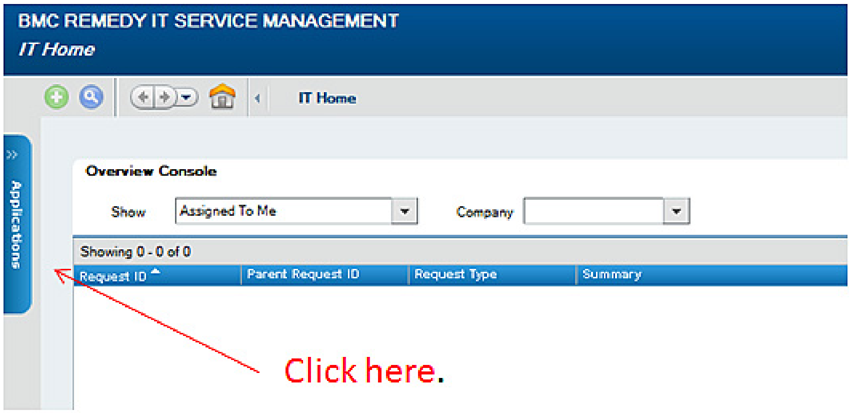
Install the Business Importer and Link with Remedy

The Business Importer performs parts of the integration work, writing data from Remedy into the FlexNet Manager Suite database.

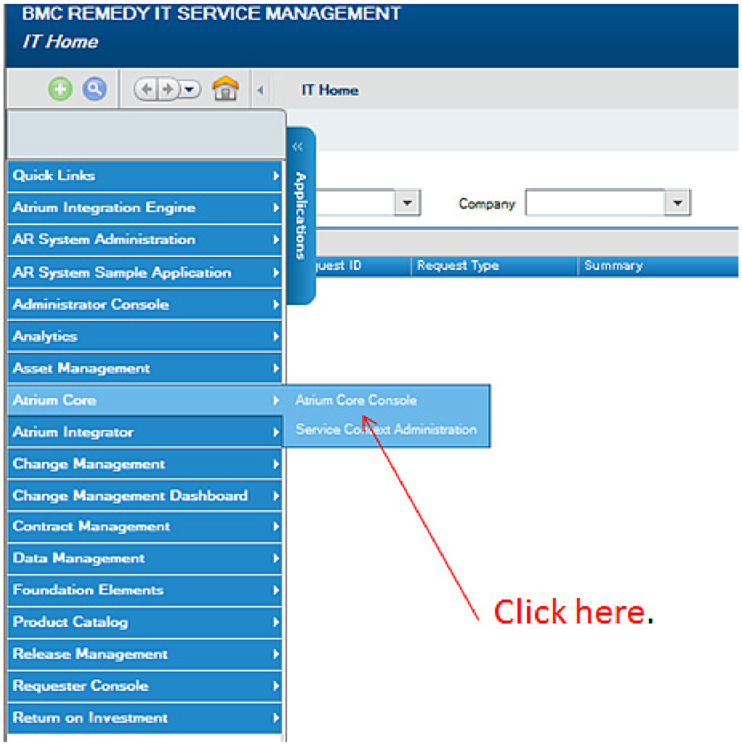
The Business Importer is installed as part of the Business Adapter Studio, a stand-alone environment you can use to build business adapters. These 'adapt' the format of data from other business systems in your computer estate for import into FlexNet Manager Suite, and are exercised by the Business Importer. You downloaded the installer for the Business Adapter Studio in *Obtaining the Adapter Components* on page 80. To enable the business import process, you also configure the reconciliation process to link with the Business Importer. Since the Atrium Integrator must trigger the Business Importer, the normal architecture is to install the Business Adapter Studio on the Atrium server.

1. On the Atrium server, navigate to and execute `fnmpbas923.exe`.
   The installation wizard for Business Adapter Studio opens.

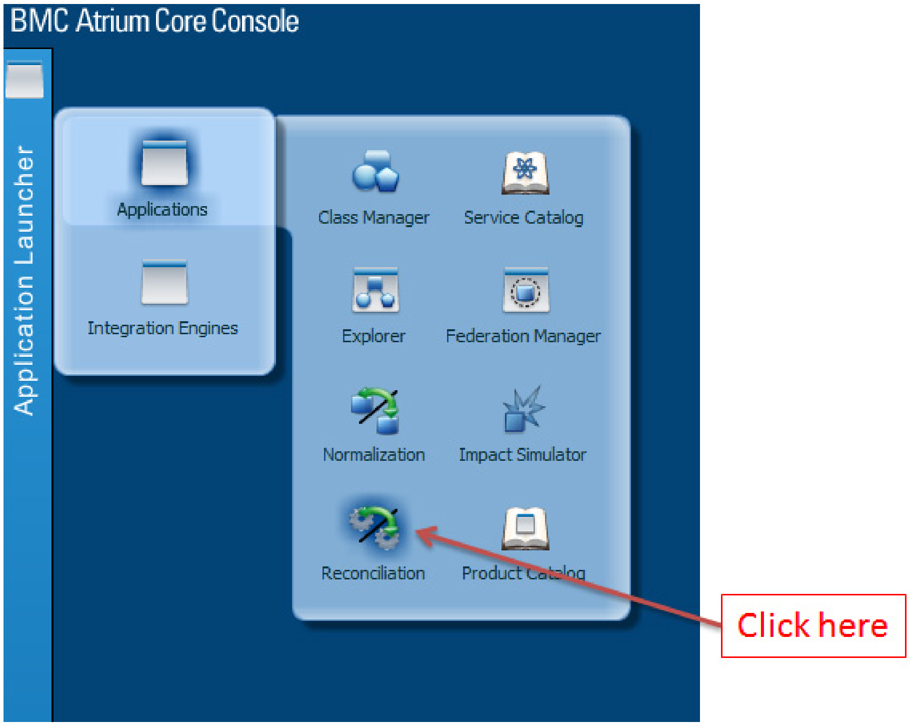
2. Complete the installation wizard using the default values it provides.
3. Login to BMC Remedy Mid-Tier using an account name and password provided by your BMC Administrator.
   The **IT Home** page appears.



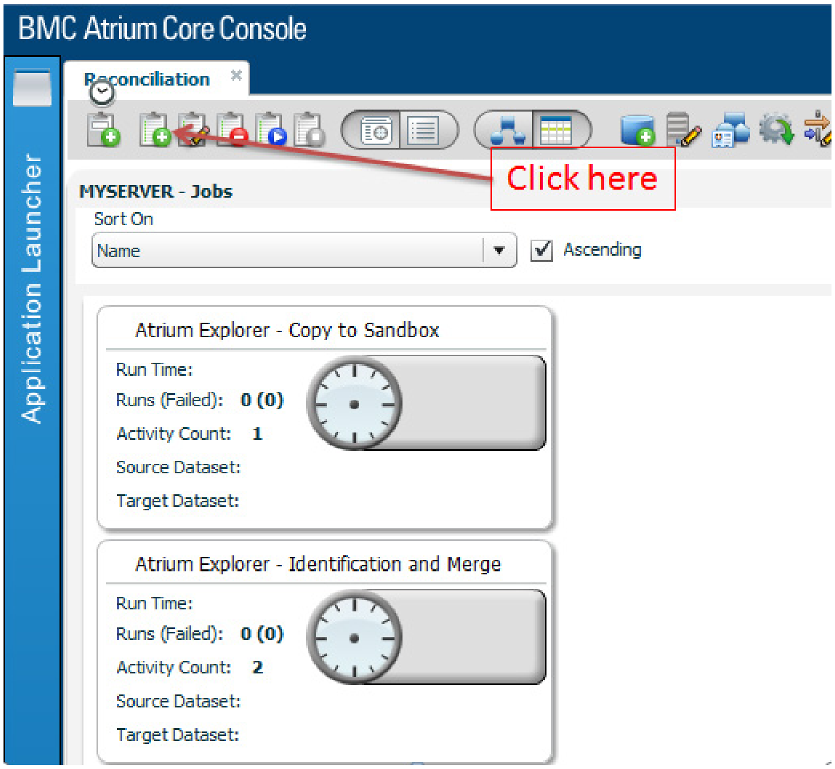
4. Click the **Applications** tab on the left boundary.
   The Applications menu expands.



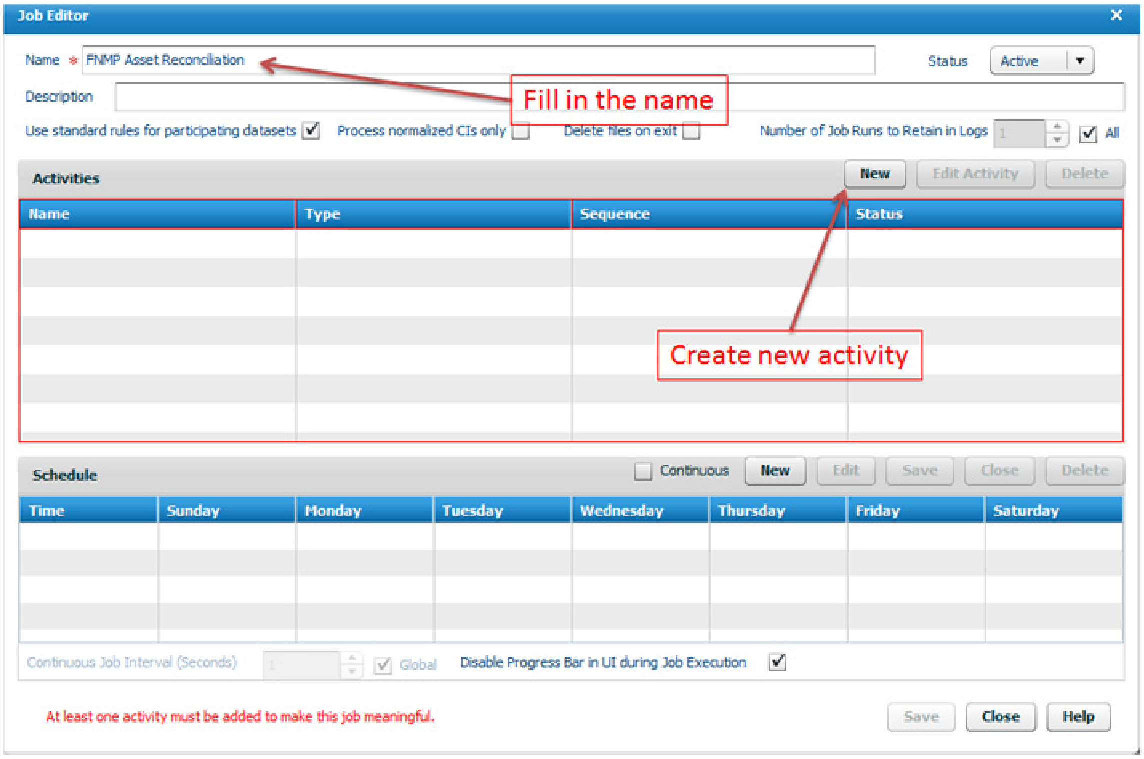
5. Select **Atrium Core**, and in the sub-menu select **Atrium Core Console**.
   The **BMC Atrium Core Console** window opens.



6. Click the **Application Launcher** tab on the left edge, then select **Applications** from the menu, and in the sub-menu select **Reconciliation**.
   The **Reconciliation** tab opens.



7. Click the new job icon in the toolbar.
   The **Job Editor** page appears.



Job Editor

Name * FNMP Asset Reconciliation

Status: Active

Description

Use standard rules for participating datasets ☑ Process normalized CIs only ☐ Delete files on exit ☐ Number of Job Runs to Retain in Logs 1 ☑ All

Activities

Name	Type	Sequence	Status

Schedule ☐ Continuous

Time	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

Continuous Job Interval (Seconds) 1 ☑ Global Disable Progress Bar in UI during Job Execution ☑

At least one activity must be added to make this job meaningful.

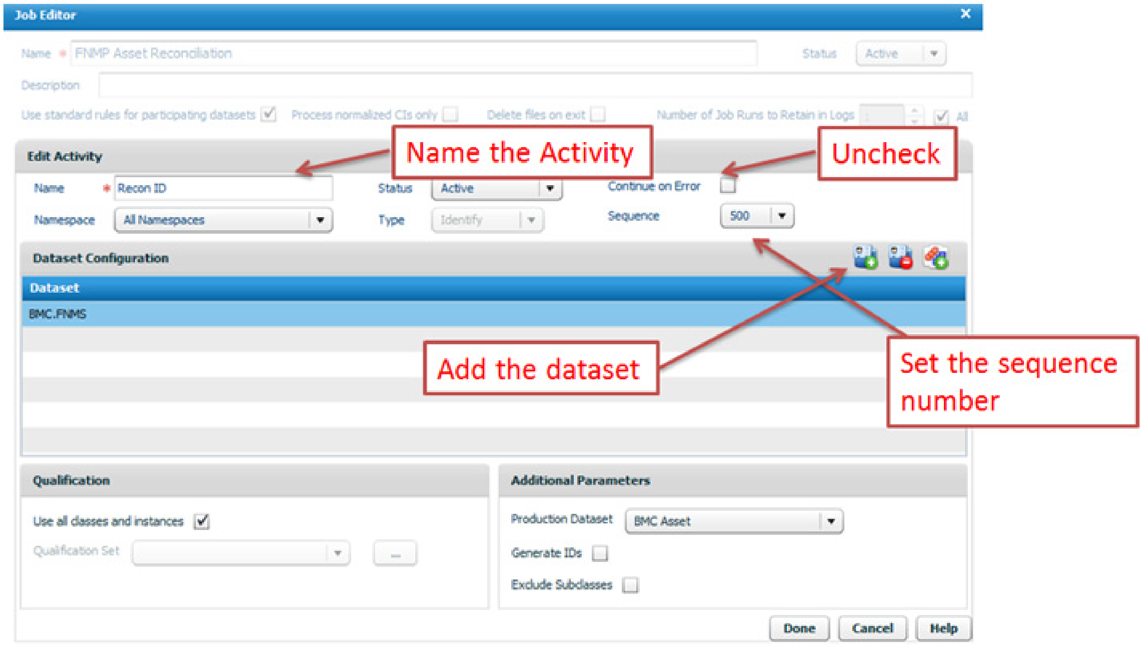
Fill in the name

Create new activity

8. In the **Name** field, provide a name for this job that you will recognize later.

   Suggested value: `FNMP Asset Reconciliation`.

9. In the **Activities** group, click **New**.
   The **Activities** group changes to the **Edit Activity** area.



Job Editor

Name * FNMP Asset Reconciliation

Status Active

Description

Use standard rules for participating datasets ☑ Process normalized CIs only □ Delete files on exit □ Number of Job Runs to Retain in Logs ☑ All

Edit Activity

Name * Recon ID

Status Active

Continue on Error □

Namespace All Namespaces

Type Identify

Sequence 500

Dataset Configuration

Dataset
BMC.FNMS

Qualification

Use all classes and instances ☑

Qualification Set

Additional Parameters

Production Dataset BMC Asset

Generate IDs □

Exclude Subclasses □

Name the Activity

Uncheck

Add the dataset

Set the sequence number

10. Complete the details for the first (Identify) activity:

a) In the **Name** field, provide a name you will recognize later.

Suggested value: `Identify`.

a) Ensure that the **Continue on Error** check box is clear.

Recall that the system is configured to send emails on any errors in exporting the assets to FlexNet Manager Suite.

b) Set the **Sequence** number to control ordering of activities in the job.

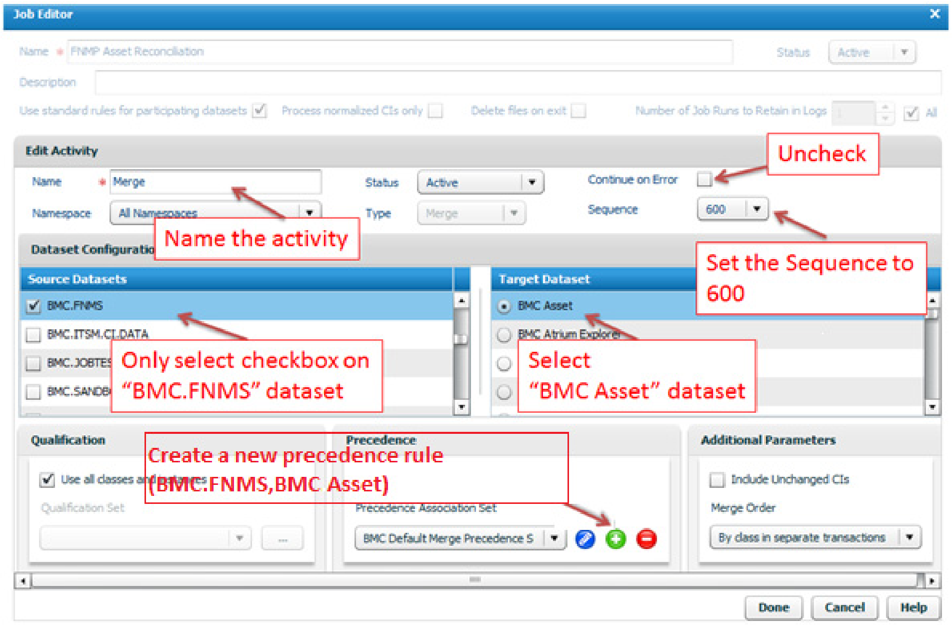
Recommendation: Number in hundreds to allow for later changes. Suggested value: `500`.

a) In the **Dataset Configuration** area, click the first icon to add a new dataset, and select the one you created for this exchange.

The suggested value was `BMC.FNMS`.

b) Click **Done** to save details of this activity.
The **Activities** group reappears, showing this first activity in the list.

11. Once again, in the **Activities** group, click **New**, and complete the details for a merge activity:



Job Editor

Name * FNMP Asset Reconciliation

Status: Active

Description

Use standard rules for participating datasets ☑ Process normalized CIs only □ Delete files on exit □ Number of Job Runs to Retain in Logs 1 ☑ All

Edit Activity

Name * Merge

Namespace: All Namespaces

Status: Active

Type: Merge

Continue on Error □

Sequence: 600

Dataset Configuration

Source Datasets	Target Dataset
☑ BMC.FNMS	● BMC Asset
□ BMC.ITSM.CI.DATA	○ BMC Atrium Explorer
□ BMC.JOBTES	○
□ BMC.SANDB	○

Qualification: ☑ Use all classes and instances; Qualification Set

Precedence: Precedence Association Set: BMC Default Merge Precedence S

Additional Parameters: □ Include Unchanged CIs; Merge Order: By class in separate transactions

Uncheck

Name the activity

Set the Sequence to 600

Only select checkbox on “BMC.FNMS” dataset

Select “BMC Asset” dataset

Create a new precedence rule (BMC.FNMS,BMC Asset)

a) Add a **Name** for the activity.

Suggested value: `Merge`.

a) Ensure that the **Continue on Error** check box is clear.

b) Set the **Sequence** number to a value higher than the identification step.

Suggested value: `600`.

c) From the list of **Source Datasets**, select only the one identified in the previous activity.

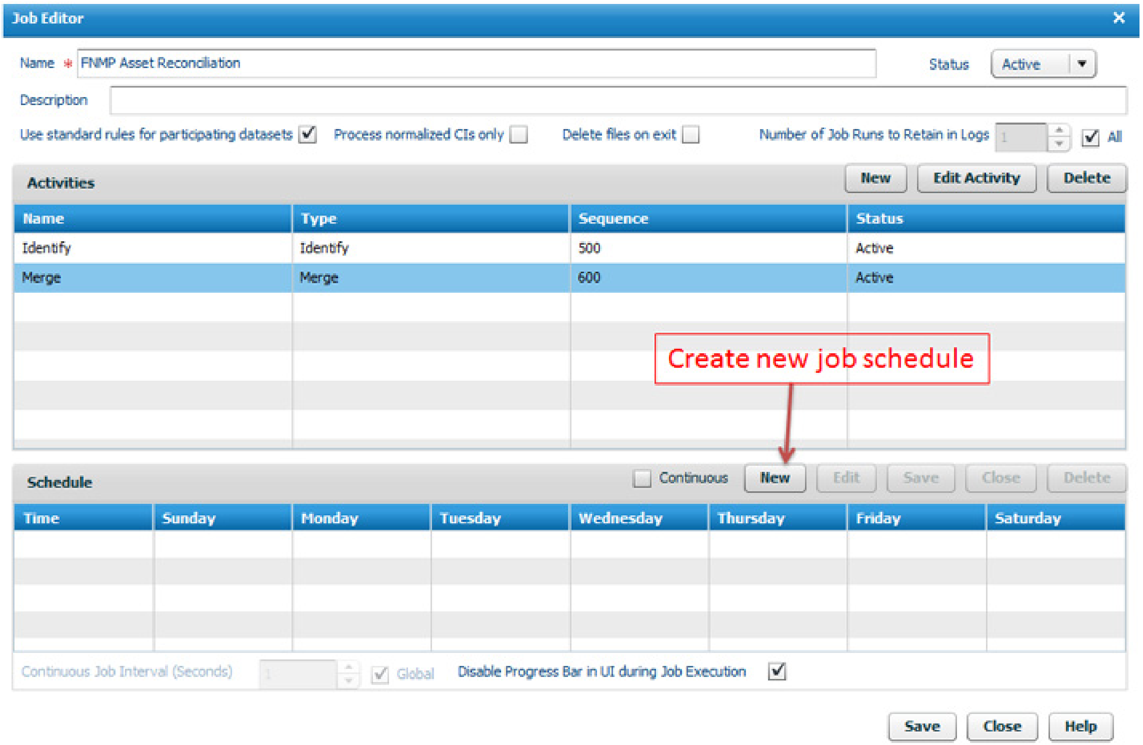
The suggested value was `BMC.FNMS`.

d) From the list under **Target Dataset**, select the `BMC Asset` dataset.

e) In the **Precedence** group, click the + icon to add a new **Precedence Association Set**.

Suggested value: `BMC.FNMS,BMC Asset`.

f) Click **Done** to save details of this activity.
The **Activities** group reappears, showing your two activities.



Name * FNMP Asset Reconciliation

Status Active

Description

Use standard rules for participating datasets ☑ Process normalized CIs only □ Delete files on exit □ Number of Job Runs to Retain in Logs 1 ☑ All

Activities

Name	Type	Sequence	Status
Identify	Identify	500	Active
Merge	Merge	600	Active

Create new job schedule

Schedule

□ Continuous

Time	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

Continuous Job Interval (Seconds) 1 ☑ Global Disable Progress Bar in UI during Job Execution ☑

12. In the **Schedule** group, click **New**.

    Set the values to suit your business processes, with a suggested frequency being once every 2-4 weeks.

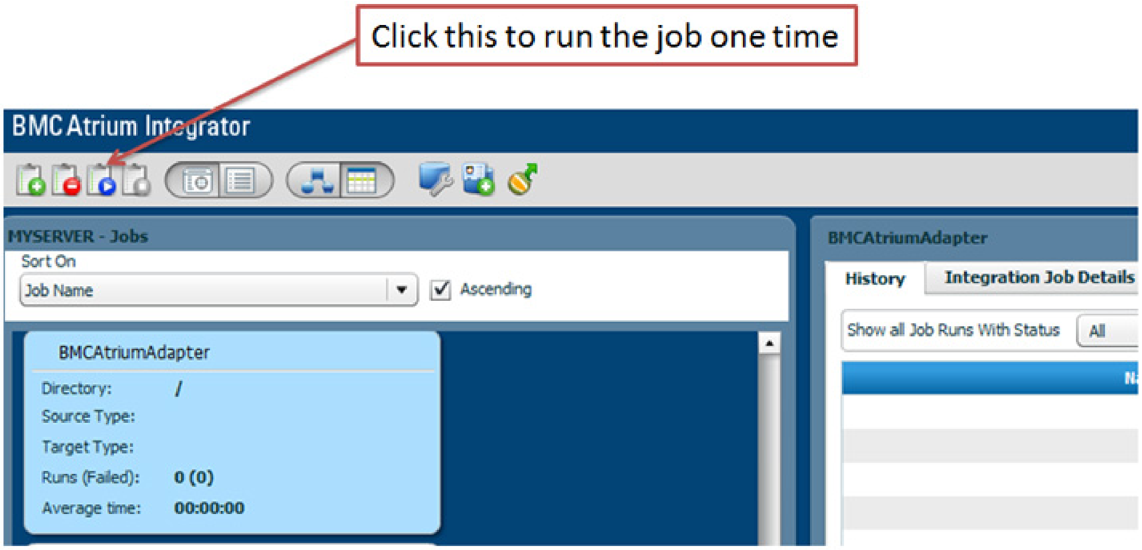
13. At the bottom of the **Job Editor**, click **Save** to store your job details in the repository.

Tuning Creation of Configuration Items

Atrium requires an initial run to populate the database, and then customization for regular scheduled operations.

In the first run of the adapter, all the products and computers exported from FlexNet Manager Suite are inserted as configuration items in the CMDB. Thereafter, matching records should be updated, and only new items inserted. We therefore run the first pass with the default settings, and then modify the settings to schedule the regular data exchange.

1. In **BMC Atrium Integrator** console, from the list of **Jobs** on the left, choose the `BMCAtriumAdapter` job.
2. In the toolbar, click the third icon (with the play button) to run the selected job once.



As there may be a large backlog of data to handle, and Spoon processes a line at a time, the first import can be quite time-consuming. Wait for the first import to complete successfully before continuing with this procedure.

3. In Atrium Integrator, update the way Atrium receives the export of computers as follows:

    a) Open the `ExportComputersTransform` transformation from the repository.

    b) Double-click the `CMDBoutput` step to expose its properties.

    c) Clear (turn off) the **Always Insert CIs** check box.

    d) Select (turn on) the **Only insert new CIs** check box.

    e) Click **OK** to close the `CMDBoutput` step, and click **Save** to store the changes to the transform.

4. Update the processing of the export of products:

    a) Open the `ExportProductsTransform` transformation from the repository.

    b) Double-click the `CMDBoutput` step to expose its properties.

    c) Clear (turn off) the **Always Insert CIs** check box.

    d) Click **OK** to close the `CMDBoutput` step, and click **Save** to store the changes to the transform.

Scheduling the Adapter

The adapter to exchange data between FlexNet Manager Suite and BMC Atrium is driven by a schedule set up in Atrium.

Continue this process in the Atrium Integrator console.

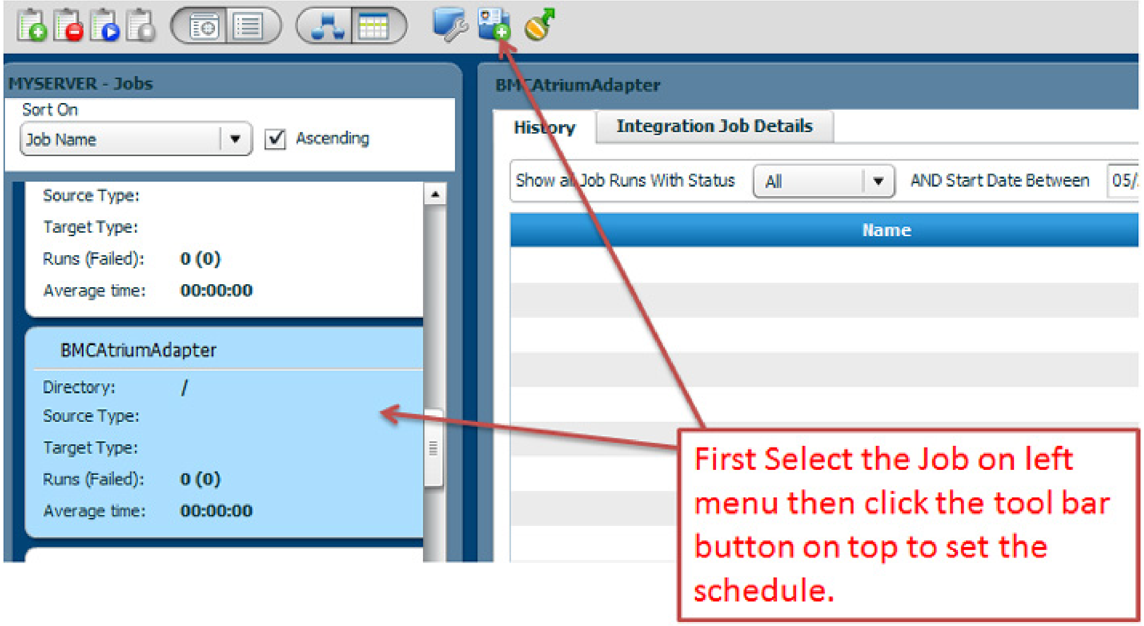
1. If you are not already in the Atrium Integrator console:

    a) Login to BMC Remedy Mid-Tier using user name and password provided by BMC Administrator.

b) Click the **Applications** tab on the left edge.
The **Applications** menu expands.

c) Select **Atrium Integrator**, and then click **Atrium Integrator Console**.

2. In the **Jobs** list on the left, select your `BMCAtriumAdapter` job.

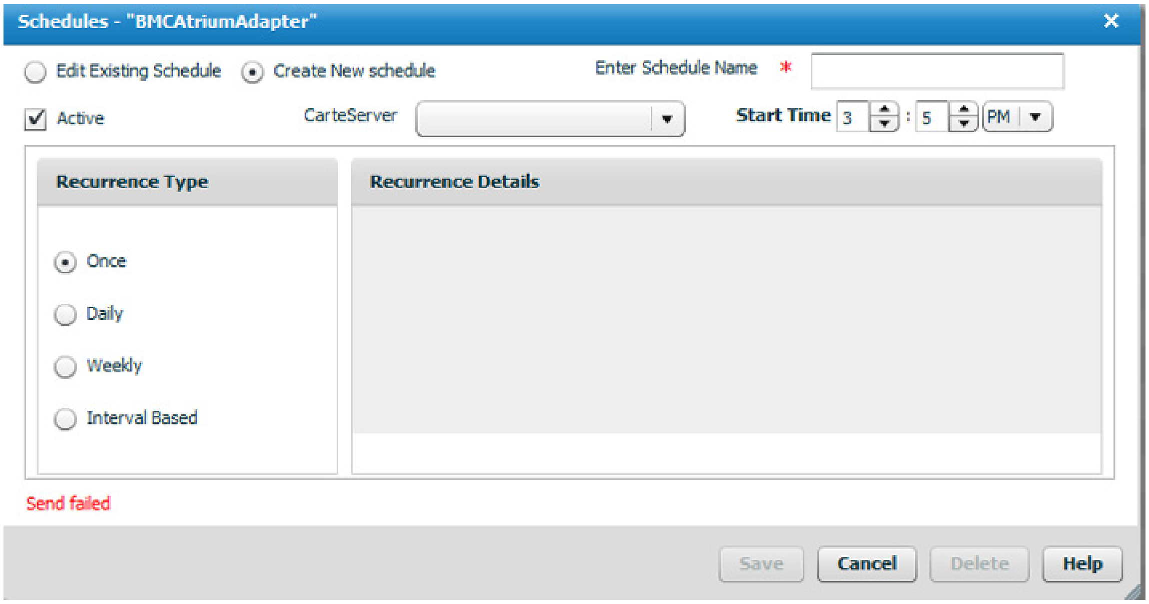


First Select the Job on left menu then click the tool bar button on top to set the schedule.

