flexera

IT Asset Management 2023 R2.4

System Reference

Legal Information

Document Name: IT Asset Management System Reference version 2023 R2.4 (for cloud implementations)

Part Number: FMS-21.4.0-SR01

Product Release Date: April 24, 2024

Copyright Notice

IT Asset Management 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

Restricted Rights Legend

IT Asset Management System Reference

This guide gathers together a range of reference material for IT Asset Management release 2023 R2.4. This forms part of a reference library that includes the chapters here, together with separate guides on larger topics such as adapters and the database schema. This grouping strikes a balance between supplying too many small guides, and a massive document of unwieldy size.

Contents

1

Adding Custom Properties

It is possible to add properties to underlying database objects, and have these custom properties displayed in the web interface. If you have an on-premises implementation, with your own database, you can implement the changes yourself, following the guidelines in this chapter. If you use a cloud-based implementation, you can use these chapters to create a detailed specification of your requirements, and then submit a change request to your support contact from Flexera (or your third-party managed service provider) to implement your specified changes on your behalf.

Custom Properties

The complexities of managing software licenses within your corporate processes inevitably means you will want additional fields to record data specific to your enterprise. This section explains how you can specify additional properties for various objects that are displayed in the property sheets and your custom reports within IT Asset Management.

Note: *If you are using a cloud-based implementation of IT Asset Management, you do not have direct access to the database, and you cannot "do it yourself". However, you may use this section to understand and specify your* ***requirements****. You can then send a support request to Flexera specifying all the details you require, and your custom property will be added on your behalf.*

The broad overview of the process is:

1. Plan your custom property, including its control type, and where it should appear in the properties of its parent object.
2. In Microsoft SQL Server Management Studio, execute specific SQL procedures to declare your customization in a "top down" manner. For example, if you want an extra field in a new section of an entirely new tab of properties, you must first declare the tab, then the section, and then the field. Objects are positioned relative to others that already exist in the database. In running the procedures, you must also refer to all objects and properties by their internal names, or by numerical mappings. Those names and mappings are included below, in Internal Property Names for Applications and the following similar topics.
3. Customizations are available immediately after the SQL procedure is run, so you can review the results immediate in the web interface for IT Asset Management. You can also immediately commence storing data in your new custom field through the web interface, as the database has been updated to store your data.
4. While the compliance database and the web interface are updated immediately, the data model for the Business

Adapter Studio installed on your inventory beacon(s) is updated by a scheduled task running overnight (central server time). This means waiting until next day before importing values with a business adapter to your new custom field.

Tip: *In general, customizations you make to the user interface and database are preserved through product upgrades. The one exception is the rare case where a product upgrade removes the 'anchor', the object used for positioning your custom control. In this case, both the anchor and your custom control disappear (although the data entered through the control is preserved and is still available in your customer reports). You can remedy this rare case by re-declaring the missing custom control relative to a new anchor. This restores your customization in the web interface, with full access to the previously-recorded data values.*

Limitations

In the web interface, a custom property is displayed in the property sheet of your chosen object, and it is automatically available in the report builder for inclusion in custom reports. However, custom properties and associated data is not available in the following:

- Standard, factory-supplied reports
- Grids in management views
- Search fields, including within property sheets
- Flexera Analytics / IBM Cognos.

As well, custom properties are always editable in property sheets (they cannot be made read-only in that context), and you cannot provide any validation to check data entered into the custom control.

In declaring internal names for your custom properties, you should adopt a stringent naming convention that starts with your own company name-space (a consistent abbreviation for your enterprise name, such as `AE` for Acme Enterprises). You'll next find it convenient to name the object type that you are adding (such as `Asset`). Finally, add the individual name of the property. Separate each of these naming elements with an underscore. Use only characters in the following ranges: `a-z`, `A-Z`, `0-9`, and underscore (_). (Specifically do not use a dot or dash.) A valid example name is:

```
AE_Asset_ChargeBackValue
```

Warning: *Do not use a naming convention that **starts with** the database object name and **uses a dot** as a separator. This combination produces obscure errors. Starting with your own name space makes it safe, and using an underscore separator also makes it safe.*

Objects You Can Customize

The following database objects support the addition of custom properties. When you are adding a custom property to any of these objects, you refer to them by the `TargetTypeID` listed here.

Note: *If you are using a cloud-based implementation of IT Asset Management, you do not have direct access to the database, and you cannot "do it yourself". However, you may use this section to understand and specify your **requirements**. You can then send a support request to Flexera specifying all the details you require, and your custom property will be added on your behalf.*

You can also make your custom property specific to only certain sub-types within each object (where available). For example, you may want to add a custom property to your assets, except that you don't want the custom property to appear on records of routers or switchers. You can exclude these two kinds of assets using the `TargetSubTypeID` included in the listing below. You reference the target sub-types using the numbers in the list (for example, refer to a workstation with the value `1`).

Table 1: Objects supporting customization

Object	TargetTypeID	TargetSubTypeID
Application	13	
Asset	9	**1.** Workstation **2.** Server **3.** Monitor **4.** Desk **5.** Chair **6.** Printer **7.** Router **8.** Switch **9.** Telephone **10.** Cellphone **11.** Laptop **12.** Mobile Device

Object	TargetTypeID	TargetSubTypeID
Contract	10	**1.** General **2.** Lease **3.** Hardware Maintenance And Support **4.** Software License **5.** Software Maintenance And Support **6.** Blanket Purchase **7.** Consulting Services **8.** Insurance **9.** Rent **10.** Subscription **11.** Microsoft Business And Service Agreement **12.** Microsoft Select Agreement **13.** Microsoft Select Plus Agreement **14.** Microsoft Select Enrolment **15.** Microsoft Select Plus Enrolment **16.** Microsoft Enterprise Agreement **17.** Microsoft Enterprise Agreement Subscription
Purchases	20	**1.** Not Set **2.** Software **3.** Hardware **4.** Service **5.** Other **6.** Software Upgrade **7.** Software Maintenance **8.** Disk Kit **9.** Hardware Maintenance

Object	TargetTypeID	TargetSubTypeID
Software License	12	1. Enterprise 2. Device 3. Node Locked 4. User 5. Concurrent User 6. Appliance 7. Client Server 8. OEM 9. Evaluation 10. Run Time 11. Processor 12. Site 13. Named User 14. Core 15. Core Points 16. Oracle DB Processor 17. Oracle DB Named User Plus 18. Processor Points 19. Oracle DB Legacy 20. Enterprise Agreement 21. SAP Named User 22. MS Server Processor 23. CAL 24. Tiered Device 25. IBM PVU 26. IBM Authorized User 27. IBM Concurrent User 28. IBM Floating User 29. Custom Metric 30. One Point Per Processor

Object	TargetTypeID	TargetSubTypeID
		31. IBM RVU **32.** IBM UVU **33.** MS Server Core **34.** Oracle Application User **35.** SAP Package **36.** MS SCCM Client Device **37.** MS SCCM Client User **38.** MSDN
Computer	14	**1.** Computer **2.** VM Host **3.** VM **4.** Remote Device **5.** Mobile Device **6.** VDI Template
User	15	
Vendor	24	

Controls You Can Add

When you declare a custom property to add to a database object, you must also declare the kind of control that is to appear in the property sheet for that object, within the web interface of IT Asset Management.

You can add an entirely new tab to the properties, or within any tab (new or existing) you can add a new section (a grouping for other controls). Both of these cases, tab and section, are special cases in that each has its own SQL procedure for its declaration. These do not need numeric references.

For other controls that you can add within a section (or, for that matter, within a tab without an intervening section if you wish), you specify them by a numeric reference called the `UIFieldTypeID`. The available controls and their `UIFieldTypeID` are shown below.

Prompts (the text beside the control telling the operator what to do) are declared as part of the declaration of each custom property.

Note: *If you are using a cloud-based implementation of IT Asset Management, you do not have direct access to the database, and you cannot "do it yourself". However, you may use this section to understand and specify your* ***requirements****. You can then send a support request to Flexera specifying all the details you require, and your custom property will be added on your behalf.*

Table 2: Supported UI controls

Control	UIFieldTypeID	Comments
Integer	3	A small single-line field combined with up and down arrows (a spinner), where operators may type in an integer value or use the arrows to 'spin up' the number required.
Text box	4	A standard, single-line text field for entering a value.
Text area	5	A rectangular area where the operator can enter free-form text.
Date	6	Provides a date entry field complete with date icon. If the operator clicks the icon, a date picker calendar appears.
Drop-down list	8	A pull-down list of fixed values from which an operator may choose. You declare the values for the list when adding the custom property.
Check box	9	Boolean: saves a 1 in the database when the check box is checked (ticked), and zero otherwise.

Positioning Your Custom Control

Within the web interface for IT Asset Management, you declare the position of your newly-added custom property in relationship to something that already exists in the interface layout. This prior control may be one that came as standard in the factory-supplied interface, or may be another custom control that you have previously declared. We call this previously-existing control the 'anchor' for your newly-added custom property.

Note: *If you are using a cloud-based implementation of IT Asset Management, you do not have direct access to the database, and you cannot "do it yourself". However, you may use this section to understand and specify your* ***requirements****. You can then send a support request to Flexera specifying all the details you require, and your custom property will be added on your behalf.*

Your custom property can have various positional relationships with its anchor. These relationships are declared with numeric values.

Table 3: Positioning of custom controls relative to anchor (mandatory)

Positioning	UIInsertTypeID	Comments
Before	1	Before the anchor
After	2	After the anchor
Start of	3	At the start of an existing tab or section (not applicable for other types of anchor)

When you specify the anchor (the existing control from which your custom control is positioned), you do so by its name. If the anchor is another custom control that you declared earlier, you use exactly the name you specified then. If it is a factory-supplied control that is part of the standard web interface for IT Asset Management, you must use the internal database name for the anchor control. See the following sub-topics for the available objects, their controls as they appear in the English-language standard web interface (these values may be localized), and the internal and unchanging

database name for the same control that you must reference as an anchor.

With the web interface, all properties pages support two columns of controls. When you insert a custom control, the columns reflow to accommodate the change, subject to the additional setting for each of the controls on the page.

For example, you can specify whether the custom control occupies just one column, or spans across two columns.

Table 4: Column span for custom controls

Columns	Width	Comments
Single column	1	Left or right column of the layout (default)
Double column	2	Always left aligned

For single-column controls, you may have a preference for whether the control appears on the left, or on the right, of the property page.

Table 5: Alignment of controls within page

Alignment	Position	Comments
Next available spot	0	The "don't care" option, where the control takes either left or right side based on its positioning in relation to the anchor (default).
Left column	1	This control is forced to the left. If its positioning is "After" an anchor that is already in the left column, the adjacent right side is left blank. Flow resumes after this control.
Right column	2	This control is forced into the right column. If its positioning is "After" an anchor that is already in the right column, the left side of the next line is left blank, and this control occupies the right.

Internal Property Names for Applications

The properties for applications are represented in the tables below, with a separate table for each tab displayed in the web interface (or UI). The first column (for sections with their own heading) and second column (for fields and other kinds of UI controls displayed within each section) show the labels displayed in the default culture `en-US`. The right-most column displays the identity of that control within the underlying system. You must use these identity names when you reference any UI control as an anchor for relative positioning of your new custom control.

Note: *If you are using a cloud-based implementation of IT Asset Management, you do not have direct access to the database, and you cannot "do it yourself". However, you may use this section to understand and specify your* ***requirements****. You can then send a support request to Flexera specifying all the details you require, and your custom property will be added on your behalf.*

Limitations

Of all the database objects supporting custom values, only applications have the following limitations:

- The Business Importer does not current support importing data into custom fields for applications. When you create

a custom property for applications, it is available for data entry within the web interface, and the data may be output in custom reports. This restriction is only that the bulk import of data using the Business Importer is not currently supported.

- The following tabs within application properties are excluded from customization:
  - **Evidence** tab (database identity Tab_Evidence)
  - **Precedence** tab (database identity Tab_Precedence)
  - **Usage** tab (database identity Tab_Usage)

Tab label: General

Database identity: Tab_General

Section	Control	Internal Name
Identification		Section_Identification
	Product	
	Table	Table
	Publisher	Publisher
	Version	Version
	Edition	
	Name	Name
	Source	Source
	FlexeraID	FlexeraID
	Classification	Classification
	Application category	Category
Details		Section_Details
	Status	Status
	Release date	ReleaseDate
	Supported until	SupportedUntil
	Extended support until	ExtendedSupportUntil
	Information	Notes

Tab label: Licenses

Database identity: Tab_Licenses

Section	Control	Internal Name
License consumption order		Section_LicenseConsumptionOrder

Section	Control	Internal Name
	Grid	LicenseOrder
	Automatically manage license priorities	ManagedLicenses

Tab label: Devices

Database identity: Tab_Computers

Section	Control	Internal Name
Related devices		Section_Computers
	Grid	RelatedComputers

Tab label: History

Database identity: Tab_History

Section	Control	Internal Name
History of changes to this application		Section_History
	Grid	ApplicationHistory

Internal Property Names for Assets

The properties for assets are represented in the tables below, with a separate table for each tab displayed in the web interface (or UI). The first column (for sections with their own heading) and second column (for fields and other kinds of UI controls displayed within each section) show the labels displayed in the default culture `en-US`. The right-most column displays the identity of that control within the underlying system. You must use these identity names when you reference any UI control as an anchor for relative positioning of your new custom control.

Note: *If you are using a cloud-based implementation of IT Asset Management, you do not have direct access to the database, and you cannot "do it yourself". However, you may use this section to understand and specify your* ***requirements****. You can then send a support request to Flexera specifying all the details you require, and your custom property will be added on your behalf.*

Tab label: General

Database identity: Tab_General.

Section	Control	Internal Name
General		Section_General
	Name	Name
	Asset type	AssetTypeId

Section	Control	Internal Name
	Linked inventory	LinkedComputer
	Serial number	SerialNumber
	Asset tag	Asset_tag
	Manufacturer	Manufacturer
	Part number	PartNumber
	Model	AssetModel
	Master asset	MasterAssetName
	Status	Status
	Category	CategoryPath
End of life		Section_EndOfLife
	Retirement date	RetirementDate
	Resale price	ResalePrice
	Retirement reason	Reason
	Disposal date	DisposalDate
	Recipient	Recipient
	Written off value	WrittenOffValue
Last inventory		Section_LastInventory
	Electronic	Electronic
	By	Electronic_Created_By
	Physical	Physical
	By	Physical_Created_By
	Installed on	Installed
	Information	Note

Tab label: Hardware

Database identity: Tab_Hardware

Tip: *The* ***Hardware*** *tab appears only when the asset record is linked to an inventory device record.*

Section	Control	Internal Name
Hardware		Section_Hardware
	Operating system	OperatingSystem

Section	Control	Internal Name
	Service pack	ServicePack
	Processors	Processors
	Cores	Cores
	Threads	Threads
	Sockets	Sockets
	Partial No Of Processors	PartialNoOfProcessors
	Processor type	ProcessorType
	Clock speed (MHz)	ClockSpeed
	RAM (GB)	RAM
	Disk (GB)	Disk
	Hard drives	HardDrives
	Display adapters	DisplayAdapters
	Network cards	NetworkCards
	Assigned chassis type	AssignedChassisType
	Inventory Chassis Type	InventoryChassisType

Tab label: Ownership

Database identity: Tab_Ownership

Section	Control	Internal Name
User		*Do not reference.*
	Assigned	
	Calculated	Calculated
	Last logged on	LastLoggedOn

Tab label: Financial

Database identity: Tab_Financial

Section	Control	Internal Name
Financial		Section_Financial
	Acquisition mode	AcquisitionMode
	Delivery date	DeliveryDate
	Warranty type	Warranty

Section	Control	Internal Name
	End of warranty	EndOfWarranty
Lease information *(see note 1)*		Section_LeaseInformation
	Lease agreement	LeaseAgreement
	Lease number	LeaseNumber
	Start date	StartDate
	End date	
	Lease price	LeasePrice
	Buyout	BuyOut
	Payment	Payment
	Period	Period
Lease Termination *(see note 1)*		Section_LeaseTermination
	Date	TerminationDate
	Retirement reason	Reason
Depreciation *(see note 2)*		Section_Depreciation
	Current value	CurrentValue
	Residual value	ResidualValue
	Depreciation method	DepreciationMethod
	Period (years)	PeriodYears
	Rate	RatePercentage
Charges		Section_Charges
	Amount	Amount
	Frequency	Frequency

Note:

1. *These sections and the fields they contain are applicable only when* ***Acquisition mode*** *is set to* `Leased`.
2. *This section and the fields it contains are applicable only when* ***Acquisition mode*** *is set to* `Purchased`.

Tab label: Sub-assets

Database identity: Tab_Sub_Assets

Section	Control	Internal Name
Sub-assets		Section_SubAssets
	Grid	SubAssets

Tab label: Contracts

Database identity: Tab_Contracts

Section	Control	Internal Name
Related contracts		Section_RelatedContracts
	Grid	AssociatedContracts

Tab label: Purchases

Database identity: Tab_Purchases

Section	Control	Internal Name
Related purchases		Section_RelatedPurchases
	Grid	AssociatedPurchases

Tab label: Documents

Database identity: Tab_Documents

Section	Control	Internal Name
Related documents		Section_Documents
	Grid	DocumentChanges

Tab label: History

Database identity: Tab_History

Section	Control	Internal Name
History of changes to this asset		Section_AssetsHistory
	Grid	History
	Created by	CreatedBy
	Creation date	CreationDate
	Last updated by	LastUpdatedBy
	Last updated date	LastUpdatedDate

Internal Property Names for Computers

The properties for computers (presented in the web interface as inventory devices) are represented in the tables below, with a separate table for each tab displayed in the web interface (or UI). The first column (for sections with their own heading) and second column (for fields and other kinds of UI controls displayed within each section) show the labels displayed in the default culture en-US. The right-most column displays the identity of that control within the

underlying system. You must use these identity names when you reference any UI control as an anchor for relative positioning of your new custom control.

Note: *If you are using a cloud-based implementation of IT Asset Management, you do not have direct access to the database, and you cannot "do it yourself". However, you may use this section to understand and specify your* ***requirements****. You can then send a support request to Flexera specifying all the details you require, and your custom property will be added on your behalf.*

Limitations

The Licenses tab (database identity: Tab_Licenses) within computer properties is excluded from customization.

Tab label: General

Database identity: Tab_Summary.

Section	Control	Internal Name
General		Section_Summary
	Status	Status
	Inventory device type	InventoryDeviceType
	Name	Name
	Compliance status	ComplianceStatus
	Linked asset	LinkedToAsset
	Domain name	Domain
	Inventory role	Role
	Manufacturer	Manufacturer
	Model	ComputerModel
	IP address	IPAddress
	MAC address	MACAddress
	Serial number	SerialNumber
	Chassis number	ChassisNumber
	Category	CategoryPath
Last inventory source		Section_InventorySource
	Last inventory date	LastInventoryDate
	Last inventory source	LastInventorySource
	Connection name	InventoryConnectionName
	You may override inventory values	OverrideInventoryValue
Service Provider		Section_ServiceProvider

Section	Control	Internal Name
	Located in service provider's datacenter cloud	LocatedServiceProviderCloud
	Service Provider	ServiceProvider

Tab label: Hardware

Database identity: Tab_Hardware.

Section	Control	Internal Name
Hardware		Section_Hardware
	Operating system	OperatingSystem
	Service pack	ServicePack
	Processors	Processors
	Cores	Cores
	Threads	Threads
	Sockets	Sockets
	Partial no of processors	PartialNoOfProcessors
	Processor type	ProcessorType
	Clock speed (MHz)	ClockSpeed
	RAM (GB)	RAM
	Disk (GB)	Disk
	Hard drives	HardDrives
	Display adapters	DisplayAdapters
	Network cards	NetworkCards
	Assigned chassis type	AssignedChassisType
	Inventory Chassis Type	InventoryChassisType
	* You may override inventory values	OverrideInventoryValue2

Tab label: Applications

Database identity: Tab_Applications.

Tip: *The* ***Applications*** *tab is displayed only when* ***Last inventory source*** *(in the* ***General*** *tab) has a value other than* `Manual`.

Section	Control	Internal Name
Applications installed on this device		Section_Applications
	Grid	Applications

Tab label: Ownership

Database identity: Tab_Ownership.

Section	Control	Internal Name
User		*Do not reference.*
	Assigned	
	Calculated	Calculated
	Last logged on	LastLoggedOn

Tab label: VM Properties

Database identity: Tab_VMProperties.

Section	Control	Internal Name
Virtual Machine Properties		Section_VMProperties
	Host	Host
	Friendly name	VM_Name
	Guest full name	VM_GuestFullName
	UUID	VM_UUID
	Location	VM_Location
	VM type	VM_Type
	Pool	VM_Pool
	Total memory (GB)	VM_TotalMemory
	Memory usage (GB)	VM_MemoryUsage
	CPU usage (MHz)	VM_CPUUsage
	Last known state	VM_LastKnownState
	Host affinity enabled	VM_AffinityEnabled
	Threads (max)	
	CPU affinity	
	Core affinity	

Section	Control	Internal Name
	Partition number	
	Partition ID	

Tab label: Virtual Machines

Database identity: Tab_VirtualMachines.

Section	Control	Internal Name
Virtual machines		Section_VirtualMachines
	Grid	VirtualMachines

Tab label: Virtual Desktop Templates

Database identity: Tab_VdiTemplates.

Section	Control	Internal Name
Virtual Deskop Templates accessed from this device		Section_VdiTemplates
	Grid	VdiTemplates

Tab label: History

Database identity: Tab_History.

Section	Control	Internal Name
History of changes to this device		Section_History
	Grid	ComputerHistory
	Created by	CreatedBy
	Creation date	CreationDate
	Last updated by	LastUpdatedBy
	Last updated by	LastUpdatedDate

Internal Property Names for Contracts

The properties for contracts are represented in the tables below, with a separate table for each tab displayed in the web interface (or UI). The first column (for sections with their own heading) and second column (for fields and other kinds of UI controls displayed within each section) show the labels displayed in the default culture `en-US`. The right-most column displays the identity of that control within the underlying system. You must use these identity names when you reference any UI control as an anchor for relative positioning of your new custom control.

Note: *If you are using a cloud-based implementation of IT Asset Management, you do not have direct access to the database, and you cannot "do it yourself". However, you may use this section to understand and specify your* ***requirements****. You can then send a support request to Flexera specifying all the details you require, and your custom property will be added on your behalf.*

Tab label: General

Database identity: Tab_General.

Section	Control	Internal Name
Identification		Section_Identification
	Contract no	ContractNo
	Status	Status
	Description	Description
	Contract type	ContractType
	Purchase program	PurchaseProgram
	Select applications level	ApplicationLevel
	Select systems level	SystemsLevel
	Select servers level	ServersLevel
	Initial platform quantity	InitialPlatformQuantity
	Replaced by	ReplacedContractBy
	Category	CategoryPath
Events		Section_Events
	Evergreen	Evergreen
	Start date	StartDate
	Next renewal date	NextRenewalDate
	Expiry date	ExpiryDate
	Review date	ReviewDate
	Last renewed date	LastRenewedDate
Payments		Section_Payments
	Global amount	GlobalAmount
	Monthly amount	MonthlyAmount
	Information	Comments

Tab label: Ownership

Database identity: Tab_ Ownership.

Section	Control	Internal Name
Ownership		Section_Ownership

Tab label: Vendors

Database identity: Tab_Vendors.

Section	Control	Internal Name
Other vendors		Section_OtherVendors
Additional vendors		Section_Vendors
	Grid	Vendors
Third-party vendors		Section_3rdPartyVendors
	Grid	Contract3rdPartyVendors

Tab label: Assets

Database identity: Tab_Assets.

Section	Control	Internal Name
Assets		Section_Assets
	Grid	Assets

Tab label: Purchases

Database identity: Tab_Purchases.

Section	Control	Internal Name
Related purchases		Section_RelatedPurchases
	Grid	Purchases

Tab label: Licenses

Database identity: Tab_Licenses.

Section	Control	Internal Name
Related licenses		Section_RelatedLicenses
	Grid	Licenses

Tab label: Responsibilities

Database identity: Tab_Responsibilities.

Section	Control	Internal Name
Responsibilities		Section_Responsibilities
	Grid	Responsibilities

Tab label: Payment schedules

Database identity: Tab_Payment_Schedule.

Section	Control	Internal Name
Payment schedules		Section_Payment_Schedule
	Grid	PaymentSchedules

Tab label: Terms and conditions

Database identity: Tab_Terms_and_Conditions.

Section	Control	Internal Name
Terms and conditions		Section_Terms_and_Conditions
	Grid	TermsConditions

Tab label: Documents

Database identity: Tab_Documents

Section	Control	Internal Name
Related Documents		Section_Documents Grid
	Grid	DocumentChanges

Tab label: History

Database identity: Tab_History.

Section	Control	Internal Name
History of changes to this contract		Section_History
	Grid	History
	Created by	CreatedBy
	Creation date	CreationDate

Section	Control	Internal Name
	Last updated by	LastUpdatedBy
	Last updated by	LastUpdatedDate

Internal Property Names for Licenses

The properties for software licenses are represented in the tables below, with a separate table for each tab displayed in the web interface (or UI). The first column (for sections with their own heading) and second column (for fields and other kinds of UI controls displayed within each section) show the labels displayed in the default culture `en-US`. The right-most column displays the identity of that control within the underlying system. You must use these identity names when you reference any UI control as an anchor for relative positioning of your new custom control.

Note: *If you are using a cloud-based implementation of IT Asset Management, you do not have direct access to the database, and you cannot "do it yourself". However, you may use this section to understand and specify your* ***requirements****. You can then send a support request to Flexera specifying all the details you require, and your custom property will be added on your behalf.*

Tab label: Compliance

Database identity: Tab_Compliance.

Section	Control	Internal Name
Compliance		Section_Compliance
	Compliance status	ComplianceStatus
	Shortfall/Availability	Available
	Breach reason	BreachReason
Entitlements and consumption		EntitlementsAndConsumption
	Licensed from PO	PurchasedFromPO
	Allocated	NumberAllocated
	Extra entitlements	ExtraEntitlements
	Used	Used
	Raw consumption	RawConsumption
	Raw usage count	RawUserCount
	NUP minimum	NUPMinimum
	Peak consumed	PeakConsumption
	Raw installations	RawInstallation
	PUR savings	PURSavings
	Total entitlements	TotalPurchased

Section	Control	Internal Name
	Consumed entitlements	AdjustedConsumption
	Consumption as at	ConsumedDate

Tab label: Identification

Database identity: Tab_Identification.

Section	Control	Internal Name
Identification		Section_Identification
	Publisher	Publisher
	Name	Name
	License type	LicenseType
	Subject to true up	SubjectToTrueUp
	Copy Version and Edition from the most recent application	CopyVersionEdition
	Version	Version
	Edition	Edition
	Duration	Duration
	Purchased as	PurchasedAs
	Expiry date	ExpiryDate
	Status	LifeCycle
	Retirement date	RetirementDate
	Retirement reason	RetirementReason
	Resale price	ResalePriceLink
	Metric	SoftwareLicenseMetricID
	Set Compliance status manually	ManuallySetComplianceStatus
	Resources consumed	ResourcesConsumed
	SAP type	SAPType
	Measurement date	MeasurementDate
	Tier type	TierType
	Tier code	TierCode
	Processors limit	ProcessorsLimit
	Cores limit	CoresLimit

Section	Control	Internal Name
	Legacy type	LegacyType
	Maximum sockets	MaximumSockets
	Minimum users	MinimumUsers
	Apply user limit per processor core	ApplyUserLimitPerCore
	Minimum processors	MinimumProcessors
	Points rule set	PointsRuleSet
	Category	CategoryPath
	Serial number	SerialNumber
	Notes	Notes
License keys		Section_LicenseKeys
	Rule	RuleType
	License key	LicenseKey

Tab label: Applications

Database identity: Tab_Applications.

Section	Control	Internal Name
Licensed software		Section_LicensedSoftware
	Title	ApplicationTitle
	Highest version	HighestVersion
Applications included in this license		Section_IncludedApplications
	Grid	Applications

Tab label: Purchases

Database identity: Tab_Purchases.

Section	Control	Internal Name
Purchase price		Section_PurchasePrice
	Override unit price	PurchasePrice
Purchases		Section_PurchaseOrderLineItems
	Grid	Purchases

Tab label: Financial

Database identity: Tab_Financial

Section	Control	Internal Name
Charges		Section_Charges
	Amount	Amount
	Frequency	Frequency
Resale		Section_Resale
	Resale price	ResalePrice
	Recipient	Recipient

Tab label: Contracts

Database identity: Tab_Contracts.

Section	Control	Internal Name
Related contracts		Section_Contracts
	Grid	Contracts

Tab label: Consumption

Database identity: Tab_Consumption.

Section	Control	Internal Name
Normal user equivalents		Section_UserEquivalents
	Infrequent user	InfrequentUser
	External user	ExternalUser
Bulk user counts		Section_BulkUserCounts
	Additional infrequent users	AdditionalInfrequentUsers
	Additional external users	AdditionalExternalUsers
Related users		Section_RelatedEmployees
Related devices		Section_RelatedComputers
Alternate bulk user count		Section_AlternateBulkUserCount
	Non-inventoried users	AlternateNonInventoriedUsers
Users related to this license		Section_OracleUsers
	Grid	OracleUserConsumption

Tab label: Restrictions

Database identity: Tab_Restrictions.

Section	Control	Internal Name
Scope restrictions		Section_ScopeRestrictions
	Restrict to OS	Restrict_OS
	Grid	Restrictions

Tab label: Ownership

Database identity: Tab_Ownership.

Section	Control	Internal Name
Ownership and access rights		Section_Ownership

Tab label: Documents

Database identity: Tab_Documents

Section	Control	Internal Name
Related documents		Section_Documents
	Grid	DocumentChanges

Tab label: History

Database identity: Tab_History.

Section	Control	Internal Name
History of changes to this license		Section_History
	Grid	History
	Created by	CreatedBy
	Creation date	CreationDate
	Last updated by	LastUpdatedBy
	Last updated date	LastUpdatedDate

Internal Property Names for Purchases

The properties for purchases are represented in the tables below, with a separate table for each tab displayed in the web interface (or UI). The first column (for sections with their own heading) and second column (for fields and other kinds of UI controls displayed within each section) show the labels displayed in the default culture `en-US`. The right-

most column displays the identity of that control within the underlying system. You must use these identity names when you reference any UI control as an anchor for relative positioning of your new custom control.

Note: *If you are using a cloud-based implementation of IT Asset Management, you do not have direct access to the database, and you cannot "do it yourself". However, you may use this section to understand and specify your* ***requirements****. You can then send a support request to Flexera specifying all the details you require, and your custom property will be added on your behalf.*

Tab label: General

Database identity: Tab_General.

Section	Control	Internal Name
Purchase details		Section_PurchaseDetails
	Item	SequenceNumber
	Description	Description
	Part no./SKU	SKUDetails
	Purchase quantity	PurchaseQuantity
	Quantity per unit	QuantityPerUnit
	Effective quantity	EffectiveQuantity
	Publisher	Publisher
	Status	ItemStatus
	Purchase type	Type
Related contract		Section_RelatedContract
	Contract	Contract
Maintenance		Section_Maintenance
	Purchase includes support, maintenance, or other service Ö	MaintenanceOrServiceAgreemen
	Agreement date	AgreementDate
	Expiry date	ExpiryDate
Volume purchase program		Section_VolumePurchaseProgram
	Product pool	ProductPool
	Product points	ProductPoints
Notes		Section_Notes
	Comments	Comments

Tab label: Financial

Database identity: Tab_Financial

Section	Control	Internal Name
Invoice		Section_Invoice
	Invoice number	InvoiceNumber
	Invoice date	InvoiceDate
Shipping		Section_Shipping
	Shipping date	ShippingDate
	Shipping location	ShippingLocation

Tab label: Ownership

Database identity: Tab_Ownership.

Section	Control	Internal Name
Ownership		Section_Ownership

Tab label: Assets

Database identity: Tab_Assets.

Section	Control	Internal Name
Assets		Section_Assets
	Grid	Assets

Tab label: Licenses

Database identity: Tab_Licenses.

Section	Control	Internal Name
Related licenses		Section_Licenses
	Allocate assigned entitlements	AllocateAssignedEntitlements
	Grid	Licenses

Tab label: Documents

Database identity: Tab_Documents.

Section	Control	Internal Name
Related documents		Section_Documents

Section	Control	Internal Name
	Grid	DocumentChanges

Tab label: History

Database identity: Tab_History.

Section	Control	Internal Name
History of changes to this purchase order		Section_History
	Grid	History
	Created by	CreatedBy
	Creation date	CreationDate
	Last updated by	LastUpdatedBy
	Last updated by	LastUpdatedDate

Internal Property Names for Users

The properties for users are represented in the tables below, with a separate table for each tab displayed in the web interface (or UI). The first column (for sections with their own heading) and second column (for fields and other kinds of UI controls displayed within each section) show the labels displayed in the default culture `en-US`. The right-most column displays the identity of that control within the underlying system. You must use these identity names when you reference any UI control as an anchor for relative positioning of your new custom control.

Note: *If you are using a cloud-based implementation of IT Asset Management, you do not have direct access to the database, and you cannot "do it yourself". However, you may use this section to understand and specify your* ***requirements****. You can then send a support request to Flexera specifying all the details you require, and your custom property will be added on your behalf.*

Tab label: General

Database identity: Tab_General.

Section	Control	Internal Name
Identification		User
	Title	UserTitle
	First name	FirstName
	Middle name	MiddleName
	Last name	LastName
	Suffix	Suffix

Section	Control	Internal Name
	Full name	Fullname
Employment		Section_Employment
	Job title	JobTitle
	Employee ID	EmployeeID
	Enployment Status	EmploymentStatus
	Manager	Manager
	Status	ComplianceUserStatus
Account		Section_Account
	Account name	AccountName
	Domain name	Domain
	Last inventory source	LastInventorySource

Tab label: Details

Database identity: Tab_Details.

Section	Control	Internal Name
Enterprise groups		Section_EnterpriseGroups
Contact		Section_Contact
	Phone	Phone
	Fax	Fax
	Mobile	Mobile
	Email	Email
Address		Section_Address
	Street address	StreetAddress
	City	City
	State/Province	State
	Country	Country
	Postal code	PostalCode

Tab label: Hardware

Database identity: Tab_Hardware.

Section	Control	Internal Name
Assets		Section_Assets
	Grid	Assets
Devices		Section_Computers
	Grid	Computers

Tab label: Software

Database identity: Tab_Software.

Section	Control	Internal Name
Related software licenses		Section_Software
	Grid	Software

Tab label: Responsibilities

Database identity: Tab_Responsibilities.

Section	Control	Internal Name
Responsibilities		Section_Responsibilities
Licenses		Section_Licenses
	Grid	Licenses
Contracts		Section_Contracts
	Grid	Contracts

Tab label: Documents

Database identity: Tab_Documents.

Section	Control	Internal Name
Related documents		Section_Documents
	Grid	DocumentChanges

Tab label: History

Database identity: Tab_History.

Section	Control	Internal Name
History of changes to this user		Section_History
	Grid	History

Section	Control	Internal Name
	Created by	CreatedBy
	Creation date	CreationDate
	Last updated by	LastUpdatedBy
	Last updated by	LastUpdatedDate

Internal Property Names for Vendors

The properties for vendors are represented in the tables below, with a separate table for each tab displayed in the web interface (or UI). The first column (for sections with their own heading) and second column (for fields and other kinds of UI controls displayed within each section) show the labels displayed in the default culture `en-US`. The right-most column displays the identity of that control within the underlying system. You must use these identity names when you reference any UI control as an anchor for relative positioning of your new custom control.

Note: *If you are using a cloud-based implementation of IT Asset Management, you do not have direct access to the database, and you cannot "do it yourself". However, you may use this section to understand and specify your* ***requirements****. You can then send a support request to Flexera specifying all the details you require, and your custom property will be added on your behalf.*

Tab label: General

Database identity: Tab_General.

Section	Control	Internal Name
Identification		Section_Identification
	Name	Name
Contact information		Section_ContactInformation
	Phone	PhoneNumber
	Fax	Fax
	Email	Email
	Web	Web
Address		Section_Address
	Street address	StreetAddress
	City	City
	State/Province	State
	Country	Country
	Postal code	PostalCode

Tab label: Purchases

Database identity: Tab_Purchases.

Section	Control	Internal Name
Related purchases		Section_PurchaseOrderLineItems
	Grid	VendorPurchases

Tab label: Assets

Database identity: Tab_Assets.

Section	Control	Internal Name
Related assets		Section_AssociatedAssets
	Grid	VendorAssets

Tab label: Contracts

Database identity: Tab Contracts.

Section	Control	Internal Name
Related contracts		Section_AssociatedContracts
	Grid	VendorContracts

Tab label: History

Database identity: Tab_History.

Section	Control	Internal Name
History of changes to this vendor		Section_History
	Grid	History
	Created by	CreatedBy
	Creation date	CreationDate
	Last updated by	LastUpdatedBy
	Last updated by	LastUpdatedDate

Creating a New Properties Tab

Execute the following in SQL Server Management Studio against your FNMSCompliance database.

Note: *If you are using a cloud-based implementation of IT Asset Management, you do not have direct access to the database, and you cannot "do it yourself". However, you may use this section to understand and specify your* ***requirements****. You can then send a support request to Flexera specifying all the details you require, and your custom property will be added on your behalf.*

Syntax:

```
EXEC dbo.AddTabToWebUIPropertiesPage
        @TargetTypeID = TargetTypeID,
        @ExcludeTargetSubTypeIDs = 'TargetSubTypeID,TargetSubTypeID,...',
        @Name = 'My_Unique_Name',
        @CultureType = 'ISOCultureCode',
        @DisplayNameInPage = 'Prompt value',
        @UIInsertTypeID = UIInsertTypeID,
        @RelativeTabName = 'RelativeTabName'
```

where

@TargetTypeID	Mandatory. Integer that identifies the type of object to which you are adding a custom property. For supported objects and their integer equivalents for `TargetTypeID`, see Objects You Can Customize.
@ExcludeTargetSubTypeIDs	Mandatory. A comma-separated list (enclosed in single quotation marks) of integer subtype IDs. For the default case of no exclusions, give this parameter an empty list: `@ExcludeTargetSubTypeIDs = '',` Many types of target objects have subtypes (for example, assets may be workstations, routers, and so on). By default, a custom property added to an object (identified by its `TargetTypeID`) is added to all subtypes of that object. However, you can exclude any subtypes you choose with this parameter. For supported subtypes and their integer equivalents for *`TargetSubTypeID`*, see Objects You Can Customize.
@Name	Mandatory. The internal name (used in code and database) of the new tab you are adding. This name must be unique across all tabs in the system (including the factory-supplied tabs, and including all database objects). For this reason, it is strongly recommended that you adopt a stringent naming convention, such as a company name space, an object type, and a tab name. The name is limited to 256 characters, and only alphanumeric ASCII characters (`A-Z`, `a-z`, and `0-9`) are acceptable. Using any other characters results in an error message: `Input @Name contains invalid character` ***Tip:*** *Keep in mind that this name is internal, and not displayed to operators. For a name visible to others, see* ***@DisplayNameInPage*** *below, which supports a wider range of characters.*

@CultureType	Default value is `en-US`. Value is a five-character ISO code for culture (enclosed in single quotation marks). The permitted values are available at http://msdn.microsoft.com/en-us/goglobal/bb896001.aspx.
@DisplayNameInPage	Mandatory. This is the label (enclosed in single quotation marks) displayed on the tab in the web interface for IT Asset Management when the culture setting for the interface matched the one you declare in `CultureType`. You can also provide localized values for this label using different culture settings, for which see Localizing Display Names of Custom Properties.
@UIInsertTypeID	Mandatory. An integer indicating the position of this new tab relative to the tab identified in `RelativeTabName`. For integer values and their meaning, see Positioning Your Custom Control. Note that in this case of creating a new tab, the value `3` is not relevant.
@RelativeTabName	Mandatory. The internal name of the anchor tab relative to which you are positioning your new custom tab. For internal names of factory-supplied tabs, see subtopics of Positioning Your Custom Control.

Creating a New Section Within a Tab

Execute the following in SQL Server Management Studio against your FNMSCompliance database, referencing an existing tab.

Note: *If you are using a cloud-based implementation of IT Asset Management, you do not have direct access to the database, and you cannot "do it yourself". However, you may use this section to understand and specify your* ***requirements****. You can then send a support request to Flexera specifying all the details you require, and your custom property will be added on your behalf.*

Syntax:

```
EXEC dbo.AddSectionToWebUIPropertiesPage
        @TargetTypeID = TargetTypeID,
        @ExcludeTargetSubTypeIDs = 'TargetSubTypeID,TargetSubTypeID,...',
        @Name = 'My_Unique_Name',
        @CultureType = 'ISOCultureCode',
        @DisplayNameInPage = 'Prompt value',
        @TabName = 'tabInternalName',
        @UIInsertTypeID = UIInsertTypeID,
        @RelativePositionTo = 'RelativePositionTo'
```

where

@TargetTypeID	Mandatory. Integer that identifies the type of object to which you are adding a custom property. For supported objects and their integer equivalents for `TargetTypeID`, see Objects You Can Customize.