3. In the tool bar, select the icon to manage the job schedule ( ).

4. Select **Active**.

5. Select the **Create new schedule** option.
The **Schedules - "BMCAtriumAdapter"** page appears.



Schedules - "BMCAtriumAdapter"

○ Edit Existing Schedule ● Create New schedule

Enter Schedule Name *

☑ Active

CarteServer

Start Time 3 : 5 PM

Recurrence Type

● Once

○ Daily

○ Weekly

○ Interval Based

Recurrence Details

Send failed

6. Complete the details for the regular operation of the adapter (this schedule controls data flow both ways, for import and export):

   a) Give the schedule a meaningful name in the top right corner (**Enter Schedule Name**).

   b) Use the **Start Time** spinners to dial up the time of day to start operation of the adapter.

      Tip • *This is local time on the BMC Remedy AR System server.*

   c) In the **Recurrence Type** area, specify how frequently you want the job to run.

      Recommendation: The process is quite time-consuming. In a stable environment, consider a monthly data exchange (**Interval Based**).

   d) Depending on your choice, complete the **Recurrence Details**.

   e) Click **Save**.

The integration between FlexNet Manager Suite and BMC Atrium is now operational.

However, before the next scheduled run, you should consider any further customization that may be necessary, especially in relation to large data sets.

Verifying Data Export to BMC Atrium

This process confirms that data has been exported from FlexNet Manager Suite and successfully imported by Atrium.

This procedure uses the Atrium Core application (not Atrium Integrator).

1. If you are not already in the Atrium Core console:

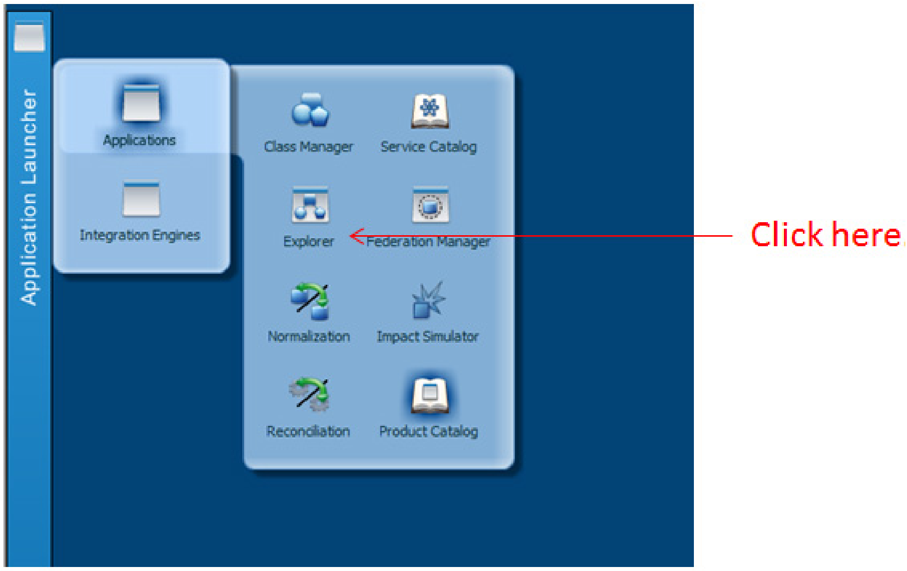
a) Login to BMC Remedy Mid-Tier using user name and password provided by BMC Administrator.

b) Click the **Applications** tab on the left edge.
The **Applications** menu expands.

c) Select **Atrium Core**, and then click **Atrium Core Console**.

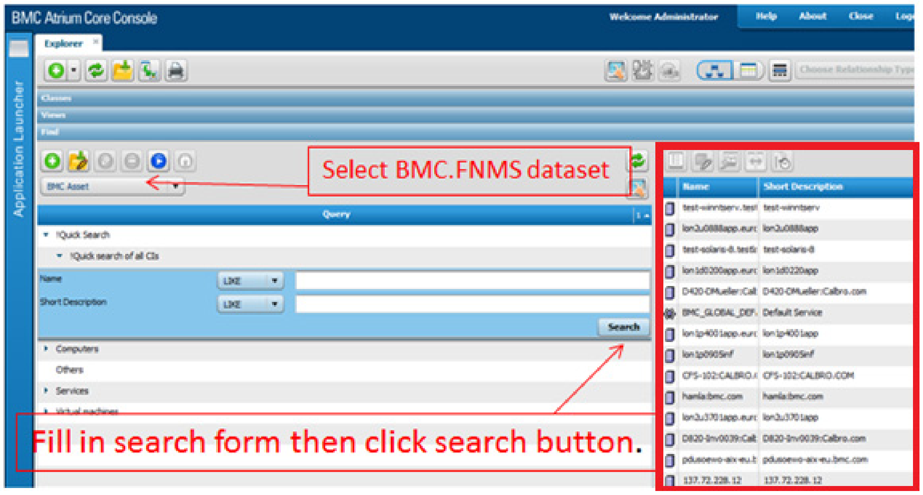
A new window opens with the **Application Launcher** tab on the left edge.

2. Click the **Application Launcher** tab, and in the fly-out menu, click **Applications**.



Click here.

3. In the sub-menu, click **Explorer**.
The **Explorer** tab opens, with multiple accordion folds. Ensure that the **Find** fold is open.



Select BMC.FNMS dataset

Fill in search form then click search button.

Name	Short Description
test-winntserv.test	test-winntserv
lon2u0888app.eurc	lon2u0888app
test-solaris-8.test	test-solaris-8
lon1d0200app.eurc	lon1d0220app
D420-DMueller:Cal	D420-DMueller:Calbro.com
BMC_GLOBAL_DEF.	Default Service
lon1p4001app.eurc	lon1p4001app
lon1p0905nf	lon1p0905nf
CFS-102:CALBRO.C	CFS-102:CALBRO.COM
hamla:bmc.com	hamla:bmc.com
lon2u3701app.eurc	lon2u3701app
D820-Inv0039:Cal	D820-Inv0039:Calbro.com
pdusoewo-aix-eu.b	pdusoewo-aix-eu.bmc.com
137.72.228.12	137.72.228.12

4. In the **Find** area, use the option list on the left below the tool icons to select the `BMC.FNMS` dataset.
5. Optionally choose an object type to search for, or expand the **Quick search of all CIs**.
6. Optionally, add data (or partial values) to the **Name** and **Short Description** fields (and controls for matching) to narrow the search for particular configuration items, or leave the fields blank for a full listing of all results.
7. Click **Search**.
   A list of results (shown inset) is displayed. Validate that the data is as expected.

3

Additional Customization of the Adapter

Topics:

The following notes help with minor changes to configuration, especially for future changes to configuration. The one case that may be critical to your initial implementation is the first topic, covering preparations for large datasets.

Preparing for Large Datasets

Large numbers of data records require a customized environment variable.

If your integration between FlexNet Manager Suite and BMC Atrium will exchange large numbers of records (for example, 100,000 computer records or more), you need to configure an environment variable on the Atrium server.

1. Ensuring that you are logged onto the Atrium server with Administrator privileges, open the Windows start menu, and right-click **Computer**.
2. From the context menu, select **Properties**.
   The Control Panel page of system properties appears.
3. In the navigation bar, select **Advanced system settings**.
   The **System Properties** dialog appears. Confirm that the **Advanced** tab is selected.
4. Click **Environment Variables...**.
   The **Environment Variables** dialog appears.
5. In the **System Variables** (lower) group, click **New...**.
   The **New System Variable** dialog appears.
6. Insert the **Variable name**:

   ```
   _JAVA_OPTIONS
   ```

7. Insert the **Variable value**:

   ```
   -Xms512m -Xmx15360m
   ```

   This configures the Java heap size above 15GB.
8. Click **OK** three times to close the various dialogs, and you can close the Control Panel window.

Updating Connection Details

When system passwords change, you need to update the adapter's configuration file.

Connection details are stored, encrypted, in the adapter's configuration file (by default located at `C:\Program Files\Flexera\BMCAtriumAdapter\ FNMPExports\Utility\ FNMPDataExportUtility.exe.config` on the Atrium server). The encryption process happens the first time the adapter runs after the configuration file is edited. Once the process has run, the connection strings are unreadable and not editable, something similar to the following:

```
<connectionStrings configProtectionProvider="DataProtectionConfigurationProvider">
    <EncryptedData>
      <CipherData>
        <CipherValue>AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAeNhcGMAVK0uVGNTq
Og/WbQQAAAACAAAAAAADZgAAwAAAABAAAACTO9kpn6BptpLvsXExg1UBAAAAAASAAACgA
AAAEAAAAELCyiwz5AXZw9xZXfEPiAAAAgAAPJQYe+G9AfScFMJTYgA0NDbAgZdRA9nB91
DN42A1xjeCskUs9+KNjVU1PSFRV4ujta40evf3IOZy5odyHsIrJRCKOOGdhDb1wh4ISEk
pJk/QDna6LeCbbtXXsQK2LoAHQc/plz77UkQZxnxkL5ElPIGHl6AojEXT2F5NGjElJX6G
XbUXQDkNnDfi2o6XI/CDbX8gCuMonY1cTLYGe6+AQPpDgcY3rA02ZFOs7/Zb0cKOw7IoZ
dB6H8OIvrClSzqNkVBd3YfLhP/KOrkQFP8orqj54BJW74E1v3VUZnte1ESgLA5MYb/F9A
h3M5xi2Q6ITXOQmVRGESrivsdqr6nyGz5APx2yVBuEcoVhpOMYURbEbBSW+6/aydg8nY1
DrcMzkPlXiZ0CQs4yZYHSWt3+bFEN30xh6XKFH7Tpc8e5y9Tq1Bhwk+Kw6AFfZhcdewAp
J4ZkGAr4ixE6gNqWXnomk2puKNFImhJfqLLIuQx9T00Uu2bjJ9Y+dUlrcuov12hQXOCh9
```

```
nqOdmeIQHPXgw8//eHYOzwy2TgMcq2M2LxXRbQqCTeWtlP1ucJVyct9gnJlk2L3FSBNgM
u1C9mncR7MsGDBWoQMXnwC5+Y6P1GT45sPOedBQvIQITj7MCuzfgDD1R9c7w1gejTGmrl
3Kmf33tGPMRYPV7CPNET3DUV9AGQUAAAAaIEkjyqfExrbeiYJvG2usqYO3Nk=</CipherValue>
      </CipherData>
    </EncryptedData>
  </connectionStrings>
```

Follow this procedure to replace the connection strings and/or passwords.

1. Open the adapter's configuration file in a flat text (or XML) editor.

   Default location: `C:\Program Files\Flexera\BMCAtriumAdapter\ FNMPExports\Utility\ FNMPDataExportUtility.exe.config`

2. Locate the `connectionStrings` element (similar to that shown above), select it all, and delete it.

3. Copy the following example, paste into the same place in the configuration file, and replace the placeholder values (including their containing brackets) as described below (you may also remove the white space from the connection strings):

   ```
   <connectionStrings>
       <clear />
      <add name="FNMPDBConnection"
                 connectionString="Data Source=[FNMPDBServerName];
                                  Initial Catalog=FNMSCompliance;
                                  User ID=[accountName1];Password=[somePassword];
                                  Integrated Security=false;
                                  Connection Timeout=60;"/>
      <add name="IntermediateDBConn"
                 connectionString="Data Source=[intermediateDBServerName];
                                  Initial Catalog=IntermediateDB;
                                  User ID=[accountName2];
                                  Password=[anotherPassword];
                                  Integrated Security=false;
                                  Connection Timeout=60;"/>
     </connectionStrings>
   ```

   where:

Placeholder	Notes
`[FNMPDBServerName]`	The database server (and optionally, the database instance name) containing the FlexNet Manager Suite compliance database. You may use: • A dot (`.`) when the database is installed on the same server where the adapter executable runs • The flat name of the server • The IP address of the server • Where the database instance is not the default instance on that server, any of the above with a slash (`/`) separator, and the database instance name.
`FNMSCompliance`	This is the recommended database name for the compliance database. If you used a different name, substitute the name you used.
`[accountName1]`	The user name that the executable uses to connect to the compliance database.
`[somePassword]`	The (newly current) password for the above account.

Placeholder	Notes
`[intermediate DBServerName]`	The database server (and if necessary, instance) for the intermediate database. The same formats are supported as described above for `[FNMPDBServerName]`.
`[accountName2]`	The user name that the executable uses to connect to the intermediate database.
`[anotherPassword]`	The password for `[accountName2]`.

4. Save the edited configuration file.
   The file is saved in plain text. The next time the adapter runs, this section of the configuration file is automatically encrypted. If you are unhappy about leaving the file unencrypted until then, trigger an additional run of the adapter by adding a one-time schedule (see *Scheduling the Adapter* on page 99).

Updating Email Details

The adapter sends emails as a heads-up when an error occurs.

With data flows in both directions between FlexNet Manager Suite and BMC Atrium, errors may occur in either export or import. The emails for the two directions are defined separately:

- For export from FlexNet Manager Suite, the data export utility generates the emails, with settings in its configuration file. If the details of your email server are changed, you need to update the adapter's configuration file. It also defines who receives the alerts, and the address from which alerts originate. You can modify any of those detail with this procedure.
- Similarly, for imports from BMC, the email is generated by Atrium based on settings in that system. Those settings are also summarized in this procedure, since it is likely that a change in your infrastructure or personnel affects both kinds of email alerts equally.

We start with emails for exports from FlexNet Manager Suite; and then continue with those for imports to the same system (that is, data is exported from BMC Atrium).

1. Open the adapter's configuration file in a flat text (or XML) editor.

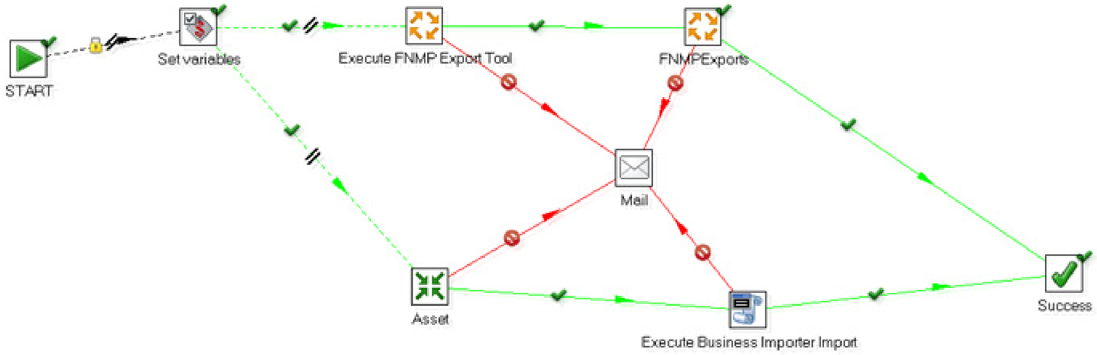
   Default location: `C:\Program Files\Flexera\BMCAtriumAdapter\ FNMPExports\Utility\ FNMPDataExportUtility.exe.config`
2. Locate the `<appender>` element for the `SmtpAppender` node in the file and update the settings marked here with placeholders:

```
<appender name="SmtpAppender" type="log4net.Appender.SmtpAppender">
 <to value="toaddress@somedomain.com"></to>
 <from value="fromaddress@somedomain.com"></from>
 <subject value=" Atrium Integration Adapter Error"></subject>
 <smtpHost value="smtp.somedomain.com"></smtpHost>
```

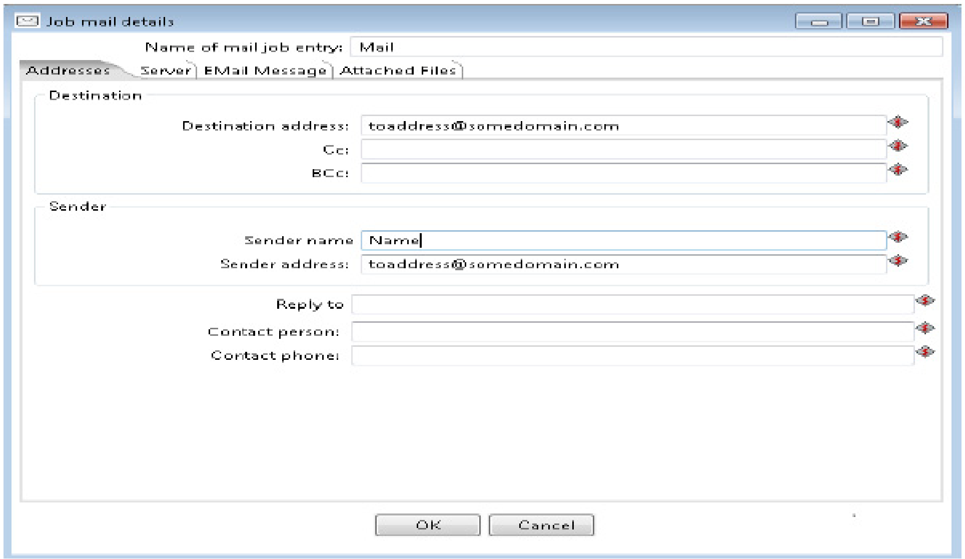
   where:

Value	Notes
toaddress @somedomain.com	Valid email address for the person who is to receive alerts when errors occur during export.
fromaddress @somedomain.com	Valid email address for an account known on the specified SMTP host that is identified as the sender of the email when an error occurs.
smtp.somedomain.com	The fully qualified domain name of the SMTP email server.

3. Save the edited file.
4. Open the Atrium Integrator Spoon tool.
5. Select **File** > **Open...**, and select `BMCAtriumAdapter.kjb` from the repository.
   A map of the job is displayed.



6. To configure email alerts for errors occurring on the Atrium side of the adapter:
   a) In the `BMCAtriumAdapter` job, right-click on the `Mail` step in the center.
   b) From the option menu, select **Edit job entry**.
      The **Job mail details** dialog appears.



Name of mail job entry: Mail

Destination address: toaddress@somedomain.com

Sender name: Name

Sender address: toaddress@somedomain.com

a) In the **Addresses** tab, insert the following:

- **Destination address** - the person who is to receive the email alert for each error.
- **Sender name** - the value to appear on the emails as the sender.
- **Sender address** - The email address from which the email will appear to come. This must be a valid email account known to the mail server (specified next).

b) Switch to the **Server** tab, and complete the **SMTP Server** details.

c) Click **OK**.

The revised email settings for data transfer in both directions take effect for the next run of the adapter.

Turning Off Email Alerts for Export Errors

You can disable email alerts for problems arising during export from FlexNet Manager Suite.

This procedure provides a quick way to temporarily disable emails sent by the data utility during data export from FlexNet Manager Suite for transfer to BMC Atrium. This procedure turns the emails off without removing the settings details.

Tip *• For emails generated by BMC Atrium about data preparation for import to FlexNet Manager Suite, there is no single "off switch". To stop those emails, re-do the settings to remove the email details (as described in Updating Email Details on page 107).*

1. Open the adapter's configuration file in a flat text (or XML) editor.

   Default location: `C:\Program Files\Flexera\BMCAtriumAdapter\ FNMPExports\Utility\ FNMPDataExportUtility.exe.config`

2. Locate the following section (towards the end of the configuration file):

```
<root>
  <level value="ALL" />
  <appender-ref ref="RollingFileAppender" />
  <appender-ref ref="SmtpAppender" />
</root>
```

3. Comment out the `appender-ref` element for the `SmtpAppender`:

   XML comments are surrounded by `<!-- -->`:

```
<root>
  <level value="ALL" />
  <appender-ref ref="RollingFileAppender" />
  <!-- appender-ref ref="SmtpAppender" / -->
</root>
```

4. Save the amended file.

Email alerts for export errors are suspended until you reverse this process, re-editing the file to remove the comment tags and restored the original value.

Changing the Log File

You can rename or relocate the log file generated by the adapter for exports from FlexNet Manager Suite.

By default, five copies of the log file, each capped at 10MB, rotate in the nominated folder (total disk space requirement is therefore about 50MB for logging). You can customize the location and base name of the log file as follows:

1. Open the adapter's configuration file in a flat text (or XML) editor.

   Default location: `C:\Program Files\Flexera\BMCAtriumAdapter\ FNMPExports\Utility\ FNMPDataExportUtility.exe.config`

2. Locate the following section in the file:

```
<appender name="RollingFileAppender" type="log4net.Appender.RollingFileAppender">
  <file value="C:\Program Files\Flexera\BMC Atrium Adapter\Log.txt" />
  ...
</appender>
```

3. Edit the `value` attribute with the new path and file name.

   Tip • *Keep in mind that the file name will be expanded with dating information to differentiate the various logs.*

4. Save the modified file.

4

Appendices

Topics:

The following topics provide data mapping between attributes recorded in the FlexNet Manager Suite database and those recorded in the BMC Atrium CMDB. In addition, some values are generated as part of the export and import processes, and are included in the following listings.

Two separate data objects are exported from FlexNet Manager Suite: `Computers`, and `Products`. A much smaller dataset is imported from Atrium into FlexNet Manager Suite, updating information about assets to show their role and owner.

Appendix 1: Export of Computers

For optimal performance, exports are differential (that is, only data changed since the last export is included). The following table shows:

- Columns from the `ComplianceComputers` table (and related tables) in FlexNet Manager Suite that are exported
- The equivalent column name in the `ComputerSystemChanges` table of the intermediate database (in case you wish to inspect the dataset there)
- The matching column in the `BMC_ComputerSystem` table in the Atrium CMDB.

Note •

1. *Some values are not read from the source database, but inserted as fixed values. These are shown in double quotation marks, and explained in the associated comments.*
2. *Some values exported from FlexNet Manager Suite to the intermediate database are not transferred to BMC Atrium. Instead, they are used to manage the exported data, removing records not relevant to the CMDB and so on.*

`Compliance Computer`	**`ComputerSystem Changes`**	**`BMC_ ComputerSystem`**	**Comments**
Not applicable.	*GeneratedTokenID*	`TokenID`	`TokenID` takes different formats for different types of computers. In the following examples, the placeholder *X* represents any hexadecimal digit (0-9, A-F). • For physical hosts and computers, the `TokenID` is *`hostname`*`:`*`DNSdomainName`*. • For some virtual machines, `TokenID` takes the form of a prefix, a colon, and a UUID: • For VMware, `VI-UUID:`*`XXXX-XX-XX-XX-XXXXXX`* • For Hyper-V, `HYPERV-ID:`*`XXXX-XX-XX-XX-XXXXXX`* **Note •** *With Hyper-V, the UUID is only available on the physical machine, so `TokenId`*

Compliance Computer	ComputerSystem Changes	BMC_ComputerSystem	Comments
			is only set for virtual machines that have been successfully linked to their hosting physical machines. • For Xen (including Oracle VM), XEN-ID:*XXXX-XX-XX-XX-XXXXXX* • For KVM (including RedHat Enterprise Virtualization), KVM-ID:*XXXX-XX-XX-XX-XXXXXX*. TokenID is generated dynamically, checking the value in ComputerType. For virtual machines, the value in ComputerToVirtualMachine_VMTypeID is used to determine the prefix, and the UUID is attached from ComputerToVirtualMachine_UUID.
Compliance ComputerID	ComputerID	CITag	The identifier for the computer in the FlexNet Manager Suite database, passed to BMC as the configuration item tag. (This is passed back again if Asset records are imported by FlexNet Manager Suite.)
ComputerName	ComputerName	HostName	The name of the computer, limited to 256 characters.
	“DNS”	Name Format	A fixed string value determining the data type for Name.
[ComplianceDomain] QualifiedName	Domain	Domain	The fully qualified domain name for the domain where this computer exists. Extracted from ComplianceDomain.QualifiedName using ComplianceComputers.ComplianceDomainID as a foreign key. Used when there is no value for FlatDomain received from FlexNet Manager Suite.
[ComplianceDomain] FlatName	FlatDomain	Domain	The unqualified name of the domain containing the computer. Extracted from ComplianceDomain.FlatName using ComplianceComputers.ComplianceDomainID as a foreign key. Where this value is present, it is stored in BMC_ComputerSystem.Domain; if

Compliance Computer	**ComputerSystem Changes**	**BMC_ ComputerSystem**	**Comments**
			not, the qualified domain name is used instead.
[Compliance ComputerStatus] DefaultValue	ComputerStatus	(Not stored.)	The current status of this computer within the inventory records. Values include: • `New` (this is the first appearance of this computer in inventory) • `Ignored` (an operator has marked this computer to be ignored) • `Registered` (this inventory record is linked to an asset record) • `Awaiting Inventory` (a dummy computer record, allowing its associated asset record to exist, which will be removed when an operator links the asset record to a real inventory item) • `Discarded`. These values are extracted from `ComplianceComputerStatus.DefaultValue` based on the foreign key `ComplianceComputers.ComplianceComputerStatusID`. ***Tip •*** *The `DefaultValue` is the US English representation of the values, not localized.* The status is used to make sure that computers that are `Awaiting Inventory`, `Discarded`, or `Ignored` are not transferred to BMC Atrium.
Not applicable.	"Workstation"	CapabilityList Primary Capability	Fixed literal string inserted in the BMC database.
[Compliance ComputerType] DefaultValue	ComputerType	isVirtual	The boolean `isVirtual` is set true whenever the `ComputerType` in the intermediate database is either `VMHost` or `Virtual Machine`. Possible values are extracted from

Compliance Computer	ComputerSystem Changes	BMC_ ComputerSystem	Comments
			ComplianceComputerType. DefaultValue based on the foreign key ComplianceComputers. ComplianceComputerTypeID .
InventoryDate	InventoryDate	(Not stored.)	The date inventory was last collected from this computer.
Manufacturer	Manufacturer	ManufacturerName	Values from FlexNet Manager Suite are provided to avoid data normalization by the CMDB that may modify names.
SerialNo	SerialNo	SerialNumber	The serial number of the computer.
[VirtualMachine] Compliance ComputerID	ComputerTo VirtualMachine_ Compliance ComputerID	(Not stored.)	Populated only for virtual machines. A foreign key that links records in the VirtualMachine table to the matching records in the ComputerCompliance table. Used here in the calculation of TokenID.
[VirtualMachine] UUID	ComputerTo VirtualMachine_ UUID	(Not stored.)	Populated only for virtual machines. The UUID (Universally Unique Identifier) of the virtual machine. Used to match virtual machine properties to their associated ComplianceComputer. Used here in the calculation of TokenID for virtual machines.
[VirtualMachine] VMTypeID	ComputerToVirtual Machine_ VMTypeID	(Not stored.)	Populated only for virtual machines. A foreign key to the VMType table, which identifies the kind of virtual machine, including: • VMware • Hyper-V • LPAR • WPAR • nPar • vPar • SRP • Zone • Unknown.

Compliance Computer	ComputerSystem Changes	BMC_ ComputerSystem	Comments
			Used here in the calculation of TokenID.
Not applicable.	UpdatedDate	(Not stored.)	Used when identifying changed data for differential exports to BMC Atrium.
Compliance Computer TypeID	Compliance Computer TypeID	(Not stored.)	This value is used to filter out remote devices (those from which inventory cannot be collected) and VDI templates, which are not relevant to the CMDB.
Not applicable.	ViewID	(Not stored.)	Deprecated and not in use.
Not applicable.	TenantID	(Ignored.)	Always contains a zero value (a string of 16 zeroes) for an on-premises implementation.
Not applicable.	“BMC_ COMPUTERSYSTEM”	ClassID	Fixed literal string inserted in these records to identify the data class in the data set.
Not applicable.	"*CompanyName*"	Company	A fixed value that must match the company identifier embedded in the Atrium CMDB.
Not applicable.	"*DataSetID*"	DataSetID	Value attached to the dataset considered. Must match the value embedded in the Atrium CMDB.

Appendix 2: Export of Applications/Products

The term 'product' is used in two slightly different ways in FlexNet Manager Suite and BMC Atrium.

- In FlexNet Manager Suite, the base unit of software is called an "application", which is specific to a particular release (version) and edition. A 'product' is the same software unit, but across multiple versions (and perhaps editions as well). In our terms, then, FlexNet Manager Suite exports application records. To confuse things further, the underlying database table is called SoftwareTitle.
- In Atrium, a "product" is the base unit of software, and related to an individual installation. The CMDB also offers a catalog of the products available for installation. In BMC terminology, Atrium imports product records, from which the Product Catalog may also be updated.

The following table shows:

- Columns from the SoftwareTitle table (and related tables) in FlexNet Manager Suite that are exported
- The equivalent column name in the ProductChanges table of the intermediate database (in case you wish to inspect the dataset there)
- The matching column in the BMC_Product in the Atrium CMDB. This indirectly populates the Product Catalog.

Note •

1. *Some values are not read from the source database, but inserted as fixed values. These are shown in double quotation marks, and explained in the associated comments.*
2. *Some values exported from FlexNet Manager Suite to the intermediate database are not transferred to BMC Atrium. Instead, they are used to manage the exported data, removing records not relevant to the CMDB and so on.*

`SoftwareTitle`	**`ProductChanges`**	**`BMC_Product`**	**Comments**
Not applicable.	*GeneratedTokenID*	`ParentCITag`	The token ID generated for the computer system on which this product is installed is assigned to the `ParentCITag` column in BMC. For details about the generated `TokenID`, see *Appendix 1: Export of Computers* on page 112.
`SoftwareTitleID`	`ApplicationID`	(Not stored.)	Used internally in data management.
`FullName`	`ApplicationName`	`ShortDescription`	By default, the full name of the application is the concatenation of the product, version, and edition fields (for example, “Microsoft Office Professional 2010”). However, an operator may overwrite this with any preferred value.
`[SoftwareTitle Version] VersionName`	`ApplicationVersion`	`MarketVersion`	The marketing version number (such as "2010"). Extracted from `SoftwareTitleVersion. VersionName` using `SoftwareTitle. SoftwareTitleVersionID` as a foreign key.
(Generated ID)	`FlexeraID`	`TokenID`	`FlexeraID` is a unique identifier for application records used for unambiguous identification across various Flexera Software solutions. The format is either: • For applications published in the FlexNet ARL, `arl://MGS-APP-00000000000` (where the 11-digit number varies by application) • For applications defined locally, `app://`*`serverGUID`*`/LOCAL-`

SoftwareTitle	**ProductChanges**	**BMC_Product**	**Comments**
			APP-*SoftwareTitleID* (where *serverGUID* depends on the identity of the central application server, and *SoftwareTitleID* is the value of SoftwareTitleID assigned to this application). The result is presented as the TokenID in the BMC_Product table, and as SignatureID in the Product Catalog.
[SoftwareTitle Publisher] PublisherName	Publisher	ManufacturerName	Identifies the company that produces the application/ product (for example, "Microsoft"). Extracted from SoftwareTitlePublisher. PublisherName via SoftwareTitleProduct. SoftwareTitlePublisherID, using SoftwareTitle. SoftwareTitleProductID as a foreign key.
[Compliance Computer] ComputerName	ApplicationTo Installation_ ComputerName	SystemName	Name of the computing device reported in inventory (such as "vincent-ltp"). Extracted from ComplianceComputer. ComputerName using the foreign key InstalledSoftware. ComplianceComputerID. The view InstalledSoftware is itself linked to the application by the foreign key SoftwareTitleID.
ComputerID	InstallationTo Computer_ ComputerID	(Not stored.)	The computer identifier, extracted in much the same way as described for ComputerName. Used internally for data management.
[Compliance Connection] LastImportDate	InstallationTo Computer_ InventoryDate	(Not stored.)	The date of the most recent inventory collection that recorded the installation of this application on the current computer. When no other value is available, this defaults to the date of data export from FlexNet Manager Suite. Used internally for data management.
CategoryID	InstallationTo Application_ Category	Category	First level of application categorization.