@ExcludeTargetSubTypeIDs	Mandatory. A comma-separated list (enclosed in single quotation marks) of integer subtype IDs. For the default case of no exclusions, give this parameter an empty list: `@ExcludeTargetSubTypeIDs = '',` Many types of target objects have subtypes (for example, assets may be workstations, routers, and so on). By default, a custom property added to an object (identified by its `TargetTypeID`) is added to all subtypes of that object. However, you can exclude any subtypes you choose with this parameter. For supported subtypes and their integer equivalents for *`TargetSubTypeID`*, see Objects You Can Customize.
@Name	Mandatory. The internal name (used in code and database) of the new section you are adding. This name must be unique across all sections in the system (including the factory-supplied sections, across all database objects). For this reason, it is strongly recommended that you adopt a stringent naming convention, such as a company name space, an object type, optionally a tab name, and a section name (example: `MyCoLicenseChargebackGeneral`). The name is limited to 256 characters, and only alphanumeric ASCII characters (`A-Z`, `a-z`, and `0-9`) are acceptable. Using any other characters results in an error message: `Input @Name contains invalid character` ***Tip:*** *Keep in mind that this name is internal, and not displayed to operators. For a name visible to others, see* ***@DisplayNameInPage*** *below, which supports a wider range of characters.*
@CultureType	Default value is `en-US`. Value is a five-character ISO code for culture (enclosed in single quotation marks). The permitted values are available at http://msdn.microsoft.com/en-us/goglobal/bb896001.aspx.
@DisplayNameInPage	Mandatory. This is the label (enclosed in single quotation marks) displayed above the section in the web interface for IT Asset Management when the culture setting for the interface matched the one you declare in `CultureType`. You can also provide localized values for this label using different culture settings, for which see Localizing Display Names of Custom Properties.
@TabName	Optional when `@UIInsertTypeID = 3`, and ignored for any other value. Provides the internal name the tab at the start of which the new section is to be inserted. If used, the tab name must be 80 characters or less, and enclosed inside single quotation marks. (If `TabName` is not specified, the value of `RelativePositionTo` is used.) For internal names of factory-supplied controls, see the subtopics under Positioning Your Custom Control.
@UIInsertTypeID	Mandatory. An integer indicating the position of this new section relative to the control identified in `RelativePositionTo`. For integer values and their meaning, see Positioning Your Custom Control. Note that in this case of creating a new section, the value `3` means at the start of the tab identified in `TabName`, meaning that `RelativePositionTo` is irrelevant in that case.

@RelativePositionTo	Mandatory when `UIInsertTypeID` has a value other than `3`; and when `@UIInsertTypeID = 3`, one of `RelativePositionTo` or `TabName` is required. The internal name of the anchor control relative to which you are positioning your new custom section. For internal names of factory-supplied controls, see the subtopics under Positioning Your Custom Control. When used with `@UIInsertTypeID = 3`, `RelativePositionTo` must be the name of a tab that has already been defined (and not any other kind of control).

Creating Other Custom Properties

Execute the following in SQL Server Management Studio against your FNMSCompliance database. The relative anchor from which positioning is determined must already be defined.

Note: *If you are using a cloud-based implementation of IT Asset Management, you do not have direct access to the database, and you cannot "do it yourself". However, you may use this section to understand and specify your* ***requirements****. You can then send a support request to Flexera specifying all the details you require, and your custom property will be added on your behalf.*

Syntax:

```
EXEC dbo.AddPropertyToWebUIPropertiesPage
        @TargetTypeID = TargetTypeID,
        @ExcludeTargetSubTypeIDs = 'TargetSubTypeID,TargetSubTypeID,...',
        @Name = 'InternalFieldName',
        @CultureType = 'ISOCultureCode',
        @DisplayNameInPage = 'Prompt value',
        @DisplayNameInReport = 'Column heading',
        @TabName = 'MyTabName',
        @UIInsertTypeID = UIFieldTypeID,
        @UIFieldTypeID = UIFieldTypeID,
        @RelativePositionTo = 'RelativePositionTo',
        @SequenceNumber = 'IntegerCount',
        @Position = Position,
        @Width = Width,
        @DataSource = 'List, Of, Values',
        @DataSourceDelimiter = ',',
        @Required = 0,
        @StringLength = IntegerMaxLength,
        @ReadOnly = 0
```

Note: @DataSource *values in the custom property drop-down list are displayed in the following order in the user interface: numerical, alphanumeric, alphabetical. For example: 1, 1000, 1a, 4000, Extra, Main, Return required, Shared.*

@TargetTypeID	Mandatory. Integer that identifies the type of object to which you are adding a custom property. For supported objects and their integer equivalents for `TargetTypeID`, see Objects You Can Customize.
@ExcludeTargetSubTypeIDs	Mandatory. A comma-separated list (enclosed in single quotation marks) of integer subtype IDs. For the default case of no exclusions, give this parameter an empty list: `@ExcludeTargetSubTypeIDs = '',` Many types of target objects have subtypes (for example, assets may be workstations, routers, and so on). By default, a custom property added to an object (identified by its `TargetTypeID`) is added to all subtypes of that object. However, you can exclude any subtypes you choose with this parameter. For supported subtypes and their integer equivalents for *`TargetSubTypeID`*, see Objects You Can Customize.
@Name	Mandatory. The internal name (used in code and database) of the new custom property you are adding. This name must be unique across all properties in the system (including the factory-supplied properties, across all database objects). For this reason, it is strongly recommended that you adopt a stringent naming convention, such as a company name space, an object type, and a property name, (example: `MyCoLicenseDailyCharge`). The name is limited to 256 characters, and only alphanumeric ASCII characters (`A-Z`, `a-z`, and `0-9`) are acceptable. Using any other characters results in an error message: `Input @Name contains invalid character`  ***Tip:*** *Keep in mind that this name is internal, and not displayed to operators. For names visible to others, see* ***@DisplayNameInPage*** *and* ***@DisplayNameInReport*** *below, both of which support a wider range of characters.*
@CultureType	Default value is `en-US`. Value is a five-character ISO code for culture (enclosed in single quotation marks). The permitted values are available at http://msdn.microsoft.com/en-us/goglobal/bb896001.aspx.
@DisplayNameInPage	Mandatory. This is the label (enclosed in single quotation marks) displayed as a prompt in the web interface for IT Asset Management when the culture setting for the interface matched the one you declare in `CultureType`. You can also provide localized values for this label using different culture settings, for which see Localizing Display Names of Custom Properties.
@DisplayNameInReport	Mandatory. This is the label (enclosed in single quotation marks) displayed as a column heading in custom reports you prepare, when the culture setting for the interface matched the one you declare in `CultureType`. You can also provide localized values for this label using different culture settings, for which see Localizing Display Names of Custom Properties.

@TabName	Optional when `@UIInsertTypeID = 3`, and ignored for any other value. Provides the internal name the tab in which the new control is to be inserted. If used, the tab name must be 80 characters or less, and enclosed inside single quotation marks. (If `TabName` is not specified, the value of `RelativePositionTo` is used.) For internal names of factory-supplied tabs, see the subtopics under Positioning Your Custom Control. For details about creating custom tabs, see Creating a New Properties Tab.
@UIInsertTypeID	Mandatory. An integer indicating the position of this new section relative to the control identified in `RelativePositionTo`. For integer values and their meaning, see Positioning Your Custom Control. Note that in this case of creating a new section, the value `3` means at the start of the tab identified in `TabName`, meaning that `RelativePositionTo` is irrelevant in that case.
@UIFieldTypeID	Mandatory. An integer indicating the kind of control used to display your custom property. For integer values and their meaning, see Controls You Can Add.
@RelativePositionTo	Mandatory. The internal name of the anchor control relative to which you are positioning your new custom section. For internal names of factory-supplied controls, see the subtopics under Positioning Your Custom Control.
@SequenceNumber	Optional (when omitted, the default value is null). Where two or more custom properties are declared with the same anchor in their **@RelativePositionTo** parameters, they are ordered by the sequence number. If there is no sequence number declared, they are ordered by the execution order of the SQL commands.
@Position	Optional (when omitted, the default value is `0`). The alignment of your custom control within the two-column layout of a properties page. For the integer values and their meanings, see Positioning Your Custom Control.
@Width	Optional (when omitted, the default value is `1`). The number of columns spanned by this control in the two-column layout of a properties page. For more information, see Positioning Your Custom Control.

@DataSource	Mandatory when `UIFieldTypeID = 8`, and otherwise ignored. Within single quotes, this is an ordered list of the values to be displayed in numerical, alphanumeric, alphabetical order within the option list. By default, the list is comma-separated, but see **@DataSourceDelimiter**. Values (between delimiters) may include white space, and leading white space on a value is ignored. Every value must be unique. One of the values may be a null, creating a blank row in the drop-down list in the web interface. First example: `@DataSource = 1, 1000, 1a, 4000, Extra, Main, Return required, Shared'` Second example: `@DataSource = ', Apples, Oranges, Ripe Pears, Tangerines'` The second example creates a drop-down list with the first position blank (this displays an empty value until the operator selects another value from the list). ***Restriction:*** *When you add a custom drop-down list (when `UIFieldTypeID = 8`), it is not possible to localize the values for the individual options within the custom drop-down list. (This is in contrast to adding a custom option within a drop-down list included in the standard product: the standard lists allow for customization of any options, including added custom options; whereas drop-down lists that are in entirety custom cannot be localized.)*
@DataSourceDelimiter	Optional (when omitted, the default value is the comma `,`). A single ASCII character (a punctuation character is expected) that does not occur in your data set and is used as a delimiter between values in **@DataSource**. The separator character must be enclosed in single quotation marks. If `UIFieldTypeID` has any value other than `8`, this parameter is ignored.
@Required	Optional (when omitted, the default value is zero). May have the following values: • `0` means that input to the custom field in the web interface is optional, such that the value may be left blank. • `1` means that in the web interface, the custom field is mandatory, and may not be left blank by an operator completing the enclosing tab. This integer value must *not* be surrounded by quotation marks. ***Note:*** *This parameter affects only data input through the web interface of IT Asset Management. It does not have any effect, for example, on data imports using the Business Importer.*
@StringLength	Optional (when omitted, the default value is `256`). Ignored unless the **@UIFieldTypeID** has either of the values `4` (text box, or field) or `5` (text area, multi-line). Specifies the maximum length of the input string in the web interface. The largest permissible string length is 4000 bytes.

@ReadOnly	Optional (when omitted, the default value is zero). May have the following values: • `0` means that the control is read/write, and can be updated in the web interface. • `1` means that the control is read-only, and cannot be changed in the web interface. This value is illegal if `@Required = 1`, and will produce an error when executed.

Localizing Display Names of Custom Properties

Execute the following in SQL Server Management Studio against your FNMSCompliance database, once the custom properties already exist in the database.

Note: *If you are using a cloud-based implementation of IT Asset Management, you do not have direct access to the database, and you cannot "do it yourself". However, you may use this section to understand and specify your* ***requirements****. You can then send a support request to Flexera specifying all the details you require, and your custom property will be added on your behalf.*

Syntax:

```
EXEC dbo.CustomPropertyUpdateDisplayName
        @Name = 'My-Unique-Name',
        @CultureType = 'ISOCultureCode',
        @DisplayNameInPage = 'Prompt value',
        @DisplayNameInReport = 'Column header'
```

where

@Name	Mandatory. The internal name (in code and database) of your custom property, declared when you added it to the database. Never localize this internal name.
@CultureType	Mandatory. A five-character ISO culture name (enclosed in single quotation marks). The permitted values are available at http://msdn.microsoft.com/en-us/goglobal/bb896001.aspx. This is the culture for the localized values in this declaration. Notice that localized values are only displayed in IT Asset Management when your enterprise has installed the corresponding language pack that provides localized values for the factory-supplied controls as well.
@DisplayNameInPage	Mandatory. This is the localized label (enclosed in single quotation marks) displayed on the tab in the web interface for IT Asset Management when the culture setting for the interface matched the one you declare in `CultureType`.
@DisplayNameInReport	Mandatory. The localized label (enclosed in single quotation marks) that is available for you to include in custom reports that display this custom property.

You can repeat this procedure as often as required to define localized display names in all cultures in use in your enterprise.

Tip: *The appearance of the web interface for different locales is controlled in two separate places:*

- *The formatting of dates and numeric values is taken from the settings on your web browser.*
- *To change the language presented in the web interface, go to the* ***My IT Asset Preferences*** *page (****Administration > IT Asset Management Settings > My IT Asset Preferences****). Options for other languages are only present when your enterprise has purchased appropriate language pack options.*

This separation allows for the common case where a single language (such as English) may have different date formats (such as 12/31/19 and 31/12/19) in different parts of the world.

Removing a Custom Property

If you have incorrectly defined a custom property, you can execute one of the following in SQL Server Management Studio against your FNMSCompliance database, referencing the faulty custom property.

Note: *If you are using a cloud-based implementation of IT Asset Management, you do not have direct access to the database, and you cannot "do it yourself". However, you may use this section to understand and specify your* ***requirements****. You can then send a support request to Flexera specifying all the details you require, and your custom property will be added on your behalf.*

Warning: *Removing a custom property (tab, section, or other control) in one of the following ways deletes the control from the web interface of IT Asset Management, removes the custom property from the custom report builder, and optionally can also delete the property from the underlying database. If you specify removal of the data from the database (`@DeleteFromDB = 1`), any custom reports previously built that included the custom property will fail to load until you modify the report definition to remove this custom property.*

Deletion is specific to each individual custom property you have previously declared, and does not cascade down to other items they may contain. For example, suppose you had declared `My.LicenseTab.Charges`, containing `My.LicenseSection.Monthly`, in which there was a custom control `My.Chargeback.Amount`. Subsequently you delete `My.LicenseTab.Charges`. Because the tab disappears, obviously everything it contained is also 'hidden'. However, `My.LicenseSection.Monthly` and `My.Chargeback.Amount` have not been deleted, and are waiting in the database. You can repeat the creation process for these controls using the same name for each (this performs an update), and declaring a new anchor point for them in the web interface (for instance, in this example you might move them into the **Financial** tab). Thereafter these controls reappear in the web interface, and display any data previously saved through the custom controls.

Syntax:

```
EXEC dbo.RemoveTabFromWebUI
        @TargetTypeID = TargetTypeID,
        @Name = 'My-Unique-Name'
```

```
EXEC dbo.RemoveSectionFromWebUI
```

```
        @TargetTypeID = TargetTypeID,
        @Name = 'My-Unique-Name'
```

```
EXEC dbo.RemovePropertyFromWebUI
        @TargetTypeID = TargetTypeID,
        @Name = 'My-Unique-Name',
        @DeleteFromDB = 0
```

where

@TargetTypeID	Mandatory. Integer that identifies the type of object from which you are removing the custom property. For supported objects and their integer equivalents for `TargetTypeID`, see Objects You Can Customize.
@Name	Mandatory. The internal name (in code and database) of the custom property you are removing. You defined this name when you create the custom property.
@DeleteFromDB	Optional (default is zero when omitted). Boolean. When omitted or given the value zero, the custom property (and the data it may contain if it has already been in use) remains in the database, although it is no longer available in the web interface of IT Asset Management. When this parameter is set to `1`, the custom property is removed (along with any stored values) from the database. This parameter is only available when removing properties, as tabs and sections do not have any data component stored in the database.

2

Inventory Beacon Credentials for Other Computers

Your inventory beacons are the front line for remote execution tasks required by IT Asset Management. These tasks may include:

- Adopting a device (that is, automatically installing the FlexNet Inventory Agent on the target device)
- Discovery, including discovery of the device itself on the network, and probing the device to see what other services are available on this device
- Remote inventory gathering from devices where the FlexNet Inventory Agent is not (and perhaps cannot be) installed locally, including:
  - Oracle Database servers, which may require accounts with access to certain tables and views, and accounts to access the Oracle Network Listener on legacy versions
  - ESXi servers and any supervising VirtualCenter servers
  - UNIX-like servers (including UNIX, Linux, and MacOS); and so on.

For all these kinds of tasks, the inventory beacon may need credentials to gain access to target devices.

Each inventory beacon has a local Password Manager that you may use to record the credentials needed by this individual inventory beacon to access the devices it is targeting. Each Password Manager is completely independent, and stores only those additional credentials that *this* inventory beacon requires for remote execution tasks. Further, no Password Manager communicates accounts or passwords to the central application server.

Each Password Manager offers two places for storing credentials:

- By default, all credentials (user name and password pairs) are stored directly in the FlexNet Beacon vault (for technical details, see Password Manager Security Overview).
- When integration with CyberArk has been configured, the individual credentials may be stored in CyberArk, and the FlexNet Beacon vault now contains a reference to that credential, including a query string that uniquely returns exactly that one credential. For an overview of CyberArk configuration, see Configuring CyberArk for Use with Password Manager.

On each inventory beacon, you can create records in the Password Manager using the graphical user interface (GUI) or

using a command-line utility. The Password Manager GUI is documented in the online help, under the inventory beacon topics; and see also the general notes in Password Manager in Operation and Managing Key Pair Authentication. For details of the command-line utility, see Command-Line Updates to Password Manager.

Supported versions for CyberArk integration

The inventory beacons for IT Asset Management 2023 R2.4 support integration with CyberArk Enterprise Password Vault and Credential Provider version 9.8.

Password Manager in Operation

The order of attempting credentials

When an inventory beacon needs a credential for a remote execution task, it tests credentials in the following order until either one credential succeeds or there are no further credentials to test. The credential test consists of an attempt to log into the target device using the credential under test.

Tip: *The storage place of a credential, either in the CyberArk Vault or the local FlexNet Beacon vault, has no effect on the testing order, which is as follows.*

1. The inventory beacon's own administrator credentials (the credentials under which the BeaconEngine service is running on the inventory beacon). This may be helpful for Windows devices in the same domain, if this account has rights on the target device, since this method of authorization does not require managing credentials in Password Manager.
2. If the target device is a VirtualCenter server configured to use SSPI for credentials (which is the default setting), the inventory beacon tries Windows integrated authentication.
3. If there is at least one credential known to Password Manager that has a filter declared, and the filter includes the target device, this matched credential is tried next. If there are multiple filtered credentials where the filters match the target device, they are ordered from the one having most matching filter definitions to the one having least, and tried in that order. (If there are multiple matching definitions that have the same number of filter matches, the order in which they are tried is indeterminate.)
4. Credentials known to Password Manager that do not have any filters declared are tried next, in alphabetical order of the logical name.

If the list of available credentials is exhausted without any match, the discovered device record is marked with an alert in the **Discovered Devices** list in the web interface for IT Asset Management, and for inventory tasks, the failure also appears in the **Inventory Errors on All Discovered Devices** report (**Reporting > Inventory Reports > Inventory Errors on All Discovered Devices**).

Limiting the number of credentials

It is best practice to limit the number of entries in the Password Manager on each inventory beacon, both for performance and to avoid possible inventory failures.

If you have large numbers of credentials in your Password Manager, the performance of remote execution tasks will be adversely affected. It is recommended that you limit the number of credentials in Password Manager to those that are

required, and that you review Password Manager periodically and remove any credentials that are no longer in use. (You can use the **Delete...** button in the Password Manager on each inventory beacon to remove selected credentials.)

Having too many credentials sharing the same account name may cause inventory failures, due to the following logic:

- *Context:* The remote access lockout feature of Microsoft Windows shuts out access to an account for which the number of failed password attempts exceeds a set limit within a time-out period. The limit is defined in `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Parameters\AccountLockout\ MaxDenials`. Once an account is locked out, it will not function for remote execution until the time-out period expires, after which the account is reset and the lockout feature restarts.
- *Issue:* To access each target device, the inventory beacon tries each credential of the appropriate type from Password Manager in turn, until one succeeds or there are no more credentials (of the correct account type) to try. If you store *many* credentials with the same **User** name but different passwords (for example, `SystemsUser/password1`, `SystemsUser/password2`, `SystemsUser/password3`), trying each one in turn on the same device may eventually cause account lockout: if the number of passwords for the same user name is more than the limit for retries on this individual device, the account gets locked out for some time. If the lock-out is triggered, discovery or inventory collection times out during the lock-out period.

To avoid this problem, use any of the following approaches, as may be appropriate for your environment:

- Within the Password Manager, use the **Filter** to specify the device(s) to which individual account name and password pairs apply.
- Set `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Parameters\AccountLockout\MaxDenials` on target devices to be greater than the number of duplicated account names. For example, if you have 20 accounts called `SystemUser` listed in the password store, set `MaxDenials=21`.
- Add the local Administrator account for each target device to the local Password Manager, as this account is not locked out.
- Change the account names on individual managed devices to remove duplication.
- For each duplicate account name in your enterprise, set an identical password, so that only one account name/password entry is required in each Password Manager.

Password Manager Security Overview

Password Manager may operate in either of two modes:

- It may manage and store credentials in the FlexNet Beacon vault, an encrypted area of the registry on the inventory beacon Windows server, as described below.
- Where integration with CyberArk is detected (by the presence of the CyberArk Credential Provider installed on the inventory beacon), Password Manager can request use of credentials already saved in a CyberArk vault. In this mode, Password Manager stores references to the credentials, including query strings that allow for the recovery of the appropriate credential for a given purpose.

When CyberArk is detected, its use may be disabled for a given inventory beacon, in which case storage remains available in the FlexNet Beacon vault.

As well, when CyberArk integration is detected and enabled, individual credentials may be designated for storage either in the FlexNet Beacon vault or in the CyberArk vault. This mix-and-match capability is convenient for mixed

environments, such as one that requires CyberArk credentials for use in production, but allows credentials stored locally in the FlexNet Beacon vault for testing purposes, to reduce administrative overhead.

Tip: *The CyberArk option remains available when either of the following is true:*

- *The CyberArk Credential Provider is detected on the same inventory beacon*
- *There is at least one credential saved in Password Manager on the inventory beacon for which the* ***Vault*** *value is* `CyberArk`.

In either mode, whether storing credentials locally or saving references to credentials available in CyberArk, the FlexNet Beacon vault makes use of the same security technologies, as described here.

Independence

Each Password Manager is completely independent on its own inventory beacon. (Further, no Password Manager communicates accounts or passwords to the central application server, although of course error reporting is centrally available.)

On each inventory beacon, you can create records in Password Manager using a graphical user interface or using a command-line utility. These methods operate independently. For example, even if an operator disables CyberArk integration in the graphical user interface, you can still manage integration with CyberArk (such as adding records of credentials available there) through the command-line interface.