`SoftwareTitle`	**`ProductChanges`**	**`BMC_Product`**	**Comments**
			The full path of categorization is extracted from the `GroupEx` table using `SoftwareTitle.CategoryID` as the foreign key. The full path is then broken up on its `/` separators, and the first level is passed to `Category` in the `BMC_Product` table.
`CategoryID`	`InstallationToApplication_Category`	`Type`	Second level of application categorization. (See `Category` for details.)
`CategoryID`	`InstallationToApplication_Category`	`Item`	Third level of application categorization. (See `Category` for details.)
Not applicable.	`ViewID`	(Not stored.)	Deprecated and not in use.
	`TenantID`	(Ignored.)	Always contains a zero value (a string of 16 zeroes) for an on-premises implementation.
Not applicable.	`EvidenceVersion`		Deprecated and not in use.
Not applicable.	“`BMC_PRODUCT`”	`ClassID`	A string literal that identifies the class in the BMC dataset.
Not applicable.	"*`CompanyName`*"	`Company`	A fixed value that must match the company identifier embedded in the Atrium CMDB.
Not applicable.	"*`DataSetID`*"	`DataSetID`	Value attached to the dataset considered. Must match the value embedded in the Atrium CMDB.
Not applicable.	*FullName*:*VersionName*	`Name`	Full name separated by a colon from the market version of the software product (for example, “Microsoft Office Professional:2010”).
Not applicable.	“`ProductName:Version`“	`NameFormat`	A string literal describing the format of the `Name` attribute.
Not applicable.	*FullName* on *ComputerName*	`Description`	Composite of the software name and device name (such as “Microsoft Office Professional 2010 on vincent-ltp”).
Not applicable.	“`Deployed`”	`Status`	A fixed value, since the records are extracted from the installed software listings.
Not applicable.	*Calculated value*	`MarkAsDeleted`	Boolean (0 = no, 1 = yes). Whether or not this product has been deleted from the computer system under consideration. Note that

SoftwareTitle	ProductChanges	BMC_Product	Comments
			this flag is raised only for the first import after the product is deleted; thereafter, further imports do not contain this row.

Appendix 3: Import of Assets

The following table shows:

- Columns from the BMC_ComputerSystem view (and AST:ComputerSystem form) in Atrium CMDB that are exported
- The equivalent column name in the Assets table of the intermediate database (in case you wish to inspect the dataset there)
- The matching columns in the FlexNet Manager Suite database where this information is imported.

BMC_ComputerSystem	Assets	ComplianceComputer	Comments
Reconciliation Identity	Reconciliation Identity	Not stored.	Used internally to link the BMC_ComputerSystem view and AST:ComputerSystem form.
Domain	DOMAIN	[ComplianceDomain] QualifiedName and FlatName	The domain extracted from BMC Atrium is processed to provide a dot-separated qualified name in the form flexerasoftware.com. The same value is processed again to provide a flat domain name (such as flexerasoftware). Where these values match existing entries in the ComplianceDomain table, no further action is taken with them (other than to return the ComplianceDomainID key); but where they are not already known, they are added to this table for use in FlexNet Manager Suite. As well, the ComplianceDomainID value returned form this update is then written into the ComplianceComputer record, which may update the domain for the current computer.
HostName	HostName	ComputerName	The name of this computer. Based on a matching ComplianceComputerID, this value may be over-written with the value imported from Atrium.

BMC_ComputerSystem	**Assets**	**ComplianceComputer**	**Comments**
CITag	CITag	Compliance ComputerID	The identity for the computer record. Where this matches an existing record, its other columns may be updated. Otherwise, a new record is created for ComplianceComputer in FlexNet Manager Suite.
	InstanceId	Not stored.	Deprecated and not in use.
ImportLog table	LastImportDate	Not stored.	Calculated from data in the ImportLog table.
[AST: ComputerSystem] System Role	System Role	Custom property Role in the **Details** tab (**User** group) of the inventoried computer properties.	Both systems record roles for servers, but the FlexNet Manager Suite role has various licensing implications, and so is not updated by imports from Atrium. Instead, a read-only custom field is added to FlexNet Manager Suite to accept the asset role for each BMC_ComputerSystem. The role is displayed in the **Details** tab of the computer properties. Values include: - Lab - Development - SIT - UAT - Training - Performance - Contingency - Production.
[AST: ComputerSystem] Reconciliation Identity	Reconciliation Identity	Not stored.	Used internally to link the BMC_ComputerSystem view and AST:ComputerSystem form.
[AST: ComputerSystem] Assigned To	Assigned To	AssignedUserID	The ID associated with ComplianceUser.UserName, written here to link the ComplianceComputer to the user record.
	Modified Date	Not stored.	Deprecated and not in use.
[AST: ComputerSystem] Owner_name	Owner_name	[ComplianceUser] UserName	The full name of the user assigned to this computer. FlexNet Manager Suite defaults to a concatenation of

`BMC_ComputerSystem`	`Assets`	`ComplianceComputer`	Comments
			title, first name, middle name, last name and suffix, so any of these parts may be present, based on the record in BMC Atrium. Where this matches an existing user record, no further action is taken with it; otherwise a new user record is created. In either case, the ID for this user is returned, and written into the `AssignedUserID` column. ***Tip* •** *Beware of small changes in record data from Atrium causing near-duplicate records to represent the same person. Also note that when new records are created, you may wish to update additional details about this person in FlexNet Manager Suite.*
`[AST: ComputerSystem] AssetLifecycle Status`	`AssetLifecycle Status`	`Compliance Computer StatusID`	A translation of the BMC Atrium status values into the nearest equivalent status value supported in FlexNet Manager Suite. This is then mapped by the Business Importer into the correct ID for the status value before being written into the `ComplianceComputer` record.
`[AST: ComputerSystem] DNS Host Name`	`DNS Host Name`	Not stored.	Host name from the `BMC_ComputerSystem` is used in preference to this one.
`[AST: ComputerSystem] Domain`	`Domain_1`	Not stored.	Domain from the `BMC_ComputerSystem` is used in preference to this one.
`[AST: ComputerSystem] Instance Id`	`Instance Id`	Not stored.	Used internally.
`[AST: ComputerSystem] Data Set Id`	`Data Set Id`	Not stored.	Used internally.

Part IV

Using the HP Universal Discovery Adapter

Topics:

You can use the HP Universal Discovery (HPUD) adapter tool to collect and import inventory data from HP Universal Discovery System to FlexNet Manager Suite. The HPUD adapter fetches all hardware, software, and virtualization information from the HPUD system and stores it in the compliance database maintained by FlexNet Manager Suite. The HPUD adapter support is available only with FlexNet Manager Suite version 2015 or later.

Note • *The HPUD adaptor tool works only with version 10.10 and 10.11 of the HP Universal Discovery System. These versions do not support Solaris 11 zones, for which reason the HPUD adapter cannot import these zones. Since host serial number and zone name are used for rationalizing duplicate inventory records across different inventory sources, this may mean that a device imported through the HPUD adapter cannot be merged with another record of the same device imported through another inventory source that supports zones information.*

1

Selecting a Configuration

Topics:

- *Architecture and Working of the HP Universal Discovery Adapter*
- *The Adapter Executable*

This chapter gives an overview of the architecture, working, and configuration of the HPUD adapter. Choose the appropriate configuration for your enterprise before you implement the adapter.

Architecture and Working of the HP Universal Discovery Adapter

The HPUD adapter architecture has the following main components:

- **HPUD Server:** This server has the HPUD uCMDB database and HP discovery tools installed on it. The HPUD uCMDB database stores the HPUD inventory details.
- **Staging Server:** This server contains the Flexera HPuCMDBStage.exe tool and a staging database. The HPuCMDBStage.exe tool retrieves topology maps by executing TQL queries on HPUD uCMDB. The retrieved topology maps are stored in a staging database (uCMDB_Staging) present on the staging server. The query execution is performed using the web service interface of HPUD uCMDB.
- **FlexNet Manager Suite:** The Compliance Reader component residing on this server fetches data from the staging database and loads it into the FlexNet Manager Suite database when you run an inventory import.

The following diagram shows the architecture and working of the HPUD adapter:

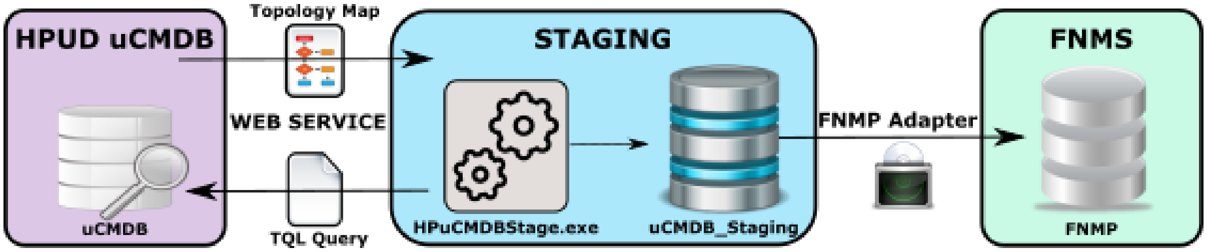


Figure 5: Architecture and working of the HPUD adapter

1. The HPuCMDBStage.exe tool extracts the inventory information through the web API of HPUD uCMDB. This tool executes Topology Query Language (TQL) queries on the HPUD uCMDB database and retrieves responses in the form of topology maps. With the default configuration, this tool stores the retrieved topology maps directly into the staging database without storing them in XML files. However, you can configure this tool to write the retrieved topology maps to XML files and then write data from those XML files to the staging database.

   Note • *One topology map is generated for each TQL query executed on HPUD uCMDB.*

2. The Compliance Reader component of FlexNet Manager Suite collects data from the staging database. When you run an inventory import on the FlexNet Manager Suite server, the data extracted from the staging database is written to the compliance database on the application server

The Adapter Executable

HP Universal Discovery (HPUD) adapter uses the ucmdb-api to obtain the version of the uCMDB instance. It uses the UcmdbService web service API to retrieve the topology maps by executing TQL queries on the uCMDB instance. You can retrieve the WSDL definition of the `UcmdbService` web service API from `http://<uCMDB Server>/axis2/services/UcmdbService?wsdl`. The configuration of the staging tool includes the TQL queries that will be executed on the `HPUD uCMDB` database. You don't have to import the TQL queries into the `HPUD uCMDB` database and there is no need to deploy and maintain the TQL queries on every HPUD instance of your enterprise.

Note • *The URL to retrieve the WSDL definition of the `UcmdbService` web service API may use 'https' instead of 'http'. It may also include a port number. Please check with your HPUD system administrator for more details on the web service URL.*

The tool to query the web API has the following two parts:

- `HPuCMDBStage.exe` — A console program capable of querying the`UcmdbService` API of HPUD and writing the results into an SQL Server database, and optionally to XML files on the local file system. This program supports command line arguments, available using `HPuCMDBStage.exe -h`. Here is the list of available command line options.

```
-h                              This help
-x <settings file>            Settings file
-s <address or URL>           uCMDB server address or URL
-u <user name>                uCMDB export user name
-p <password>                 uCMDB export user password
-c <connection string>        SQL Server staging database connection string
-m <staging method>           Set staging method (stage/staged/prestaged/stream)
-f <staging file path>        Set the staging file path
```

- `FNMPuCMDBSettings.xml` — The self-documenting configuration file for`HPuCMDBStage.exe` which contains TQL queries executed against HPUD, and can include connection settings for HPUD and SQL Server.

In operation, the executable, `HPuCMDBStage.exe`, extracts the inventory data from HPUD and saves it for further processing. The value of the `method` parameter determines how the data is saved. The `method` parameter can have the following values:

- `Stage` — This option enables you to save inventory data from HUPD to a series of XML files on the staging server **(HPUD to XML)**. The XML files are not imported into the staging database, but you can review the extracted inventory data. This option also enables you to collect inventory data from the HPUD servers that are not directly connected to the staging server. You can manually copy and upload the inventory XML files to the staging server. Also see the `Prestaged` method below.

- `Staged` — This options enables you to write the data extracted from the HPUD servers to XML files and then copy the information to the staging database **(HPUD to XML/SQL)** where it can be imported into FlexNet Manager Suite for use in compliance calculations.

- `Prestaged` — This option takes information stored in XML files (from the Stage and Staged method) and loads it into a staging SQL database **(XML to SQL)**. Inventory is not gathered from HPUD in this mode.

- `Stream` — This option enables you to extract the inventory information from HPUD and load it into the staging database directly, without storing it in XML files on the staging server **(HPUD to SQL)**. You can import these files to into FlexNet Manager Suite for use in compliance calculations.

You can set the default values for the `method` parameter and all other parameters in the `FNMPuCMDBSettings.xml` file. The adapter tool uses the default parameter values when you use it without other command-line options. The settings file is self-documented and the matching command-line options are available using `HPuCMDBStage.exe -h`.

Files Created by the Staging Tool

The following files are created when the staging tool performs a read/write operation on the local disk of the staging server. You can view the contents of the following files to review HPUD configuration item details that have been extracted.

Filename	Content
`Computer.xml`	Details of computers and their properties.
`Virtualization.xml`	Virtualization and partitioning details.
`InstalledSoftware.xml`	Installed software evidence linked to each computer.

2

Installation and Configuration

Topics:

For **on premises** implementations, the adapter XML files are included as a part of the FlexNet Manager Suite installation. You need to download the staging tool, the configuration file, and the staging schema for this tool from *https://flexerasoftware.flexnetoperations.com*. You also require this download for **cloud** implementations of FlexNet Manager Suite. You do not need to make any changes to your central application server, but additional components are required in the download.

You need credentials supplied by Flexera Software to access this download. Details of the download are included in *Creating the Staging Database Tables* on page 129.

- The HPUD adapter suits FlexNet Manager Suite releases 2015 and later for on premises delivery.
- The build number for this adapter is 10.3.0.12006 (or higher). You can identify this number by right-clicking the `HPuCMDBStage.exe` file in Windows Explorer, selecting **Properties**, and looking at the **Details** tab.

Save the zipped archive to a suitable temporary location, and unzip it.

Full details of setting up the HPUD adapter are included in this chapter. The following chapter (see *HPUD Operation* on page 134) covers testing the completed installation and validating the results.

Download Tier 1 Adapter Archive

The Tier 1 adapter archive includes content for many adapters, and is updated on the Flexera Software website from time to time.

Start this procedure using a web browser on a computer that has good network accessibility from all the machines needing installations for your adapter.

1. Using the credentials supplied by Flexera Software with your order confirmation (or as renewed since), log into the Product and License Center at *`https://flexerasoftware.flexnetoperations.com`*.
2. On the first page, select `FlexNet Manager Platform`, and on the resulting second page, select the product again.
3. In the list of versions, click the product name for the version you are using (typically the most recent version).
4. In the list of components to download, select the `Adapter Tools.zip` archive, and save this to a convenient location (such as `C:\Temp` on a central, accessible server).
5. Right-click the zip archive, and choose **Extract All...**.

The folders are now available for the range of adapters in the Tier 1 archive.

Selecting a Staging Server

Multiple configuration options are there to configure the staging server. You can install the staging server on any of the following types of computers:

- A dedicated stand-alone server or a virtual machine.
- Any other suitable machine in your enterprise, such as a print server.
- An inventory beacon.
- The central application server where FlexNet Manager Suite is installed (for on-premises installations).

The following are the requirements for a staging server:

- Microsoft Windows 2008 or later.
- Access to an instance of Microsoft SQL Server 2008 or later to host the staging database. You can implement the staging database on the staging server or on a separate database server.
- Microsoft .NET 4.0 run time environment installed.
- Efficient network access to each HPUD server in your enterprise, using the HTTP or HTTPS protocols.

Creating the Staging Database Tables

Once your selected staging server is able to access the Microsoft SQL Server instance, you can use the provided script to create the staging database and set up the appropriate database tables. You can achieve this through SQL Server Management Studio or from the command line as described in the following procedure:

1. Download and install the Tier 1 adapter archive. For more information, see *Download Tier 1 Adapter Archive* on page 129.
2. Navigate through the unzipped archive to `Tier 1 Adapter Tools` > `HP Universal Discovery Tools` > `SQL`.
3. If necessary, copy the script `uCMDB_staging.sql` from the `SQL\` folder of your unzipped adapter archive to a temporary folder on your staging server.
4. Open a command prompt on the staging server.
5. In the command prompt window, execute the following command, as amended:

    ```
    sqlcmd -S ServerName\InstanceName -i TemporaryPath\uCMDB_staging.sql
    ```

    where:

    - The database `uCMDB_staging` is created with all necessary tables, indices, and so on.
    - *ServerName* is the name of the server hosting the SQL Server 2008 or later. You can install the database on the staging server or on any remote server. You can use the name of the staging server or its IP address or a "`.`" (dot) if you are running the staging script on the same server as the database instance.
    - *InstanceName* is the name of the database instance to use for the database staging tables. You can omit this parameter if the default instance name is being used.
    - *TemporaryPath* is the location where you saved the SQL procedure.

    Example:

    ```
    sqlcmd -S 192.100.0.20\Development -i C:\temp\uCMDB_staging.sql
    ```

The staging database is now ready for operation.

Configuring HP Universal Discovery System

You must configure HPUD adapter settings to include a user which can perform the Run Legacy API and the Run Query by Definition actions on the HPUD system. Following are the steps to create and configure a user in HPUD:

1. In the HPUD interface, click **Security**, **Roles Manager**.
2. Create a new role for FlexNet Manager import.
3. Select the role that you just created and click **General Actions** and assign the Run Query By Definition and the Run Legacy API actions to this role.
4. In the HPUD interface, click **Security**, **Users and Groups**.
5. Create a new user for FlexNet Manager import.
6. Click the **Roles** tab and assign the role that you created in Step 2 to this user.
7. Click the **Permissions Overview** tab and review the assigned permissions for this role.
8. Click **Data Flow Management**, **Universal Discovery**, **Zone-Based Discovery** and expand the **Mapping Options** section in **Preferences**.
9. Check the **Raw OS Installed Software** and set the **Include** option to `name=.*`. This enables HPUD adapter to capture all raw installation evidence.

You have created and configured an HPUD user for use in the HPUD staging tool. You must add this user to the `FNMPuCMDBSettings.xml` file.

Installing and Configuring the Staging Tool

This procedure includes many separate sub-processes to complete the setup of the HPUD adapter. Following are the steps to install and configure the HPUD adapter:

1. Ensure that the account under which the adapter executable will run has read/write/execute permissions on this `uCMDB_staging` database. Authentication may be through Windows authentication or SQL Server authentication. Using Windows authentication, the default account is the username running the `HPuCMDBStage.exe` adapter. The username and password for SQL Server authentication are specified in the database connection string, which you may supply in `FNMPuCMDBSettings.xml`, or override with the `-c` option on the command line.
2. To extract information from the HPUD system, the `HPuCMDBStage.exe` tool must have the required permissions to query the HPUD system. You must create a user in the HPUD system and assign it with permission to perform the Run Legacy API and the Run Query by Definition actions. You must specify this user in the `FNMPuCMDBSettings.xml` file. For more information on configuring the HPUD system, see *Configuring HP Universal Discovery System* on page 130.
3. Navigate to the fixed location on the inventory beacon where the inventory reader must find configuration files to control its uploads. Verify the existence of the following path: `C:\ProgramData\Flexera Software\Compliance\ImportProcedures\Inventory\Reader\HP Universal Discovery`.
4. Create a folder to contain the adapter executable and its configuration file. Location is not critical; a suggested path is under `C:\Program Files\Flexera Software` (for version 2015 and later, or for a stand-alone server that is not an inventory beacon or application server). In your chosen location, create a folder such as `HPUDAdapter`.
5. From the `HPuCMDBStage\` folder within your unzipped archive, copy both `HPuCMDBStage.exe` and `FNMPuCMDBSettings.xml` to your newly created folder (such as `C:\Program Files\Flexera Software\HPUDAdapter`).
6. Open your copy of `FNMPuCMDBSettings.xml` in a text editor of choice, and review the self-documenting comments within that file. Modify the following values as required. Ensure that you specify the IP address of your HPUD server. You can use the command line options to configure `HPuCMDBStage.exe`. Use the '-h' option to get the list of available options.

    - Update values in the first element describing the downstream connection to the HPUD server, including the IP address, port, the account name and password for access. Keep a record of the account name and password for registering with HPUD. The default values are:

      ```
      <server protocol="http" address="10.200.20.138"
      port="8080" username="exportuser" password="Pa$$w0rd" timeout="3600"/>
      ```

      Tip • *If you do not wish to record the password in the plain text configuration file, you can use a script to retrieve the password from an encrypted store, and supply it as a command-line option when starting the `HPuCMDBStage.exe` tool. Here is an example:*

```
HPuCMDBStage.exe -p <password>
```

- Update the second element for the connection to the staging database. The default values are:

```
<database connection-string="Server=.;Database=uCMDB_Staging;Trusted_Connection=yes;"/>
```

- Update the third element to configure whether, and where, the executable should save XML files of the inventory collected from HPUD. The default value stores any XML files below the location of the executable: `<staging path="StagingData" method="stream"/>` You may wish to redirect the path setting for easier access for human inspection.
- Make sure that the user details in the `<Server>` section match with the user configured in the HPUD system.
- Save the settings file.

7. Validate the HPUD adapter operation. For more information about validating the adapter, see *Validating HPUD Adapter* on page 134.
8. Assuming that you do not wish to trigger the adapter manually every time it needs to run, start Windows Task Scheduler, and create a basic task to run the adapter.

***Important** • It is suggested that you schedule the adapter to run at times which cannot overlap with the inventory reader uploading the results to the application server. Check the schedule for the compliance reader on the central application server, and avoid this time slot.*

By default, the inventory import (starting with the reader) is triggered around 2 am. Therefore you might consider scheduling this task for some time such as 10 pm daily. The command line for the scheduled task (assuming that you have saved your preferred settings) is simply to invoke the executable. Any parameters not specified on the command line are taken from the settings file in the same folder as the executable. Here is an example:

```
HPuCMDBStage.exe -x <settings file>
```

This completes the configuration of the adapter executable.

3

Operation and Validation

Topics:

This section describes the normal operation of HPUD adapter and also lists the steps to validate its operation. This section has the following tasks:

- HPUD Operation: See *HPUD Operation* on page 134
- Validating HPUD Adapter: See *Validating HPUD Adapter* on page 134

HPUD Operation

Normal operation of the HPUD adapter relies on the following sequence of events:

1. According to the scheduled instructions, the adapter reads the current content of the HPUD database and stages the new data in the staging database after removing the previously stored data. A flag stored in the staging database indicates the status of the data extraction. The resulting status is flagged within the database. For more information on installing and configuring the staging tool, see *Installing and Configuring the Staging Tool* on page 131

2. Following the schedule on the central application server, and provided that the staging status is `Success`, the import reader uploads this content to the inventory database.

3. The next compliance import brings the final data set into the compliance database, where it is automatically taken into account for compliance calculations. As always, the inventory records must be recognized by the Application Recognition Library, and you must have the resulting application records linked to the appropriate license, for compliance calculations to proceed.

Validating HPUD Adapter

To validate the adapter operation:

1. Manually trigger the adapter executable.

   You can specify a value for the `method` parameter if you need to override the default settings set in the settings XML file. For example:

   ```
   'C:\Program Files\Flexera Software\HPUD\HPuCMDBStage.exe' -f 'C:\temp' -m staged
   ```

   This will write XML files under your `C:\temp` directory for review. It will also write data into the staging database.
2. Inspect the saved XML files to validate the inventory gathered.
3. Use SQL Server Management Studio to validate that the data is written to the staging database. Also review the `StagingState` property in the `uCMDBStagingDatabaseConfiguration` table in the staging database. Possible values are `Running`, `Failed`, or `Success`. This value must be `Success` before the HPUD data can be uploaded from the staging database to the central inventory database.
4. Wait until the next inventory import and calculations have run. You might have to wait overnight for this.
5. Use FlexNet Manager Suite to validate that new evidence has been recovered. Identify which evidence has been recognized by the ARL and which new rules are required. Link the applications to appropriate licenses.

Part V

Oracle Enterprise Manager Adapter

Topics:

Oracle Enterprise Manager monitors and manages other Oracle software installed on customer sites. Therefore it is a useful aid in gathering inventory from Oracle systems.

The Oracle Enterprise Manager (OEM) adapter, from Flexera Software, connects to Oracle Enterprise Manager, and extracts a file of connection information for the Oracle systems it monitors. This file is in a standard format (`TNSNames.ora`) used by Oracle. (In fact, if you already have files of the same name generated by Oracle, you can simply copy these to the appropriate folder on an inventory beacon. This may be a viable alternative to using this adapter.)

When the file is saved to a special location on an inventory beacon (and the inventory rules for this inventory beacon allow processing of `TNSNames.ora` files), the connection information in it can be used by FlexNet Manager for Oracle, a separately-licensed option for FlexNet Manager Suite, to collect software inventory information. This document covers the set up and configuration to use the adapter as part of an Oracle inventory solution.

Terminology

Throughout, the Flexera Software adapter is referred to as the **OEM adapter**. The database for Oracle Enterprise Manager (to which the OEM adapter connects) is referred to as the **OEM repository**.

1

Understanding the Oracle Enterprise Manager Adapter

Topics:

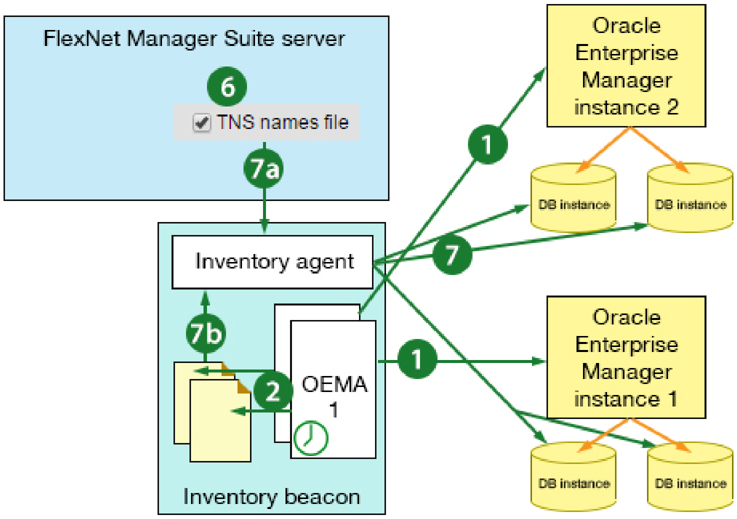
As well as providing a functional overview of the straight-forward OEM adapter itself, this chapter provides additional background about the other aspects of your system that collaborate to provide Oracle introspection and inventory reporting. As well, this chapter covers the prerequisites, content, and delivery of the OEM adapter.

How the Adapter Assists in Inventory Gathering

The OEM adapter provides an alternative or additional method of discovering Oracle database servers in your computing estate. Discovery (by one means or another) is a prerequisite for collecting software inventory from the Oracle servers.

The OEM adapter is installed on a convenient computer that has network access to Oracle Enterprise Manager and its OEM repository. Typically, this computer may be a FlexNet Beacon. Multiple instances of the OEM adapter may be installed (on one or more computers), each of which can connect with one instance of Oracle Enterprise Manager.

Numbers in the diagram (where "OEMA" identifies the OEM adapter) correspond to the process description below:



The process runs as follows:

1. The Windows scheduled task triggers the OEM adapter, which contacts its instance of Oracle Enterprise Manager, and collects the connection data from the OEM repository.
2. The OEM adapter formats the data into a `TNSNames.ora` file, and saves it to a special location on the inventory beacon.

   You may configure the name of the saved file, and, if multiple instances of the OEM adapter save files to the same inventory beacon, you *must* modify the file names so that the files do not overwrite each other.

Note • *This completes the function of the OEM adapter. The remainder of the process is standard operating procedure for FlexNet Manager Suite with the FlexNet Manager for Oracle option installed. If you already have an operational system, several of these steps may already be completed.*

3. You set up a special account with read-only permissions on your Oracle database for all the tables and views needed for Oracle introspection. One helpful practice is to use the same purpose-driven set of credentials on all servers. A utility is available to help created the account(s) required with the correct permissions.
4. The same credentials must be recorded in the Password Store on each of the relevant inventory beacons.
5. In the web interface for FlexNet Manager Suite, you define a subnet (or several subnets) that contain the Oracle database servers of interest; and you assign to these subnet(s) the inventory beacon(s) where the `TNSNames.ora` files are being saved. (Assignments are distributed automatically to inventory beacon(s), along with rules.)
6. Continuing n the web interface, you create a rule with an action including Oracle inventory collection, with the option to use TNS name file selected. The rule also has a target which matches the subnet(s) you are interested in. This rule is automatically distributed to inventory beacon(s).
7. On the inventory beacon(s) of interest:

    a) The rule and assignment are received.

    b) The inventory beacon assesses these, concludes that it is authorized to act, and looks for a any `*.ora` file(s) in the special path on the inventory beacon.

    c) For any Oracle server which is both within the authorized subnet and listed in the `.ora` file, the inventory beacon checks for credentials in its Password Store, and tests them (from the most closely matching to the most general) until either one gets a response (success), or there are no further applicable credentials (failure).

    d) When successfully logged in, the FlexNet inventory agent running on the inventory beacon, uses the credentials to read the data necessary for Oracle introspection. It writes the data as an Oracle inventory file into its staging folder.

    e) Within a minute of completion, the regular upload process starts moving this inventory file to the central application server (or, in a multi-server installation, the inventory server).
8. After the next inventory import and resulting consumption calculations, the Oracle inventory is available in the web interface for FlexNet Manager Suite; and the Oracle servers originally identified in the `TNSNames.ora` file are visible in the **All Discovered Devices** listing, displaying `Yes` in the **Oracle** column.