Storage

Credentials entered into the Password Manager for saving the FlexNet Beacon vault are encrypted, Base64 encoded, and stored in the registry on the inventory beacon under `HKLM\SOFTWARE\ManageSoft Corp\ManageSoft\PasswordStore`. Encryption uses a key derived from a primary password. (References to credentials saved in a CyberArk vault are not encrypted, since the security aspect here belongs to CyberArk.)

Initialization

On each inventory beacon, the primary password is created the first time that Password Manager is accessed, using `CryptGenRandom` to generate a 256-bit string. This string is then encoded using Base64 and stored as a local private data object using `LsaStorePrivateData`. This is accessible only to administrators on the individual inventory beacon.

Important: *Private data objects are only as secure as the computer on which they are stored, and any operator with administrator privileges can read them. If regular domain operators are members of the Administrators group on an inventory beacon, they too will be able to view the Password Manager vault on that server. Review your user configuration to ensure that only appropriate operators are members of the Administrators group on any inventory beacon.*

Re-encryption

The following command line tool can be used if it ever becomes necessary to generate a new primary password (or security key):

```
C:\Program Files (x86)\Flexera Software\Inventory Beacon\RemoteExecution\mgspswd
```

```
--recrypt
```

With the `--recrypt` option, this tool decrypts all the credentials in the Password Manager on the inventory beacon where it runs (using the old primary password), generates a new primary password, and re-encrypts all stored credentials/references with the new primary password. For more information about the `mgspwd` utility, see Command-Line Updates to Password Manager.

Operation

In use in the FlexNet Beacon vault, the encryption and key derivation functions rely on the `CryptoAPI` support in the inventory beacon's version of Windows. The process is:

1. The primary password is retrieved using `LsaRetrievePrivateData`.
2. An encryption key is derived from it using `PBKDF2` with `HMAC-SHA-256`, using the AES cypher and a 256-bit key. This uses the default `CryptoAPI` provider of type `PROV_RSA_AES`.
3. The key derivation function generates an additional 128-bit initialization vector.

All ciphers are used in CBC mode with PKCS #5 / RFC 1423 padding.

Configuring CyberArk for Use with Password Manager

For integration with CyberArk, FlexNet Beacon expects to find the CyberArk Credential Provider installed on the inventory beacon.

Note: *In the CyberArk AIM CD image, this is the installer for `Credential Provider`. Do not use the `Application Server Credential Provider`.*

Credential Provider must be installed before FlexNet Beacon can display the data needed to configure CyberArk to recognize the inventory beacon as an authorized requester of credentials. (The check is that the 32-bit `CPasswordSDK.dll` is present in `%windir%\System32` on a 32-bit device, or `%windir%\SysWOW64` on a 64-bit device.)

Once both the CyberArk Credential Provider and (of course) FlexNet Beacon are installed and operational on the inventory beacon server, your CyberArk administrator can register the application with the level of security required by your corporate operating procedures. For complete details, see the CyberArk documentation (such as the *Credential Provider and ACSP Implementation Guide* included with your CyberArk installation); but here is a summary that illustrates the relationships between the data provided by FlexNet Beacon and the configuration points provided by CyberArk.

CyberArk supports multiple processes for configuration, including manual interaction and automated processes. This summary assumes the manual process for clarity. However, as you need to configure this integration independently on each inventory beacon, your CyberArk administrator may well prefer to set up automated processes.

The manual process is most conveniently started from the inventory beacon in question, having the FlexNet Beacon interface open (this requires administrator privileges on the inventory beacon), as well as a web browser that can access your CyberArk implementation.

To configure integration between FlexNet Beacon and CyberArk Application Identity Manager (AIM):

1. Complete the installation of the CyberArk Credential Provider on this inventory beacon, if necessary following instructions provided by CyberArk.

   Do not install the `Application Server Credential Provider`. In the CyberArk AIM CD image, open the folder for `Credential Provider`, and run `setup.exe` from that folder.

   ***Tip:** As part of the installation process, be sure to configure the Credential Provider for access to your CyberArk server hosting the relevant CyberArk Vault.*

2. Use Windows Explorer to validate that `CPasswordSDK.dll` is present in `%windir%\System32` on a 32-bit device, or `%windir%\SysWOW64` on a 64-bit device. If CyberArk Application Identity Manager version 13 or later is being used, also validate that `libeay32.dll` is present in the same directory; otherwise, Password Manager will not display the CyberArk option because `CPasswordSDK.dll` version 13 or later has a dependency on `libeay32.dll`.

   If this is not already the case following installation of the CyberArk Credential Provider on this inventory beacon, you can find a copy of `CPasswordSDK.dll` (this 32-bit version, and not the equivalent 64-bit version) in the *`Credential-Provider-installation-path`*`\ApplicationPasswordSdk` folder. Copy this to the appropriate location for the architecture of your inventory beacon server.

   ***Important:** Do not copy the 64-bit version. The Password Manager on the inventory beacon requires the 32-bit version (even though the inventory beacon user interface, which is built in .NET, appears to recognize the 64-bit version, if you copied that by mistake).*

3. In FlexNet Beacon, navigate to the **Password management** page.

   This page displays the values needed for insertion into CyberArk.

4. In your web browser, log into the CyberArk Password Vault Web Access (PVWA).

5. From the top navigation bar of PVWA, select the **Applications** tab.

6. If this is the first registration of the FlexNet Beacon application, create the base application record in PVWA:

   a. In the top right of the **Applications List** page, click **Add Application**.

   b. From FlexNet Beacon, copy the value of the **Application ID**, and paste it into the **Name** field in the **Add Application** dialog in the PVWA.

      ***Tip:** The default value, `Flexera_FlexNetBeacon`, follows the guidelines in the CyberArk documentation. If your environment requires that you must change this default, edit the `[Registry]\PasswordStore\CyberArkAppId` setting on the inventory beacon. If you change the registry value, restart FlexNet Beacon before copying the new value to the PVWA.*

   c. Complete the remaining details in the **Add Application** dialog, in line with your corporate protocols.

      For **Location**, it is typical to select the `\Applications` folder, but this is not mandatory.

      ***Tip:** It is best practice to not add time restrictions or an expiration date for this application's access to*

the CyberArk Vault. This is because inventory gathering may be scheduled at any time of day, typically after hours when systems are lightly loaded.

d. Click **Add**.

The application is added and displayed in the **Application Details** page.

e. In the **Application Details** page, below the **Authentication** tab, select **Allow extended authentication restrictions**.

This enables you to specify multiple machines, OS users, path values, and hash values for a single application.

f. Under the **Authentication** tab, click **Add**, and from the drop-down list, choose `OS User`.

g. From FlexNet Beacon, copy the value of the **Inventory beacon service account**, paste into the **OS User** field in PVWA, and click **Add**.

h. To test that the configuration is successful, click **Test CyberArk integration...** in the **Password management** page of FlexNet Beacon.

A separate **Test CyberArk Integration** dialog appears.

i. Enter a query for which the credential already exists in CyberArk (depending on how the credential is secured, specifying `Object=`*`accountName`* may be sufficient, provided that the answer is exactly one credential), and click **Test**.

After a moment, the dialog displays the results:

- For success, several attributes of the account in CyberArk (but not, obviously, its password) are displayed.
- For failure, the error message received from CyberArk is displayed unchanged. You should fix the error(s) and repeat the testing until successful.

Tip: *If the Credential Provider service is not running on the inventory beacon, the response waits 30 seconds and then times out.*

For example messages and likely remedial actions, see Typical Errors and Fixes.

The basic application record now exists in CyberArk. This application record can now be referenced by all your inventory beacons that need access to credentials stored in CyberArk. As you register additional inventory beacons in PVWA (hint: click **Search** to populate the application list under the **Applications** tab), validate that:

- The OS user account that runs the inventory beacon engine is unchanged. If a different account has been used on a particular inventory beacon, update the application record in PVWA with the additional OS user.

Of course, this requires logging in to each inventory beacon to check the relevant details on the **Password management** page of FlexNet Beacon there.

7. Register each inventory beacon as a device using the registered application details in PVWA:

a. In the **Application Details** page for your FlexNet Beacon application in PVWA, select the **Allowed Machines** tab.

b. At the top of this tab, click **Add**.

c. In the **Add allowed machine** dialog, enter the host name or fully-qualified domain name for this inventory beacon in the **Address** field.

Tip: *The IPv4 address of the inventory beacon is another method of identification; but beware of using this where dynamic IP address allocations may alter the IP address of your inventory beacons over time.*

Repeat this registration of the machine details for each relevant inventory beacon.

8. Configure the required safe and its memberships:

   a. If it does not already exist, create the safe that will store credentials needed by your inventory beacons (navigate to **Policies > Access Control (Safes)**, and click **Add Safe**). If the safe already exists, select it from the list of safes, and in the bottom right, click **Members**.

   b. In the **Members** tab, click **Add Member**, search for the application you saved, select it in the list of results (the default privilege levels are adequate, and must include **Retrieve accounts**), and click **Add**.

Tip: *Look for the success message at the bottom of this dialog, and then click* ***Close****. Once the dialog is closed, the list of members is updated and displays your application as a member permitted to access this safe.*

9. Ensure that the credentials needed for the remote execution activities of every inventory beacon are recorded as CyberArk "accounts".

   a. In the **Account** tab of PVWA, click **Add Account**.

   b. In the **Add Account** page, in the **Store in Safe** drop-down, select the safe you created for credentials accessed by inventory beacons.

   c. Select the **Device Type** and resulting **Platform Name** for this credential.

   Here are suggested mappings between the **Account type** saved in the FlexNet Beacon Password Manager and these two fields in PVWA:

Account type (FlexNet)	Device Type (PVWA)	Platform Name (PVWA)	Typical Format
`Windows domain account`	`Operating System`	`Windows Domain Account`	`Domain\Account`
`Local account on Windows device`	`Operating System`	`Windows Server Local Accounts` (or `Windows Desktop Local Accounts` if you are targeting desktop computers)	`Account`
`SSH account (password)`	`Operating System`	`Unix via SSH`	`Account`
`SSH account (key pair)`	`Operating System`	`Unix via SSH Keys`	DSA key file

Account type (FlexNet)	Device Type (PVWA)	Platform Name (PVWA)	Typical Format
`Account on VMware ESX server`	`Operating System`	`VMWare ESX Account API`	`Domain\Account` or `Account`
`Account on VMware VirtualCenter`	`Application`	`VMWare vCenter Shared Accounts`	`Domain\Account` or `Account`
`Password for Oracle listener`	`Directory` (This choice does not require a user name for the credential.)	`[None]`	Only requires a password
`Account on Oracle database`	`Database`	`Oracle Database`	`Account`
`Oracle VM management API account`	`Operating System`	`VMWare ESX Account API` or `Unix via SSH` (see note)	`Account`

Note: *The `Oracle VM management API account` is likely to be a local account on Linux, and currently PVWA does not offer a matching platform name. Your CyberArk administrator may perhaps create a custom platform name for your use; or you can use other platform names (such as either of the two suggested above) that provide the correct set of data fields.*

d. Complete the remaining properties for the credential, and click **Save**.

- **Address** may identify the device that requires this credential, either using an IP address, a host name, or a fully-qualified domain name. In cases where this isn't particularly meaningful (such as a Windows domain password), enter free text such as the domain name.
- **Name** must be unique in context (for example, within the safe), and is typically used in query strings. For this reason, you may prefer to provide a custom, simpler name rather than use the one that is automatically generated.
- For `Unix via SSH Keys`, after saving, click **Add SSH Key** (upper right), and provide the required details.

e. Take note of the query string parameters that uniquely identify this credential.

The query strings must be entered in FlexNet Beacon Password Manager, and are used to request each credential from CyberArk. If the query string does not return a single, unique credential, the request fails. Query string elements that may be used include:

- `Safe` — where the credentials for inventory beacon use are stored
- `Address` — the same value you provided for this parameter for the credential, identifying the target device where the credential is to be used
- `Object` — shown in the **Account Details** page in PVWA as the **Name**.

These values should be sufficient to uniquely identify the credential. For other elements possible in a query string, see the CyberArk documentation.

Repeat for as many credentials as required. This completes configuration of CyberArk. The last remaining step is to record the various credentials in Password Manager on the various inventory beacons from which they will be used for remote execution.

10. On each inventory beacon in turn, access Password Manager, and create records of the credentials used from this inventory beacon.

    For details, see the online help for the Password Manager.

Removing CyberArk Integration

It may become necessary to remove CyberArk integration from one or more inventory beacons as organizational structures evolve.

Tip: *To disable the CyberArk integration only temporarily, so that it is easy subsequently to resume integrated operations, open the FlexNet Beacon interface, select the* ***Password management*** *page, and clear the* ***Use CyberArk Credential Provider on this inventory beacon*** *check box. This disables the use of all credentials stored in CyberArk from this inventory beacon, until you re-enable it by selecting this check box again.*

Within the Password Manager interface, the option to select `CyberArk` option for the **Vault** setting remains available as long as either of the following is true:

- CyberArk integration is detected on this inventory beacon (through the files installed on the same device as part of the CyberArk Credential Provider)
- Any credential is saved in Password Manager for which the **Vault** setting is currently `CyberArk`.

This suggests the following best practice for removing CyberArk integration from an inventory beacon. This process must be repeated separately on each affected inventory beacon.

To remove CyberArk integration from an inventory beacon:

1. Log into the FlexNet Beacon interface (administrator privileges required), select the **Password management** page, and click **Launch Password Manager**.
2. Look at the credentials in the **Current credentials** group (on the left side).

   Every credential with a blank value in the **Account** column is either a `Password for Oracle listener` (which requires no account name, regardless of where the credential is stored), or is a credential saved in the CyberArk Vault.
3. Select each of these credentials in turn so that its properties are displayed in the **Editor** group, and check the **Vault** setting for each one.
4. When the **Vault** setting is `CyberArk`, choose either of the following paths, depending on the strategic requirements:
   - If this inventory beacon must still conduct the same remote execution task against the same target device, and therefore still requires a credential, switch the **Vault** setting to FlexNet Beacon, and update the remainder of the (newly displayed) fields with values that are now to be saved locally on this inventory beacon. Click **Save** when you have finished editing the credential.

- Where the selected credential is no longer required in any form, in the **Current credentials** group click **Delete...** to remove the selected reference to CyberArk from the FlexNet Beacon vault, and confirm.

Tip: *If the case is that every credential for this inventory beacon is stored in CyberArk, and none of these is required in future, use* ***Delete All...****.*

5. When all affected credentials are either removed or updated, exit Password Manager.
6. If these credentials were for use *solely* from this inventory beacon, notify your CyberArk administrator that these credentials may now be removed from CyberArk. (However, be alert for credentials still needed by other inventory beacons.)
7. The appropriate person may now uninstall the CyberArk Credential Provider from this inventory beacon.

Since the CyberArk installed files have been removed from the inventory beacon, and there are no credentials remaining in Password Manager that reference CyberArk in the **Vault** setting, the user interface of the Password Manager is different when you open it hereafter. The **Vault** setting no longer appears, and all interactions are with credentials stored locally in the FlexNet Beacon vault.

Typical Errors and Fixes

When testing the integration between an inventory beacon and a CyberArk Vault, the following are the more common errors that may occur, and the kinds of fixes you might investigate.

Tip: *If fetching a credential fails during production, an error is reported in the web interface of IT Asset Management, on the* ***Status*** *tab of the discovered device properties for the target device. The error report includes the entire error message received from CyberArk, as listed below.*

Error	Investigate
`PDKTC006E Failed to connect to provider (Reason=[select timed out], Rc=[-1])`	The Credential Provider installed on the inventory beacon could not connect with the CyberArk AIM installation. Check your network access.

Error	Investigate
APPAP004E Password object matching query [*queryStringUsed*] was not found (Diagnostic Info: 5). Please check that there is a password object that answers your query in the Vault and that both the Provider and the application user have the appropriate permissions needed in order to use the password.	No credential saved in CyberArk matches the query you issued. If this message is received during testing, check the value in the **Enter test query string** field in the **Test CyberArk Integration** dialog. Check this value against the matching properties of the test credential (which must already exist in CyberArk). The upside of this error (and following ones) is that Password Manager on the inventory beacon is communicating successfully with the local Credential Provider, which in turn is communicating with the CyberArk Vault. When you fix the query value, it is likely that the result will be successful.
APPAP227E Too many password objects matching query [*queryStringUsed*] were found: (Safe=*safeName*;Folder=*folderName*;Object=*accountName* and Safe=*safeName*;Folder=*folderName*;Object=*account2Name*) (Diagnostic Info: 41)	The query you issued is not sufficiently specific, and could be answered by more than credential in CyberArk. It is mandatory that each query to CyberArk can be answered by exactly one credential. If this message is received during testing, improve the value in the **Enter test query string** field in the **Test CyberArk Integration** dialog (typically by adding another parameter, such as the `Object` value that specifically identifies the CyberArk account name).
APPAP133E Failed to verify application authentication data: OSUser "*userName*" is unauthorized	The requested credential is secured by the requesting application *and* the username running that application on the inventory beacon, and the current username does not match (any of) the one(s) registered in CyberArk. Either switch to the correct username running the `BeaconEngine.exe` file on the inventory beacon, or update the OSUser names listed in CyberArk.

Error	Investigate
`APPAP133E Failed to verify application authentication data: Path "executablePath" is unauthorized`	The requested credential is secured by the requesting application *and* the file path where the executable is running on the inventory beacon, and the current file path does not match (any of) the one(s) registered in CyberArk. Most commonly you need to update the file paths listed in CyberArk for the application (assuming that there are different installation paths on different inventory beacons).
`APPAP133E Failed to verify application authentication data: Hash "executableHash" is unauthorized`	The requested credential is secured by the requesting application *and* the hash of the executable on the inventory beacon, and the run-time hash of the current executable does not match (any of) the one(s) registered in CyberArk. Most commonly this happens after a version upgrade of the FlexNet Beacon code on the inventory beacon, and you need to record the new hash in CyberArk for the latest version of the executable.

Managing Key Pair Authentication

Private-public key pair authentication for SSH can be more secure than password authentication, although there are some general guidelines you should follow to manage your key pair authentication:

- Keep the private key private. Do not store it in a public location.
- Always save it with a well-chosen passphrase.
- Be aware that although Password Manager, when using the default FlexNet Beacon vault, keeps a duplicate of the private key file (in an encrypted form), the original is still required if you need to reconfigure Password Manager.

As an additional security step, OpenSSH offers a `from` option that allows you to enter extra details on the public key, limiting the hosts for which the public key will work. Notice that this may render the target device inaccessible if the network configuration of the inventory beacon changes. To mitigate this, you can include additional hosts, or create another private-public key pair to allow access in this circumstance. See the OpenSSH documentation for further details.

Testing key pair credentials

If you want to test SSH credentials, you need two programs: an SSH client and an SSH agent. SSH clients attempt to obtain key pair values from the SSH agent. If authentication fails, the client will test authentication by prompting for a password.

Table 6: Suitable test programs for OpenSSH and PuTTY

Key format	SSH client	SSH agent	Testing notes
OpenSSH	`ssh`	`ssh-agent`	Use `ssh` with high verbosity to see which methods of authentication are enabled in the authentication process.
PuTTY	`PuTTY.exe`	`Pageant`	You can adjust settings in `PuTTY.exe` to enable or disable the `Pageant` and `keyboard-interactive` (password) authentication types during authentication tests.

Command-Line Updates to Password Manager

The `mgspswd` command-line utility allows rapid updating of the Password Manager on an individual inventory beacon. It is installed on each inventory beacon in *`InstallDir`*`\RemoteExecution` (by default `C:\Program Files\Flexera Software\Inventory Beacon\RemoteExecution\mgspswd.exe`). It is an alternative to the GUI presentation of the Password Manager.

This utility supports either:

- Storing credentials in a local vault (encrypted within the Windows registry), and recalling them on demand from there
- Storing a query string that can uniquely access one existing credential saved in a CyberArk Vault, where integration with CyberArk has been detected (by the presence of the CyberArk Credential Provider installed on the inventory beacon). In this case, you may switch between storage types for individual credentials: for example, you can reference CyberArk for production passwords, but use the FlexNet Beacon vault to store credentials for test environments that have lower administrative overheads.

Syntax:

```
mgspswd.exe --add  logical-name  [options...]
mgspswd.exe --delete  logical-name
mgspswd.exe --help
mgspswd.exe --list [ logical-name]
mgspswd.exe --matches  [options...]
mgspswd.exe --recrypt
mgspswd.exe --reset
```

where *`logical-name`* is the friendly name given to the current credential (name/password pair).

For some credential types, certain parameters can be specified in alternate ways. Parameters for which this occurs are:

- Domain: the Windows domain within which the specified credential applies. This is not available as a separate parameter in the command line. You may insert the domain into the account name parameter, using the normal format of `domainName\sampleUser`.
- Password: while this value *can* be provided in the command line, it will be visible there in plain text. If omitted as a

command-line parameter, it can be entered interactively, in which case the password characters are masked.

- Other mandatory parameters (such as account and the logical name, when saving to the FlexNet Beacon vault) also prompt interactively for values if they are omitted from the command line.