Prerequisites for the OEM Adapter

The OEM adapter must be installed on a computer with network access to Oracle Enterprise Manager, and requires read access to certain tables and data views there (the required permissions are listed in *Grant Permissions to Account* on page 146).

The OEM adapter requires that, on the computer where it will execute, the Oracle client version 12.1 is installed. For a Windows 64-bit computer, use 64-bit ODAC 12c Release 1 (12.1.0.1.0).

To take advantage of the information gathered by the OEM adapter, there are also the following requirements on the remainder of the system. Once the OEM adapter saves a `TNSNames.ora` file, the subsequent gathering of Oracle inventory requires that:

- The FlexNet inventory agent can access each Oracle system. This can be achieved either by installing the FlexNet inventory agent on the target Oracle server(s), or by remote execution (zero touch inventory gathering).
- You have FlexNet Manager for Oracle, a separately licensed option for FlexNet Manager Suite.

Tip • You can check whether your implementation has this option licensed in the web interface for FlexNet Manager Suite:

1. *Navigate to the system menu (⚙▾ in the top right corner) > **FlexNet Manager Suite License**.*
2. *Check the list of **Subscribed and purchased products** on the right, looking for a card for FlexNet Manager for Oracle. If the card is present, you have this option licensed.*

Components

The OEM adapter is supplied as an installer, called `OEMAdapter.exe`, which installs the following:

- OEM adapter executable and dependencies
- OEM adapter configuration file
- OEM adapter scheduled task, which can be created during installation if desired.

Download Tier 1 Adapter Archive

The Tier 1 adapter archive includes content for many adapters, and is updated on the Flexera Software website from time to time.

Start this procedure using a web browser on a computer that has good network accessibility from all the machines needing installations for your adapter.

1. Using the credentials supplied by Flexera Software with your order confirmation (or as renewed since), log into the Product and License Center at *`https://flexerasoftware.flexnetoperations.com`*.
2. On the first page, select `FlexNet Manager Platform`, and on the resulting second page, select the product again.
3. In the list of versions, click the product name for the version you are using (typically the most recent version).
4. In the list of components to download, select the `Adapter Tools.zip` archive, and save this to a convenient location (such as `C:\Temp` on a central, accessible server).
5. Right-click the zip archive, and choose **Extract All...**.

The folders are now available for the range of adapters in the Tier 1 archive.

2

Installing the Adapter, and More

Topics:

- *Installing the OEM adapter*
- *Other Setup Activities*

This chapter covers the fairly simple process of installing the OEM adapter. However, once installed, the OEM adapter is only a small part of gathering Oracle database inventory. Therefore the remainder of this chapter assumes a fairly new implementation of FlexNet Manager Suite, and introduces the other kinds of set up necessary for a working system of Oracle introspection. Some of the latter may already be in place within your enterprise.

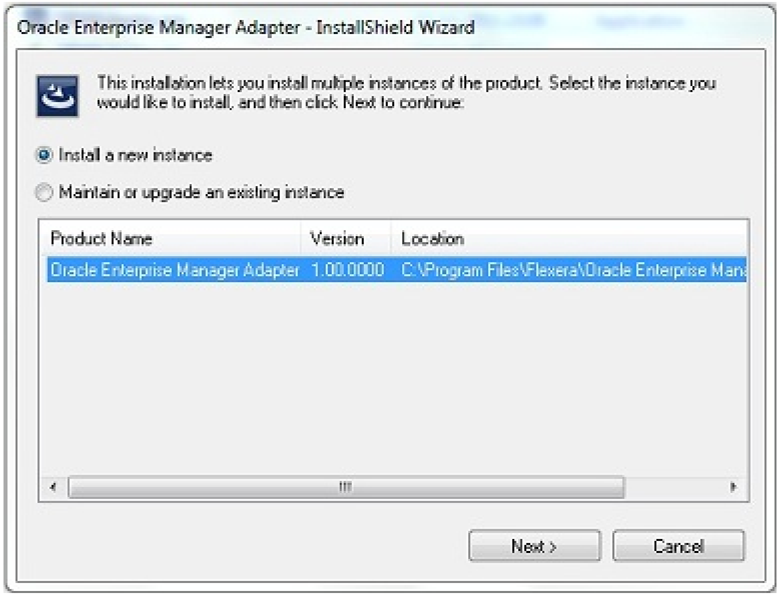
Installing the OEM adapter

The OEM adapter is normally installed on an inventory beacon that has high-speed network access to the Oracle server to which the adapter must connect. It is possible to install multiple instances of the OEM adapter on the same computer, each configured to access a different OEM repository. Each instance of the OEM adapter can access exactly one OEM repository.

1. From the unzipped Tier 1 adapter archive, navigate into the `Oracle Enterprise Manager Adapter` folder.

   For details about downloading the archive, see *Download Tier 1 Adapter Archive* on page 129.

2. In Windows Explorer, execute the `OEMAdapterInstaller.exe` installer from that folder.
   If this is the first installation of the OEM adapter on this computer, the welcome page appears, and you can click **Next**. If another instance of the OEM adapter is already installed on this computer, the following screen appears.



Oracle Enterprise Manager Adapter - InstallShield Wizard

This installation lets you install multiple instances of the product. Select the instance you would like to install, and then click Next to continue.

● Install a new instance

○ Maintain or upgrade an existing instance

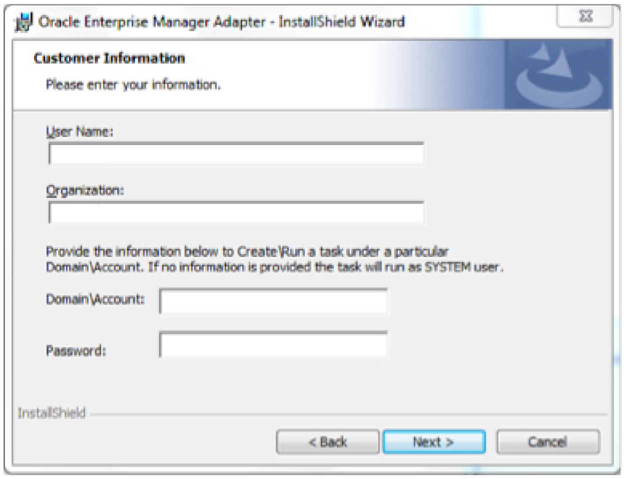
Product Name	Version	Location
Oracle Enterprise Manager Adapter	1.00.0000	C:\Program Files\Flexera\Oracle Enterprise Man

3. If this page appears, choose either of the following:

   a) For an additional installation, click **Install a new instance**, and click **Next**. Follow the remaining instructions below.

   b) To change one of the instances previously installed, select the instance from the list on this page, click **Maintain or upgrade an existing instance**, and click **Next**.

   The **Customer Information** page appears.



Oracle Enterprise Manager Adapter - InstallShield Wizard

Customer Information

Please enter your information.

User Name:

Organization:

Provide the information below to Create\Run a task under a particular Domain\Account. If no information is provided the task will run as SYSTEM user.

Domain\Account:

Password:

4. Identify the license owner, and the enterprise name, in the **User Name** and **Organization** fields respectively.
5. Provide credentials for the Windows scheduled task that triggers operation of the OEM adapter.

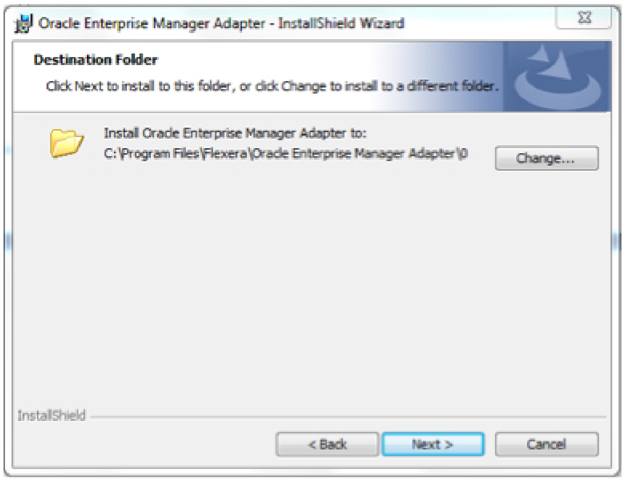
   Tip • *This account need not be the same as the one to access the OEM repository (identified in a later page).*

   Leave the fields blank to run the scheduled task as the local `SYSTEM` account on this computer.

   Note • *To change timing or frequency of the scheduled task, use the Microsoft Windows facilities as usual. They are not configurable through the installer.*

6. Click **Next >**.

   The **Destination Folder** page appears.



Oracle Enterprise Manager Adapter - InstallShield Wizard

Destination Folder

Click Next to install to this folder, or click Change to install to a different folder.

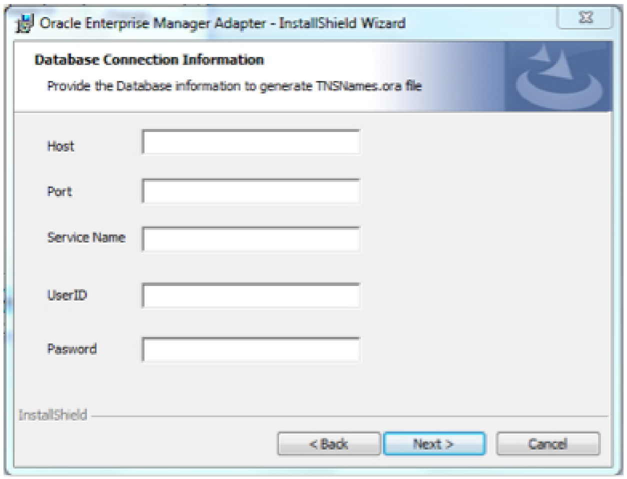
Install Oracle Enterprise Manager Adapter to:
C:\Program Files\Flexera\Oracle Enterprise Manager Adapter\

7. Optionally, click **Change...** to modify the supplied installation location.

***Note •** If you are installing multiple instances of the OEM adapter on this computer, you must modify the path so that each is installed in a separate folder. The InstallShield wizard offers the default folder `0` for the first instance. Increment this value to give a unique folder for each instance.*

***Important •** By default the OEM adapter saves the `.ora` file in the directory where it is executing (that is, the one you specify here). This is useful as a holding bay where the files may be manually inspected for initial testing/verification; but no automated processing of the `.ora` files occurs from this location. For automated operation after initial testing, you must reconfigure the file path and name as described in Configure Data Staging on page 158.*

8. When satisfied with the location, click **Next >**.
   The **Database Connection Information** page appears.



Oracle Enterprise Manager Adapter - InstallShield Wizard
Database Connection Information
Provide the Database information to generate TNSNames.ora file
Host
Port
Service Name
UserID
Pasword
InstallShield
< Back
Next >
Cancel

9. If necessary, confer with your Oracle DBA to complete details about the Oracle database from which this instance of the OEM adapter collects inventory:

   a) In the **Host** field, identify the server where Oracle Enterprise Manager is installed.

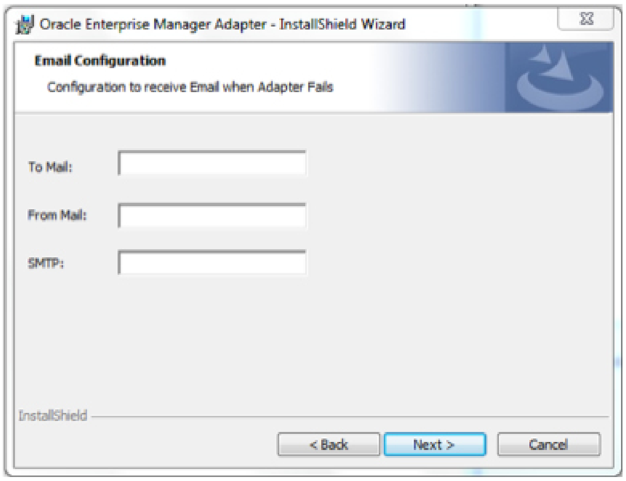
   b) The **Port** is used by the OEM adapter to access Oracle Enterprise Manager.

   c) **Service Name** identifies the Oracle service (for this database instance) that was defined when Oracle was installed.

   d) **UserID** is the account name (with its related **Password**) accepted by Oracle Enterprise Manager for login by the OEM adapter.

      Suggestion: `FNMS-OEMadapter`.

   When satisfied, click **Next >**, and the **Email Configuration** page appears.



Oracle Enterprise Manager Adapter - InstallShield Wizard

Email Configuration

Configuration to receive Email when Adapter Fails

To Mail:

From Mail:

SMTP:

10. To receive email alerts when the OEM adapter encounters any errors:

    a) Enter your email address as the **To Mail** value.

    b) In **From Mail**, enter the email address from which the error alerts should come.

    a) In the **SMTP** field, enter the fully qualified domain name (or IP address) of the email server.

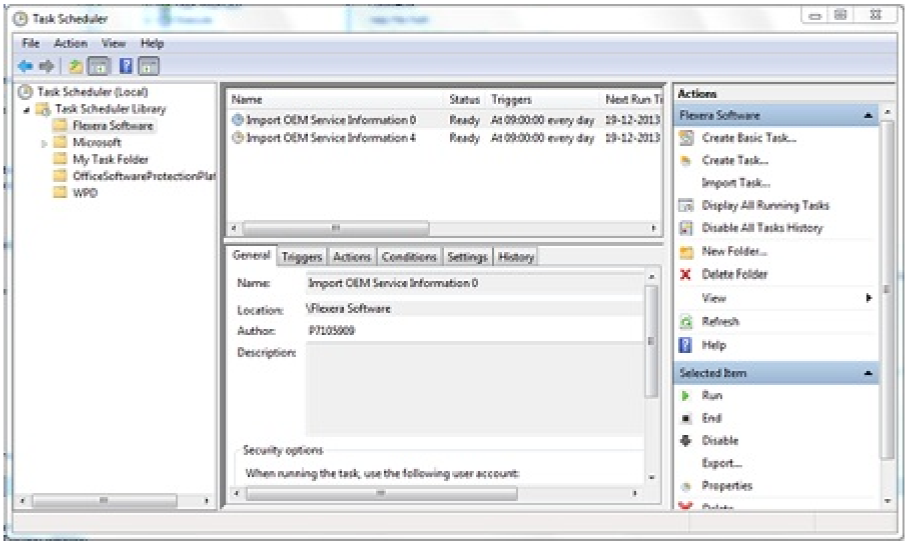
    *Tip • Ensure that the **To Mail** and **From Mail** addresses are both recognized by this email server.*

    a) Click **Next >**.

    The **Ready to Install the Program** page appears.

11. When satisfied, click **Install**.

The installer writes the OEM adapter executable (`OEMAdapter.exe`) into your chosen folder, and records your other settings in a configuration file (`OEMAdapter.exe.config`) saved in the same folder. It also creates a scheduled task to run the OEM adapter. To allow for multiple instances of the OEM adapter on the same computer, the scheduled tasks are named based on the numbering of the installed instances. The default scheduled task for the initial installation on a computer looks like this:



Name	Status	Triggers	Next Run Ti
Import OEM Service Information 0	Ready	At 09:00:00 every day	19-12-2013
Import OEM Service Information 4	Ready	At 09:00:00 every day	19-12-2013

Name:	Import OEM Service Information 0
Location:	\Flexera Software
Author:	P7105909
Description:	

Security options

When running the task, use the following user account:

You may update the scheduled task(s) as usual through the Microsoft interface.

Before using the OEM adapter in production, you must also do all of the following:

- Confirm that the account used to access Oracle Enterprise Manager (suggested name `FNMS-OEMadapter`) has adequate permissions to read from the OEM repository (see *Grant Permissions to Account* on page 146)
- Modify the location where the `.ora` file is saved for production use (see *Configure Data Staging* on page 158)
- Ensure that the appropriate subnets containing target Oracle servers are identified and assigned to the inventory beacon where the `TSNNames.ora` file is saved (see *Assign Beacon to Subnet* on page 151)
- Configure the collection of Oracle inventory in the web interface for FlexNet Manager Suite (see *Configure Collection of Oracle Inventory* on page 152)
- Set up accounts on each Oracle server with adequate permissions to gather inventory, with target machines being based on the contents of the .ora file created by the OEM adapter (see *Inventory-Gathering Accounts on Oracle Servers* on page 148)
- Register the same account(s) in the Password Store for each relevant inventory beacon (see *Save Inventory Account in Password Store* on page 151).

Grant Permissions to Account

Allow adequate access to the OEM repository.

The account used to access Oracle (suggested name `FNMS-OEMadapter`) must have adequate permissions to read from tables the OEM repository. If you are setting up multiple instances of the OEM adapter, repeat the process for each instance (you may be using a common account name for all of them to access each distinct OEM repository, or providing each with a unique account name). This task is normally completed by an Oracle DBA.

There are two possible ways to provide appropriate permissions:

1. Either: Make the account (suggested name `FNMS-OEMadapter`) a member of the `EM_ALLVIEWER` role; or
2. Give the account read permissions on the following:

Option	Description
Target views	• `MGMT_Targets` • `MGMT_TARGETS_LOAD_TIMES` • `MGMT_TARGET_TYPES`
Target properties views	• `MGMT_Targets` • `MGMT_TARGET_PROPERTIES`

3. When permissions have been set up, test as follows:

   a) Log in using the account name (suggestion: `FNMS-OEMadapter`) and password registered during the installation process.

   b) Execute the following query against the OEM repository:

   ```
   SELECT * from mgmt$Target
   ```

   Note the number of rows returned.

   c) Log out, and log in using a `DBroot` account (or other account known to have full permissions); and repeat the same query.
   The same number of rows should be returned by the root account with full permissions, and by the account for the OEM adapter with more limited and specific permissions.

4. Repeat the process and testing for any further instances of the OEM adapter accessing other distinct instances of OEM repository.

Other Setup Activities

The OEM adapter is relatively straight-forward, and, as triggered by the Windows scheduled task set up during installation, regularly collects data from Oracle Enterprise Manager and saves the data as a `TNSNames.ora` file in a special location on the inventory beacon.

However, by itself, when the goal is to gather inventory from Oracle database servers, the `TNSNames.ora` file is just the first step. Many other 'cogs in the machine' need to do their part for the `.ora` file to be beneficial:

- You need a defined set of sites and subnets, most easily obtained by Active Directory import through an inventory beacon.
- The subnet(s) containing your Oracle servers of interest must be assigned to the inventory beacon that collects the `TNSNames.ora` file, and which subsequently collects the Oracle inventory (see *Assign Beacon to Subnet* on page 151).
- An inventory rule must be defined that associates the target subnet with an action including inventory collection using the `TNSNames.ora` file, and schedules its execution (see *Configure Collection of Oracle Inventory* on page 152).

- Quite separately from the account that queries the OEM repository to create the `TNSNames.ora` file, one or more separate accounts must be defined that connect to the Oracle database servers and collect the inventory data. There are two parts to configuring these accounts:
  - The accounts must be created and given adequate permissions on each Oracle database server (there is a utility available to assist with this, as described in *Inventory-Gathering Accounts on Oracle Servers* on page 148)
  - The same accounts must be registered in the Password Store on the inventory beacon (see *Save Inventory Account in Password Store* on page 151).

Strictly speaking, none of these are part of the OEM adapter; but all are integral to the process of Oracle introspection, and therefore are at least summarized in the following sections.

Inventory-Gathering Accounts on Oracle Servers

The inventory beacon collects inventory from Oracle servers using accounts that must be registered at both ends.

This process sets up the inventory-gathering accounts on the Oracle servers.

1. Using your credentials supplied as part of order confirmation/fulfilment, log in to the Flexera Software knowledge base available through the Support pages of the Flexera Software website, and access the following knowledge article:

   ```
   https://flexeracommunity.force.com/customer/articles/en_US/INFO/Q200934
   ```

2. Scroll to the bottom of the article, and click the link `CreateOracleAuditUserQ200934.sql`.

   This downloads a SQL script that an Oracle DBA can review. (Ignore the text content which is for earlier products.)

3. In a flat text editor, modify lines 8 and 9 of the script, replacing the default user name and password with credentials of your choosing; and save the modified script.

   To minimize configuration and maintenance effort, the same credentials can be implemented on each Oracle database server. Keep a note of these credentials, as you must also record them in the Password Store on the inventory beacon.

4. An Oracle DBA can run the script on each target Oracle database server.

   This (re)creates the user name on each run, and grants read access to a numbers of tables and views (shown below) that are required for Oracle database introspection.

The downloaded SQL script gives the account read-only access to the following tables and views:

- `applsys.fnd_app_servers`
- `applsys.fnd_nodes`
- `applsys.fnd_product_installations`
- `applsys.fnd_application_tl`
- `applsys.fnd_user`
- `applsys.fnd_responsibility`

- `apps.fnd_user_resp_groups`
- `SYS.DBA_USERS`
- `SYS.V_$PARAMETER`
- `SYS.V_$INSTANCE`
- `SYS.V_$DATABASE`
- `SYS.V_$OPTION`
- `SYS.DBA_FEATURE_USAGE_STATISTICS`
- `SYS.DBA_ENCRYPTED_COLUMNS`
- `SYS.DBA_TABLESPACES`
- `ODM.ODM_MINING_MODEL`
- `ODM.ODM_RECORD`
- `DMSYS.DM$OBJECT`
- `DMSYS.DM$MODEL`
- `DMSYS.DM$P_MODEL`
- `DVSYS.DBA_DV_REALM`
- `LBACSYS.LBAC$POLT`
- `OLAPSYS.DBA$OLAP_CUBES`
- `SYS.DBA_AWS`
- `SYS.DBA_SEGMENTS`
- `SYS.DBA_CUBES`
- `SYS.GV_$INSTANCE`
- `SYS.GV_$PARAMETER`
- `MDSYS.ALL_SDO_GEOM_METADATA`
- `SYS.V_$SESSION`
- `SYSMAN.MGMT_LICENSE_DEFINITIONS`
- `SYSMAN.MGMT_ADMIN_LICENSES`
- `SYSMAN.MGMT_LICENSES`
- `SYS.DUAL`
- `SYSMAN.MGMT_LICENSE_CONFIRMATION`
- `SYSMAN.MGMT_TARGETS`
- `SYSMAN.MGMT_$TARGET`

- `SYSMAN.MGMT_VERSIONS`
- `SYSMAN.MGMT_INV_COMPONENT`
- `SYSMAN.MGMT_FU_REGISTRATIONS`
- `SYSMAN.MGMT_FU_STATISTICS`
- `SYSMAN.MGMT_FU_LICENSE_MAP`
- `SYS.DBA_REGISTRY`
- `SYS.V_$LICENSE`
- `SYS.DBA_TABLES`
- `CONTENT.ODM_DOCUMENT`
- `SYS.V_$VERSION`
- `SYS.USER_ROLE_PRIVS`
- `SYS.USER_SYS_PRIVS`
- `SYS.ROLE_SYS_PRIVS`
- `MDSYS.SDO_GEOM_METADATA_TABLE`
- `SYS.DBA_INDEXES`
- `SYS.DBA_LOBS`
- `SYS.DBA_OBJECTS`
- `SYS.DBA_RECYCLEBIN`
- `SYS.DBA_MINING_MODELS`
- `SYS.REGISTRY$HISTORY`
- `SYS.DBA_TAB_PARTITIONS`
- `SYS.DBA_TAB_SUBPARTITIONS`
- `SYS.DBA_LOB_PARTITIONS`
- `SYS.DBA_LOB_SUBPARTITIONS`
- `SYS.V_$ARCHIVE_DEST_STATUS`
- `SYS.DBA_SQL_PROFILES`
- `SYS.DBA_ADVISOR_TASKS`
- `SYS.DBA_SQLSET`
- `SYS.DBA_SQLSET_REFERENCES`
- `SYS.DBA_FLASHBACK_ARCHIVE`
- `SYS.DBA_FLASHBACK_ARCHIVE_TS`

- `SYS.DBA_FLASHBACK_ARCHIVE_TABLES`
- `SYS.V_$BLOCK_CHANGE_TRACKING`
- `SYS.V_$CONTAINERS`

Save Inventory Account in Password Store

The accounts that collect Oracle inventory must exist in the Password Store.

This process must be completed on the inventory beacon.

1. Log into the inventory beacon, using an account that has administrator privileges.
2. Start the FlexNet Beacon software from the Windows Start menu.
3. In the navigation bar on the left, select the **Password management** page, and click **Launch Password Store**.
   The separate interface for the Password Store opens.
4. In the **Current Password Store** group, click **New**.
   The controls in the **Editor** group are activated.
5. Provide a **Logical Name** to identify this set of credentials.

   Logical naming allows you to have one account name, but with different passwords on different servers. For more, see the online help.
6. For **Account Type**, choose **Account on Oracle database**.
7. Complete the **User** (account name) and **Password**.
8. Click **View/Edit filter...**.
   The **Password Store: Password Filter** dialog is displayed.
9. Use the **Oracle service names** filter to create a comma-separated list of the Oracle services for which this account/password pair should be used.

   The services names are visible in your `.ora` file. Filtering the account/password pair to apply only to specified Oracle services provides maximum efficiency for login and introspection.
10. Click **Apply** to close the dialog, and continue to save the entry in the Password Store.

    Repeat the process as required until all your Oracle machines are covered by one or more entries in the Password Store. Thereafter you may exit the FlexNet Beacon interface.

Assign Beacon to Subnet

Before rules can take effect, inventory beacons must know their subnets.

It is a requirement for an operational system that your subnets are assigned to appropriate inventory beacons. This summary covers only making adjustment for the collection of Oracle inventory.

1. In the web interface for FlexNet Manager Suite, navigate to **Discovery & Inventory** > **Subnets** (in the **Network** group).

   This page is populated with the sites and subnets in your enterprise after you have completed an import from Active Directory (for details, consult *FlexNet Manager Suite Help > Inventory Beacons > Active Directory Page > Importing from Active Directory*).

2. Expand the appropriate site(s), using the + expander icon, until you can see the subnet that includes your Oracle database servers.

   Tip • If the subnet does not yet appear in the listing, you can add its details manually. In the row for the appropriate site, click the + sign on the right-hand end, and enter the subnet IP address in the new row that appears.

3. If the **Beacon name** column shows `Unspecified`, click the editing (pencil) icon at the right-hand end of this row.
   The row becomes editable. Beware of accidentally over-writing the IP address, which initially has focus.
4. In the **Beacon name** column, use the option list to choose the appropriate inventory beacon from the list of those registered so far; and click the blue disk icon on the right to save your change.

Your chosen inventory beacon is now authorized to work in the subnet that contains your Oracle database servers. Now continue and create a rule that dictates what should happen within that subnet.

Configure Collection of Oracle Inventory

Rules must be established that allow Oracle inventory collection, including the use of `.ora` files.

Inventory rules are created on the Discovery and Inventory Rules page of the web interface for FlexNet Manager Suite. Each rule has three parts:

- A *target* that identifies the machines on which the rule is to be exercised
- *Actions* that are the be performed on those target machines
- The *schedule* on which the rule is to be applied.

When an action includes permission to use a `.ora` file, the relevant inventory beacon uses the locally-available `.ora` file to 'filter' the related target and identify the Oracle database servers from which inventory should be collected. For example, if you target an IP range that covers your server room, and include the action setting to apply the `.ora` file to this range, then Oracle inventory is collected only from the matching machines listed in the `.ora` file.

Once a discovery and inventory collection has been completed, the individual Oracle database servers are identified in the list of discovered devices. Should you wish to change to tighter targeting rules, you can use this information to redefine the target until (if you wish) it identifies exactly the Oracle database servers and no others. Keep in mind that if you use this approach, you will need to adjust the target each time you vary the list of your Oracle database servers. To reduce this manual maintenance, keep a target that specifies an appropriate IP range (or ranges), and applies the `.ora` file to this to identify the individual Oracle database servers from which to collect inventory.

1. In FlexNet Manager Suite, navigate to **Discovery & Inventory** > **Discovery and Inventory Rules** (in the **Discovery** group).
2. If you do not already have a target to reach the Oracle servers from which you want to gather inventory:

   a) On the left side, select the **Targets** tab.

***Fastpath* • In the hints area across the top of the page, click *Create targets*. These hints can guide you through the process.**

a) On the right side, click **Create a target**.

b) The page appearance changes, allowing you to define a target.

c) Complete the details requested, with particular attention on **Define machines to target**.
Notice that:

- After you complete each definition for this control, a + icon is displayed that allows you to add more to your definition of target machines. Use these lines to define a target sufficiently broad to capture your Oracle database servers.
- If you do not want the FlexNet inventory agent to be installed on these Oracle servers, be sure to select **Do not allow these targets to be adopted**.
- You should likely also select **Do not allow application usage tracking on these targets**, since you may not want to collect a large quantity of file evidence from these servers.

d) Click **Save**.

3. If you already have an action for Oracle discovery, you can check its settings; or create a new one:

a) Click the **Actions** tab (or in the hints section, click **Create actions**).

b) If you wish to collect hardware inventory for these servers (perhaps to assist with licensing calculations for your Oracle database license), expand the General accordion and select **Gather hardware and software inventory**.

*Tip • You may want to clear the check box for **Discover devices**, if you are limiting this action specifically to inventory collection for known Oracle database servers identified in your `.ora` file.*

c) Expand the **Oracle database** accordion.

d) If you are confident that every Oracle database is identified in your `.ora` file, you may clear the check box for **Discover Oracle databases**. Alternatively, if you may have rogue servers, leave this check box selected, and the network within your declared target will be probed for other Oracle servers.

e) Select the check box for **Also gather Oracle database inventory**.
Additional controls, if not already visible, are exposed.

f) Ensure that the **Port scan** check box is selected, and if necessary use the + icon to add additional ports to the list until every port listed in your `.ora` file is included.

g) Ensure that the **SNMP scan** check box is selected.

h) It is critically important that you select the **TNS names file** check box.

This setting authorizes the relevant inventory beacons to apply any `.ora` files found in the 'magic path' to the target used for this action (in the rule soon to be completed). This is the mechanism that most efficiently restricts probing to the relevant servers.

i) Adjust other settings in other parts of the accordion to suit your environment, and click **Save**.

The action is stored, ready for inclusion in a rule.

4. Select the **Rules** tab (or in the hints area, click the third **Create rules** step).

a) Do either of the following:

- If you have an existing rule to review or modify, click the edit (pencil) icon on its right-hand end.
- Click **Create a rule** (upper right) to define a new rule (as described in the following steps).

A rule builder work area appears above the list of existing rules.

b) Return to the **Actions** tab (for example, using the link in the rule builder), and on the row for your edited Oracle action, click **Add to rule builder**.
The name of your action appears in the rule builder.

c) Return to the **Targets** tab (for example, using the link in the rule builder), and on the row for your edited target, click **Add to rule builder**.

5. On the right side of the rule builder, click **Schedule**.

The rule builder changes to display controls for scheduling:

a) Choose a value from the **Frequency** option list.

This control sets the style of scheduling. For example, the `Daily` option lets you make further choices about a pattern of days, and does not enforce inventory collection every day.

Option	Notes
`Once`	A single-shot trigger for inventory collection the next time the declared time window occurs (you cannot nominate a future date). For example, if it is 4pm when you set a `Once` schedule for 8am, commencing within 4 hours, inventory collection occurs next morning. Compare with `As soon as possible`. Keep in mind the propagation delays for your changed instructions, as described there.
`Daily`	An additional option list, **Every**, appears so that you can choose the pattern of days you want. For example, you may wish to trigger inventory collection every second day. Choose a value from this list as well.
`Weekly`	New check boxes appear that allow you to choose specific days within the weekly cycle when inventory should be collected. For example, you may want collection on Sunday and Wednesday every week. Select (check) the boxes for the days you prefer.
`Monthly`	New controls appear that allow you to specify a pattern within the month: • Choose the option for **On day** to nominate a particular day within the month (such as the third Saturday). Make your choices from the two option lists adjacent. Notice that the option `Last` chooses the fifth

Option	Notes
	occurrence in months long enough to have one, and otherwise takes the fourth occurrence. • Choose the option for **On date** to nominate the calendar date within the month.
`As soon as possible`	This is a single shot trigger which causes each FlexNet inventory agent to randomize an inventory collection within the time window starting when it receives this setting and lasting for the interval you specify in the **Commence within** controls. 💡 *Tip • Any change you save to these settings is first collected by the inventory beacons on the schedule specified by the* ***Beacon settings*** *(further down this web page), by default checked every 15 minutes. Each inventory beacon then prepares new instructions for all the installed FlexNet inventory agents that it manages, which come calling to collect their latest package every 12 hours. Because of these propagation periods,* `As soon as possible` *on average means starting about six hours from now, or worst case a little over 12 hours from the time you save this change.*

6. In the rule builder, click **Save as**, give the rule a name you will recognize later in lists, and click **Save**.

If you accepted the default `Enabled` setting, the rule is ready to run on the schedule you have established. (Remember to allow around 15 minutes for the new rule to be distributed to your inventory beacons.) When the rule is executed on an inventory beacon, if a `.ora` file exists in the 'magic path', the systems that lie both within the inventory beacon's assigned subnet(s) and within the `.ora` file are targeted for Oracle inventory collection.

3

Modifying the Adapter

Topics:

This chapter covers changes you can make to the operations of the OEM adapter by modifying its configuration file. Read the first topic for general guidance on the editing process, and choose the detailed subtopics based on what changes you need.

Reconfiguring the OEM Adapter

During installation, you recorded your preferred settings for the OEM adapter, and no further work is required in the installation process.

If you later wish to modify the configuration of an instance of the OEM adapter:

1. In Windows Explorer, navigate to the correct folder for the appropriate instance of the OEM adapter.

   Keep in mind that there may be several instances installed on a computer, accessing distinct instances of Oracle Enterprise Manager.
2. In your preferred plain text (or XML) editor, open the OEM adapter’s configuration file, called `OEMAdapter.exe.config`.

   It is strongly recommended that you make a backup copy of the original configuration file, so that you can easily revert your changes if there are problems.
3. Use the following subtopics to guide your changes to the configuration file.
4. When you have finished, save the updated file.
5. Either wait until the next scheduled run of the OEM adapter, or use the Windows scheduled task interface to trigger an immediate test run to confirm that your changes work as expected.

Updating Connection Details

You can change all details and credentials for connection to the Oracle Enterprise Manager instance.

When the OEM adapter is first run, it encrypts the connection details used to access the Oracle Enterprise Manager implementation. Therefore, the process to edit the connection details is different, based on whether the OEM adapter has been run since the connection details were last recorded. You can tell whether this is so by looking in the configuration file.

The details are all stored in the `<connectionStrings>` element of the configuration file.

1. If the OEM adapter has been run since the details were recorded in the configuration file, the encrypted `<connectionStrings>` element appears similar to this:

   ```
   <connectionStrings configProtectionProvider="DataProtectionConfigurationProvider">
       <EncryptedData>
         <CipherData>
           <CipherValue>AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAeNhcGMAVK0uVGNTqOg/WbQQAAAACAA
              AAAAADZgAAwAAAABAAAACTO9kpn6BptpLvsXExg1UBAAAAAASAAACgAAAAEAAAAELCyiwz5A
              XZw9xZXfEPiAAAAgAAPJQYe+G9AfScFMJTYgA0NDbAgZdRA9nB91DN42A1xjeCskUs9+KNjV
              U1PSFRV4ujta40evf3IOZy5odyHsIrJRCKOOGdhDb1wh4ISEkpJk/QDna6LeCbbtXXsQK2Lo
              AHQc/plz77UkQZxnxkL5ElPIGHl6AojEXT2F5NGjElJX6GXbUXQDkNnDfi2o6XI/CDbX8gCu
              MonY1cTLYGe6+AQPpDgcY3rA02ZFOs7/Zb0cKOw7IoZdB6H8OIvrClSzqNkVBd3YfLhP/KOr
              kQFP8orqj54BJW74E1v3VUZnte1ESgLA5MYb/F9Ah3M5xi2Q6ITXOQmVRGESrivsdqr6nyGz
              5APx2yVBuEcoVhpOMYURbEbBSW+6/aydg8nY1DrcMzkPlXiZ0CQs4yZYHSWt3+bFEN30xh6X
              KFH7Tpc8e5y9Tq1Bhwk+Kw6AFfZhcdewApJ4ZkGAr4ixE6gNqWXnomk2puKNFImhJfqLLIuQ
              x9T00Uu2bjJ9Y+dUlrcuov12hQXOCh9nqOdmeIQHPXgw8//eHYOzwy2TgMcq2M2LxXRbQqCT
              eWtlP1ucJVyct9gnJlk2L3FSBNgMu1C9mncR7MsGDBWoQMXnwC5+Y6P1GT45sPOedBQvIQIT
              j7MCuzfgDD1R9c7w1gejTGmrl3Kmf33tGPMRYPV7CPNET3DUV9AGQUAAAAaIEkjyqfExrbei
              YJvG2usqYO3Nk=</CipherValue>
         </CipherData>
       </EncryptedData>
     </connectionStrings>
   ```

   In this case, delete the entire element and replace it with a plain text version as shown below.

2. If the OEM adapter has not been run since the configuration file was changed, it has a plain text form similar to this (line wrapping has been added here for legibility). Replace the italicized placeholders with your own values, as described below. Keep the `connectionString` attribute all on one line.

   ```
   <connectionStrings>
       <add name="OEMConn"
           connectionString="Data Source=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
                            (HOST=HostIPAddress)
                            (PORT=PortNumber))
                            (CONNECT_DATA=(SERVICE_NAME=ServiceName)));
           User Id=AccountName;Password=UnencryptedPassword;"></add>
       </connectionStrings>
   ```

   where the following details should be confirmed with your Oracle administrator:

   - *HostIPAddress* is the IP address of the Oracle server hosting Oracle Enterprise Manager
   - *PortNumber* is the port that the OEM adapter should use to query Oracle Enterprise Manager
   - *ServiceName* is the service name established when Oracle was installed on the Oracle server
   - *AccountName* is the account (user name) that the Oracle administrator has established for accessing Oracle Enterprise Manager
   - *UnencryptedPassword* is a plain text rendition of the password for the same account (remembering that all the connection details are encrypted together as soon as the OEM adapter runs next).
3. Save the amended configuration file.
4. Use the Windows schedule task interface to run a test of the OEM adapter to confirm that the connections details were correctly entered.

Configure Data Staging

Modify the name and location of the file of Oracle connection information.

The OEM adapter collects information about connections to Oracle systems from which you need to gather inventory information. This file is in a standard Oracle format, used for their `TNSNames.ora` file. It is saved by default where the OEM adapter is executing; but the default location for the OEM adapter does not support automated processing of the `TNSNames.ora` file when the inventory beacon applies rules for gathering Oracle inventory. For the OEM adapter to function seamlessly, you must customize the location as you edit the `OEMAdapter.exe.config` file.

Another reason to customize the file name is if you have multiple instances of the OEM adapter running from the same computer (or if anyone manually adds `TNSNames.ora` files to the processing directory). Each instance must write to a unique file name, so that one output does not over-write the other.

These settings live in an `add` element with the `key` attribute of `ConnectionInfoFile`.

1. In the `OEMAdapter.exe.config` file, locate the appropriate element. Its default values are similar to the following:

   ```
   <add key="ConnectionInfoFile"
        value="C:\Program Files\Flexera\Oracle Enterprise Manager Adapter\TNSNames.ora"></add>
   ```

   This reflects the default location of the OEM adapter, and must be modified.
2. Replace the `value` string with the new file path and name.

Important • *The file name **must** use the extension `.ora`.*

You may use a mapped drive on the local computer to specify a network path. This example shows the file in the recommended 'magic path' on the inventory beacon where the `.ora` file is automatically processed to 'filter' the target supplied from the central application server. Also notice the customized file name to avoid naïve overwriting of the `TNSNames.ora` file name by other copies saved here:

```
<add key="ConnectionInfoFile"
     value="C:\ProgramData\Flexera Software\Repository\TNSNames\TNSNames-01.ora"></add>
```

3. Save the amended configuration file.
4. Use the Windows schedule task interface to run a test of the OEM adapter to confirm that the staging file is saved according to your revised specifications. Check the location (recommended: `C:\ProgramData\Flexera Software\Repository\TNSNames\`) for the presence of a saved file immediately after the test run (if you wait too long, the resulting file may be automatically uploaded and removed from this staging location).

Managing Email Alerts

You can turn the alerts off, changes their addresses, or switch email servers.

The OEM adapter can send email alerts any time that it encounters an error. Manage the emails with the following XML elements in the `OEMAdapter.exe.config` file.

1. To stop email alerts entirely comment out the *reference* to the `SmtpAppender` like this:

```
<appender-ref ref="RollingFileAppender"></appender-ref>
<!-- <appender-ref ref="SmtpAppender"></appender-ref> -->
```

(The character sequence `<!--` starts an XML comment, and the sequence `-->` closes the comment.)

Tip • *Using this technique is preferable to deleting this line entirely, as this can be easily reversed if you decide to reinstate email alerts in future. Notice that, depending on the configuration you saved, there may be other elements between the `<appender-ref>` tags.*

2. To reconfigure the email alerts, locate the `<appender>` element, and modify three of its child elements by changing the example values shown in italics here:

```
<appender name="SmtpAppender" type="log4net.Appender.SmtpAppender">
   <to value="toaddress@somedomain.com"></to>
   <from value="fromaddress@somedomain.com"></from>
   <subject value="OEM Adapter Error"></subject>
   <smtpHost value="smtp.somedomain.com"></smtpHost>
</appender>
```

3. Save the amended configuration file.
4.

Configure Logging

Change the path and file name for logging by the OEM adapter.

The log file name and location cannot be set during installation of the OEM adapter, but after installation, they can be modified as follows:

1. In the `OEMAdapter.exe.config` file, locate the `<appender>` element called `RollingFileAppender`.
2. Edit the child `<file>` element by replacing the drive, path and file name shown as placeholders here:

```
<appender name="RollingFileAppender"
          type="log4net.Appender.RollingFileAppender">
   <file value="drive:\path\path\logfilename.txt"></file>
</appender?
```

3. Save the amended configuration file.

After the next run of the OEM adapter, inspect the log in your new location.

Part VI

ServiceNow Integration with FlexNet Manager Suite

Topics:

You can exchange a limited set of data between these systems to provide a consistent view of your hardware and software estate, and related contracts.

ServiceNow provides cloud-based IT Service Management, while FlexNet Manager Suite is focused on Software Asset Management. To help provide a unified view of your management data, there are three parts to the integration of these systems, each of which is available independently of the others:

- Data on hardware assets, application installations, and contracts can be exported from FlexNet Manager Suite and imported into ServiceNow
- Data on assets and contracts can be exported from ServiceNow and imported into FlexNet Manager Suite
- To provide a "single pane of glass", the web interface for FlexNet Manager Suite may be displayed within the ServiceNow interface, with high-level menu items integrated into the ServiceNow menu bar. (This is not a *requirement*, and separate operation of the two products remains supported.)

This release of the ServiceNow integration package supports the following versions of ServiceNow:

- Eureka
- Dublin
- Fuji.

1

Architecture, Components, and Prerequisites

Topics:

This chapter provides the background information you need to get started. It is valuable to understand the concept of 'sources of truth' which guides the flow of information between the two systems:

- FlexNet Manager Suite is authoritative on hardware assets, application installations, and contracts.
- ServiceNow is authoritative on assets and contracts.

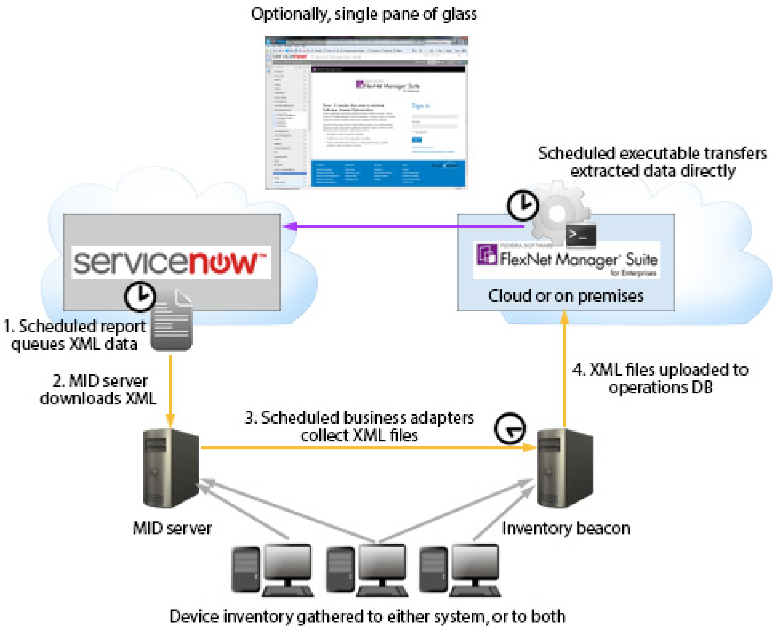
These ideas drive the default configuration of exports and imports. (You can also customize the exports, as described in a later chapter.)

Architecture and Operation

This big-picture overview provides context for understanding both configuration and operation of the integration package.

While ServiceNow is cloud-based, FlexNet Manager Suite may be either cloud-based or implemented on your premises. The following architecture applies to both these implementation models.

__Tip__ • While you may, of course, continue to use ServiceNow and FlexNet Manager Suite in separate browser windows, you may also choose to use FlexNet Manager Suite within the ServiceNow page to provide a 'single pane of glass' for managing your assets, licenses, and contracts.


Optionally, single pane of glass
Scheduled executable transfers extracted data directly
servicenow
FlexNet Manager Suite
Cloud or on premises
1. Scheduled report queues XML data
2. MID server downloads XML
4. XML files uploaded to operations DB
3. Scheduled business adapters collect XML files
MID server
Inventory beacon
Device inventory gathered to either system, or to both

The following two processes (illustrated in the diagram) operate completely independently:

- The import of computer and contract data from ServiceNow into FlexNet Manager Suite (gold arrows).
- The export of computer, application, and contract data from FlexNet Manager Suite into ServiceNow (purple arrow).

Import from ServiceNow

1. On a schedule you configure in ServiceNow, two export jobs drop all hardware asset and contract records into XML files.
2. The MID server collects the XML files from your ServiceNow cloud server.

> ***Note •*** *The MID server collects the XML data in chunks, such that the XML file is not well formed until the last chunk has been received. The business adapters in the next step should only be run once the export from ServiceNow is complete.*

3. On a separate schedule (configured to allow plenty of time for the previous steps to complete), two business adapters execute on your FlexNet inventory beacon, transferring the XML files to the beacon.
4. Once the files are complete on the beacon, they are uploaded to your central application server (either your own on premises server, or your cloud server). Should there be any networking problems, a catch-up job will re-try the uploads overnight.
5. Within a few minutes of arriving on the central application server, your files are queued for import into the operations database.

Export to ServiceNow

1. On a schedule (configurable for on premises implementations), an executable on the central application server reads application, computer, and contract data from the FlexNet Manager Suite operations database. (For cloud implementations, this schedule triggers at 3am on Sunday mornings.) At other times, an administrator can also trigger an import manually from the **ServiceNow Integration** page (**Settings** > **System Settings** > **ServiceNow**; and also see *Configuring Integration through the Web Interface* on page 170).
2. The data is transferred directly to staging tables set up in the ServiceNow instance.
3. A series of requests are also set up for ServiceNow. As these requests are processed, a set of transforms migrate the data from the staging tables to the operational database for ServiceNow.

Note • *A standard implementation of ServiceNow has some performance issues with step 3 in particular, and most especially on the initial data transfer, which is the largest one. (A worked example: data exported from FlexNet Manager Suite to the staging tables in four hours then took seven days to transform into ServiceNow.) You can register a support request with ServiceNow to create some additional indexes on the ServiceNow database tables (in the worked example, the indexes provided a 450% performance improvement, reducing the upload time from 7 days to 37 hours). For details on the indexes to request, see Additional ServiceNow Indexes for Performance on page 196. For further performance improvement, subsequent exports from FlexNet Manager Suite are differential downloads, so that ServiceNow only needs to process records that contain change of some kind (additions, updates, and deletions).*

Functionality Summary

This summary applies to the following product configuration:

- Integration package release number: 2.0 (includes `FlexNet Manager Suite Integration.v2.0.xml`)
- Minimum version of FlexNet Manager Suite: 9.2 SP2 and SP3 (these support command-line tool only). Release 2014 (10.0) or later required for integration of the web interface for FlexNet Manager Suite within ServiceNow.

- Integration supports either the on premises implementation or the cloud implementation of FlexNet Manager Suite.

This chart summarizes the capabilities of the ServiceNow integration package, once configured. in the chart, "Export" means transfer from FlexNet Manager Suite to ServiceNow, and "Import" means transfer from ServiceNow to FlexNet Manager Suite. (The items listed in this table make a standard set used for comparing functionality of different adapters.)

Database Item	Export to ServiceNow	Import from ServiceNow	Comments
Computers (Hardware inventory & Assets)	Yes	Yes	
Users	No	No	Not relevant to this integration.
Application installations	Yes (with related computer IDs)	No	
Installer evidence	No	No	Not relevant to this integration.
File evidence	No	No	Not relevant to this integration.
Product codes	No	No	Not relevant to this integration.
Flexera ID	Yes (as application property)	No	
Usage	No	No	Not relevant to this integration.
Virtualization	No	No	
Clusters	No	No	
Licenses and license counts	No	No	FlexNet Manager Suite is considered the source of truth for licenses.
Oracle LMS	No	No	Not relevant to this integration.
Contracts	Yes	Yes	

Components Included

When you download the ServiceNow Integration pack, it includes the following components:

- An `Exporter` folder which, in its entirety, must be copied to the appropriate folder on your central application server for on-premises implementations (see *Installation of the Integration Package* on page 169). The folder contains:
  - The `Database` folder, which includes three SQL queries to extract content from your operations database for export to ServiceNow

  - The executable `fnmp_servicenow_export.exe`, responsible for extracting data from the FlexNet Manager Suite operations database and importing it into ServiceNow
  - Its configuration file `fnmp_servicenow_export.exe.config`
  - `log4net.xml` configuration file
  - Five DLL files, namely
    - `DevExpress.Data.dll`
    - `DevExpress.Xpo`
    - `Flexera.Broker`
    - `Flexera.LicenseManagement`
    - `FlxCore.dll`
    - `FixDotNetClient.dll`
    - `log4net.dll`
    - `ManageSoft.Database.dll`
    - `Newtonsoft.Json.dll`
    - `SNC.Integration.JSON.dll`.
- An `Importer` folder, containing two business adapters, `ServiceNowAssets.xml` and `ServiceNowContracts.xml`, to import content from ServiceNow into FlexNet Manager Suite.
- A `ServiceNow` folder, containing `FlexNet Manager Suite Integration.v2.0.xml`.
- A copy of this documentation. The same content is also released within the online help for FlexNet Manager Suite.

Prerequisites

The following are requirements for implementing and operating the integration package for ServiceNow and FlexNet Manager Suite.

- You must have a license issued by Flexera Software that permits use of the integration package. This license authorizes all communications in both directions through the integration package.
  - To check in the web interface for release 2014 or later, navigate to the system menu (⚙▾ in the top left corner) > **FlexNet Manager Suite License**. Check under the **Subscription details** on the left.
  - For release 9.2, use the license activation wizard.
- You need a functional implementation of FlexNet Manager Suite.
- You must have an operational inventory beacon that communicates with your central application server.

- You have an operational ServiceNow instance. This release of the ServiceNow integration package supports the following versions of ServiceNow:
  - Eureka
  - Dublin
  - Fuji.
- Your ServiceNow instance must have the Software Asset Management plugin installed and enabled.
- You need a MID server configured for your ServiceNow implementation.

  ***Tip** • If you prefer, your FlexNet Beacon and ServiceNow MID server can be implemented on the same physical computer.*

- You need to download the ServiceNow integration package from the Flexera Software Product and License Center:
  - Navigate to `https://flexerasoftware.flexnetoperations.com`.
  - Log into the **Product and License Center** with your customer credentials, and navigate through `FlexNet Manager Platform` to find the ServiceNow integration package.
  - Download the zipped archive to a convenient location (such as `C:\temp`) and extract the files. The following instructions assume that you have this archive available to you.
- You most likely need to request a support ticket with ServiceNow to add several indexes to your database to improve performance in the data transfer from FlexNet Manager Suite to ServiceNow. For details, see *Additional ServiceNow Indexes for Performance* on page 196.

***Important** • Managed Service Providers (MSPs) and other operators of cloud implementations must modify the standard configuration of the data export tool to allow for use in multi-tenant environments. See Command-line Tool for Export to ServiceNow on page 184.*

2

Installation and Configuration

Topics:

Because the system involves a number of different servers, there are several installation and configuration steps. These are described in full in the following topics. For greater clarity, the first topic below includes a summary list of which topics apply for cloud-based implementations of FlexNet Manager Suite, and which apply for on premises implementations.

Installation of the Integration Package

This table summarizes which of the processes must be completed for a cloud-based implementation of FlexNet Manager Suite, and which for an on premises implementation (where you have your own central application server). Regardless of the implementation type, the integration package installation may overwrite any ServiceNow customizations like changes to transform maps or system properties. It is strongly recommended that you reapply such customizations after you install this integration package.

Procedure	For cloud-based implementations	For on premises implementations	For details, see:
1. Install the exporter executable and associated files	Not required	Yes	See below on this page
2. Configure the ServiceNow settings in FlexNet Manager Suite	Yes	Yes (for the 2014 release and later)	*Configuring Integration through the Web Interface* on page 170
3. Optimize data transfer settings	Not required	Optional	*Configure the Utility for Export from FlexNet Manager Suite* on page 172
4. Import the customization package into ServiceNow	Yes	Yes	*Configuring ServiceNow for Integration* on page 174
5. Configure the ServiceNow customization	Yes	Yes	*Configuring ServiceNow for Integration* on page 174
6. Install and schedule the business adapters	Yes	Yes	*Configure Imports from ServiceNow* on page 179

To install the exporter executable on your central application server (only for on premises implementations of FlexNet Manager Suite):

1. Navigate through your unzipped archive (see download instructions in *Prerequisites* on page 166) to the `Export` subdirectory, and identify the following:

    - The `Database` folder, which includes three SQL queries to extract content from your operations database for export to ServiceNow
    - The executable `fnmp_servicenow_export.exe`, responsible for extracting data from the FlexNet Manager Suite operations database and importing it into ServiceNow
    - Its configuration file `fnmp_servicenow_export.exe.config`
    - `log4net.xml` configuration file
    - Five DLL files, namely
        - `DevExpress.Data.dll`
        - `DevExpress.Xpo`
        - `Flexera.Broker`
        - `Flexera.LicenseManagement`

  - `FlxCore.dll`
  - `FixDotNetClient.dll`
  - `log4net.dll`
  - `ManageSoft.Database.dll`
  - `Newtonsoft.Json.dll`
  - `SNC.Integration.JSON.dll.`

2. Copy the entire contents of the `Export` subdirectory to `*installation-directory*\DotNet\bin\ServiceNowExport` on your central application server.

   Note • *If you have scaled up to a multi-server implementation, copy these files to this folder on your processing server (or where your implementation has scaled up to separate this into multiple servers, on your batch server).*

Tip • *The PowerShell scripts used when installing FlexNet Manager Suite (since the 2014 release) have already created a Windows scheduled task `Export to ServiceNow` to export data (the exports are dependent on the license term being present). This means that, even if the appropriate license term is added later, exports are already configured. By default, the export is scheduled for 3am Sunday morning. You may update that scheduled task if you wish to change the default schedule.*

Now continue with the remainder of the configuration tasks, as listed above for your particular configuration.

Configuring Integration through the Web Interface

You can set up the integration with ServiceNow through the web interface of FlexNet Manager Suite (from version 2014, and later).

You need to complete this configuration once to commence operations. Repeat if your ServiceNow details change at any time in the future. You may also use this same page to trigger an immediate export from FlexNet Manager Suite to ServiceNow.

Tip • *This configuration does not affect the reverse path, importing data from ServiceNow. For that, you must configure the jobs on ServiceNow (see Configuring ServiceNow for Integration on page 174), and set up the business adapters on an inventory beacon to import the results (see Configure Imports from ServiceNow on page 179).*

1. Ensure that you are logged in with an account that is a member of the `Administrator` role, and that this account has the **Configure** access right.
2. In the web interface of FlexNet Manager Suite, navigate to the system menu (⚙▼ in the top right corner) > **System Settings**, and select the **ServIceNow** tab.

   Tip • *The **ServiceNow** tab is only visible when your account has the access rights described above.*

3. In **Instance URL**, enter the protocol and path to your ServiceNow website (for example, `https://myServer.service-now.com`).
4. In **Username**, enter the account you use to access ServiceNow. This must exactly match the **User ID** value shown in the ServiceNow interface:

   a) Navigate in ServiceNow to **FlexNet Manager Suite integration** > **Credentials**.

   b) Identify the correct set of credentials from the list presented, and click on the hyperlinked **User** name on the right (not the credentials **Number**).

   c) In the **User** properties, the **User ID** is the top left field.
5. Copy the **Token** value from ServiceNow and paste into the **Token** field on this page. To find the token:

   a) Navigate in ServiceNow to **FlexNet Manager Suite integration** > **Credentials**.

   b) Identify the correct set of credentials from the list presented, and click on the credentials **Number**.

   c) In the **Credentials** properties, click **View Token** and copy the value from the pop-up dialog. (If there is no token value yet, first click **Generate Token**, and don't forget to click **Update** to save the modified details.)
6. If necessary, change the default selection of data types to export to ServiceNow. It is best practice to export all available data types. (ServiceNow only returns hardware asset records and contract details for synchronization. FlexNet Manager Suite is considered the source of truth for applications and licensing, and so includes installed applications in its export.)

   Warning • *Hardware inventory details are critical to the ServiceNow data set. Contracts and applications both have dependencies on assets in that system. If you clear the **Hardware inventory** check box, you may produce gaps in the contract and software records imported into ServiceNow.*

7. Click **Save**.
   When details are complete, the export from FlexNet Manager Suite to ServiceNow is triggered on a regular schedule. In the cloud implementation of FlexNet Manager Suite, the export is triggered at 3am on Sunday mornings. For on premises implementations, the default is also 3am on Sunday mornings, but you may edit the Windows scheduled task `Export to ServiceNow` to modify the schedule. If you need to trigger an additional export (for example, when commencing integrated operations), continue with the rest of this process.
8. Optionally, click **Export** to trigger an immediate export using the settings saved in the fields displayed above. This button, and the scheduled task, trigger the command-line executable `fnmp_servicenow_export.exe`, which;

   - Extracts the required data from your operations database

- Segments the data for easier transmission across the Internet, minimizing time-out risks with ServiceNow
- Reassembles the data into staging tables in ServiceNow.

Separate transforms on your ServiceNow instance then map the data from the staging tables to your operational ServiceNow CMDB.

Tip • *If you have an on premises implementation of FlexNet Manager Suite, you may also run the executable from the command line (see Command-line Tool for Export to ServiceNow on page 184).*

For a cloud implementation, configuration of exports from FlexNet Manager Suite to ServiceNow is now complete (and uses the default values listed below). In on premises implementations, you may also configure:

- The number of database records included in each transferred segment (default 8000)
- The number of retries if ServiceNow returns a connection failure (default 10)
- The length of time to wait for ServiceNow to respond before timing out (default 30 seconds)
- The maximum number of records of each data type to be included in the transfer (default 500,000 for inventory and contracts; for applications, the default is 2,000,000)
- The file name, path, and roll-over triggers for your log files (log files are not available through the cloud implementation, and for logged details you need to contact Flexera Software Support.).

For details of configuring these additional elements in an on premises implementation, see *Configure the Utility for Export from FlexNet Manager Suite* on page 172.

Configure the Utility for Export from FlexNet Manager Suite

Optionally, the configuration file for the export utility may be customized to suit your environment.

The configuration file called `fnmp_servicenow_export.exe.config` is located in the same folder as the executable called `fnmp_servicenow_export.exe`, which performs the data export from FlexNet Manager Suite to ServiceNow.

1. Save a backup copy of the configuration file, for example as `fnmp_servicenow_export.exe.config.orig` so that you can revert if there are problems.
2. Open the configuration file `fnmp_servicenow_export.exe.config` in your preferred XML or plain text editor. (Do not edit in a word processor that cannot save plain text files.)
3. Optionally, customize the values for data segmentation, retries, and timeout when the executable is accessing ServiceNow.

Path/element	Attribute	Default	Notes

`<configuration>`
`  <fnmpGroup>`
	`maxRetries`	`"10"`	When ServiceNow is not responding, the executable

Path/element	Attribute	Default	Notes

`<fnmp>`
`<baseConfig />`
			will retry each connection (one for each data segment being transferred) this many times. If this number of retries is reached for any data segment, the export fails. Each successful connection resets the count to zero. (If this attribute is omitted, the executable uses a default value of 10.)

`<configuration>`
`<fnmpGroup>`
`<fnmp>`
`<baseConfig />`
	`timeout`	`"30000"`	The maximum number of milliseconds to wait for a response from ServiceNow. This timer is applied independently to each connection request (at least one per data segment being transferred). If ServiceNow does not respond within this interval (default 30 seconds), a new request is issued (up to `maxRetries`). (If this attribute is omitted, the executable uses a default value of 30000.)