The parameters and options available in the command line include the following (in alphabetical order):

Options	Notes
`--account` *`account-name`*	Specify the account name (often called the user name) to use. This parameter is only used with the `--add` parameter, and only when `--vault` is not specified as `CyberArk`. In its simplest form, this parameter specifies only the account name (or user name). Three compound formats are also supported: • To specify a domain account within a particular domain (including for an account that is `--type WindowsDomain`), you may identify the domain using the common backslash-separated format, such as *`domainName`*`\`*`sampleUser`*. • To specify the *`sampleUser`* account name for Windows computers that are *not* part of a domain, use the string literal "localhost" by entering `localhost\`*`sampleUser`*. • You can include the string variable `$(MachineName)` as part of the account name (followed by a backslash separator). At run-time, the name of the computer on which remote execution is being performed will replace `$(MachineName)`. For example: `--account $(MachineName)\Administrator`

Options	Notes
`--add logical-name [options...]`	Add a new credential (account/password pair) with the specified logical name, account name, and password. • If you repeat the command line with the `--add` parameter and a *`logical-name`* that already exists in Password Manager, the existing record is updated. • If you omit the account name or password parameters, and they are required, the utility prompts for the missing parameters. (Be aware for Oracle listener access, only a password is required and is mandatory, and there is no account name.)  ***Tip:*** *If the logical name includes spaces, enclose it in double quotation marks.* Example: `mgspswd --add "Local admin account" --account $(MachineName)\Administrator` Since the `--password` option is omitted, the utility prompts for the password value, masking the characters as they are entered. This example presumes the *unusual* situation of using a common administrator password across a range of devices. A more common example might specify an exact match for a particular device name, such as: `mgspswd --add "myDevice admin" --account myDevice\Administrator`  ***Note:*** *When the parameter --vault CyberArk is included in the command line, the command is adding to Password Manager a reference to an existing credential saved in an appropriate CyberArk vault and safe. Commands from the inventory beacon cannot change the content of CyberArk. The saved reference allows Password Manager to request the appropriate credential from CyberArk at an appropriate time.*
`--cyberark-query ["]query-string["]`	This parameter is only used with the `--add` parameter, and only when the parameter `--vault CyberArk` is included in the command line. Specify the exact query string expected by CyberArk for it to return the required credential. Of course, the credential itself (account name and password pair) must already exist in the appropriate CyberArk vault and safe (the vault is specified when CyberArk integration is first configured, and the safe may optionally be specified as part of the query string). If the query string contains any white space, it should be enclosed in double quotation marks (otherwise, these are optional). Details of the query string are specific to your implementation of CyberArk, and must be obtained from your CyberArk administrator. (See also `privilege-cyberark-query`.)
`--delete logical name`	Removes the credential (account and password record) with the specified logical name from Password Manager. Notice that when the vault setting is for the default (omitted, or set to `FlexNetBeacon`), the credentials themselves are removed from the FlexNet Beacon vault; but when the parameter `--vault CyberArk` is included in the command line, the *reference* to the CyberArk record (including the query string) is removed, but the credential itself (account name and password pair) is *not* removed from CyberArk. Removal of credentials from CyberArk must be performed by a CyberArk administrator.

Options	Notes
`--filter-dnsdomains` *list*	If this credential should only be used for a limited set of target devices, you can specify the DNS domains of affected managed devices here as a comma-separated list of domain names. If multiple filters are specified, target devices that match any of the specified criteria will use the credentials. Credentials matched through a filter are tried before unfiltered credentials. This option is only used with the `--add` or `--match` parameters.
`--filter-dnsnames` *list*	If this credential should only be used for a limited set of target devices, you can specify the DNS names of target devices as a comma-separated list of names. If multiple filters are specified, target devices that match any of the specified criteria will use the credentials. Credentials matched through a filter are tried before unfiltered credentials. This option is only used with the `--add` or `--match` parameters.
`--filter-ipaddresses` *list*	If this credential should only be used for a limited set of target devices, you can specify the IPv4 addresses of target devices as a comma-separated list of addresses. If multiple filters are specified, target devices that match any of the specified criteria will use the credentials. Credentials matched through a filter are tried before unfiltered credentials. This option is only used with the `--add` or `--match` parameters.
`--filter-macaddresses` *list*	If this credential should only be used for a limited set of target devices, you can specify the MAC addresses of target devices as a comma-separated list of addresses. Both the Windows and UNIX formats are valid. Example: `00:01:b0:c4:e6:10,00-AF-F7-CD-F9-10` If multiple filters are specified, target devices that match any of the specified criteria will use the credentials. Credentials matched through a filter are tried before unfiltered credentials. This option is only used with the `--add` or `--match` parameters.
`--filter-names` *list*	If this credential should only be used for a limited set of target devices, you can specify the device names of target devices here as a comma-separated list. For example: `accounts-laptop,finance-desktop` If multiple filters are specified, target devices that match any of the specified criteria will use the credentials. Credentials matched through a filter are tried before unfiltered credentials. This option is only used with the `--add` or `--match` parameters.
`--filter-netbiosdomains` *list*	If this credential should only be used for a limited set of target devices, you can specify the NetBIOS domain names of target devices as a comma-separated list of domain names. If multiple filters are specified, target devices that match any of the specified criteria will use the credentials. Credentials matched through a filter are tried before unfiltered credentials. This option is only used with the `--add` or `--match` parameters.

Options	Notes
`--filter-oracleservicenames` *`list`*	This parameter only applies to accounts of type `OracleDatabase`. If this credential should only be used for a limited set of target devices, you can specify a comma-separated list of the Oracle service names to which the credential applies. Use only with the `OracleDatabase` and `OracleListener` account types. For example, `ORA001,TestORA` If multiple filters are specified, target devices that match any of the specified criteria will use the credentials. Credentials matched through a filter are tried before unfiltered credentials. This option is only used with the `--add` or `--match` parameters.  ***Tip:*** *Oracle names may match on individual parts of the service name. It may be helpful to specify the fully qualified service name in the Oracle service names filter to avoid unintentional matches. To use a filter to match service names with multiple suffixes, you can specify each fully qualified service name in the filter, separated by commas.*
`--help`	Displays a list of parameters.
`--list`	Lists all credentials within the password store. All elements are shown (logical name, account name, and password) with the password displayed in masking characters. If the optional *`logical-name`* for a credential is supplied, the utility displays the credential with the specified logical name.
`--matches`	Identifies all the credentials in the Password Manager that match (and therefore may be applied to) a device. To narrow the specification, you may add the --type option with one valid value, and any of the filter options (also described in this listing): • --filter-names • --filter-dnsnames • --filter-dnsdomains • --filter-netbiosdomains • --filter-ipaddresses • --filter-macaddresses • --filter-oracleservicenames.
`--password` *`password`*	Specify the password to use. Only permitted for the default FlexNet Beacon vault (that is, omit when --vault CyberArk is specified). Note that if you do not wish to see the password echoed in plain text on the command line, you may omit this parameter, and the utility will prompt for it, and mask it as it is entered. Passwords are required for the `OracleListener` account type, and are optional for all other account types. This option is only used with the `--add` parameter.

Options	Notes
`--privatekeyfile` *path*	The name and location of a source file containing the private key for SSH, for use with the default FlexNet Beacon vault. (This parameter is not relevant when `--vault CyberArk`, since CyberArk then owns management of the public/private key pair, and simply returns the private key on demand through the appropriate query string.) Using the default FlexNet Beacon vault, the private key file is read from the specified path, and added to the Password Manager. The private key can be in the OpenSSH project's format (generated using `ssh-keygen`) or the PuTTY format (generated using `PuTTYgen.exe`). The corresponding public key must be in place on the target device before SSH login using a private-public key pair. If you are using OpenSSH on target devices, the public key is expected in `~/.ssh/authorized_keys`. Use `mgspswd.exe --list` *`logical-name`* to obtain the public key to add to `~/.ssh/authorized_keys`. Other SSH implementations may require that the public key be stored elsewhere. This option is only used: • With the `--add` parameter • For the `SSHKeyPair` type of credential • When the vault is *not* CyberArk.
`--privilege-cyberark-query ["]`*`query-string`*`["]`	Specify the exact query string expected by CyberArk for it to return the credential required to escalate privileges on the target device. Of course, the escalation credential itself (account name and password pair) must already exist as a separate credential in the appropriate CyberArk vault and safe. If the query string contains any white space, it should be enclosed in double quotation marks (otherwise, these are optional). (See also `cyberark-query`.) This option is only used: • With the `--add` parameter • When the parameter `--vault CyberArk` is included in the command line • For the SSHPassword or `SSHKeyPair` types of credential.

Options	Notes
`--privilege-password` *password*	You can specify that login should be attempted with elevated privileges on target devices running UNIX-like operating systems. This is the password used to gain those elevated (`root`) privileges. (See also `--privilege-password-prompt` and `privilege-prefix`, which are used in conjunction with this.)  ***Tip:*** *If `sudo` on the target device(s) is configured to allow escalation of privileges without requiring an interactive password, just omit this parameter.* This option is only used: • With the `--add` parameter • For the `SSHPassword` or `SSHKeyPair` types of credential • When the vault is *not* CyberArk.

Options	Notes
`--privilege-password-prompt` *text*	For UNIX-like devices on which login should be attempted using elevated privileges, specify the exact prompt for which FlexNet Beacon should wait before issuing the value of the `--privilege-password` parameter.  ***Tip:*** *The `sudo` tool typically issues a prompt similar to this:* `[sudo] password for userName:` *You could enter this entire value, since you know the* ***User*** *name for this login, in the `--privilege-password-prompt` parameter; but (assuming that this credential is reused across multiple servers) this approach is at risk because of variations across different versions of UNIX-like operating systems. A risk-free alternative is to use the following special settings:* 1. *Supply the elevation command with an option to declare a specialized password:* `--privilege-prefix "sudo -p flxpwd:"` *The `-p` option instructs `sudo` to issue the specified prompt (for a password) when it is invoked by the FlexNet Beacon engine.* 2. *for this parameter, enter* `--privilege-password-prompt "flxpwd:"` *(or exactly the prompt value you specified in the field described above).* 3. *Be sure to also specify the `--privilege-password password` parameter.* *When invoked by the FlexNet Beacon engine, `sudo` now issues a known prompt, which in turn is recognized by the FlexNet Beacon engine, and inventory collection can proceed.* This `--privilege-password-prompt` option is only used: • With the `--add` parameter • For the `SSHPassword` or `SSHKeyPair` types of credential • When the vault is *not* CyberArk.
`--privilege-prefix` *prefix*	For UNIX-like devices on which login should be attempted using elevated privileges, specify the valid privilege elevation command (such as `sudo` or `priv`) here. This option is ignored if the credential matches any Windows devices. This option is only used: • With the `--add` parameter • For the `SSHPassword` or `SSHKeyPair` types of credential • When the vault is *not* CyberArk.

Options	Notes
`--recrypt`	Decrypts all passwords in the Password Manager vault using the current primary password (security key), replaces the primary password with a new one, and re-encrypts all the passwords in the Password Manager vault with the new security key, using the strongest available algorithm. For more information, see Password Manager Security Overview.
`--reset`	Clears the Password Manager vault on this inventory beacon, and resets the internal security key.
`--type` *type*	The credential type. This must be one of: • `OracleDatabase`: to connect to an Oracle Database instance • `OracleListener`: to connect to a server running Oracle listener services (for this type, a password is mandatory and no account name can be defined) • `OracleVMManagerApiAccess`: an account that can access the API for Oracle VM Manager • `SSHKeyPair`: to connect to managed devices using SSH, when SSH on the target devices is configured to require a key-value pair for login • `SSHPassword`: to connect to managed devices using SSH, when SSH on the target devices is configured to require a password for login • `VMwareESX`: to connect to a VMware ESX server • `VMwareVirtualCenter`: to connect to a VMware Virtual Center server • `WindowsDomain`: when tasks being remotely executed on Windows devices should run as a domain user • `WindowsLocal`: when tasks being remotely executed on Windows devices should run as a local computer user. This option is only used with the `--add` or `--matches` parameters.
`--vault` *vault-type*	The kind of vault used for storing the credentials. May be omitted when CyberArk integration has not been detected on this inventory beacon (in which case the FlexNet Beacon vault is used). When CyberArk is available, the *vault-type* must be one of: • `CyberArk` (the normal choice in a production environment where CyberArk has been detected) • `FlexNetBeacon` (note: no white space). ***Tip:*** *Values are case insensitive.* This option is only used with the `--add` or `--matches` parameters.

Example 1: Set a new record to retrieve the credential for a Windows domain (account and password) saved in CyberArk (this command is entered all on one line):

```
mgspswd --add WDomain07 --vault CyberArk --type WindowsDomain
--cyberark-query Safe=PasswordSafe;Folder=Root\
Applications;Object=WinDomain07-DLPW
```

Because the domain name is a property of the credential returned from CyberArk, no specification of the domain is needed in this command line, and you rely on naming conventions in the *logical-name* to track the purpose of this credential.

Example 2: Specify the same credential saved in the FlexNet Beacon vault (when CyberArk integration has not been detected on this inventory beacon):

```
mgspswd --add WDomain07 --type WindowsDomain --account ourDomain\winSvcAcct
--password qwerty1
```

Notice that you cannot specify the domain name as a distinct command line parameter, but you can specify it as part of the account data as shown. You may choose to enter the password interactively (with character masking) by omitting it from the command line.

Example 3: Using the default FlexNet Beacon vault (when CyberArk integration has not been detected on this inventory beacon), save a local account on a target device named `myDevice`:

```
mgspswd --add myDevicePW --type WindowsLocal --account svcUser --password qwerty!
--filter-names myDevice
```

Each call to `mgspswd` with the --add parameter (and using the default vault) stores on the inventory beacon:

- A logical name for the account/password pair (when this is a new logical name, a new entry is created; and when it is an existing logical name, the current entry is updated)
- Optional filters to restrict the use of this credential to specific target devices (in this example, the `filter-names` parameter ensures that this credential is attempted only on the device of the matching device name)
- An account name (or username) on the target device
- The account password on the target device, encrypted using a private key unique to each inventory beacon, with the private key stored as a private data object on the inventory beacon. The private key is automatically initialized with the first save to the Password Manager on each inventory beacon.

Example 4: Register the same local credentials, saved in CyberArk, for use on the target device named `myDevice`:

```
mgspswd --add myDevicePW --type WindowsLocal --filter-names myDevice --vault
CyberArk
--cyberark-query Safe=PasswordSafe;Folder=Root\Applications;Object=LocPW-myDevice
```

In this case, the account name (username) and password are already saved within CyberArk, and we supply the query string that returns this credential for use. Notice that the same filter now means that the inventory beacon only attempts to retrieve this credential from CyberArk when it is targeting the matching device.

Notes

Tip: *For Windows-based computers, use `net use \\machineName\ipc$` to test that login credentials work before*

adding them to Password Manager.

3

Customizing Dashboards for IT Asset Management

IT Asset Management includes Flexera Analytics, a technology that allows you to prepare reports and to customize dashboards, either for your enterprise or for personal use. Each dashboard may contain one or many widgets, which graphically represent selected aspects of the underlying information saved in the system.

The system ships with a library of widgets pre-configured with useful data. Any widget you like can be 'pinned' in the personal pin list of widgets that you prefer, and then included in custom dashboards. For each one, you can customize the data visualization, as well as what data is displayed.

This chapter provides a brief introduction to Flexera Analytics, how you can create and customize dashboards and control the way the data is updated.

Launching Flexera Analytics

Accessing Flexera Analytics requires that the operator is assigned to a role that includes the `Analytics User` privilege. To check, go to the **IT Asset Accounts** page (**Administration > IT Asset Management Settings > IT Asset Accounts**), select the **Roles** tab, and identify a role (or more than one if required) that allows the `Analytics User` privilege in the **Business reporting portal** section. By default, 60 operators may be assigned to roles that grant this privilege. (If you need more, please contact your Flexera representative.) Switch to the **All Accounts** tab and check the **Role** column for the operators of interest to ensure that they are assigned to a role conveying this privilege.

These operators can directly access Flexera Analytics within IT Asset Management.

To launchFlexera Analytics within IT Asset Management:

1. Go to any of the following dashboards by searching for the dashboard names or by hovering over **Dashboards** and select the dashboard you want to open:

   - **Software Assets**—This dashboard displays widgets related to managing your software assets, including information about products at risk, unlicensed installations, unauthorized installations, and licenses at risk of over-consumption.

   - **Hardware Assets**—This dashboard provides data to help you manage your hardware assets, including

information about assets by type, new assets, and inventory devices with duplicate names.

- **My Analytics Home**—You can set your home page to display one of the existing dashboards, or you can set the `My Analytics home` page to display a customized dashboard. (By default, `My Analytics home` is set to display the **Software Assets** dashboard.)

Flexera Analytics Data

The data used by Flexera Analytics is automatically copied from the compliance database to the data warehouse database. There are 3 ways that updating this data is triggered.

- The `Inventory import and license reconcile` as scheduled in IT Asset Management Settings **Defaults** > **Inventory**. At the completion of this task, a separate task will be triggered to copy the appropriate data from the compliance database to the data warehouse database.
- The `Data warehouse export` scheduled task.
- **Partial exports**. Creating, editing and deleting of some objects in the interface for IT Asset Management will trigger the partial copying of data from the compliance database to the data warehouse database. By default IT Asset Management will poll for changes every 60 seconds and if a change, or changes to data have been detected in the compliance database, a partial export will be triggered to copy those changes to data warehouse database; unless a partial export has been executed within the last 15 minutes.

Using the Widget Library

Flexera Analytics includes a library of commonly used widgets that can be used on customized dashboards. Pin or save the widgets to your **My Pins** list, and reference these widgets later when creating or modifying a dashboard.

To open the Widget Library and pin widgets:

1. In the side navigation bar, click **Team Content**, then click **Flexera Boards**
2. Click **Widget Library**.
3. To pin a dashboard widget, click the white background (not on a graph or other data representation) of the widget to select it, and click the Pin icon (thumbtack) that appears beside the widget.

   A message appears briefly at the top of the page, confirming that the item has been added to your **My Pins** list. You may now use your pinned items when creating a new dashboard (see Creating a Dashboard).

Creating a Dashboard

Flexera Analytics allows you to create dashboards that display real-time data about applications, software licenses, assets, and inventory.

To create a dashboard:

1. In the side toolbar, click **New** (with the plus sign), then click **Dashboard**.

   This displays the **Select a template** page

2. From the **Dashboard** pane, select a page-layout type, then select a layout template from the right pane, and click **OK** (or you can double-click the template).

   A blank template appears on the screen.

3. If necessary, click the **Sources** icon from the side toolbar to display the **Selected sources** fly-out.

4. Click the **Add a source** icon (plus sign) to add a source.

5. In the dialog that appears, ensure that **Team Content** is selected on the left, and click **Flexera Data Models**.

6. Ignore any text saying that the folder is empty, and click **Open**.

   The Flexera Data Models folder (containing the `Flexera Relational Model` and the `Flexera Dimensional Model`) appears in the **Selected sources** pane.

7. Click the **Pins** icon (upper-right, in the toolbar) to display previously saved or pinned widgets.

8. To add a pinned widget, select the widget and click **Add** (at the bottom of the **My pins** panel), or drag and drop the widget onto the template.

   Tip: *You can also drag and drop data model attributes to a widget to customize the way data is displayed on the dashboard. See Customizing a Dashboard for more details.*

9. To save your new dashboard:

   a. Click the disk icon (upper left, in the toolbar) to display the **Save as** dialog.

   b. On the left side, choose either:

      - **Team content** to save your dashboard where others can access it
      - **My content** to save your dashboard in a private area.

   c. In the **Save as** field, give the dashboard a meaningful name.

   d. Click **Save**.

Customizing a Dashboard

Once you add a widget to a dashboard, Flexera Analytics provides functionality to customize it. You can filter, sort, and change appearance for a widget. Each widget has its own set of parameters that you can customize. (See the Analytics online help for more about using the general user interface.)

In addition, some widgets allow you to affect the data presented in other widgets. For example, the **Publisher filter** widget enables you to filter which publishers' data will appear in other widgets that report on certain application measures. The following procedure provides an example of how to customize a widget.

To customize widgets on a dashboard:

1. Ensure you are in edit mode (click the pencil icon in the upper-left corner).
2. Select the widget by clicking its background, then click the arrow in the upper-right corner of the box.

   The box expands showing additional parameters to customize.
3. Click an item on the right side (such as **Categories** or **Value**) to expose a fly-out with icons for additional control. Use these icons to filter, sort, summarize, or change the way information appears in the widget.

   For example, here's how you can customize the **Publisher filter** widget, which then acts on the data available in other widgets. Still in edit mode:

   a. Select the **Publisher filter** widget, and click the arrow in the upper-right corner of the box that outlines the widget.

      The box expands showing additional parameters to customize.

   b. Select the **Publisher** parameter, then choose one or more of the following menu items:

      - **Sort** icon—This menu item allows you to sort the list of publishers in an ascending or descending order if desired.
      - **Filter** icon—This menu item allows you to select or clear publishers that appear in a filtered list. There are five publishers in the filtered list by default. If you limit the list to one publisher, you must change the visualization type to **Hierarchy**. Use the **Visualization types** icon in the lower-right corner to do this.
      - **Top or Bottom**—This menu item allows you to maintain an extensive filtered list but only report on the top five/ten or bottom five/ten values.