`<configuration>`
`<fnmpGroup>`
`<fnmp>`
`<records />`
	`numRecords`	`"8000"`	The number of records in each XML-based data segment to be transferred from FlexNet Manager Suite to ServiceNow. To avoid time-out issues with ServiceNow, the total exported data (in each of the three available types) is chunked into segments containing a maximum of this number of records.

4. Ensure that the record limits for each data type are adequate to your needs.

   `<configuration><fnmpGroup><fnmp>` includes XML elements for each of the `<application>`, `<contract>`, and (hardware) `<inventory>` data types. Each of these elements has a `maxRows` attribute that defaults to `"500000"`. With these values, if any of your selected data types exports more than a half-million rows, the data export is truncated at that record count. Since the ordering of rows is determined by the database, when there are excess rows, you may have a different set returned for each report. Therefore ensure that this limit is adequate to your needs. (If this attribute is omitted, the executable uses a default value of only 10,000 rows.)

5. Optionally, configure the location for your log files.

Path/element	Attribute	Default	Notes

`<configuration>`
`  <log4net>`
`    <appender>`
`      <file />`
	`value`	`"C:\Temp`
		`\Log`
		`\ServiceNow_`
		`date``.log"`
		(All on one line)
			You may adjust the path to store the log files to suit your preference. For on premises installations, you may also change the file name.

With the other default values for the elements under appender, multiple passes on the same date append to the end of the log file, and a new log file is commenced each day.

Configuring ServiceNow for Integration

ServiceNow runs specific reports to export data to FlexNet Manager Suite.

Configuring ServiceNow requires importing an update set (an XML file), setting up credentials, and customizing the schedules. The schedule for extracting data from ServiceNow, and the MID server on which the extracted data is staged, are both defined within ServiceNow. You can also use the following processes to monitor the progress of exported data for testing.

Warning • *The ServiceNow documentation warns you to import update sets outside business hours, as this process can impact the performance of your ServiceNow instance.*

1. In ServiceNow (logged in as a ServiceNow administrator), navigate to the **System Update Sets** group, and select **Retrieved Update Sets**.

   A list of already registered updated sets is displayed.
2. Below the list, click on the **Import Update Set from XML** link.
3. Under **(1) Choose file to upload**, click **Browse...** and navigate in your unzipped integration download to the `ServiceNow` subdirectory. Select `FlexNet Manager Suite Integration.v2.0.xml`.
4. Click **Upload**.
   When your file finishes uploading, it displays in the list of **Retrieved Update Sets** with a **State** of `Loaded`.
5. Click on the hyperlinked **Name** of your update set.
   The **Retrieved Update Set** properties are displayed.
6. ***Tip*** • *You cannot commit the update set until you have previewed it and addressed any problems.*

   Below the properties, click **Preview Update Set**.
   The **Progress** (preview completion) page indicates when problems are detected with a red `Error` display in the **Completion code** field. You must address any problems that may be detected (see section 3.1 and 3.2 in *http://wiki.servicenow.com/index.php?title=Transferring_Update_sets#Preview_Remote_Update_Sets*).
7. When there are no (remaining) errors, click **Commit Update Set**.

8. Optionally, validate the setting for the system property `com.fnmp.execute.deleteaction.installation`.

   When you remove an application from, FlexNet Manager Suite, the item record is marked for deletion and sent to ServiceNow. This property determines whether the application records marked by FlexNet Manager Suite for deletion are deleted from ServiceNow, or held pending a later decision to delete them. By default, the property is `false`, so that during initial testing, there is no surprise from records being deleted unexpectedly. To check the current setting of this system property in ServIceNow:

   a) Navigate to **System Definition** > **Tables**, and search for the `sys_properties` table.

   b) Click **Show Records in List**.

   c) Search for `%fnmp`, and select the record for `com.fnmp.execute.deleteaction.installation`.

   The default value is `false`. Optionally, switch this option on now, or there is a reminder to attend to it later.

9. Optionally, validate the setting for the system property `com.fnmp.execute.deleteaction.inventory`.

   When you delete an inventory item from FlexNet Manager Suite, the item record is marked for deletion and sent to ServiceNow. This property determines whether the inventory items marked by FlexNet Manager Suite for deletion are deleted from ServiceNow, or held pending a later decision to delete them. By default, the property is `false`, so that during initial testing, there is no surprise from records being deleted unexpectedly. To check the current setting of this system property in ServIceNow:

   a) Navigate to **System Definition** > **Tables**, and search for the `sys_properties` table.

   b) Click **Show Records in List**.

   c) Search for `%fnmp`, and select the record for `com.fnmp.execute.deleteaction.inventory`.

   The default value is `false`. Optionally, switch this option on now, or there is a reminder to attend to it later.

10. Optionally, set the Source Of Truth for inventory and installation records that ServiceNow imports from FlexNet Manager Suite

    When multiple applications contain different data about the same inventory, contract, or installation items, the source of truth setting determines which application is the definitive source of data. For example, ServiceNow and FlexNet Manager Suite can have different records for the same asset or inventory item. To configure this setting:

    a) Navigate to **System Definition** > **Tables**, and search for the `sys_properties` table.

    b) Click **Show Records in List**.

    c) Search for `%fnmp`, and set the following properties to `true` to make FlexNet Manager Suite as the source of truth for inventory, contract, and installation export types:

    - com.fnmp.source.inventory.insert
    - com.fnmp.source.contract.insert
    - com.fnmp.source.installation.insert
    - com.fnmp.source.inventory.update
    - com.fnmp.source.contract.update
    - com.fnmp.source.installation.update

This finalizes the update set in your ServiceNow instance. You can now attend to credentials and scheduling.

11. Still in ServiceNow, expand the **Flexera FlexNet Manager Suite integration** group, and select **Credentials**. The **Credentials** page lists all currently available credentials. (If an appropriate account to use for this integration package does not yet exist, create it.)

12. Click the hyperlinked credential **Number** to open the properties for this account.

    To avoid passing credentials across the network, ServiceNow provides an integration **Token** to be used for remote access.

    - If the **Token** field is currently empty, click **Generate Token**, and continue with the next step.
    - When the **Token** field displays a row of dots, click **View Token**.

    A pop-up modal dialog appears, displaying the current value of the token for the selected credentials.

13. Copy the value of the token (and then dismiss the dialog), and save the token in a secure and convenient location. You need to supply this token both to the web interface for FlexNet Manager Suite, and also (for on premises implementations) in the command line of the executable, any time that you choose to run it manually.

14. In ServiceNow, in the **Flexera FlexNet Manager Suite integration** group, select **Scheduled Jobs**. The list of standard jobs to export data from ServiceNow is displayed, including:

    - `Export Assets From ServiceNow`
    - `Export Contracts from ServiceNow`.

    Each of these jobs needs to be configured for your environment.

15. Click on the scheduled job you want to configure. The **Scheduled Script Execution** properties are displayed.

    Note • *The MID server collects the exported XML data in chunks, such that the XML file is not well formed until the last chunk has been received. Be sure to schedule sufficient time for the exports to complete before the scheduled import from the MID server into FlexNet Manager Suite.*

16. Change the value in the **Run** field to suit your business needs.

    These data exports may be large, based on the size of your ServiceNow repository. A common practice would be to choose `Weekly` exports.

    Tip • *ServiceNow by default limits its exports from the database to 10,000 records (for a general discussion, see https://community.servicenow.com/community/blogs/blog/2014/08/01/increasing-the-export-limit, but note that this page discusses limits for exports from the ServiceNow user interface grids, which are not relevant to the database exports to FlexNet Manager Suite). You can modify the maximum number of records exported from ServiceNow by having your ServiceNow administrator add the property `glide.db.max_view_records` and setting a new limit. For instructions on adding the property, see http://wiki.servicenow.com/index.php?title=Adding_a_Property#gsc.tab=0.*

17. Use the additional fields that appear to complete your scheduling.

    For example, for `Weekly` exports, set the **Day**, and **Time** of day, when the export should occur. An off-peak time is preferable.

18. In the **Run this script:** pane, edit the value to match your MID server.

    Tip • *While editing the MID-server-ID field, be careful not to change the database view name (`u_fnmp_asset` in the following example). The database view name matches the declared file name in ServiceNow.*

    The script looks similar to the following (this example is for exporting assets), and you should replace the placeholder *MID-server-ID* with the ID of the appropriate MID server:

    ```
    var ae = new FNMPFileExportProbe("MID-server-ID","u_fnmp_asset");
    ae_generateExport();
    ```

    Tip • *To validate the code of your MID server, navigate to the **MID server** group, and select **Servers**. This list shows the **Host name** (so you can check on the correct server), its **Status** (`Up` in a green field is good), and also the **Name**, which is the ID to insert in the export script.*

19. Click **Update** to save your changes.

    Repeat the configuration of MID server and schedule for all the export tasks.

20. Click the adjacent **Execute Now** button to test the export from ServiceNow. (This same button can be used to run the export if you chose the **Run** `On Demand` setting for the schedule.)

    Tip • *If you left the default value for the system property `com.fnmp.execute.deleteaction`, any items marked for deletion by FlexNet Manager Suite are not deleted in this test. Remember that for production runs, you should set this system property to `true`. Once this is true, the deletion clean-up occurs after each subsequent data integration, including acting on any records from past imports that carry the flag (set by FlexNet Manager Suite) that marks them for deletion. This means that even with differential data transfers, where FlexNet Manager Suite exports the data items only once each, items previously marked for deletion when the system property may have been `false` are still correctly processed (and therefore deleted) when you turn the property to `true`.*

    If you wish to monitor data flow, continue the rest of this process.

21. From the **ECC** group, select **Queue**. In the **Queues** list, click the **Created** column header to show the most recently-created entries at the top of the list.

    The MID server agent collects content from this queue.

22. Click the hyperlinked date/time under **Created** to inspect the contents of each package.

    Because data is chunked, the first segment just lays down the opening elements for the XML file; subsequent segments add data entries; and the last closes the XML elements and reports success. You can check the files in your preferred XML reader, or in Microsoft Internet Explorer.

The configuration is now completed on the ServiceNow side. Data exported from ServiceNow will now be delivered to your chosen MID server. From here it must be collected by two business adapters running on a

FlexNet Beacon for import into FlexNet Manager Suite. For configuration of the business adapters, see *Configure Imports from ServiceNow* on page 179.

Modify Data Mappings from FlexNet Manager Suite to ServiceNow

Transforms within ServiceNow are responsible for mappings between database columns.

Transform maps work on the imported data waiting in the staging tables in ServiceNow and map the columns into database columns in the main ServiceNow database. Most of the time, the default mappings (loaded when you imported the XML Update Set to ServiceNow) are appropriate; but if you have customized your implementation of ServiceNow, you may need to make matching changes to the appropriate transforms.

Tip • *For the complete set of default transforms supplied for integration, see Transform Maps for ServiceNow Integration on page 191.*

1. In ServiceNow, in the **Flexera FlexNet Manager Suite integration** group, select **Transform Maps**.
   A list of available **Table Transform Maps** is displayed. The following transforms (listed alphabetically) are included in the integration package:

   - `FNMP - Application -> Installed Application Transform` updates the actual installation records of particular software against individual computers.
   - `FNMP - Application -> Software Installation Transform` updates the records of software that can be installed.
   - `FNMP - Application -> Software Model Transform` sets up the preliminary model information for applications newly-arrived from FlexNet Manager Suite.
   - `FNMP - Contracts -> Contracts Transform` provides field mappings between contract records in the two databases.
   - `FNMP - Contracts -> Contract Instance Transform` refreshes the contract records in ServiceNow.
   - `FNMP - Inventory -> Computer Model Transform` sets up new computer models from the current import.
   - `FNMP - Inventory -> Computer Transform` takes data from the hardware inventory and asset records from FlexNet Manager Suite and updates the corresponding computer records in ServiceNow.
2. Click the hyperlinked name of your chosen transform map from the list.
   The properties of the **Table Transform Map** are displayed.
3. In the top section, ensure that the source (staging) table and the target (destination) table are correctly identified.
4. In the **Field Maps** tab (low on the page), ensure that the **Source field** and **Target field** values are correct in each case, and validate the **Coalesce** setting.

Coalesce is set to `true` on the key field (or fields) used for record matching between the two databases. If the field(s) with **Coalesce** `true` have the same value(s) incoming as already in the ServiceNow database, the record is updated. If the corresponding value in the incoming dataset is unique, a new record is created.

Run a test import and inspect the destination tables to validate that your modified transforms are correct.

Configure Imports from ServiceNow

ServiceNow shares information only about contracts and hardware assets to help synchronize data views between the products.

ServiceNow runs two export tasks to queue data about hardware assets and contracts for collection by the MID server of your choice. Once these have been collected by the MID server, two business adapters running on your inventory beacon collect this data and upload it to your central operations database. You must configure these business adapters before they can operate. (This process applies to the FlexNet Beacon since version 2014.)

Tip • *It is possible that the FlexNet Beacon software and the ServiceNow MID server are installed on the same physical machine. If they are on separate machines, network access allowing a file share is required.*

1. In the unzipped archive you downloaded for the ServiceNow integration package, open the `Importer` folder and locate the two business adapters `ServiceNowAssets.xml` and `ServiceNowContracts.xml`.
2. Copy these two files to an inventory beacon that has network access to your MID server, and save them into the business adapters folder, `%`*`CommonAppData`*`%\Flexera Software\Beacon\BusinessAdapter`.

   Note • *Version 2.0 of the ServiceNow integration package contains modified source to target column mappings for* `ServiceNowContracts.xml`*. Before replacing the existing file with the new version, you may wish to review the impact of this change on existing data. For more information about changed column mappings, see Imports from ServiceNow to FlexNet Manager Suite on page 195. Reapply any ServiceNow customizations after you copy these files.*

3. In the interface for the inventory beacon, select the **Business Importer** page from the **Connections** group.
   The two adapters appear in the list of available business adapters.
4. For each of these adapters in turn, select the adapter from the list, and click **Edit...**.
   The Business Adapter Studio opens, editing this adapter.
5. In the Business Adapter Studio, select the top level import node (like **ServiceNowAssets** or **ServiceNowContracts**) and locate the **File name** field. If you have already exported data to MID server, you may browse the XML file.
   The following example from the assets adapter shows a path when the MID server and inventory beacon are installed on the same physical computer:

   ```
   C:\ServiceNow\MyMIDServerFolder\agent\u_fnmp_asset.xml
   ```

The following example from the contracts adapter shows a path when the MID server is separate, using UNC format:

```
\\MyMIDServer\ServiceNow\MyMIDServerFolder\agent\u_fnmp_contracts.xml
```

Tip • Be sure not to change the database view name while editing the MID server name in the script.

6. Save each modified adapter; and when both are completed, close Business Adapter Studio.
7. *Tip • If you have not already created the schedule(s) on which you want to run these adapters, switch to the **Scheduling** page of the inventory beacon interface, and create them now. You may run these two business adapters on the same schedule, as the beacon queues them and runs them one after the other.*

   Select each adapter in turn, and:

   a) Click **Schedule...**

   b) From the list of available schedules, select the one for each adapter.

   c) Click **OK** (and repeat for the other adapter).
8. In the **Business Importer** page of the inventory beacon, ensure that your two modified adapters are marked as **Enabled**.
9. At the bottom of the inventory beacon interface, click **Save** to store your changes.
10. Optionally, if you have test data waiting for a system check, you can select each adapter in turn, and click **Execute Now**.
11. You can check the status of this job from the **Data Inputs** or **System Tasks** page of FlexNet Manager Suite.

3

Operational Overview

Topics:

Standard operation is automatic.

The exchange of data between ServiceNow and FlexNet Manager Suite happens in a pair of independent processes driven by schedules (either Windows scheduled tasks or schedules built in to the products).

As these schedules by default run on a weekly cycle, you may expect the data to be synchronized after the weekly runs and up until either system is subsequently updated. If you require an intermediate synching after an important update to one system or the other, you can also trigger either process manually.

The process flows are described in the following topics. For on premises implementations of FlexNet Manager Suite, the command-line utility is also documented here, should you choose to run that directly.

Note • *The export from FlexNet Manager Suite to ServiceNow is delayed if any of the following processes are already running:*

- *A previous instance of the export to ServiceNow*
- *An inventory import*
- *A license reconciliation.*

Process for Exports from FlexNet Manager Suite to ServiceNow

Here is how the export of data to ServiceNow works.

Prerequisites

- You have licensed the option for ServiceNow integration from Flexera Software. As previously mentioned (see *Prerequisites* on page 166), this license authorizes these exports, as well as all other communications through the integration package.
- The executable `fnmp_servicenow_export.exe` is installed in *`installation-directory`*`\DotNet\bin\ServiceNowExport` (see *Installation of the Integration Package* on page 169).
- The configuration file `fnmp_servicenow_export.exe.config` in installed in the same directory, and customized to your requirements (for example, for logging).
- A Windows scheduled task `Export to ServiceNow` is configured to trigger the executable (by default, weekly at 3am on Sunday morning).
- The "emergency switch" has not been turned off to interrupt exports (this is a value in the `ComplianceSettings` database table, `EnableServiceNowExport`, available for on premises implementations.) Naturally, this defaults to on, allowing exports.

Process Summary

1. The scheduled task triggers the executable (`fnmp_servicenow_export.exe`).

   ***Tip •** From version 2014, you can also trigger an export through the web interface for FlexNet Manager Suite (see Configuring Integration through the Web Interface on page 170).*

2. The executable validates all the command-line parameters. If validation fails, it exits with code `4` on the command line, or shows an error in the web interface.
3. It checks for a valid license from Flexera Software for the ServiceNow integration option (in a multi-tenant environment, this must be recorded against the `tenantUId` supplied as a command-line parameter). If not, it exits with code `3` on the command line, or shows an error in the web interface.
4. If the command-line parameter `-connectionTest` is true, the executable passes the supplied credentials to ServiceNow. If the credentials are good, ServiceNow passes back a connection code, and the executable exits with code `0` on the command line. Failure will result in 2 being written to the command line. When `-connectionTest` is true, execution ends at this point.
5. For regular exports, the connection test is completed as described above, and then the `-changes` parameter is assessed. If this is missing, or is present and set to `true`, the only information is processed in the remaining steps is what has changed since the baseline date (by default, this is the date of the last export run by the batch scheduler on the central server; but you can override this with either the `-changesSince` parameter or

the `-changesPeriod` parameter). For more information about these parameters, see *Command-line Tool for Export to ServiceNow* on page 184.

6. The (normally differential) data export, transfer, and import processes are run for all data types (unless any of them have been specifically excluded) in the following order:

    a. Hardware inventory and assets

    b. Contracts

    c. Applications and installation details.

    Caution • *In ServiceNow, contracts and applications/installations both have dependencies on computers/assets. It is recommended that you do not exclude the hardware inventory and asset transfer, as this may result in unpredictable gaps in other records when the correct dependencies cannot be established.*

    To prevent timeout issues, each data set is split into segments for transfer to ServiceNow. Each segment is identified with a transaction id, transaction type, and the data in an XML file. ServiceNow returns a connection code for each segment of data. If the connection code shows a failure, the segment is retransmitted for a maximum number of tries (the default value of 10 can be changed in the configuration file). If the number of maximum tries is reached, the program exits (skipping any remaining exports) and returns `1` to the command line.

    While success continues, the executable waits for each stage to complete before commencing the next stage (although it does not wait for a result after the last segment of application data is uploaded).

7. In ServiceNow, the inbound data segments are written to staging tables based on the transaction type:

Transaction type	**Import Set Table**
`application_export`	**FNMP Application Import**
`contract_export`	**FNMP Contract Imports**
`inventory_export`	**FNMP Inventory Imports** (note that this is hardware inventory)
`connection_test`	Not applicable
`export_status`	Not applicable

    You can see the requested imports by expanding the **FlexNet Manager Suite** group in the ServiceNow navigation bar, and selecting **FNMP Requests**. The imported data is visible a little further down, in the **Import Set Tables** group.

8. When the data is present, one or more Transform Maps are executed to map the fields in the import set tables to fields in ServiceNow database tables. For instance, the **FNMP Inventory Imports** set has columns that are matched with the `Computers` table in ServiceNow. Contract data is first mapped into the `Contracts` table in ServiceNow, and a second Transform Map is run to provide links to relevant Configuration Items (Computers). (For a complete set of the transform mappings, see *Transform Maps for ServiceNow Integration* on page 191.)

Transaction type	Ultimate Data Table(s)
`application_export`	**Software Models**, **Discovery Models**, and **Software Installations** (all in the Software Asset Management group). The process updates the tables in the order above, as later tables reference the previous one.
`contract_export`	**Contracts** (Navigate to **Contracts Management** > **Contracts** > **All**) ***Tip •*** *Check the* ***Contract used by*** *field near the bottom of the page of contract properties. This references all computers linked to the contract.*
`inventory_export`	**Computers** (Navigate to **Computers Table.Configuration** > **Base Items** > **Computers**)

9. When all the data has been transformed into the ServiceNow database, a check of the system properties `com.fnmp.execute.deleteaction.installation` and `com.fnmp.execute.deleteaction.inventory` is made. When true (expected for normal operations, but requiring your configuration as described in *Configuring ServiceNow for Integration* on page 174), all the affected tables in ServiceNow are checked for items on which the data flag `u_fnmp_isdeleted` has been raised by FlexNet Manager Suite. These items are now deleted to bring the datasets into alignment.

At the completion of this process, data exported from FlexNet Manager Suite is reflected in your ServiceNow data set.

Command-line Tool for Export to ServiceNow

This tool extracts data from the FlexNet Manager Suite operations database, and imports it into ServiceNow.

Note • *It is not normally required to execute this tool manually. It is normally run on a schedule set in the web interface for FlexNet Manager Suite.*

```
fnmp_servicenow_export.exe
                -tenantUId tenant-identifier
                -token ServiceNow-token
                -user ServiceNow-account-name
                -url ServiceNow-URL
                -endpoint ServiceNow-endpoint
                -changes Boolean
                -changesPeriod –numberOfDays
                -changesSince YYYY-MM-DD
                -connectionTest true
                -withApplications Boolean
                -withContracts Boolean
                -WithHardwareInventory Boolean
```

Option	Sample value	Notes
-tenantUId	`0000000000000000`	Mandatory for cloud implementations.
-token	32 random alpha-numeric characters	Mandatory. This can be copied from the ServiceNow web interface (navigate to **FlexNet Manager Suite Integration** > **Credentials**, click on the appropriate credentials **Number** to display details, and in the top right, click **View Token**; and you can copy from the pop-up dialog).
-user	`account@mydomain.com`	Mandatory. This is the user name specified in the appropriate credentials set within ServiceNow, the same credentials from which the token is collected.
-url	`https://`*`myserver`*`.service-now.com`	Mandatory. The URL to your ServiceNow server instance in the cloud. Use the HTTPS protocol.
-endpoint	`fnmp.do`	Mandatory. Leave the value unchanged. This is the integration point for ServiceNow. ***Tip** • While the endpoint is visible in ServiceNow and could be modified there, it is also recorded in the operations database for FlexNet Manager Suite. Changing it in ServiceNow without a matching change in the operations database will cause automated exports to fail.*
-changes	`true`	Optional, with a default of `true` if it is not present. When this option is not present, or when it is present with a value of `true` (case insensitive), only differential changes are included in all data exports to ServiceNow. For the baseline from which differences are measured, see the -changesSince or -changesPeriod options. When this option is present and has a value `false`, all data (from the relevant period) is exported from the relevant tables in FlexNet Manager Suite (which is likely to make the data transforms in ServiceNow considerably more time-consuming). ***Tip** • You may use the default value even for an initial data transfer, or for the first transfer after an upgrade from an earlier version of the executable. Initial transfers are always the full data set, with differential transfers following thereafter.*
-changesPeriod	`-5`	Optional (no effect when not present). When present, specifies the time window (as a number of days) before the export for which changes are included in the export. The negative sign is

Option	Sample value	Notes
		mandatory. The example means that changes in the previous five days are exported. Days are measured as complete 24-hour blocks prior to the time of the export.
-changesSince	`2015-02-28`	Optional. Use only when necessary to override the default date, which is the last run of the batch scheduler on the central batch server for FlexNet Manager Suite (or, for smaller on premises implementations, the server hosting this functionality). The format is `YYYY-MM-DD`.
-connectionTest	`true`	Optional. If this option is provided, and has a value `true` (case insensitive), no data exports are attempted. The executable attempts a connection to ServiceNow, and reports the results (`0` or `2` as described below) on the console. In order to transfer data from FlexNet Manager Suite to ServiceNow, this parameter must either be omitted, or set to `false`.
-withApplications	`false`	Optional. When omitted, it is assumed to be `true`, and applications are included in the export. If set to `false`, applications are excluded from the export. Note that at least one of these `-with*` parameters must be true.
-withContracts	`false`	Optional. When omitted, it is assumed to be `true`, and contracts are included in the export. If set to `false`, contracts are excluded from the export. Note that at least one of these `-with*` parameters must be true.
-WithHardware Inventory	`false`	Optional. When omitted, it is assumed to be `true`, and hardware devices are included in the export. If set to `false`, hardware devices are excluded from the export. Note that at least one of these `-with*` parameters must be true. ***Caution •*** *In ServiceNow, contracts and applications/installations both have dependencies on computers/assets. It is recommended that you do not exclude the hardware inventory and asset transfer, as this may result in unpredictable gaps in other records when the correct dependencies cannot be established.*

Result Codes

`0`	Success

1	A data upload to ServiceNow has failed through the maximum number of retries set in the configuration file.
2	Failed to connect to ServiceNow. Check the URL and account details for connection.
3	There is no valid license for execution of the export utility recorded for the current `tenantUId`.
4	Incomplete set of command-line parameters provided (see syntax listing above). Help information is displayed in this case.
5	An invalid `tenantUId` was provided (and in the configuration file, the `requireTenantUId` parameter is set to `true`).
6	Export from FlexNet Manager Suite cannot proceed because ServiceNow has not completed previous pending imports.

Examples

This example runs a connection test (because of the final parameter). Although the defaults are set for all data transfers (applications, contracts, and hardware inventory), no exports occur. The command-line result is `0` for good credentials, and `2` for bad credentials.

```
fnmp_servicenow_export.exe
       -token DbBNKvekyBVGFxPXe7TAGooeyb5bsByB
       -user myname@mycorp.com
       -url https://myserver.service-now.com
       -endpoint fnmp.do
       -connectionTest true
```

The following example restricts the export to collect only hardware inventory. Because it is not mentioned, the default true value includes hardware inventory, with the other two possibilities excluded. Because the `-changes` and `-changesSince` options are both missing, only hardware changes occurring since the last scheduled export are included.

```
fnmp_servicenow_export.exe
       -token DbBNKvekyBVGFxPXe7TAGooeyb5bsByB
       -user myname@mycorp.com
       -url https://myserver.service-now.com
       -endpoint fnmp.do
       -withContracts false
       -withApplications false
```

This example covers all export types from FlexNet Manager Suite, but including only changes that occurred in the previous 3 days, measured back in 24-hour blocks from the time the export is run. Assuming this example is run at 1pm Thursday, it will export only changes from 1pm Monday to 1pm Thursday (all times are local on the batch server). Strictly, `-changes true` is redundant, since this is the default.

```
fnmp_servicenow_export.exe
       -token DbBNKvekyBVGFxPXe7TAGooeyb5bsByB
       -user myname@mycorp.com
       -url https://myserver.service-now.com
       -endpoint fnmp.do
       -changes true
       -changesPeriod -3
```

Process for Exports from ServiceNow to FlexNet Manager Suite

Here is how the import of data from ServiceNow works.

Prerequisites

- You have licensed the option for ServiceNow integration from Flexera Software. As previously mentioned (see *Prerequisites* on page 166), this license authorizes all communications through the integration package.
- You have loaded the XML customization into your ServiceNow instance (see *Configuring ServiceNow for Integration* on page 174).
- Your MID server is accessible across the network from your inventory beacon, and the inventory beacon is configured with appropriate credentials for read access the MID server.
- The two supplied business adapters for reading asset data and contract data from ServiceNow are configured on your inventory beacon.

Process Summary

1. Two scheduled reports are run independently in ServiceNow.

   Note • *These reports are visible in **System Definition** > **Scheduled Jobs** as **Export Assets from ServiceNow** and **Export Contracts from ServiceNow**. Click the report names to make them active or inactive, to modify the schedule, or to execute either one immediately.*

2. When either report is triggered, the following steps occur in the installation folder of your MID server:

   a. Any previous XML file is overwritten on your MID server, replaced with the opening tags for the new XML file.

   b. Data is transferred to the MID server in segments (to prevent time-out issues with large files), and gradually assembled into the XML file.

   c. The closing tags for the XML file are written.

3. Two independent business adapters from FlexNet Manager Suite on independent schedules connect from your inventory beacon to your MID server, and collect the latest XML files saved there, converting them into the intermediate data form required for business data uploads.

   Important • *The business adapters will fail if they are run while the XML files on the MID server are incomplete. Be sure that the schedules for export from ServiceNow, and the running of the business adapters, allow sufficient time for the export to be completed. For large datasets, this may be several hours.*

4. As the adapters complete their run, the dataset is uploaded immediately to the application server for FlexNet Manager Suite. (There is also a catch-up upload task that runs overnight to retry any failed uploads.)
5. On a separate schedule, the batch server starts a business import job. This imports data from the data package into the FlexNet Manager Suite database.

At the completion of this process, data exported from ServiceNow is reflected in your FlexNet Manager Suite data set.

4

Appendices

Topics:

- *Transform Maps for ServiceNow Integration*
- *Imports from ServiceNow to FlexNet Manager Suite*
- *Additional ServiceNow Indexes for Performance*

The following topics provide details on data transfers between FlexNet Manager Suite and ServiceNow (either direction), and list some enhancements you can request for indexing on your ServiceNow database that greatly improve performance of the integration between the products.

Transform Maps for ServiceNow Integration

These transform maps map hardware, license, and contract data collected from FlexNet Manager Suite into ServiceNow.

Data collected from FlexNet Manager Suite is initially held in staging tables within ServiceNow, and must be transformed for insertion in the operational tables in your ServiceNow implementation. The following tables show the standard transformations from the staging tables (the source) to operational tables (the target) within ServiceNow. Items marked [Script] involve a data transformation, which may involve finding the foreign key to another record, or conversion of units such as bytes to MB.