      You may also add attributes from the data model to add additional filtering to this widget. For example, you might continue as follows.

   c. In the side toolbar, ensure that the **Flexera Data Models** are visible in the `Sources` fly-out, and click the arrow-head beside **Flexera Dimensional Model** to expand it and display its members.

   d. Similarly, expand **Applications**, then expand **Application Classification**, and then drag the **Application Classification** attribute to the **Local Filters** field on the right of the expanded widget.

   e. Use the menu to select the specific type of application you want to include. The filter excludes those that you do *not* select.

   f. To edit the title of a widget, click its background to expose the menu of icons, and choose the pencil (`Edit the title`). Type your replacement title, and click elsewhere to finish.

   g. When you are satisfied with all your changes, save them by clicking the arrow in the upper-right corner of the box enclosing the expanded widget, and then click the diskette icon top left (`Save`).

4

Importing Inventory Spreadsheets and CSV Files

Among many supported methods of importing software and hardware inventory details, IT Asset Management supports the import of inventory information in either comma-separated value (`.csv`) files, or Microsoft Excel spreadsheets (`.xlsx` files). To differentiate these from other imports of spreadsheet information (such as for purchases, enterprise groups, and user assignments), these are called 'inventory spreadsheets', a convenient term covering both file formats (unless specifically excepted).

Inventory spreadsheets can be imported in either of two ways:

- A 'one-off' import through the web interface of IT Asset Management
- A repeatable, scheduled import through an inventory beacon.

This chapter covers the processes for both kinds of imports of inventory spreadsheets.

The formats of these imported files are fixed, and defined by downloadable templates. The documentation of each of these templates, and the mapping of all the spreadsheet columns to the compliance database, is included in the *Inventory Spreadsheet Templates* topic in *IT Asset Management Schema Reference*.

Overview of Inventory Spreadsheets

You can upload inventory data from spreadsheet files to IT Asset Management in two ways:

- A one-off, one-time upload that you might need to do for workflow validation, demonstration or proof-of-concept purposes.
  The data will be saved on the application server and created as a unique connection. Once data from this connection is processed, the connection is disabled (and cannot be re-enabled). Inventory from this connection ages, and eventually appears in the **Out-Of-Date Inventory** list. To clear the stale data, delete the connection: everything imported (only) from that connection is then removed from the operations databases.
- Ongoing import of spreadsheet inventory data.
  To set up this workflow, you must use an inventory beacon. This workflow allows scheduled, repeated imports, and data updates, in just the same way as regular inventory imports from other (non-spreadsheet) data sources.

Tip: *Importing `.xlsx` files requires that you have installed a 32-bit version of Microsoft Access Database Engine on the inventory beacon that performs the import. (This requirement does not apply to `.csv` files.)*

Remember: *The complete set of inventory data should be uploaded whenever updating the data previously imported from a pre-existing spreadsheet. That is, all original rows, along with the new rows of inventory data, get imported every time the data has to be updated. As with all other inventory imports, records that disappear from the source connection (in this case, your spreadsheets) are automatically removed from the operations databases to maintain synchronization with the data source.*

If you need to collect inventory from a source to which IT Asset Management cannot directly connect (for example, a source inaccessible for security or any other reasons), you can export this software, hardware or virtualization data from your source into a comma-separated value (`.csv`) or Excel (`.xlsx`) file. This inventory spreadsheet acts as an intermediate file for data upload through an inventory beacon. For repeated use, you can use custom, in-house scripts to populate these spreadsheets with current data, saving them to your chosen upload folder. You can then schedule regular imports from your upload folder to keep the data current.

The following inventory types are supported:

- Computers and VMs
- CAL usage inventory, called access evidence
- Installation evidence
- File evidence
- Windows Management Instrumentation (WMI) evidence
- Oracle evidence
- Virtual machine (VM) pool data
- Cluster evidence
- Cluster group data
- Cluster host affinity rule data
- File evidence and installer evidence for remote access (use these templates for lists of all users who can access a cloud-based service, or virtual desktops and applications).

Each type of inventory has a fixed file format, and you can access the corresponding inventory templates in either of two ways:

- You can download them from the web interface of IT Asset Management
- You can open them from an inventory beacon, where local copies are automatically synchronized with the central application server.

You can then populate your chosen templates with your inventory data, and import the completed files back into IT Asset Management.

Important: *Within your spreadsheets, the number of columns, the column names, and the column order cannot be modified from those included in the template files supplied for each data type. Any such difference between the*

corresponding template and your spreadsheet results in an import failure. This also means that your spreadsheets must correspond with the current versions of the appropriate templates. From time to time, the templates are updated with additional columns or other changes to support new functionality. Updated templates are always distributed with the current release of IT Asset Management, matching the data import requirements for that release. The changes are announced in the Features by Release documentation for the corresponding release, as an alert that existing spreadsheets need to be upgraded to match the latest templates. If you upgrade IT Asset Management to a release including changed templates, and then attempt an inventory spreadsheet import based on an outdated template, the import fails. For example, for a mismatch in the expected number of columns, the error message looks similar to this:

```
Import failed. Error: Malformed spreadsheet file detected: Header row
contains unexpected number of columns: xx. Expecting yy columns in file
```

*Any such errors are displayed in the **Last completed import** section of the **Inventory Data** tab of the **Data Imports** page (**Data Collection > IT Assets Inventory Tasks > Data Imports**).*

One-Off Import of an Inventory Spreadsheet

You can manually upload your inventory data from a spreadsheet to IT Asset Management.

The **Inventory Data One-Off Upload** page (accessible through the **Inventory Data** tab of the **Data Imports** page (**Data Collection > IT Assets Inventory Tasks > Data Imports**)) is for a single import of inventory data from spreadsheets. If you need to repeat the inventory data import (for example to make a data correction), you must first delete the previously set-up connection (the corresponding server-side copy of the spreadsheet is automatically deleted as well). For details, see Deleting Spreadsheet Inventory Data from the Database.

Each one-off upload creates a separate import connection. This is true even if two uploads have an identical name: updates are not possible through the web interface, and two uploads of the same name produce two identically-named connections, functioning separately.

***Important:** Do not allow two or more one-time connections for inventory spreadsheets to exist at the same time, even with different import names. Data from every upload is persisted on the central application server, and is imported afresh from the central spreadsheet copies into the operations databases for every inventory import and license calculation. Having multiple connections to spreadsheets that contain the same computers (for example, from the mandatory `Computer` spreadsheet, reflected in lists of inventory devices) can cause data 'toggling' between imported values, based on the which connection for spreadsheet data was most recently processed. Therefore you must delete the previous one-off upload connection before uploading a newer batch of inventory data.*

Scheduling regular imports of inventory spreadsheets is not supported through the web interface: it is a one-time connection. Once data from this connection is processed, the connection is disabled (and cannot be re-enabled). Inventory from this connection ages, and eventually appears in the **Out-Of-Date Inventory** list. To clear the stale data, delete the connection: everything imported (only) from that connection is then removed from the operations databases. (In contrast, you can arrange regularly scheduled spreadsheet imports through an inventory beacon, as described in Setting Up Scheduled Imports of Inventory from Spreadsheets.)

***Tip:** One-off import of an inventory spreadsheet in the `.xlsx` file format requires that you have installed a 32-bit version of Microsoft Access Database Engine on your central batch server (or central server fulfilling that function). In a*

multi-server implementation, the import request is received by the web application server, placed in the MSMQ message queue, and then executed by the batch server — which is why the driver must be on the batch server. (This requirement does not apply to `.csv` files.)

The data from an individual spreadsheet file may affect several database tables in IT Asset Management. For more details about the template's column names and their corresponding database fields, see the corresponding section in *IT Asset Management Schema Reference*.

***Remember:** Within your spreadsheets, the column names and column order cannot be modified from those supplied in the template files. Any such change results in an import failure. (Here, for a one-off import, you may rename the spreadsheet files themselves, since their purpose is made clear by the field through which you identify and upload each spreadsheet. In contrast, when the same templates are used on an inventory beacon for scheduled uploads, the file names as well as the column names and column order must all be maintained.)*

To perform a one-off upload of an inventory spreadsheet:

1. Go to the **Data Imports** page (**Data Collection > IT Assets Inventory Tasks > Data Imports**).
2. Click the **Inventory Data** tab, and ensure that the inventory data **Name** which is marked as `Primary` has a **Task status** of `Completed`.

   At least the first occurrence of the primary inventory import must have completed successfully before any secondary source (including the one-off inventory spreadsheet) can be imported. If you attempt a one-off import of an inventory spreadsheet before the first successful primary import, it results in an error `Inventory import failed because data has not been imported from the primary data source`. This is because of the rules for data merging, explained in more detail in Making a Data Source Connection the Primary One. By default, the primary connection is the internal inventory connection, named **IT Asset Management** in this list. If it has not yet completed, a member of the `Administrator` role can run an import and compliance calculation manually by going to the **Reconcile** page (**Data Collection > Process Data > Reconcile**). Be sure to select **Update inventory for reconciliation** (available only to administrators) before clicking **Reconcile**.
3. Click **One-off upload**.
4. Download the `InventoryTemplates-version.zip` file, and populate the template that you need with the inventory data.

   ***Tip:** When you process spreadsheets uploaded through the **Application virtualization** section, there are two possible paths:*

   - *You may want to record consumption against existing users on their computers that are already recorded in the operations databases. In this case, be certain that the user's ID from the central database is exactly recorded in the `UserID` column in (either or both of) the spreadsheets used for **Application virtualization**, which are identified in the **Access shown by file evidence** or **Access shown by installer evidence** fields of this **Inventory Data One-Off Upload** page. When the user is matched, the installation is recorded against a computer that the user 'owns' (that is, is linked as either the assigned user or calculated user).*
   - *You may want to create new records for remote devices and remote users who are not already recorded in your operations databases. To do this, make sure that both these statements are true:*
     - *The `Computer` spreadsheet (identified in the **Computers and VMs** field) contains data in both the `ComputerName` and `LastLoggedOnUser` columns.*

- *The value in that `LastLoggedOnUser` column matches the value in the `UserID` column in (either or both of) the spreadsheets used for **Application virtualization**, which are identified in the **Access shown by file evidence** or **Access shown by installer evidence** fields.*

5. In the **Upload name** field, enter a name for this inventory data upload.

   It is best practice to specify an easily recognizable name, because this name is used in lists of data connections. In particular, when the time comes to delete the connection to this one-off import, you will value a self-evident name. Perhaps consider a name space prefix, such as `OOIIS-` or some other convention to help you isolate one-off imports of inventory spreadsheets.

6. From the **Spreadsheet type** drop-down list, select the inventory file format you are going to upload.

   ***Attention:** This control selects the kind of processing applied to your uploaded files. Your upload may include several files for different groups of inventory (one for each kind of inventory listed on this page – for example, a **Computers and VMs** spreadsheet, an **Installation evidence** spreadsheet, and a **File evidence** spreadsheet); but within a single upload, all of the uploaded files must be of the same file type (for example, all `Excel XLSX with headers`).*

7. For each kind of inventory that you wish to import from spreadsheets:

   a. Next to the field for the appropriate data type, click **Browse**, and select the matching `.csv` or `.xlsx` template-based file containing your inventory data.

      ***Restriction:** As a minimum, you must upload one file through the **Computers and VMs** field. This file is mandatory because it contains computer names and serial numbers, plus the `Processor Cores` field required for license optimization.*

   b. Next to each identified data file, click **Upload**.

      "`File uploaded successfully`" message displays.

   c. Repeat the identification and upload process for all files included in this named upload.

8. Scroll down to the bottom of the web page, and click **Start processing**.

   The name of your inventory connection is added to the table at the bottom of the page, and `In progress` gets displayed in the **Status** column. When the inventory file is fully processed, and the license reconciliation is finalized, `Completed` displays instead. Note that all inventory sources are reconciled, regardless of type.

Depending on the file type you imported, its data is available from the corresponding area of the web interface. For example, if you imported computer-related data, go to the **All Inventory** page (**Inventory > Inventory > All Inventory**) to view the uploaded records.

***Note:** License reconciliation does not imply that there are no validation errors: you might need to click*  *in the **Task/Step** name column to see the results of individual steps. If there are any errors, click the hyperlink and troubleshoot as needed. For example, an error that occurred during the `Import into staging` step indicates an issue with the staging, in-memory tables or an invalid spreadsheet type. File-import tasks with the `Failed` status are displayed below the ⚙▼ menu, and are indicated with a red dot:*

You can also view a detailed log of any step within the inventory import: click  to expand its task, and in the **Logs** column for the step in question, click **Download log**.

Setting Up Scheduled Imports of Inventory from Spreadsheets

On an inventory beacon, you can set up a repeatable, automatic uploading of selected spreadsheet files of inventory data to IT Asset Management.

Tip: *Importing* `.xlsx` *files requires that you have installed a 32-bit version of Microsoft Access Database Engine on the inventory beacon that performs the import. (This requirement does not apply to* `.csv` *files.)*

Important: *If it is not the first time that you are importing a spreadsheet file into IT Asset Management, the values already imported with the previous file will be updated and overwritten accordingly. Any update of a previously uploaded spreadsheet file:*

- *Updates the data saved from any modified rows.*
- *Inserts data found in new rows.*
- *Deletes the data from the operational database that was previously imported from rows that have since been removed from the new spreadsheet file.*
- *Deletes duplicate rows. If a duplicate row is identified, only a single entry is created. Identification is based on matching the values in key columns. For example, if the keys match, but some of the other data is different, the first row of the two is kept, while all duplicate rows that follow are discarded.*

Templates are available through the inventory beacon (as described below), or you can reuse templates downloaded for one-off uploads through the web interface of IT Asset Management (provided that these have not been renamed).

To schedule regular imports of inventory spreadsheets:

1. Start FlexNet Beacon.

   Note: *To run FlexNet Beacon, you must have system administrator rights.*

2. In the **Data collection** group, click **Inventory systems**.
3. Click the arrow on the **New** button, and click **Spreadsheet**.
4. From the **Spreadsheet Type** drop-down list, select the type to be used.
5. To prepare your inventory spreadsheets:

a. Click the **Templates** hyperlink displayed in the **Create Spreadsheet Source Connection** dialog box.

b. Populate the template(s) you choose with the appropriate type of inventory data.

When scheduling an automatic, scheduled inventory import, you can populate and update spreadsheet files either by a purpose-built script, or through a work flow applicable to your organization. It is good practice to edit spreadsheet files in a work folder separate from your upload folder. This prevents a clash between file updates and file uploads that could result in incomplete data being processed.

Important: *Remember to always use the latest template rather than one you may have saved previously. Also keep in mind that you cannot change the file name, nor any columns (number, names, or order) supplied in the template.*

c. Create an upload folder, and save your completed inventory spreadsheet files to that folder.

All spreadsheet files located in the one folder are included in the scheduled uploads.

Note: *The folder can also be located on a shared network drive (make sure that an account running the inventory beacon has **Read** permissions on the folder).*

6. In the **Connection Name** field, type in a name for this inventory data upload.
7. In the **Connection Folder** field, browse to the folder with inventory spreadsheets you created in the steps above.

Tip: *If you do not want to import the inventory data as yet, select the **Connection is in test mode (do not import results)** check box. (Remember that data from any secondary inventory source, including this one, cannot be imported until after the first successful import from your primary inventory source.)*

8. Select the overlapping inventory filter that you need.
9. Click **Save**.

The new connection is added to the list of available inventory connections.

10. Select the new connection, and either:

- Click **Execute Now**.
- Click **Schedule...**, and choose the schedule on which to run repeated imports from the **Connection Folder**. (For details on creating schedules, see the online help for the inventory beacon.)

Making a Data Source Connection the Primary One

If you import hardware inventory fields from multiple sources, some fields duplicated across the sources may receive conflicting data. A common case is that one inventory tool returns a particular hardware property, while another tool does not collect the same property and returns a null. There are also differences in the ways tools collect inventory, so that sometimes values vary across tools.

Tip: *This section applies only to hardware inventory values. Software inventory is always merged across all sources regardless of any source being marked as primary.*

IT Asset Management resolves conflicting hardware data in two ways:

- A non-null value received from an inventory connection designated as primary is never overwritten. (Nulls can be replaced.)
- Among values received from secondary sources, generally data with the most recent inventory date is used.

Note: *There are some other settings for the secondary sources (related to whether duplicate inventory should be merged, ignored, or ignored only if older than x days).*

For example, suppose you have three inventory sources that report these values for a single device A:

Source	Primary	Last inventory date	Cores	Threads
Source 1		10 May 2015	4	16
Source 2	Yes	12 May 2015	8	NULL
Source 3		14 May 2015	6	NULL

All non-null hardware properties from Source 2 are given priority, because it is the primary source. Thereafter, based on date, Source 3 is used, and finally Source 1. The final record for Device A shows 8 cores and a total of 16 threads.

An inventory spreadsheet is treated exactly like any other inventory connection in this regard. You can use a repeated import of an inventory spreadsheet to correct specific values reported incorrectly by other inventory tools.

Tip: *You cannot make a one-off inventory spreadsheet upload primary. After a single upload, this source is disabled, and its inventory data ages. Only a repeated upload of an inventory spreadsheet through an inventory beacon should be considered as a possible primary source. Even then, you cannot make it primary until after its first import, so that the source is recognized by the central application server.*

To make a source connection the primary one, click **Make primary** displayed on its row on the **Inventory Data** tab of **Data Imports** page (**Data Collection > IT Assets Inventory Tasks > Data Imports**) of the web interface.

Viewing Validation Errors for Uploaded Inventory Spreadsheets

Diagnose the source of any spreadsheet validation errors.

A page is available that analyzes validation errors in all uploaded inventory spreadsheets (both one-off through the web interface, and scheduled through an inventory beacon). This page is not directly available through the menus, but you can reach it in either of the following ways.

To review upload validation errors:

1. To access through the **Inventory Data One-Off Upload** page (**Data Collection > IT Assets Inventory Tasks > Data Imports** > **Inventory Data** > **One-off upload**):
    a. Scroll to the bottom to the **Last 5 uploads** list.

b. If necessary, click the **+** expander in the **Task/Step** column to reveal individual steps in the processing until the error is revealed.

c. In the **Summary** column, click the **Validation errors** hyperlink.

The **Inventory Upload Validation Errors** page displays.

2. To access through the **Data Imports** page:

a. Go to the **Data Imports** page (**Data Collection > IT Assets Inventory Tasks > Data Imports**).

b. Click the **Inventory Data** tab.

c. Identify the connection for your inventory spreadsheet import, and click the expander arrow on its far right.

The **Last completed import** section shows the count of validation errors. You may click the count (if it is more than zero).

d. If necessary, click on **Show/hide task status and history** to expose the matching panel.

e. In the **Summary** column, click the **Validation errors** hyperlink.

The **Inventory Upload Validation Errors** page displays.

3. Use the diagnostic information to locate and fix the problem. (See the online help for this page for further details.)

4. Repeat the upload using the modified spreadsheet(s).

Important: *For a one-off upload, remember that you must delete the connection for your previous upload before attempting another.*

Deleting Spreadsheet Inventory Data from the Database

To remove data imported from an inventory spreadsheet, delete its connection.

When any inventory data source is removed from IT Asset Management, the data imported exclusively from that source is removed from the database as well.

Tip: *Any data imported from multiple sources remains until its last source is removed. This means that if you want to delete from the database those inventory records that you imported only from a spreadsheet, you need only remove the connection to that inventory spreadsheet.*

You may want to delete the connection to an inventory spreadsheet for several possible reasons:

- You made a mistake with some values in a one-time import of an inventory spreadsheet. To correct this, you must first delete the previous connection to that spreadsheet, and then do a new one-off upload of the amended inventory spreadsheet.
- A one-time upload failed in some way, and is now disabled. You must delete this connection to retry.
- You accidentally have multiple connections to one-time inventory spreadsheet imports existing to exist at the same time. All but one of these must be deleted.

- Inventory imported from a one-time spreadsheet import has aged, and you want to remove it from the **Out-Of-Date Inventory** listing.
- You want to change details about a scheduled import of inventory spreadsheets through an inventory beacon.

You can delete the connections separately for:

- One-time data uploads (for details about one-off uploads, see One-Off Import of an Inventory Spreadsheet). These connections are deleted only in the web interface.
- Scheduled, repeated uploads through an inventory beacon (for more information about these uploads, see Setting Up Scheduled Imports of Inventory from Spreadsheets). Connections established on an inventory beacon must be deleted both in the web interface and separately in the FlexNet Beacon interface.

To delete a connection to an inventory spreadsheet:

1. Go to the **Data Imports** page (**Data Collection > IT Assets Inventory Tasks > Data Imports**).
2. Click the **Inventory Data** tab.

   Tip: *For connections through inventory beacons, if you do not want to delete your set-up connection, and plan to re-use it in future, select* ***Disabled*** *from the* ***Connection status*** *drop-down list displayed on its row. Manually disabled connections can be re-enabled when you are ready. (In contrast, one-off upload connections are automatically disabled after a single processing run, and cannot be re-enabled.)*

3. Click the light-gray triangle displayed at the far right of your data connection's row:

   A panel expands to reveal more details about the connection.

4. Inside this panel, click **Delete connection**.
5. Click **OK** on the confirmation dialog.
6. If this is a scheduled inventory spreadsheet import (that is, one running through an inventory beacon):
   a. Log in to the appropriate inventory beacon (for example, through Remote Desktop Connection), and start FlexNet Beacon.

      Tip: *You must log in with an account that has administrator privileges on the inventory beacon.*

   b. Ensure that the **Inventory systems** page is selected, and select the connection from the list.

      Your connection shows `Spreadsheet` in the **Type** column.

   c. Click **Delete**, and on the confirmation dialog, click **OK**.

The saved copy of your spreadsheet is removed (from the central application server for one-off uploads, or from the inventory beacon for repeatable uploads). At the next import and compliance calculation, the records created from that spreadsheet are removed from the database.

5

Sub-Capacity Licensing with IBM PVU, IBM VPC, and IBM Cloud Pak

When an application runs on a virtual machine, some licenses take into account the kind of server the virtual machine is running on. In those cases, the license rules might require that either:

- The application on the virtual machine must be licensed for the full power of the underlying host (such as counting all its processors, or cores, or threads to work out how many points/entitlements to consume from the license). This is called full capacity licensing.
- The license may take account only of that fraction of the capacity of the host server that is assigned to the virtual machine. For example, on a 16-core host server, a particular virtual machine may be limited to using only 2 cores. You then work out the license consumption using only that fraction of the host server's capacity, and this is called sub-capacity licensing.

Both IBM PVU and IBM VPC licenses support both full capacity and sub-capacity licensing. IBM have tight requirements in their license agreements for sub-capacity licensing. Most basically, you must choose between:

- Using ILMT (or related IBM tools like IBM SUA, Tivoli Asset Discovery for Distributed [TAD4D], and IBM BigFix Inventory — for simplicity this chapter focuses on ILMT). For IBM PVU licenses, this option includes the possibility of importing peak values from ILMT to incorporate them into your global license management view within IT Asset Management. (For IBM VPC licenses, there is no import of license *consumption* results from ILMT – although of course ILMT remains available as an inventory source.)
- Using IT Asset Management to calculate sub-capacity license consumption for the current reporting period.

Note: *You cannot combine sub-capacity calculations from both ILMT and IT Asset Management at the same time — one or the other is the source of truth at any moment. The following points apply:*

- *When ILMT is the source of truth, calculations made by IT Asset Management are (and must always be) at the full capacity of each host. However, when you import IBM PVU results from ILMT into IT Asset Management, the sub-capacity results imported from ILMT for any device are used for license consumption in preference to full capacity consumption for the same device.*
- *Although there can only be a single source of truth at any moment, you can of course switch over from one to the other at a moment of your choosing. Such a change-over does not even need to coincide with a boundary of a*

reporting period, as described later in this chapter.

- *When IT Asset Management is the source of truth, you may continue to import results from ILMT if you wish. These results are also sub-capacity results, since ILMT only exports sub-capacity results. However, results calculated by IT Asset Management are, in this case, always used for license consumption. The only purpose of importing ILMT results when IT Asset Management is the source of truth is to allow comparison of IBM PVU sub-capacity results from the two systems.*

  ***Tip:** It is not uncommon to find differences between the two systems. This is because sub-capacity consumption results show the peak value (summed across three mandatory 'regions' declared by IBM) over time, and the times used are different:*

  - *ILMT exports the peak values for all time. (This is different from the results displayed within ILMT, which are the peaks for the current reporting period.)*
  - *IT Asset Management recalculates daily the peak values for the current reporting period, taking into account your latest data corrections (as discussed later).*

  Therefore, if you had a peak value in Asia/Pacific region in 2015, and since then you have adjusted your configuration so that consumption is much lower, IT Asset Management reports your lower peak for the current reporting period (typically a 3 months' rolling window), while ILMT still exports the higher peak from years ago. For an apples-to-apples comparison, then, it's best to compare the values visible within ILMT with those calculated by IT Asset Management.

This chapter first gives you a high-level overview of the requirements for the two approaches, to help you make the technology choice (see Two Ways to Collect Inventory).

However, for many enterprises, this is not so much an "either-or" question as it is a "both-and", or more precisely, first one and then the other. Therefore the chapter next provides a conceptual framework for understanding the overall process of transitioning from one tool to the other (see Understanding the Transition).

Whether you are implementing just one approach, or are changing over from one to the other, there is a lot of overlap in required actions and detailed processes. Because of the overlap, the remaining topics in the chapter can be read from either point of view:

- If you are implementing only one approach, choose either Using ILMT (and Importing Results) or Using IT Asset Management, and step through the procedures there, ignoring any sections about transitioning tasks.
- If you are changing from ILMT to IT Asset Management, you may use all the topics in the order provided to ensure a smooth transition and an effective outcome.

Two Ways to Collect Inventory

Like all license consumption calculations, sub-capacity licensing using IBM PVU or IBM VPC licenses relies on incoming software and hardware inventory. IBM authorizes any of the following approaches:

- Collect inventory using the FlexNet Inventory Agent, and under tightly-prescribed conditions, import into IT Asset Management which performs the sub-capacity calculations and provides appropriate reporting. This approach is available for either IBM VPC or IBM PVU licenses.
- Collect inventory using ILMT under similar conditions (such as inventory collection every 30 minutes), allowing ILMT

to perform the sub-capacity calculations and resultant reporting; and then import the product consumption into IT Asset Management to achieve a 'single pane of glass' for compliance management. The import of license consumption *results* from ILMT is supported only for IBM PVU licenses.

- Use either product for full capacity licensing, usually with the resultant increase in licensing costs. This option is not investigated further in this chapter, which focuses entirely on your options for sub-capacity licensing.

The following table highlights the different requirements for the two approaches to sub-capacity licensing.

Requirement	IT Asset Management	ILMT
Licensing	Use a current version of IT Asset Management. ***Note:*** *IBM approval requires that you are using a version later than 2015 R1. You are currently reading documentation for release 2023 R2.4.* Be sure that you have licensed the FlexNet Manager for Datacenters product.	ILMT is licensed. If using IBM Db2 as the underlying database, assign a commercial Db2 license to ILMT, rather than using Db2 Community Edition, or the free Db2 bundled with ILMT, or similar. (Microsoft SQL Server may be used as an alternative database, and this of course also requires a commercial license.) Use a current version of IT Asset Management. Be sure that you have licensed the FlexNet Manager for Datacenters product. (This supports the IBM PVU license type and appropriate reporting.)
Approval	The certification of Flexera IT Asset Management as an IBM-approved alternative to ILMT means that no further action is required.	The standard IBM PVU or IBM VPC licenses apply.
Agents	Install the FlexNet Inventory Agent on all application servers running software under either IBM PVU or IBM VPC licenses; and also on any Hyper-V hosts that may have guest VMs running software licensed under IBM PVU or IBM VPC.	Install the ILMT agent on all application servers running software under IBM PVU or IBM VPC licenses.
Drivers	Nothing additional required.	Where ILMT is using a Db2 back-end, install a Db2 driver on each applicable inventory beacon. (No separate driver is required for an underlying Microsoft SQL Server database.)
Credentials	Normal requirements for the installed FlexNet Inventory Agents. Additional credentials may be required (as usual) for inventory collection from VMware virtual hosts.	Create an account in the database (either Db2 or SQL Server) for data collection. ***Tip:*** *This account requires the DATAACCESS privilege level.*

Requirement	IT Asset Management	ILMT
Configuration	Configure FlexNet Inventory Agent to collect the relevant data every 30 minutes.	The standard ILMT configuration collects data every 30 minutes.
FlexNet Beacon	Normal operation for data uploads from the FlexNet Inventory Agents to the central application server.	Remote data collection by the inventory beacon.
License management	Licenses can be entirely managed in IT Asset Management, using all your normal processes.	Product consumption for IBM PVU licenses calculated in ILMT may be imported into IT Asset Management, which automatically creates license records to match the consumption data from ILMT. You may then attach purchases to the licenses to automatically compare consumption against entitlements and determine your compliance status. ***Remember:*** *For IBM VPC licenses, in contrast, there is no way to import any license consumption results from ILMT into IT Asset Management.*

There are following sections that provide guidance through the processes required for each of these methods. In both cases, we assume that you have already licensed the appropriate products, and proceed from there. Either method may be used independently and without reference to the other, starting from a fresh implementation.

However, the more common scenario is that you may have ILMT already operational, and you wish to *transition* to using IT Asset Management for sub-capacity IBM PVU and VPC license calculations. Since the descriptions of the independent approaches overlap with description of the transition, we start with an understanding of the transition process. If this is not relevant to you, you can skip directly to whichever approach you have chosen to use.

Understanding the Transition

The golden rule for managing a switch from ILMT as the reporting tool for IBM sub-capacity licensing over to IT Asset Management as the replacement reporting tool is this: there can only be *one source of truth* at any given moment. Peak license consumption can be calculated by exactly one tool.

Having a single source of truth must be the case, because the calculation of peak consumption must account for all devices within a *single* calculation. You may not get the same result from two sub-capacity calculations for a few machines here and a few others there, simply summed (for example, how would you track VM reassignments that transitioned across tool boundaries?); and far more damage may come from *overlapping* calculations, where one or more servers are included in two different sets of peak calculations, double counted. So the principle is: a single source of truth.

However, this is not the same as saying that both tools cannot be *present* within the same 'sphere of influence' within your enterprise. Your operations may be divided across several 'spheres of influence'. Each of these may be:

- A separately managed region
- A distinct subsidiary in your company structure
- The time period before transition day, and the future after transition day — at first you have only ILMT providing the peak consumption calculations, and after "throwing the switch" you have only IT Asset Management providing the calculations.

Regardless of how you may divide up your enterprise, you may have both tools present within any individual sphere of influence. Having both tools present at once is possible because of this feature of IT Asset Management: where it has access to the PVU data imported from ILMT *and* its own calculations from FlexNet inventory, by default it gives preference to the ILMT imported results. Therefore, both tools may be present, but there is still a single source of truth: the ILMT calculations. (And keep in mind that, for IBM VPC licenses, there are no imports of consumption results calculated by ILMT.)

This allows for a change-over process that, at the high level, looks like this:

1. Initially only ILMT is in use. Of course, it must cover all inventory devices where sub-capacity calculations apply. ILMT sub-capacity peak consumption reports are supplied to IBM at each reporting interval (typically, each quarter). If you also have products that must be licensed at full device capacity, you can record these separately (in theory, even on a spreadsheet if you wanted to). So each quarterly report consists of sub-capacity peak value *plus* any separate full capacity value.
2. IT Asset Management is now introduced. At first, for IBM PVU points it simply connects to the ILMT database, and imports inventory data, product consumption (which IT Asset Management converts into license records), and peak consumption results. ILMT remains the source of truth, even though seen through the lens of IT Asset Management. Meanwhile, for your IBM VPC licenses, you continue to use consumption results from ILMT.
3. Now you deploy the FlexNet Inventory Agent to those same target devices where sub-capacity calculations apply. As usual, the installed FlexNet Inventory Agents gather their inventory daily, uploading it to the central application server. However, although IT Asset Management now starts internally calculating results from its own inventory, these are restricted:
   - For IBM PVU licenses, IT Asset Management does not publish these results because, by default, it honors the results calculated by IMLT and publishes those in the web interface. (In fact, once both the ILMT agent and the FlexNet Inventory Agent are uploading data, and the daily import from ILMT is operational, IT Asset Management provides a report where you can compare, for every device, both full- and sub-capacity calculation results from each of the two tools.) ILMT is still the source of truth.

     Tip: *If it should happen, during this stage, that there is a device reported only in FlexNet inventory (and not reported by ILMT), this device is excluded from sub-capacity calculations (for which ILMT alone is currently responsible), and its consumption must be calculated at full capacity.*

   - For IBM VPC licenses, since there is no import of license consumption results from ILMT, IT Asset Management calculates license consumption. Since IT Asset Management at this point is *not* in high-frequency mode, these are *full capacity* calculations (by default), and those full capacity results are displayed in the web interface. This means that, at this stage, you should continue to use ILMT for your management of VPC consumption and for reporting to IBM, since ILMT provides you with the sub-capacity results that represent your license liability to IBM. ILMT is still the source of truth.

     Tip: *Unlike the PVU case, for VPC there cannot be a built-in report comparing results from ILMT with those*

from IT Asset Management, since no results are imported from ILMT.

4. When the agent coverage across target devices is the same for both ILMT and IT Asset Management, you metaphorically "flick the switch". This includes changing the installed FlexNet Inventory Agents into high-frequency mode, so that they can track hardware changes every 30 minutes as IBM requires; and also changing the default so that results no longer come from ILMT imports, but from calculations by IT Asset Management. From now on, the results calculated from FlexNet inventory are the source of truth. Because IT Asset Management has been saving historical data for sub-capacity calculations, it is able to prepare reports based on FlexNet inventory for whichever is the longer of:
   - The time since you had FlexNet inventory available (having the FlexNet Inventory Agent installed on devices running software under a sub-capacity PVU or VPC license)
   - The current reporting period.
5. In one final piece of fine tuning, when setting the reporting period in IT Asset Management, you can also separately set **Ignore any value prior to** — the date when you want your reports to IBM to be entirely based on FlexNet inventory. You might set this for a few days *after* turning on high-frequency mode, allowing time to check that everything is working perfectly, all devices are covered and so on. On that date, you archive your final report out of ILMT (even when this is for a part of a reporting period), and from then on reports come from your new source of truth, IT Asset Management.

Here is the same process summarized into tabular form:

Project phase	ILMT is source of truth	FNMS is source of truth
1. ILMT only	Covers all subcap devices; provides reports.	
2. Integrated presentation	ILMT consumption data for IBM PVU licenses is imported into IT Asset Management. ILMT provides reports. (For IBM VPC licenses, IT Asset Management displays full capacity consumption results.)	
3. Rollout	FlexNet Inventory Agent is locally installed on the same complete set of sub-cap devices. ILMT provides reports. (For IBM VPC licenses, IT Asset Management displays full capacity consumption results.)	
4. Switch		Turn on high-frequency mode, which inverts the default so that you now prefer sub-capacity results from FlexNet inventory. IT Asset Management provides reports, starting from the effective date of your choice.

We've already noted above that your operations may be bounded within certain 'spheres of influence'. If this applies to you so that you have several such spheres to deal with, you can undertake phases 1-3 of the above transition process in one sphere, and then move on to another sphere, and continue until all required areas have the FlexNet Inventory Agent fully deployed. Then you can move your entire IT Asset Management implementation into phase 4 in one step (there is a single setting that controls high-frequency mode for an entire implementation at once, and a single schedule for IBM sub-capacity calculations worldwide).

This chapter does not cover your implementation of ILMT. It is possible that you may never implement ILMT, if IT Asset Management is your first inventory and license management tool. In what is perhaps the more common case, it is assumed that you have already implemented stage 1, using ILMT as a free-standing system. You may then start with Using ILMT (and Importing Results) (and its subtopics) for details about stage 2, the integrated presentation of ILMT results for PVU consumption within IT Asset Management.

Using ILMT (and Importing Results)

This section covers using ILMT as the source of truth for sub-capacity points calculations, and importing its IBM PVU results into IT Asset Management for license compliance calculations.

Remember: *IT Asset Management does not import license consumption results for IBM VPC licenses from ILMT. These topics are exclusively dealing with IBM PVU licenses.*

This overview assumes that you have already licensed both IT Asset Management and FlexNet Manager for Datacenters. If not:

- Validate your license for FlexNet Manager for Datacenters by navigating to the system menu in the top right corner of the web interface for IT Asset Management, and selecting **IT Asset Management License**. If the card for FlexNet Manager for Datacenters is grayed out, contact your Flexera representative for assistance.

The process overview (with appropriate pointers to the online help) covers the following:

- Operation Using ILMT
- Set Up Connections.

Operation Using ILMT

This topic describes integration with ILMT specifically for IBM PVU licenses, where results from ILMT can be imported into IT Asset Management. (There is no equivalent import of license consumption calculations from ILMT for IBM VPC licenses.)

Note: *ILMT may operate with either an IBM Db2 back end, or a Microsoft SQL Server database. In the first of these cases (Db2), integration between IT Asset Management and ILMT normally requires that you hold a commercial (paid) license for the underlying Db2 database. Your license from IBM has to be sufficient to allow third-party access to the database. Suitable examples include Db2 Workgroup Server Edition, or (for advanced features of Db2) Db2 Enterprise Server Edition, or Advanced Enterprise Server Edition. The free, bundled Db2 license for ILMT does not include these third-party access rights.*

When you are using ILMT as the calculator and reporter for sub-capacity consumption of IBM PVU licenses, the majority of your effort is in ILMT (deploying the ILMT agent, potentially licensing a commercial version of Db2, and so on), for

which see the documentation from IBM. The integration of the two systems, in summary, is as follows:

1. According to your configuration of ILMT, the ILMT agent collects inventory regularly, and ILMT tracks your historical and peak consumption for all relevant products.
2. Using an inventory beacon, you configure a connection to your ILMT database, using an account with `DATAACCESS` privileges. On the schedule you configure, IT Asset Management collects a full set of data from ILMT into the staging tables of the compliance database.

   Information collected and calculated by ILMT (and subsequently imported as finished data into IT Asset Management) includes:

   - Inventory evidence
   - Software titles (or, in the terminology used for IT Asset Management, applications, including bundles)
   - Inventory devices running the licensed software
   - The points consumed by these devices for the software.

   Note: *In this scenario, ILMT performs all required high-frequency hardware tracking, and typically reviews the data and updates peak consumption calculations only once a day. Therefore the import from ILMT to IT Asset Management does not need to be a high-frequency action: typically, a daily import of the ILMT results is all that is required.*
3. Honoring your configuration in the inventory **Settings** page to make ILMT the source of truth for IBM PVU sub-capacity licensing (that is, the **Enable frequent hardware scanning for IBM sub-capacity license calculations** is *not* selected), at the next full license reconciliation IT Asset Management provides special behaviors:

   - It imports the ILMT data from the staging tables, and integrates it for presentation within your overall inventory and license compliance picture.

     Tip: *ILMT keeps records of all your inventory devices since records began, including all those that have been marked as 'deleted' at any time in the past. While all of these records are available to your ILMT connector, IT Asset Management imports only newly-deleted devices: that is, those that were marked as deleted within the last 90 days, and which may therefore be contributing to the calculation of peak consumption in the current reporting period. Once the rolling 90-day window passes the date when a particular device was marked as deleted, it is no longer imported; and since it "disappears" from all inventory sources, the equivalent record within IT Asset Management is also removed.*
   - IT Asset Management creates IBM PVU license records that replicate the consumption data imported from ILMT. One way to identify license records created in this way is to check the **History** tab of the license properties, where the **Created by** field displays the operator name followed by `(Auto generated)`.

     Note: *IT Asset Management does not recalculate individual device consumption imported from ILMT on these licenses, but simply attaches the values calculated by ILMT. In addition, any manual entry in the* ***Overridden consumption*** *column (on the* ***Consumption*** *tab of the license properties) is ignored for rows where* ***Calculated by*** *displays `ILMT`. Think of this as a "copy/paste" operation from ILMT into the license properties in IT Asset Management — nothing changes the PVU consumption figures calculated by, and imported from, ILMT for each individual inventory device. However, the total consumption recorded for all inventory devices linked to the license record is calculated within IT Asset Management as a simple sum.*

- IT Asset Management may also create or update other records as required to keep data aligned. For example, if an application is reported from ILMT that has not previously been identified from inventory within your enterprise (and the application exists in the Application Recognition Library [ARL]), an application record is linked to the license on import.