Note • *When true, the* `Coalesce` *field identifies key fields for record matching. If a record with identical values for fields so marked already exists, it is updated (and if not, a new record is created).*

Contracts Transform

- Original data from FlexNet Manager Suite: Contracts.
- Staging table in ServiceNow: ast_contract

Source Display Name	Source Field	Target Display Name	Target Field	Coalesce
ContractNumber	u_contractnumber	Contract number	vendor_contract	TRUE
EndDate	u_enddate	Ends	ends	FALSE
ContractName	u_contractname	Description	description	FALSE
Vendor	u_vendor	Vendor	vendor	FALSE
ContractType	u_contracttype	Short Description	short_description	FALSE
IsDeleted	u_isdeleted	Is deleted	u_fnmp_isdeleted	FALSE
StartDate	u_startdate	Starts	starts	FALSE
ContractStatus	u_contractstatus	State	state	FALSE

Contract Instance Transform

- Original data from FlexNet Manager Suite: Contracts.
- Staging table in ServiceNow: ast_contract_instance

Source Display Name	Source Field	Target Display Name	Target Field	Coalesce
ContractNumber	u_contractnumber	Contract	ast_contract -> vendor_contract	TRUE

Source Display Name	Source Field	Target Display Name	Target Field	Coalesce
See note.	See note.	Configuration Item	ci_item	TRUE
ContractType	u_contracttype	Contract Type	contract_type	FALSE

***Note •** Script is run to query `cmdb_ci_computer` table. If the CI is found, the script returns the `sys_id` for REF. If the CI is not found, this field is left blank.*

Computer Model Transform

- Original data from FlexNet Manager Suite: Inventory.
- Staging table in ServiceNow: cmdb_model

Source Display Name	Source Field	Target Display Name	Target Field	Coalesce
ModelNo	u_modelno	Model number	model_number	TRUE
Manufacturer	u_manufacturer	Manufacturer	manufacturer	FALSE
ModelNo	u_modelno	Name	name	FALSE
ChassisType	u_chassistype	Type	type	FALSE
ComputerType	u_computertype	Model categories	cmdb_model_category	FALSE

Computer Transform

- Original data from FlexNet Manager Suite: Inventory.
- Staging table in ServiceNow: cmdb_ci_computer

Source Display Name	Source Field	Target Display Name	Target Field	Coalesce
ComputerID	u_computerid	FNMP Computer ID	u_fnmp_computer_id	FALSE
MaxClockSpeed	u_maxclockspeed	CPU speed (MHz)	cpu_speed	FALSE
InventoryConnection Name	u_inventoryconnection name	Inventory Connection	u_fnmp_inventory_ connection	FALSE
CalculatedUser	u_calculateduser	Calculated User	u_fnmp_calculated_user	FALSE
ChassisType	u_chassistype	Chassis type	chassis_type	FALSE
ModelNo	u_modelno	Model ID -> model_number	model_id	FALSE

Source Display Name	Source Field	Target Display Name	Target Field	Coalesce
[Script]	[Script]	RAM (MB)	ram	FALSE
[Script]	[Script]	Correlation ID	correlation_id	FALSE
IPAddress	u_ipaddress	IP Address	ip_address	FALSE
OperatingSystem	u_operatingsystem	Operating System	os	FALSE
LastLoggedInUser	u_lastloggedinuser	Last logged in user	u_fnmp_last_logged_in_user	FALSE
InventorySource	u_inventorysource	Inventory Source	u_fnmp_inventory_source	FALSE
Manufacturer	u_manufacturer	Manufacturer	manufacturer	FALSE
NumberOfProcessors	u_numberofprocessors	CPU count	cpu_count	FALSE
ComputerName	u_computername	Name	name	FALSE
SerialNo	u_serialno	Serial number	serial_number	TRUE
DiscoveredDate	u_discovereddate	Discovered date	u_fnmp_discovered_date	FALSE
NumberOfThreads	u_numberofthreads	CPU core thread	cpu_core_thread	FALSE
ProcessorType	u_processortype	CPU type	cpu_type	FALSE
AssetID	u_assetid	FNMP Asset ID	u_fnmp_asset_id	FALSE
ComputerType	u_computertype	Subcategory	subcategory	FALSE
IsDeleted	u_isdeleted	Is deleted	u_fnmp_isdeleted	FALSE
Domain	u_domain	Domain	sys_domain	FALSE
[Script]	[Script]	Disk space (GB)	disk_space	FALSE
ModelNo	u_modelno	Model number	model_number	FALSE
NumberOfCores	u_numberofcores	CPU core count	cpu_core_count	FALSE
ComputerStatus	u_computerstatus	Status (hardware_status)	hardware_status	FALSE
MACAddress	u_macaddress	MAC Address	mac_address	FALSE

Software Model Transform

- Original data from FlexNet Manager Suite: Application.
- Staging table in ServiceNow: cmdb_software_product_model

Source Display Name	Source Field	Target Display Name	Target Field	Coalesce
FlexeraID	u_flexeraid	FlexNet Manager Id	u_flexnet_manager_id	TRUE
ApplicationVersion	u_applicationversion	Version	version	FALSE
Publisher	u_publisher	Manufacturer	manufacturer	FALSE
Classification	u_classification	Type	type	FALSE
ApplicationName	u_applicationname	Flexera Application Name	u_flexera_application_name	FALSE
FlexeraID	u_flexeraid	Short description	short_description	FALSE
ProductName	u_productname	Name	name	FALSE

Software Installation Transform

- Original data from FlexNet Manager Suite: Application.
- Staging table in ServiceNow: cmdb_sam_sw_install

Source Display Name	Source Field	Target Display Name	Target Field	Coalesce
DiscoveredAt	u_discoveredat	Last Discovered	u_fnmp_last_discovered	FALSE
DisplayName	u_displayname	Display name	display_name	FALSE
ApplicationVersion	u_applicationversion	Version	version	FALSE
ApplicationID	u_applicationid	Application ID	u_fnmp_application_id	TRUE
LastScanned	u_lastscanned	Last scanned	last_scanned	FALSE
FlexeraID	u_flexeraid	Discovery model -> prod_id	discovery_model	FALSE
[Script]	[Script]	Is deleted	u_fnmp_isdeleted	FALSE
FlexeraID	u_flexeraid	Prod id	prod_id	FALSE
[Script]	[Script]	Installed on	installed_on	TRUE
DiscoveredBy	u_discoveredby	Discovered by	u_fnmp_discovered_by	FALSE
Publisher	u_publisher	Publisher	publisher	FALSE

Imports from ServiceNow to FlexNet Manager Suite

Data for hardware assets and contracts is transferred from ServiceNow to FlexNet Manager Suite.

For hardware assets and contracts, many companies prefer to regard ServiceNow as the 'source of truth', the single point of maintenance. For that reason, these records are imported from ServiceNow to FlexNet Manager Suite. On a schedule defined in ServiceNow, the data is first reported in ServiceNow, and then exported as a pair of XML files (one for each object type) in a predefined format. These files are collected by, and saved on, your MID server. On a separate schedule (configured on your inventory beacon), the inventory beacon collects and uploads these files to the central application server for import into the compliance database.

The **Use this property for matching existing data** check box is checked for the field which is used for matching existing records. If the value in this field already exists in the compliance database, the record is updated with the other values imported for this item. Otherwise, a new record is automatically created in FlexNet Manager Suite.

The following tables detail the source fields from ServiceNow, and the target (destination) fields in FlexNet Manager Suite.

Assets Imports

- Original data source: ServiceNowAssets
- Destination table in FlexNet Manager Suite database: `Asset`

Source Display Name	Source Field	Target Display Name	Target Field	Match on
Configuration Item	`alm_hardware_ci`	**Short Description**	`ShortDescription`	`FALSE`
Serial number	`ci_serial_number`	**Serial Number**	`SerialNumber`	`TRUE`
Asset tag	`alm_hardware_asset_tag`	**Asset Tag**	`AssetTag`	`FALSE`
State	`alm_hardware_install_status`	**Asset Status**	`AssetStatusID` [Note 1]	`FALSE`
Manufacturer	`ci_manufacturer`	**Manufacturer**	`Manufacturer`	`FALSE`
Model number	`ci_model_number`	**Model No**	`ModelNo`	`FALSE`
Installed	`ci_install_date`	**Installation Date**	`InstallationDate`	`FALSE`

Notes:

1. `AssetStatusID` is a foreign key to the `AssetStatus` table, from which display values are drawn.

Contracts Imports

- Original data source: ServiceNowContracts
- Destination table in FlexNet Manager Suite database: `Contract`

Source Display Name	Source Field	Target Display Name	Target Field	Match on
Contract number	contract_vendor_contract	**Contract Number**	ContractNo	TRUE
Short description	contract_short_description	**Contract Name**	ContractName	FALSE
Display name	pm_display_name	**Contract Type**	ContractTypeID [Note 1]	FALSE
State	contract_state	**Contract Status**	ContractStatusID [Note 2]	FALSE
Starts	contract_starts	**Start Date**	StartDate	FALSE
Ends	contract_ends	**End Date**	EndDate	FALSE
Description	contract_description	**Comments**	Comments	FALSE

Notes:

1. ContractTypeID is a foreign key to the ContractType table, from which display values are drawn.
2. ContractStatusID is a foreign key to the ContractStatus table, from which display values are drawn.

Additional ServiceNow Indexes for Performance

Creating additional indexes on your ServiceNow instance substantially improves the intake of data exported from FlexNet Manager Suite.

The *Fuji* release of ServiceNow enables you to create your own indexes. If you are using an older version, you may need to contact ServiceNow for a support task to add a database index on each of the following columns. A worked example suggests that, with the following indexes added, you can expect better than four-fold performance increase in the initial data transforms to finish the import of data that has been exported from FlexNet Manager Suite.

Database table	New column indexed
Contract [ast_contract]	vendor_contract
Software Model [cmdb_software_product_model]	u_flexnet_manager_id
Product Model [cmdb_model]	model_number
Computer [cmdb_ci_computer]	u_fnmp_computer_id
Software Installation [cmdb_sam_sw_install]	u_fnmp_application_id

Part VII

XenApp Server Adapter

Topics:

The Flexera Software XenApp server adapter allows you to collect software inventory from Citrix XenApp and import it into FlexNet Manager Suite. Depending on the type of virtualized applications being served, the evidence appears in either the installer evidence list (for App-V or streaming profile applications delivered through XenApp), or in the file evidence list (for file-based applications managed through XenApp).

In either case, the evidence must be linked to an application record. This can happen in either of two ways:

- Where the evidence is matched by an existing inventory rule for an application, it is automatically linked to that application record.
- Where required details (below) are incomplete, or there is no existing exact match in any existing evidence rules/records (either supplied by the Application Recognition Library or created locally in your enterprise), the evidence is left in the **Discovered Evidence** (and **All Evidence**) page with its **Assigned** property set to `No`. The fields that require matching are:
  - For installer evidence, the application name, version and publisher
  - For file evidence, the file name, version, company, and description.

Once the evidence is linked to an application, the application must be linked to a license. The license should then be linked to purchase records to determine your entitlements. Of course, these are manual tasks outside the scope of the adapter's operations.

The term *XenApp Server* is used in this documentation as a generic term to cover the differently-named control servers for different versions of XenApp:

- In version 6.x, the XenApp Server was officially named the Zone and Data Collector. One such controlling server was required per *farm*.
- In version 7.5 and later, the XenApp Server is called the Delivery Controller. One such controlling server is required per *delivery site*.

Supported versions

The adapter links the current version of FlexNet Manager Suite to one of the following versions of XenApp:

- Version 6.0
- Version 6.5
- Version 7.5
- Version 7.6.

If it happens that you have multiple of these versions of XenApp in operation (for example in different domains), you can use the same structure described in this section to link them all to FlexNet Manager Suite.

1

Architecture, Operations and Prerequisites

Topics:

This chapter provides a useful framework for your understanding of the more detailed content to follow.

Architecture and Operation

In order to track licenses for applications delivered remotely to users from a Citrix XenApp Server, FlexNet Manager Suite needs information about which users and devices have access to which applications. There are several sources of such data available from XenApp Servers, depending on the version of XenApp:

- For XenApp 6.0 and 6.5:
  - Access control lists (ACLs)
  - Streaming profiles
  - Citrix EdgeSight servers
- For XenApp 7.5 and later:
  - Access control lists (ACLs)
  - App-V 5 packages (and the applications they contain)
  - For XenApp 7.6 and later, the XenDesktop database supplied as part of XenApp that tracks application usage.

    Tip • *No usage tracking is possible for XenApp 7.5, as in this release Citrix did not include usage tracking capabilities in XenApp.*

Each of these sources is discussed in turn in the following sections.

Access control lists (ACLs)

These are lists which specify the permissions associated with an object, such as an application's executable file, on a server. The FlexNet Manager Agent for XenApp Server (XenApp Server agent) is a tool that extracts information about users and which applications they can access remotely, and transfers that information to an inventory beacon for use in licensing calculations in FlexNet Manager Suite. To do this, the XenApp Server agent must be installed:

- For XenApp 7.5 and later, on one Delivery Controller for each Delivery Site. If you have multiple Delivery Sites, you may choose either of the following:
  - Install the FlexNet XenApp Server agent on one Delivery Controller in each Delivery Site
  - Use only a single FlexNet XenApp Server agent, and provide that agent with the required network access and credentials to access all required XenApp Delivery Controllers.
- For XenApp 6.0 or 6.5, on each controlling XenApp Server (not the worker servers delivering applications) in a Citrix farm.

The XenApp Server agent is supplied as an integral part of the XenApp Server adapter.

Tip • In previous releases, the XenApp Server agent extracted Active Directory names and details of users and devices from the XenApp Server. Now, the XenApp Server agent collects only Active Directory SIDs (a large performance improvement). As a result, best practice is that your inventory beacon completes its import of Active Directory data before importing XenApp data, so that all Active Directory SIDs can be resolved against the user names, devices, and groups collected directly from Active Directory.

Streaming profiles (for XenApp version 6.0 and 6.5)

The XenApp Server agent is also able to read the contents of the `.profile` and the key executable files associated with streamed applications published to your XenApp Servers.

Tip • As the streaming profile is not stored in the XenApp Server's database, the XenApp Server agent must have at least read access to the streaming profile location to be able to read and extract this information.

As these applications are not physically installed on your XenApp Server, combining the XenApp Server agent's data from `.profile` files with EdgeSite server information may be the only way for FlexNet Manager Suite to recognise usage of such applications.

Note • FlexNet Manager Suite is only able to recognize usage of files streamed to a XenApp Server, not those streamed directly to client devices.

Citrix EdgeSight servers (for XenApp 6.0 and 6.5)

Citrix EdgeSight for XenApp monitors and profiles the usage of remote and streamed applications by users, telling you both who is using that application, and on what device. The data from EdgeSight is very valuable for FlexNet Manager Suite: you may use it for license optimization (for example, tightening access through ACL permissions to exclude users who evidently do not need to use the applications); or it may be critical for any user-based or usage-based licensing of applications delivered through XenApp 6.0 or 6.5.

EdgeSight agents may be installed on each XenApp Server, and report back to a central EdgeSight server, which can keep track of application usage on multiple XenApp machines, belonging to one or more farms. The FlexNet Beacon can connect to each EdgeSight server and collect this usage information for use in compliance calculations.

Unlike data from the XenApp Server agent, EdgeSight data does contain details of which devices access a particular application. Thus, EdgeSight data is usually more valuable to an enterprise for calculating license compliance than the data about application availability returned by the XenApp Server agent alone. If, however, your enterprise deploys streamed applications, EdgeSight usage data may need to be supplemented by XenApp Server agent information to accurately recognize these applications.

The following information is returned from the EdgeSight server:

- A list of applications and the users who use them
- The devices on which users request and run applications
- The XenApp (worker) servers from which users request applications

- The farms to which the XenApp worker servers belong.

To use EdgeSight data, you must create a database connection to the EdgeSight SQL server database.

App-V 5 packages (for XenApp 7.5 or later)

The XenApp Server agent is able to inspect the contents of App-V 5 packages and recover the name, version, and publisher of the application contained in each package. The agent also returns the user's ability to *access* these App-V packages (as recorded in the ACLs described earlier). However, in the ability to track which users actually *use* the applications, there are differences across versions:

- Version 7.5 has no technology like the EdgeSight server available in version 6.x, and so cannot report application usage
- From version 7.6, XenApp again allows tracking application usage through connection to the XenDesktop database incorporated in XenApp 7.6 and later.

VDI images (for XenApp 7.5 or later)

The same capabilities apply to VDI images. The XenApp Server agent interrogates any VDI device managed by the XenApp Server to read the applications listed in all VDI master images available (including spinning up any images that are currently dormant to inspect their applications). As with App-V packages:

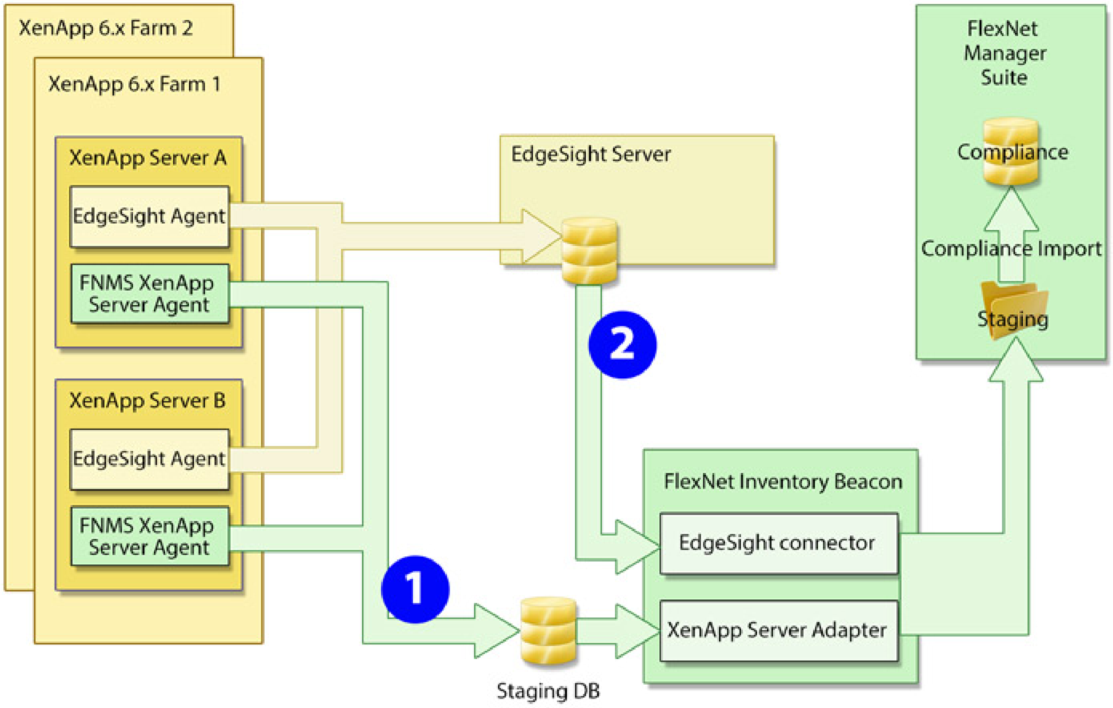
- For XenApp version 7.5, there is no ability to track who uses any VDI image, or when
- For XenApp 7.6 and later, the included XenDesktop database allows collection of application usage information.

Changed architecture across versions

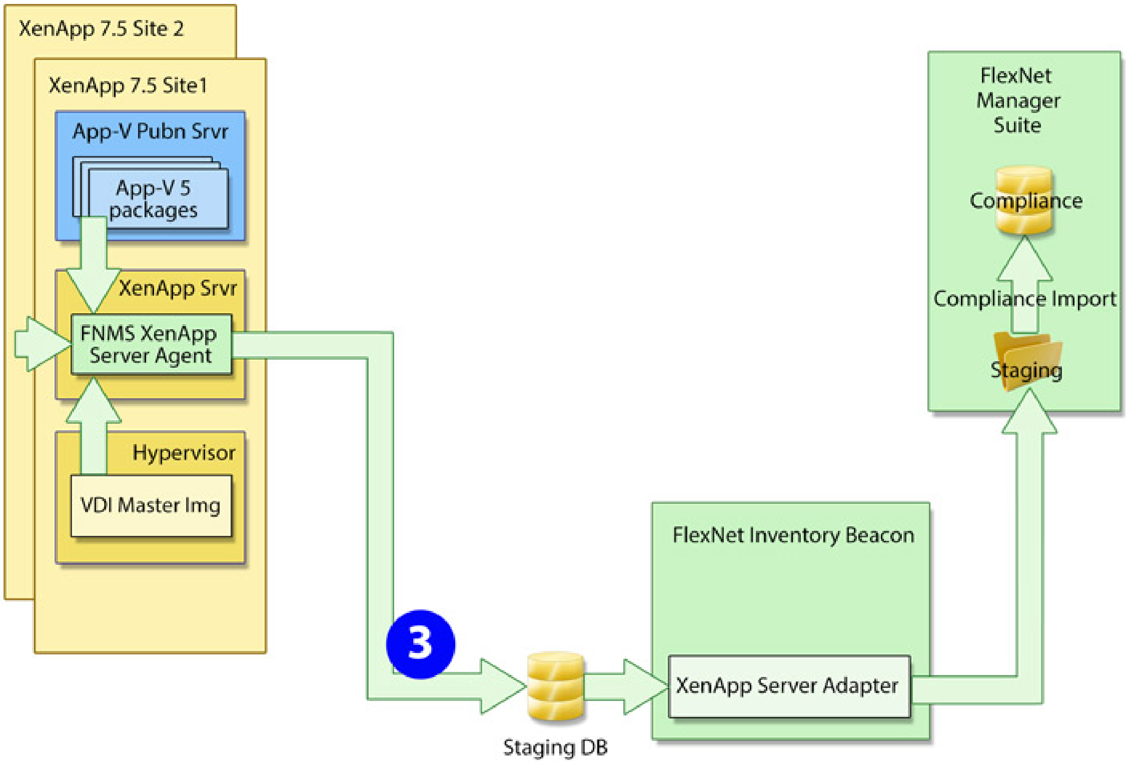
Because XenApp release 7.5 follows an extensive rewrite of the XenApp line by Citrix, the architectures of the two systems (and therefore the ways that the adapter integrates with the architecture) are quite different from version 6.x to 7.x.

***Tip •** The following three diagrams do not include the import of Active Directory data by the inventory beacon, as this is not part of the adapter itself. However, the prior import of Active Directory data (typically, by the same inventory beacon connecting to the staging database) is a prerequisite for operation of the adapter.*

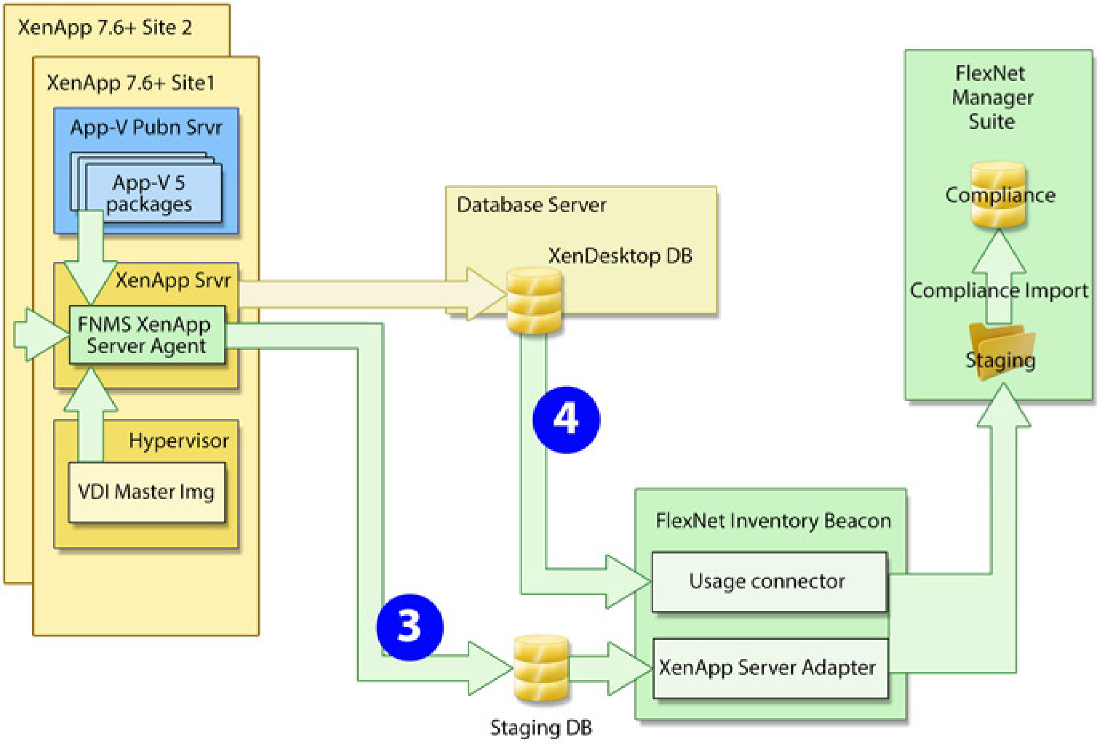
This diagram represents the architecture and data flows for the adapter connected to XenApp 6.0 or 6.5 (the key numbers are referenced in the table below):



The next diagram shows the architecture and data flows when connected to XenApp 7.5. The diagram is laid out similarly to highlight the changes in architecture, and in particular the absence of usage information:



Finally, the third diagram shows the architecture and data flows when connected to XenApp 7.6 or later. The main point to note is the return of usage information:



The following table shows the data collected by the adapter through the different channels numbered in the three diagrams.

Table 1: Lists imported by XenApp adapter

List	Case 1	Case 2	Case 3	Case 4
User SIDs, Active Directory Group SIDs	Y		Y	
File evidence (`.exe`) details – file name, version, company, description	Y	Y	Y	
Installer evidence (from App-V/streaming profiles) details – name, version, publisher	Y	Y	Y	Y
Application access rights per user SID	Y		Y	
Application usage per user		Y		Y*
Client computer SIDs (with user SIDs)		Y		Y
XenApp Servers with applications present for delivery	Y			
App-V packages (and the applications therein) managed by XenApp			Y	

List	Case 1	Case 2	Case 3	Case 4
XenApp Servers that served applications		Y		
Applications available in App-V packages (creates App-V 'evidence' records too)			Y	Y
Applications available as streaming profiles	Y			

* Application usage by users and devices for XenApp 7.6 is limited to applications delivered in App-V packages. (Usage based on imported file evidence is not collected.)

Operation with XenApp 7.5 and later

The XenApp Server agent is installed on the XenApp Server (at your discretion, on only one server that can access all other controllers for XenApp in your enterprise, or as many as one per XenApp site).

Triggered by a Windows scheduled task, the agent runs according to your settings. It may collect inventory details only from the XenApp Server on which it is installed (default), or it may collect from several controlling XenApp Servers in sequence (identified with the `-s` command line option, detailed in *XenApp Server Agent Command Line Options* on page 219).

a. VDI images

Based on information found on the XenApp Server, the agent may connect to any relevant XenApp servers hosting VDI images that contain applications. XenApp allows an administrator to nominate applications within VDI images for delivery

1. as individual applications only
2. within a VDI (delivered as a whole environment) only
3. either as individual applications or within a VDI.

The XenApp Server agent collect information on all applications within the VDI master images that are identified for individual delivery (options 1 or 3 above). The VDI evidence is returned to FlexNet Manager Suite as file evidence. (For XenApp version 7.6, usage tracking is not available for the file evidence.)

***Tip •** To track inventory delivered within an entire VDI environment (option 2 above), use XenDesktop discovery and inventory through the rules-based process in the web interface of FlexNet Manager Suite.*

b. App-V packages

Again based on information gathered from the XenApp Server, the agent connects to any Microsoft App-V publication server to inspect any App-V packages registered for delivery through XenApp. Because XenApp requires App-V version 5 (or later) for integration, the agent can interrogate the packages to identify the applications inside. The App-V evidence is returned to FlexNet Manager Suite as installer evidence.

For XenApp version 7.6 and later (but not for version 7.5), a connection is also made to the XenDesktop database from the appropriate inventory beacon. This collects details of App-V application packages that users and devices have accessed. This additional information restores the ability (missing for version 7.5) to determine which users and devices have actually used each application, as distinct from merely having access to them. This may allow for more accurate license consumption calculations in later compliance calculations for applications delivered in this way.

c. Processing

So both kinds of applications are returned to FlexNet Manager Suite as evidence:

- From VDI images, file evidence is produced that normally includes file name, version, company, and description
- For App-V packages, installer evidence is returned that normally includes application name, version, and publisher.

When the data is finally imported into FlexNet Manager Suite:

- The incoming evidence is tested against existing evidence records (including "rules" generalized with wild cards) already linked to applications (either from the Application Recognition Library or from records produced in your enterprise).
- If the incoming evidence matches any existing rule or record, it is recorded against the linked application, and its presence is recorded in the properties of the appropriate user or device as an "installation" record. For XenApp 7.6 (or later), a usage record is automatically created for each user and each device shown to have accessed the application.
- If the incoming evidence is not matched, it is displayed in evidence listings (for example, navigate to **License Compliance** > **Evidence** group > **Discovered Evidence**, selecting the **Installer evidence** tab for App-V applications and the **File evidence** tab for VDI applications). You can select the evidence in the appropriate tab, and click **Assign** to choose an application record (or create a new one) to link to the evidence. (You only need do this the first time that new evidence is reported. Once linked to the application, your evidence serves as a 'rule' for matching future imports of the same evidence.)
- Once the incoming evidence is linked to an application (either automatically or manually), license reconciliation attempts to calculate consumption through XenApp on any license linked to the application. For this to take effect, you must correctly configure at least one license attached to the application:

    1. Navigate to the **Use rights & rules** tab of the license properties.
    2. Ensure that the **License consumption rules** heading is expanded (if not, click the heading).
    3. Select **Access granted to users, or usage, consumes license entitlements** to expose additional controls.
    4. Depending on the terms of your license, choose one of **Consume one entitlement for each user** or **Consume one entitlement per device owned by each user**.
    5. For XenApp 7.5, set **Consume entitlements based on** to `Access` (because only access records are available through XenApp 7.5). For other versions tracking App-V applications, check the terms of your license to see whether usage-based licensing is acceptable for this application, and make selections accordingly.

Prerequisites

The XenApp server adapter requires the following:

- The executable and supporting files for the XenApp server agent. These are available as described in *Creating the Staging Database* on page 210.
- The XenApp Server (on which the XenApp server agent is installed) requires:
    - .NET version 4.5 or greater
    - PowerShell 2.0 or greater.
- A staging database that can run in a convenient Microsoft SQL Server instance. For example, this may be a database running on the inventory beacon, or on the XenApp Server hosting the XenApp server agent, or on your central operations databases for FlexNet Manager Suite. For more about the requirements for this database, see *Creating the Staging Database* on page 210.
- An inventory beacon (or multiple if required) that collects Active Directory data for the domain(s) where your XenApp Server(s) are located.
- An inventory beacon (possibly the same as in the previous point) that can connect to your staging database and upload the inventory to the central FlexNet Manager Suite database.
- If you are using XenApp 6.x with the recommended EdgeSight server, an inventory beacon (almost invariably the same one as in the previous point) that can connect to the EdgeSight database.
- If you are using XenApp 7.6 or later, an inventory beacon (usually the same one) that can connect to your XenDesktop database, and uploaded the imported data to your central operations databases.

The adapter links the current version of FlexNet Manager Suite to one of the following versions of XenApp:

- Version 6.0
- Version 6.5
- Version 7.5
- Version 7.6.

2

Setting Up the XenApp Server Adapter

Topics:

The XenApp server adapter is available for different versions of XenApp:

- For version 6.0 and 6.5, it augments information available from EdgeSight, particularly about streamed applications
- For version 7.5, it is the primary means of gathering inventory information from XenApp
- For version 7.6 and later, it again augments information about application usage collected from the XenDesktop database included with XenApp.

The adapter is automatically saved as part of the standard installation of FlexNet Manager Suite. Installation consists of five main activities, all described in the following topics:

- Setting up the staging database for the inventory that is collected (see *Creating the Staging Database* on page 210)
- Copying the appropriate folder for the adapter to your chosen XenApp Server(s) (see *Installing the XenApp Server Agent* on page 211)
- Ensuring an appropriate account is available to run the adapter (also in *Installing the XenApp Server Agent* on page 211)
- Setting up a local scheduled task to run the agent as you require (see *Create a Scheduled Task* on page 212)
- Setting up a connection from an appropriate inventory beacon to the staging database (see *Create Connections for Data Upload* on page 214).

Creating the Staging Database

The staging database allows the XenApp server agent to drop collected data in a conveniently close location. From here, an inventory beacon collects the data for transfer to the central operations databases for FlexNet Manager Suite.

The staging database can be installed in any convenient Microsoft SQL Server 2008 (or later) database:

- If your inventory beacon is located on a SQL server, the staging database can be on the inventory beacon.
- If your central operations databases for FlexNet Manager Suite are accessible from the XenApp Server, the staging database tables are already present on your central SQL Server.
- Where there is an inventory beacon with network access to your XenApp Server (and XenApp is running its database in SQL Server), the staging database can be installed into the same database as used by XenApp.
- If none of these suit, any other SQL Server instance that allows network access both from the XenApp server agent on the XenApp Server and from the inventory beacon.

It is a small footprint database (half a dozen tables and one stored procedure) that is size-limited by the scale of your XenApp implementation, with data being replaced at each upload. Installation is by script, as follows:

1. Using Windows Explorer, locate the temporary location where you unzipped the downloaded product archive to install FlexNet Manager Suite.
2. In your unzipped archive, navigate into the `\Citrix XenApp Server Agent` subdirectory.
3. Further navigate into the appropriate sub-folder for your version of XenApp:
    - `XenAppAgent6`
    - `XenAppAgent65`
    - `XenAppAgent75`
    - `XenAppAgent76`.

    Note • *If it happens that you have multiple different versions of XenApp (for example, in different domains), you may use a single database for all versions of XenApp. The following script is identical in all these folders.*
4. From your chosen folder, collect a copy of the database creation/update script `SetupXenAppAgentStagingDatabase.sql`.
5. On your selected SQL Server, drop a copy of this file, and execute it in SQL Server Administration Studio against your chosen database instance.

    An appropriate SQL Server database instance:

    - Is accessible from the XenApp Server(s) running the XenApp server agent
    - Is accessible from the inventory beacon responsible for uploading collected inventory to FlexNet Manager Suite

- Grants read/write access to the account running the scheduled task for the XenApp server adapter (if you choose to use Windows Authentication — if not, take note of the account name and password for database access that you will include in the database connection string)
- Grants read access to the service account running the inventory beacon engine (if you choose to use Windows Authentication — if not, take note of the account name and password for database access that you will include in the database connection string)

The script generates the SQL schema for the database, including creating the appropriate stored procedure. Takes notes for the connection string needed to connect to this database.

Installing the XenApp Server Agent

This procedure assumes you have already located the installation folder where the XenApp server adapter awaits.

1. In your unzipped archive, navigate into the `\Citrix XenApp Server Agent` subdirectory.
2. Copy the appropriate sub-folder to match your installed version of XenApp:
   - `XenAppAgent6`
   - `XenAppAgent65`
   - `XenAppAgent75`
   - `XenAppAgent76`.

   Tip • *If you have different versions of XenApp deployed in different domains, install the appropriate agents on the correct XenApp Servers. Agents for different versions may connect to a single staging database.*

3. Using your network, a memory stick, or other available means, paste the entire folder into an appropriate location on your XenApp Server(s).

   For example, you could paste the folder at the root level (such as `C:\XenAppAgent75`). Keep in mind that for version 6 and 6.5, you must install the agent on every XenApp Server. For version 7.5 (or later), you may choose to install an agent on one XenApp Server in each site, or to install on only one XenApp Server that has network access (and credentials) for all your sites.
4. Ensure that, on your XenApp Server (Delivery Controller), there is an account with sufficient privileges to run the agent in production.

   Such an account:
   - Can run a Windows scheduled task on the XenApp Server where the agent is installed
   - Has read access to the file system on the XenApp Server(s) where it is to collected inventory
   - Has read access to any file shares used to house XenApp packages (versions 6 and 6.5 only), App-V packages, or applications hosted in VDI images
   - For versions 6 and 6.5, is a Citrix Admin account (a limitation in the PowerShell API for those versions means that only a Citrix Admin can retrieve the list of XenApp Servers in a farm)

- For version 7.5 (or later), is a Citrix Read only admin account, with read access to the file systems of any other XenApp Servers sharing locally published applications for which inventory is to be collected
- Is recognized by the SQL Server hosting the shared database (if you prefer to use Windows Authentication for access to the staging database; otherwise, you may choose to include an account name and password in the connection string for the staging database).

Create a Scheduled Task

The XenApp server agent must be run locally on the XenApp Server, where it collects inventory and transfers the data immediately to the staging database. This is triggered by a Windows scheduled task on the XenApp server.

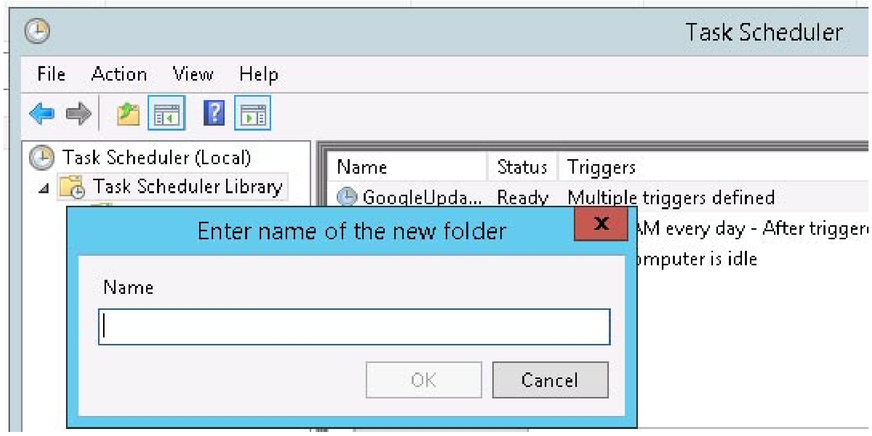
Because the XenApp server agent, as its first action for each inventory collection, clears all old data from the staging database, it is important that the XenApp server agent does not run at the same time as the inventory beacon collects data from the staging database (otherwise, corrupt or incomplete data may result). A buffer of 2 hours provides a good safety margin (depending on the scale of your XenApp implementation).

Another consideration is that you want your XenApp inventory uploaded to the central FlexNet Manager Suite database before the system import and compliance calculations take place. Typically, this process starts around 2am central server time. A two-hour upload buffer should be more than adequate.

These considerations suggest (within a single time zone) a collection schedule around 10pm, an inventory beacon connection around midnight, and everything in place for the nightly compliance calculation.

The process for setting up Windows scheduled tasks varies across different editions of Windows Server. The following example is for Windows Server 2012. Adjust for your XenApp server's conditions.

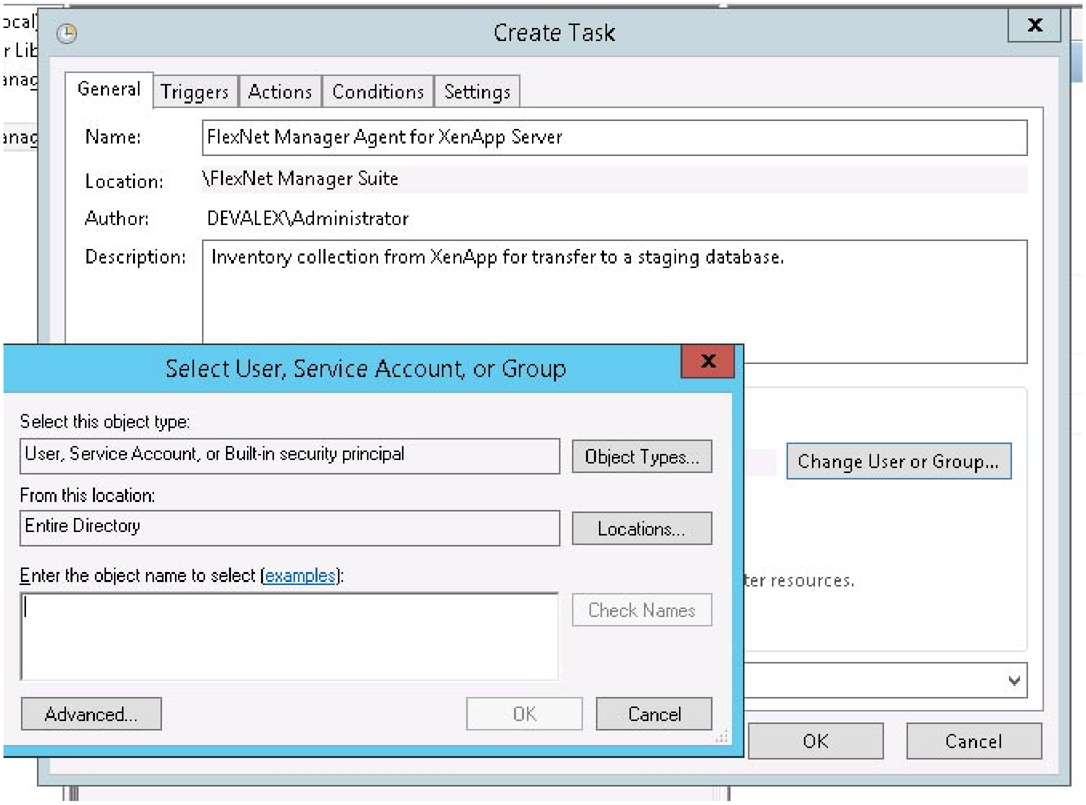
1. In Windows Explorer, navigate to **Control Panel** > **System and Security** > **Administrative Tools**, and double-click **Task Scheduler**.
   The **Task Scheduler** window appears.
2. In the navigation tree on the left, select **Task Scheduler Library**, and then in the **Actions** list on the right, click **New Folder...**.
   A dialog appears for entering the folder name.



A suggested value is `FlexNet Manager Suite`.

3. Click **OK**, and select the new folder in the navigation tree.
4. Select **Action** > **Create Task...**.
   The **Create Task** dialog appears.
5. Enter an appropriate **Name**, such as `FlexNet Manager Agent for XenApp Server`, and add any **Description** to help future maintenance of this task.

   Your description may be something like `Collects application and access information from a Citrix XenApp server and transfers this information to a staging database.`
6. Click **Change User or Group...**.
   The **Select User, Service Account, or Group** dialog appears.



Create Task

Name: FlexNet Manager Agent for XenApp Server

Location: \FlexNet Manager Suite

Author: DEVALEX\Administrator

Description: Inventory collection from XenApp for transfer to a staging database.

Select User, Service Account, or Group

Select this object type: User, Service Account, or Built-in security principal

From this location: Entire Directory

Enter the object name to select (examples):

7. Enter the account name that is to run the scheduled task, and click **OK**.

   This is the account you identified in *Installing the XenApp Server Agent* on page 211, and you can check the requirements for this account there.
8. Further down in the **Security options** group, select **Run whether user is logged in or not**.
9. Switch to the **Triggers** tab, and click **New...**.
   The **New Trigger** dialog appears.
10. Ensure that the default setting **Begin the task** `On a schedule` is selected, set the parameters for the schedule, and from the **Advanced settings** group, be sure that **Enabled** is selected.

The suggested schedule is daily at 10pm local time, but be sure that this suits the upload procedures for your enterprise.

11. Switch to the **Action** tab, and click **New...**.
    The **New Action** dialog appears.
12. Ensure that the default **Action**, `Start a program`, is selected, and browse to your local copy of `FnmpXenAppServerAgent.exe`.
13. In the **Add arguments (optional)** field, specify all the command-line arguments you need for the agent.

    All command line arguments are documented in *XenApp Server Agent Command Line Options* on page 219. For common implementations, you need to define only the connection string to the staging database. In addition, if you are using XenApp 7.5 (or later) and want a single agent to collect inventory from multiple servers, you need the option to define those servers.

    Example 1: Command line arguments to connect to your staging database.

    ```
    -d "Data Source=192.168.13.38;Initial Catalog=MyStaging;User
     ID=accountName;Password=password"
    ```

    Example 2: For XenApp 7.5 (or later), accessing multiple XenApp Servers and recording their details in a common staging database (all on one line):

    ```
    -d "Data Source=192.168.13.38;Initial Catalog=MyStaging;User
     ID=accountName;Password=password"
    -s "localhost, xda01.fqdn.com"
    ```

14. Click **OK**.
15. Optionally, make any preferred adjustments to the **Conditions** or **Settings** tabs (normally the defaults are acceptable).
16. Click **OK** to close the **Create Task** dialog.
    The new task appears in the list of scheduled tasks for this server.
17. Right-click the new task, and click **Run** in the context menu.

    This checks that the scheduled task completes successfully.
18. Validate operations in the following ways:

    - Review the log file (`FnmpXenAppAgent.log` in the same folder as the agent's executable file) for any errors or warning messages.
    - Use Microsoft SQL Server Management Studio to check the contents of the staging database.

Create Connections for Data Upload

The XenApp server agent gathers inventory information from your XenApp Server, and saves it in a staging database. Now an inventory beacon is responsible for uploading the data to the central operations databases of FlexNet Manager Suite. This requires two things:

- Defining a connection to the staging database.
- Defining a schedule to trigger collection of inventory data from the staging database.

In addition, with the exception of XenApp version 7.5, a connection to the appropriate database is strongly recommended, as this allows tracking who is actually using applications:

- If you are using XenApp 6.0 or 6.5, the connection is to your EdgeSight server database
- If you are using XenApp 7.6 or later, the connection is to the XenDesktop database included with XenApp.

Perform this process on the inventory beacon.

1. Log in to your selected inventory beacon.

   Tip • *Starting the Inventory Beacon interface requires that you are logged in with administrator privileges.*

2. If you have not already specified a schedule (or two) that can be linked to the connection(s) you are about to create, it's convenient to do so now.

   There are two connections for each version of XenApp except version 7.5, which needs only one. Remember that you already decided on the schedule for data collection on the XenApp Server (see *Create a Scheduled Task* on page 212), and the schedules on the inventory beacon need to tie in with that plan. For example, you might schedule the connection to the staging database at midnight, and the secondary connection for usage data at 12:15am. For details about creating a schedule on the inventory beacon, see *FlexNet Manager Suite Help > Inventory Beacons > Scheduling Page > Creating a Data Gathering Schedule*.
3. In the navigation pane on the left, select the **Inventory systems** page, and towards the bottom of the page, click **New...**.
   The **Create SQL Source Connection** dialog opens.

   Tip • *The **New...** button defaults to creating a connection for Microsoft SQL Server. If you use the down arrow on the split button, you can also select between **Microsoft SQL Server**, **Other**, and **Spreadsheet** connections.*

4. Complete the values in the dialog, as follows:

Control	Comments
Connection name	A descriptive name for this connection, such as `XenApp` *`ServerName`* `Staging DB`.
Source Type	Select **Citrix XenApp (Server Agent)**. (Don't be confused by **Citrix XenApp (EdgeSight)**, which you may use shortly.)
Server	Type the server name or IP address. Use the special value `(localhost)` if the database is installed on this same inventory beacon server. If the database instance you need is not the default one on the server you identify, add the instance name, separated with a backslash character. Example: `(localhost)\myInstance`
Authentication	Select one of: • **Windows Authentication** — Select this option to use standard Windows authentication to access the database server. The credentials of the account (on the inventory beacon) running the scheduled task for importing inventory

Control	Comments
	are used to access the SQL Server database. This account must be added to a security group that has access to the database. • **Windows (specific account)** — Specify an account on the inventory beacon that can make a connection to the SQL database. • **SQL Authentication** — If you select this option, you must then specify an account and password already known to SQL Server on the target database. This account is used to access the database, regardless of the local account running the scheduled task on the beacon server.
Username	The account name used for SQL authentication, or Windows (specific account). (Not required for Windows Authentication.)
Password	The password for the account name required for SQL authentication, or Windows (specific account). (Not required for Windows Authentication.)
Database	Enter the name of the database, or use the pull-down list to select from database names automatically detected on your specified server.
Connection is in test mode (do not import results)	Controls the uploading and importing of data from this connection: • When this check box is clear, the connection is in production mode, and data collected through this adapter is uploaded to the central server and (in due course) imported into the database there. • When the check box is set: • The adapter for this connection is exercised, with data written to the intermediate file in the staging folder on the inventory beacon (%*CommonAppData*%\Flexera Software\Beacon\IntermediateData) • The immediate upload that normally follows data collection is suppressed, so that you can inspect the contents of the file • The catch-up process that retries stalled uploads, normally scheduled overnight, runs as usual and uploads the file to the central server • At the central server, the file contents are discarded (and not imported into the central database).
Overlapping Inventory Filter	This control does not apply to XenApp inventory records, and you may leave it at the default setting.

5. Click **Test Connection**.
    - If the inventory beacon can successfully connect to the nominated database using the details supplied, a `Database connection succeeded` message displays. Click **OK** to close the message. Click **Save** to complete the addition. The connection is added to (or updated in) the list.
    - If the inventory beacon cannot connect, a `Database connection failed` message is displayed, with information about why that connection could not be made. Click **OK** to close the message. Edit the connection details and retest the connection.

You cannot save the connection details if the connection test fails. If you cannot get the connection test to succeed, click **Cancel** to cancel the addition of these connection details.

6. When the connection to the staging database is successful, and if you are using any XenApp version other than 7.5, repeat the steps above to define a second connection to the usage database (EdgeSight database for version 6.x, and XenDesktop database for version 7.5 and later). This time through, for all these versions set the **Source Type** control to **Citrix XenApp (EdgeSight)** (including for XenApp version 7.5 and later).
7. On the **Inventory systems** page, from the list of connections select the one to the staging database you created in this process, and below the list, click **Schedule...**.
8. In the resulting dialog, choose the schedule you wish to use for this connection, and then click **OK** to close the dialog, and **Save** to apply the selected schedule to your connection.
9. For any XenApp version other than 7.5, on the **Inventory systems** page, from the list of connections select the EdgeSight connection, and repeat the scheduling process, choosing the second schedule you created.

Don't forget, as you move this XenApp adapter into production, to ensure that **Connection is in test mode (do not import results)** is clear (not checked).

3

Command-Line Options

Topics:

For those times when you want to control execution of the command-line agent directly, this chapter covers all options.

XenApp Server Agent Command Line Options

Details for manual operation from the command line.

The FlexNet Manager Agent for XenApp Server (`FnmpXenAppAgent.exe`), or XenApp server agent, is a command line tool which runs regularly on a Citrix XenApp server to determine which end-users have the right to run applications through that server. The data it collects is sent back to FlexNet Manager Suite and processed further once the inventory import process runs.

Syntax

`FnmpXenAppAgent.exe` [*options*...]

Options

```
-d connection
-h
-i true/false
-o output_file
-s servers
-t timeout
-v 0|1
```

where

`-d` *`connection`*	A database connection string to your staging database. Refer to *http://www.connectionstrings.com/sql-server-2008* for some examples. If you do not have a staging database but connect directly to your central inventory database, create the connection string for that database. ***Note* •** *Do not use the* `-d` *option and the* `-o` *option at the same time.*
`-h`	Displays usage for the XenApp server agent.
`-i` *`true/false`*	(Default false.) Ignore errors. Used only for debugging purposes so that the adapter runs end-to-end and logs all issues in the log file (in the same directory as the agent executable).
`-o` *`output_file`*	The full path of a file to store the output of the XenApp server agent as it runs. Use such an output file for debugging purposes, to see the content collected by the agent. The output file is not required for normal operations (the collected data is uploaded directly to the staging database). ***Note* •** *Do not use the* `-d` *option and the* `-o` *option at the same time.*
`-s` *`servers`*	Option only for Citrix XenApp 7.5 (and later). For both 6.5 (which does not support this option) and 7.5 or later (where the option may be omitted), the XenApp server agent assumes that it is running on the XenApp Server from which it is to collect inventory. In both systems, therefore, you may handle multiple XenApp Servers by

	installing the XenApp server agent on each one. When the agent is installed locally on the XenApp Server from which it gathers inventory, omit this option. In Citrix XenApp 7.5 or later, you have the option for a XenApp server agent installed on a single XenApp Server to collect the inventory for all XenApp Servers in a server farm. To do this, create a comma-separated list of fully-qualified domain names or IP addresses for all the servers from which this agent should collect inventory. Inside such a list, use the keyword `localhost` to include the server on which the XenApp server agent is installed. ***Note •*** *The account executing the XenApp server agent must have permissions to connect to each of the servers named in your list.*
`-t` *`timeout`*	The timeout period (in seconds) when connecting to the staging database (default 600 seconds). If the timeout expires, the upload fails, and the data collected from the XenApp Servers on this occasion is lost. An entry is made in the log file (in the same directory as the agent executable) to record the failed connection.
`-v` *`0|1`*	(Default `1`). Sets logging messages to verbose mode. For less information, specify `-v 0`.

Examples

The following examples are split across several lines for readability. Run each example on a single command line.

Use Windows Authentication (for the account running the scheduled task, or the command line) to connect to the staging database with a ten minute (600 second) timeout:

```
FnmpXenAppAgent.exe
        -d "Server=192.168.13.38;Database=MyStaging;Trusted_Connection=yes;"
        -t=600
```

Use a particular user name and password to connect to the staging database:

```
FnmpXenAppAgent.exe
        -d "Data Source=192.168.13.38;Initial Catalog=MyStaging;User
 ID=accountName;Password=password"
```

Collecting inventory (in Citrix XenApp 7.5) from two servers, and for debugging purposes, saving the collected data in an XML file:

```
FnmpXenAppAgent.exe
        -s "localhost, xda01.fqdn.com"
        -o "c:\XenAppTest.xml"
```

4

Validation, Troubleshooting, and Limitations

Topics:

This chapter may assist in diagnosing operations of the adapter, as well as explaining certain inherent limitations in its operation.

Validation and Problem Solving

Because the XenApp server adapter has a number of moving parts, validation and problem solving may also involve multiple steps.

After initial implementation, the simplest validation is simply to inspect the additional installer evidence (for App-V applications) and file evidence (for VDI applications) that are collected by the adapter, and displayed in the web interface for FlexNet Manager Suite. Keep in mind that to complete the end-to-end process, you may need to

- Manually link the evidence to an appropriate application
- Ensure that the application is linked to a suitable license
- Record entitlements on the license, typically by linking purchases to it.

Also remember that you must be importing information from Active Directory prior to importing inventory through the XenApp server adapter.

When more detailed analysis is required, you may use the following checks.

XenApp server agent

The XenApp server agent, installed on your XenApp Server, records a log file in the same folder where it is installed. The log file is replaced at each inventory collection (that is, each time the scheduled task triggers the agent). Review the log for details of any problems. To increase the level of detail, run the agent with the command-line option `-v 1`.

To see all the information that the XenApp server agent has collected, run the agent without a `-d` option (the path to the staging database) and instead using a `-o` option with a path to a convenient local folder. This saves a plain-text XML file of the collected inventory that you can inspect in your preferred text editor. This is a valuable check point when some inventory is being returned, but particular expected applications seem to be missing. If these are missing from a file output with the `-o` option, look for reasons preventing agent access to the source information. Examples might include credentials or access rights to folders containing packages.

If you use the `-o` option, don't forget to replace it with the `-d` option for normal operations!

Staging database

When records missing from the web interface for FlexNet Manager Suite are present in the output of the XenApp server agent (see previous section), next use Microsoft SQL Server Administration Studio to inspect the contents of the staging database. Recall that the contents of the staging database are over-written with each inventory collection by the XenApp server agent. This means that there may be legitimate differences between an output file obtained in the previous section, and the database contents examined in this section. Such differences may come about if there is additional access granted to XenApp applications (the source data) in between the time the agent is run to output the test file, and the time it is run to populate the staging database. In general, however, there should be a high degree of correlation between the two data sets.

The inventory beacon intermediate file

Next, you can validate the data set that the inventory beacon collects from the staging database. Simply trigger the connection to the staging database in test mode, as described in *Create Connections for Data Upload* on page 214. This allows you to inspect the zip archive, which would otherwise be uploaded to the central server, in

`%CommonAppData%\Flexera Software\Beacon\IntermediateData` on the inventory beacon. Provided that you make comparisons before the next run of the XenApp server agent, you should find 1:1 correspondence between the data in the staging database and the intermediate file.

Uploads and imports

If the required data has made it into the intermediate file on the inventory beacon, you may need to debug uploads from the inventory beacon to the central server (for example, see *FlexNet Manager Suite Help > Inventory Beacons > Inventory Beacon Reference > Troubleshooting: Inventory Not Uploading*).

Keep in mind that there is a delay between the upload from the inventory beacon to the central server, and the appearance of the data in the web interface for FlexNet Manager Suite. There must be an inventory import and compliance calculation that occurs between these two events. If you are a member of the Administrator role, in the web interface you may manually trigger an import and compliance calculation.

Limitations

The following limitations apply to the XenApp adapter:

- Information about who has access to applications on the one hand, and about who actually uses applications on the other, is collected separately. Usage data relies on a Citrix EdgeSight server (for XenApp 6.x) or the XenDesktop database included with XenApp version 7.5 and later. Since XenApp 7.5 does not support it, no usage information is available for this version; and if you are using any other version without the appropriate database connection, again no usage information is available.
- When the XenApp server agent connects to a VDI image to read the applications it contains, and the image is not currently running, the agent attempts to start up a server running the image for interrogation (and will shut it down again afterward). This relies on the remote support of power actions (on and off, and so on) for the master image. Where these are not available, or the XenDesktop does not have sufficient resources to spin up another image, the start-up attempt fails. In these cases, the agent imports the name of the executable, but it is missing the version, company, and description details. These cases appear in the list of file evidence with the executable name, but with the name, version, company, and description columns blank.
- XenApp supports "application delivery from a remote PC" and "application delivery from the cloud". Neither of these is supported by the adapter, and no inventory is returned for such cases.
- To improve performance compared with earlier versions of the XenApp adapter, the XenApp server agent no longer collects user names and computer names through the XenApp Server. Instead, it collects only the Active Directory security IDs (SIDs) from XenApp, and relies on the separate import from Active Directory to flesh out the SIDs with more complete identities. This has two immediate implications:
    - If you are upgrading from an earlier version of the XenApp adapter, the data from the access control lists (ACLs) on the XenApp Server is removed on upgrade. An import from Active Directory is then required to populate the SIDs and other identity details. After the first inventory import in the upgraded system, the access data is repopulated.
    - If Active Directory information is not imported by an inventory beacon for the domain(s) in which the XenApp Servers (the ones hosting the XenApp server agent) are located, users and computers newly

registered in Active Directory (since the last import of Active Directory data) will not be recognized or displayed in FlexNet Manager Suite.

- For XenApp version 7.6 and later, application usage information is available only for App-V packages and applications, and streaming profile applications. Usage information is not available for file-based applications, including VDI applications delivered through XenApp.

5

Database Impacts

Topics:

- *Affected Database Tables*

This chapter provides an overview of database tables within FlexNet Manager Suite that are affected by the adapter. These brief notes can be augmented by reviewing the Schema Reference document for the current release.

Affected Database Tables

The XenApp server adapter causes data to be imported to the following database tables in the operations databases for FlexNet Manager Suite (specifically the compliance database). You may prepare custom reports against these:

- `ComplianceDomain` records the domain of the XenApp Server where the XenApp server agent is installed. If the record does not already exist, on import both the FQDN and the flat name are populated.
- `ComplianceUser` records are created for any user names identified in the XenApp inventory but not previously in the operations databases. In these new records, only the `UserName`, `SAMAccountName`, and domain are available and saved (with `ComplianceUserID` calculated automatically). (The domain is a link to the `ComplianceDomain` table, which once again is updated as required with any new records. Such updates should be rare, since domains should be identified from Active Directory.) Because the `ComplianceUser` records created from XenApp inventory imports are far from complete, you may want to enhance these with additional information.
- `ComplianceComputer` records may be created if a `ComplianceUser` record is created (or identified) that has no link to an existing `ComplianceComputer` record. For many types of license, the consumption record is linked to an individual `ComplianceComputer`, so when it is impossible to identify a computer for a particular `ComplianceUser`, a new skeleton computer record is created. These have a computer name of the form

  ```
  UserName (Remote)
  ```

  and have their type shown as `Remote Device`. (This type is identified through a foreign key to the `ComplianceComputerType` table.) They are also linked to the `ComplianceUser` for whom they are created.

  Tip • *If, in future, inventory from another source identifies the same `ComplianceUser` as either the assigned user or the calculated user for a computer identified by that inventory, then this placeholder record for the remote device is removed, and any license consumption recorded against it is moved to the newly-identified (real) computer that belongs with the user.*

- For App-V applications managed by XenApp 7.5 or later, or any XenApp applications managed by version 6.x:
  - `InstallerEvidence` records are created as the primary inventory data for the use of those applications.
  - `InstalledInstallerEvidence` records are create to link that evidence to the appropriate `ComplianceComputer` records.
  - Where the application name, version, and publisher in the `InstallerEvidence` table match an existing evidence rule for an application, an entry is created in the `SoftwareTitleInstallerEvidence` table to link (by foreign keys) the installer evidence to the application record in the `SoftwareTitle` table.
- For VDI applications managed by XenApp 7.5 or later:
  - `FileEvidence` records are created as the primary inventory data for the use of those applications.
  - `InstalledFileEvidence` records are create to link that evidence to the appropriate `ComplianceComputer` records.

    - Where the file name, version, company, and description in the `FileEvidence` table match an existing evidence rule for an application, an entry is created in the `SoftwareTitleFileEvidence` table to link (by foreign keys) the file evidence to the application record in the `SoftwareTitle` table.

- `InstalledApplications` records are created for any applications identified in the `SoftwareTitle` table, linking the application to the `ComplianceComputer` record.