Note: *Applications must exist in the ARL, or matching records must have been previously created within your enterprise, for those applications to be automatically linked to the automatically-created licenses. Where ILMT is the* ***only*** *source of information about an application, that application is not reproduced within IT Asset Management. You may request that such an application be added to the ARL. In another example of 'live' license records, if you download an updated ARL that now contains the newly-added application record, and subsequently import again from ILMT, the previously-missing application is now automatically linked to the existing license record.*

Important: *Licenses automatically created from an ILMT import are still 'live' license records that can change, just like all others. It is therefore possible for undesirable changes to be made to these license records, with confusing results. Such undesirable changes may occur automatically, or as a result of manual action. As examples:*

- ***Automatic change:*** *Remember that the license is created to show the consumption results for individual inventory devices imported from ILMT. (You can identify the imported devices on the* ***Consumption*** *tab of the license properties by adding the* ***Calculated by*** *column, which displays* `ILMT` *for these devices.) However, because this is still a 'live' license, it is possible for an inventory device that is not imported from ILMT to become attached, if this auto-generated license provides the best fit for its installation data. (For this non-imported device, the* ***Calculated by*** *column displays* `Internal`*.) Naturally, if extra device(s) get attached to the license, this increases the total consumption on the license:*
  - *The consumption figures for individual inventory devices imported from ILMT remain the same as recorded in ILMT*
  - *The extra* `Internal` *device adds its own consumption figure to the total result as well, so that the total license consumption shown in IT Asset Management is now greater than the consumption shown in ILMT.*

  If this automatic change occurs and is not what you want, you can correct it in either of two ways:
  - *Identify the rogue* `Internal` *device that has the licensed software installed but is not being managed by ILMT, and move it into management by ILMT*
  - *If you have some reason for excluding the device(s) with* `Internal` *calculations from management by ILMT, create a clone license with a changed name (for example, adding* `from FNMS inventory` *to the license name), and use group assignments or individual allocations to prioritize these non-ILMT devices onto this separate license record.*
- ***Manual change:*** *As an example of a manual change with undesirable, confusing outcomes, an operator may mistakenly apply an exemption within IT Asset Management to an inventory device imported from ILMT. Remembering that the consumption is being calculated only in ILMT, it follows that, if you need to add an exemption to an inventory device, you must apply the exemption in ILMT, your chosen source of truth. For each device where inventory is imported from ILMT, four figures are recorded internally within IT Asset Management for each installed application:*
  - *The full capacity and sub-capacity points currently consumed by the installation (in the latest import). The sub-capacity figure is selected whenever the device is eligible, and otherwise the full capacity figure is*

*used. The selected current consumption for each device is what is displayed on the **Consumption** tab of the IBM PVU license created to track consumption for this application. If you apply an exemption to such a device within IT Asset Management, the exemption does not affect this current consumption (as you can observe by the unchanged total consumption shown on that tab).*

- *The maximum full capacity and sub-capacity points historically consumed by the installation. These points are totaled and displayed in the **Compliance** tab of the license. Once again, any exemption within IT Asset Management for a device does not affect the historical consumption imported from ILMT at all.*

*Best practice (when ILMT is your chosen source of truth) is to make such configuration changes only in ILMT. If someone has mistakenly applied an exemption within IT Asset Management, copy the exemption into ILMT, and remove it from IT Asset Management. The next import into IT Asset Management causes the records there to be updated appropriately, because devices exempted in ILMT export an individual consumption figure of zero, and the peak consumption figures are also adjusted appropriately. (The exemption settings themselves are also imported from ILMT, and are automatically displayed in IT Asset Management as `Covered by related product` for the appropriate devices in the **Consumption** tab of the license properties. If you are transitioning from ILMT to IT Asset Management, you will use these exemptions as the basis for setting your own exemptions within IT Asset Management, as detailed in Additional Transition Steps.)*

4. After each import from ILMT and the subsequent full license reconciliation, IT Asset Management displays the latest **Peak consumed** figure in the **Compliance** tab of license properties, and reflects this value in the **Consumed** column of license listings.

   ***Tip:** While listings of separate licenses correctly use the peak consumption as the total for each license, the consumption for each individual device displayed in the **Consumption** tab of the license properties is its current value calculated at the last full reconciliation, regardless of any peak to which the device may have contributed at some other time.*

   While uncommon, it is possible for the peak consumption calculated by ILMT to go down: for example if, in ILMT, you add an exemption for a server. For each change in points calculated by ILMT (whether up or down), the ILMT import to IT Asset Management and subsequent reconciliation update the **Peak consumed** figure in the license properties to take account of the ILMT results. However, keep in mind that the value displayed in IT Asset Management may be the sum of the sub-capacity value calculated by ILMT and some full capacity results calculated from other inventory sources. When there are no other inventory sources, the **Peak consumed** figure matches the latest peak points imported from ILMT.

5. While ILMT remains the source of truth for sub-capacity calculations, regularly archive reports from ILMT for submission to IBM.

Set Up Connections

To gather IBM PVU consumption data from ILMT, IT Asset Management must make a direct connection to the database used for ILMT. This connection is made by an inventory beacon, which imports the necessary data, and uploads the file to the central application server for processing and display.

To configure a connection to ILMT:

1. Move to the appropriate inventory beacon, and using a local account with adequate privileges on the inventory

beacon server, open the Windows start menu, right-click **Inventory Beacon**, and select **Run as administrator**.

2. If you do not already have a convenient schedule for connecting to ILMT, set the schedule first (so it is easier to link to it later).
   a. From the **Data collection** group in the navigation bar, choose **Scheduling**, and click **New...**.
   b. Make the unique **Schedule name** distinguishable within the first few characters (for use in narrow display columns).
   c. Typically, select the radio button for daily data collection, and set the preferred time for the connection (this is local time on the inventory beacon).
   d. Click **OK** to add this schedule to the list on the **Scheduling** page.
3. Set up the connection from the inventory beacon to your ILMT database.

   ILMT supports either a Db2 database or a Microsoft SQL Server database. Choose the connection type to match your configuration:
   Using a Db2 database:
   a. Select the **Inventory systems** tab in the FlexNet Beacon interface.
   b. Below the list of connections, click the down arrow on the right of the **New** split button, and choose `Other`.
   c. Give the connection a useful name that makes sense in listings of several connections, and for **Source type**, choose `ILMT`.
   d. Set up the **Connection String** similarly to the following (but all in one line):

   ```
   Provider=driverCode.Db2COPY1;Password=password;
   Persist Security Info=True;User ID=db2admin;
   Data Source=TLMA;Location=servername:50000;
   Extended Properties=""
   ```

   where

   - The `Provider` value consists of:
     - A fixed string literal determined by the driver (for example, `IBMDADB2` using the IBM driver, or `DB2OLEDB` using the Microsoft driver)
     - A period (separator)
     - The database copy registered name (typically `TEMADB`).

     Note: *Including the database copy name may be optional after DB2 Version 9.7 Fix Pack 8 is applied, but its use resolves an earlier problem with accessing DB2.*
   - `Password` is the password that the inventory beacon should use when accessing the DB2 database
   - `User ID` is the account name that the inventory beacon should use when accessing the DB2 database (`DATAACCESS` privileges are required, since later versions of ILMT drop and recreate database tables)
   - `Location` includes the name of the server running the database, and the access port (typically 50000,

unless locally reconfigured).

Using a Microsoft SQL Server database

a. Select the **Inventory systems** tab in the FlexNet Beacon interface.

b. Click **New**. (If you clicked the down arrow on the split button to reveal all the options, choose `SQL Server`.)

c. Give the connection a useful name that makes sense in listings of several connections, and for **Source type**, choose `ILMT SQL`.

d. In the **Server** field, identify the database host server:

- Use either the host name or FQDN, or an IPv4 address
- If the database is on this same inventory beacon server, use the special value `(localhost)` including the parentheses.
- 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
```

e. Specify the **Authentication** details for the database connection.

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 an Active Directory security group that has access to the database.
- **Windows (specific account)** — Use the following two fields (enabled when you make this choice) to specify an account on the inventory beacon that can make a connection to the SQL database.
- **SQL Authentication** — Use the following two fields to specify an account and password registered as a user with database access on SQL Server . This account is used to access the database, regardless of the local account running the scheduled task on the inventory beacon server.

Tip: *The account used needs read-only privileges.*

f. Identify the **Database** name as `TEMADB`.

When you entered the **Server** details earlier, the inventory beacon attempted to connect and identify all databases present there. If this was successful, you can pull down the list and select the `TEMADB` database. If the appropriate database name is not available, you can enter `TEMADB` directly in this field.

Tip: *A default implementation of ILMT on SQL Server may offer two different databases. Because you are using ILMT (or else BigFix Inventory), choose the database called `TEMADB`. (The alternative database, called `BFEnterprise`, is for use with a different connector type, and in any case is not recommended because the combination returns very incomplete inventory.)*

4. If necessary, configure the **Overlapping Inventory Filter**.

This filter has effect even when you have declared that this connection to ILMT is to be your primary inventory source. In normal operation, to ensure that details imported from ILMT cannot be blocked by overlapping inventory from another source, select **Import the inventory from this source for possible merging**.

5. Use the **Test connection** button to validate the details provided and ensure that the connection can operate (if not, correct the problem before proceeding), and then click **Save** to add this connection to your list.
6. Select your new connection from the list, and click **Schedule...** (below the list).
7. In the **Select schedule** dialog, choose the appropriate schedule, and click **OK**. The **Next run** column shows the projected run time for the connection.
8. At the bottom of the panel, click **Save** to store your connection details and schedule.

The connection is now ready for regular imports from ILMT. By default, the IBM PVU license consumption figures calculated by ILMT have priority, and are displayed in the web interface for IT Asset Management. If you are transitioning from ILMT to IT Asset Management, this completes phase 2 of the process outlined in Understanding the Transition. Some additional steps, applying only to the transition project, now help to ensure that corresponding data eventually appears in both systems. These are described in Additional Transition Steps.

Additional Transition Steps

The following steps apply only if you are transitioning from ILMT to IT Asset Management as your source of truth for sub-capacity calculations for IBM PVU license consumption.

Steps to smooth the transition process:

1. If you have not already done so, it is time to deploy the FlexNet Inventory Agent to all target inventory devices where sub-capacity IBM PVU points may be consumed. This includes any Hyper-V hosts that may have guest VMs running IBM PVU software.

   For details, see *Gathering FlexNet Inventory*. You may use either the Adopted model or the Agent third-party deployment model. Be clear that IBM requires that you deploy the full FlexNet Inventory Agent (not any subset), and that it is locally installed on the target inventory device in question — a combination that excludes the other models in that PDF file. Rolling out the FlexNet Inventory Agent moves you into phase 3 of your transition project (as summarized in Understanding the Transition). During this phase, calculations by ILMT continue to have priority; and if a report for IBM falls due during your roll-out, supply a report from ILMT.

2. Update all the device exemptions imported from ILMT in IT Asset Management.

   Specific exemptions that prevent consumption on certain devices (such as backup servers) that you have recorded in ILMT are imported into IT Asset Management as `Covered by related product` exemptions on the matching devices. However, the sad news is that, when you later remove your ILMT connection, all information that was imported *only* from this source is automatically cleaned up as well, and this clean-up removes the exemptions that came only from ILMT. (A special clean-up exception is made for license records that were created to match imports from ILMT: these license records are *preserved* after the ILMT connection is removed. By that time, those licenses should also be receiving FlexNet inventory and consumption calculations from IT Asset Management.)

   To prevent the automatic clean-up of exemptions imported from ILMT, manually visit each one and switch the exemption type to a different value that better describes the purpose of the hardware device. Here's one way you could do this:

a. For each affected IBM PVU license in turn, open the license properties and choose the **Consumption** tab.

b. If necessary, drag the **Calculated by** column from the column chooser into the header row of the list of devices.

c. If the simple filter row is not already visible below the header row, click the blue filter icon (above the listing).

d. Enter `ILMT` in the filter field for the **Calculated by** column, click the filter icon to its right, and from the drop-down menu, choose `Contains`.

   After a moment, the list is refreshed (as a flat list now, with the hierarchy no longer visible) to show only those devices whose consumption was calculated by, and imported from, ILMT. These are the only devices you need to amend, since if devices show `Internal` (that is, calculated by IT Asset Management), any exemptions applied to them are not linked to ILMT as the inventory source.

e. In a similar way, set a filter on the **Exemption reason** column where this value `Contains related product`.

   Again the list refreshes, showing only those consumption rows imported from ILMT with an exemption of `Covered by related product`.

   Tip: *If you have many such devices, so that the list runs over multiple pages, it is handy to edit the* ***rows per page*** *value (above the list, left side) to see all these target devices in a single page.*

f. At the left end of the *header* row, tick the selection box to select all the inventory devices visible in the current page.

   When you make any selection from the list, the **Exemption reason** button above the list is enabled. Scan down the list to identify any device(s) that you think should not share the *same* exemption reason, and deselect the check box on the left of their individual row(s) (for example, some machines are `Backup, disaster recovery` devices and others are `Development` devices). Keep adjusting your selections from the list until you have a homogenous set of devices that all share the same exemption reason.

g. Click the **Exemption reason** button, and choose the appropriate item from the list. Notice that you can also add a custom exemption reason using the same drop-down, but be careful to reflect accurately the terms of your IBM license agreement.

   All selected inventory devices are assigned the same exemption reason, and in future IT Asset Management calculations of consumption for this license, these devices consume zero points. Making the exemption in this way, directly in the **Consumption** tab of an individual license, naturally affects only this one license (whereas exemptions by **Device role** can conveniently cover multiple licenses). If you need to apply a different exemption reason to other devices in this list, ensure that the correct set of devices is selected, and use the same technique to apply the new **Exemption reason**.

h. Repeat for all remaining IBM PVU licenses that were both imported from ILMT and using sub-capacity calculations, and had exemptions recorded in ILMT.

After processing all relevant IBM PVU licenses in this way, you have locally-set exemptions for all your applicable devices. Consider whether you need any audit trail of reasons that may cause you to add notes or separate documents to any of these license records.

3. Exceptional corner case: If you require separate licenses for sub-capacity and full capacity licenses for the same product(s), plan your approach now.

Typically, for a product entitled to use sub-capacity license calculations, the rows on the **Consumption** tab of the license properties provide a mix of devices, some of which use sub-capacity calculations and other full capacity, for reasons like the following:

- Sub-capacity licensing applies (of course) only to virtual machines. If the same software is installed on a stand-alone physical server, that device is calculated at full consumption; but typically this is attributed to the same license as sub-capacity calculations for virtual machines.
- IBM authorizes a very limited set of tools for sub-capacity calculations, and for the related inventory collection. If inventory for a particular device is returned either by ILMT or by the FlexNet Inventory Agent, this device may use sub-capacity calculations (from either one source or the other); but if the same device appears only in inventory collected from another source (say, BMC Discovery), then the same device must use full capacity calculations (except by special IBM permission).

For these scenarios, there is *no* need to separate the consumption rows so that all the sub-capacity calculations apply to only one license, and all the full capacity calculations apply to a separate license. Typically, the only real reason for separating licenses for full- and sub-capacity licensing is when an enterprise (perhaps through mergers and acquisitions) has some divisions using licenses with sub-capacity entitlements, and other divisions that must use full capacity licensing. If this applies to you, plan how to implement this. An approach that typically works well is to use two orthogonal kinds of enterprise groups:

- Use locations to represent (or roll up into) the three regions required by IBM for separate reporting (for details, look ahead to Configuring Regions for IBM)
- Use corporate units (or less commonly, cost centers) to create separation between the relevant company divisions.

Assuming that both kinds of division span multiple of the IBM-defined regions, each product may need separate licenses, differently configured in each of your FullCap Division and your SubCap Division. You achieve the correct roll-up results when you assign ownership of inventory devices (and optionally, license purchases) to both of:

- The appropriate location for roll up into the mandatory IBM regions, and
- The appropriate corporate unit to establish membership in either your FullCap Division or your SubCap Division as appropriate.

However, you may find it more attractive to swap to Plan B: negotiate with IBM to move FullCap Division to sub-capacity licensing as well!

Using IT Asset Management

This overview assumes that you have already licensed both IT Asset Management and FlexNet Manager for Datacenters. If not:

- Validate your license for FlexNet Manager for Datacenters by navigating to the system menu in the top right corner of the web interface for IT Asset Management, and selecting **IT Asset Management License**. If the card for FlexNet Manager for Datacenters is grayed out, contact your Flexera representative for assistance.

The process overview (with appropriate pointers to the online help) covers the following:

- The legal and technical prerequisites for operating in the mode where IT Asset Management is both the inventory source and the calculator for sub-capacity license consumption (often called "high-frequency" for short, based on the requirement for hardware inventory checks every 30 minutes), described in Requirements for IT Asset Management

Sub-Cap

- An overview of how high-frequency mode works (see Operation in High-Frequency Mode)
- Preparing to report license status for each of the three Regions declared by IBM (see Configuring Regions for IBM)
- Setting up and initializing the appropriate sub-capacity licenses (see Configure Appropriate Licenses)
- Seeding the automatic checking of hypervisors for changes in VMs that may impact these licenses (see Set Up Virtual Inventory Tracking)
- Initializing inventory collection from target devices where software linked to an IBM sub-capacity license may run (see Set Up and Collect Inventory, and Reconcile)
- At last, turning on the automated management of high-frequency mode and sub-capacity license calculations by IT Asset Management (see Turn on High-Frequency Mode)
- Archiving reports and associated materials for possible use in a future audit (see Reporting to IBM).

Requirements for IT Asset Management Sub-Cap

The full set of requirements for using IT Asset Management to calculate sub-capacity license calculations for IBM PVU or IBM VPC licenses divides into two groups: the legal conditions that IBM writes into the amended license agreement; and the mechanical requirements imposed by the product. An understanding of these requirements helps to guide you through the processes described in the subsequent pages.

IBM requirements for sub-capacity licensing

Using IT Asset Management data (derived from the 2015 release or later) to determine an IBM PVU or VPC license position on IBM software is acceptable by IBM for sub-capacity reporting in place of IBM License Metric Tool (ILMT), Tivoli Asset Discovery for Distributed (TAD4D), IBM Software Usage Analysis (SUA), or IBM BigFix Inventory. This requires all of the following:

- You must be using IT Asset Management 2015 or later. (This documentation is for 2023 R2.4.) This product must be installed, configured and maintained correctly.
- You must have licensed (and use) the FlexNet Manager for Datacenters product.
- Inventory must be collected by the FlexNet Inventory Agent, which must be installed directly on the device(s) running (or that may run) the software that is linked to an IBM PVU or VPC license. The method of installation of the FlexNet Inventory Agent does not matter, and installation may be achieved (for example) by any of:
  - Adoption, using separate inventory rules targeting the servers in question to trigger the adoption
  - Deploying the FlexNet Inventory Agent independently, using your own preferred infrastructure tools
  - Including the FlexNet Inventory Agent in the base image used to clone virtual machines that may run relevant software
  - Installing the FlexNet Inventory Agent manually.
- You must increase the frequency of *hardware* inventory scanning on the target device so that it is performed every 30 minutes. The **IBM reporting and archiving settings** section of the **Inventory Settings** page contains the settings for the required increased frequency of hardware scanning.

- On the same frequency, you must check the virtual hosts (VMware vCenter, Oracle VM Manager and Hyper-V servers) for any changes affecting VMs running the software attached to an IBM sub-capacity license.

Product prerequisites for higher frequency processing of IBM sub-capacity licenses

The following technical conditions must be satisfied before you can use this facility:

- **Permission:** You have licensed the FlexNet Manager for Datacenters product.
- **Inventory:** Inventory must be collected by the FlexNet Inventory Agent locally installed on the devices that are (or may be) running software attached to IBM sub-capacity licenses. While you may use targets to trigger adoption of these devices, you may alternatively use any deployment method. For example, you might install the FlexNet Inventory Agent in your base image used to clone virtual machines.
- **License with consumption:** You have at least one IBM PVU or IBM VPC license linked to one or more applications that have installations shown in inventory. Since both IBM PVU and IBM VPC licenses are points-based licenses, you must be using the appropriate points rule: this may be one of the standard points rule sets downloaded with the Application Recognition Library, or (if you have negotiated a custom points rule with IBM), you may have a locally-customized points rule (remembering that where Flexera points rules and local points rules overlap, the local rule has priority).
- **Reconciled:** You have run a **Reconcile** since inventory was collected, triggering consumption from the above license. The initial gathering of this inventory and calculation of consumption is used to trigger operation of two specialized and normally hidden targets, described next.
- **Targets:** IT Asset Management automatically maintains two relevant targets:
  1. Servers running VMware vCenter and Oracle VM Manager (OVM Manager) — these servers (which may also manage clusters) do not expose their VM management to a locally-installed FlexNet Inventory Agent. For IT Asset Management to gather data about the VMs under management, this target must be used for inventory gathering by the inventory beacon remotely accessing the management API (this is called 'direct' inventory gathering, since it does not involve any form of the FlexNet Inventory Agent).

     Tip: *Note that the list of these servers is not visible on the* ***Discovery and Inventory Rules*** *page in the web interface, but functions as a hidden target called* `Known vCenter or OVM Manager servers` *in the web interface: its use is controlled through a check box titled* ***Collect inventory from VMware vCenter or Oracle VM Manager servers*** *which is described in IBM High-Frequency Scanning in the* ***IBM reporting and archiving settings*** *section of the* ***Inventory Settings*** *page.*

     When the **Collect inventory from VMware vCenter or Oracle VM Manager servers** check box is selected and the default `Known vCenter or OVM Manager servers` option is selected in the accompanying drop down list, this automated inventory, used to track the movement of virtual machines, is collected on the same frequency as the inventory of the devices running software attached to an IBM sub-capacity license. This target is populated from the discovered device records, so you must ensure discovery of all relevant servers before the target becomes effective. This hidden target automatically updates to include all discovered vCenter or OVM Manager servers, so that newly-installed sub-capacity-related software can be identified in dynamic environments. Alternatively, you may prefer to create your own custom target(s) that identify only those vCenter or OVM Manager servers known to manage VMs running PVU-related software. This enables you manage scanning, limiting it to only managers of inventory devices relevant to IBM sub-capacity licenses; but it also means that if you later deploy additional managers of VMs that should consume from IBM sub-capacity licenses, you must remember to update your targets to avoid creating an audit risk. (For details about creating

targets, see the Creating a Target page.)(For details about creating targets, see the *Creating a Target* page in the online help.)

> ***Remember:*** *Remote inventory gathering from virtual hosts requires that you have saved credentials for these servers in the Password Manager on the inventory beacon accessing the servers. (For details about using the Password Manager, see Password Management Tab.)(For details about using the Password Manager, see the Password Management Page page in online help.)*

> ***Tip:*** *IBM also allows that, in addition to VMware and Oracle hosts, several other virtual hosts are acceptable for sub-capacity consumption of IBM PVU licenses. However, these hosts do not require the same direct inventory gathering as the VMware vCenter and Oracle VM Manager servers described above. For example:*
>
> - *For Hyper-V hosts, the FlexNet Inventory Agent locally installed on the host server is able to collect all required information about the deployment of VMs. These hosts are automatically included in the target described below, rather than in the target for direct inventory.*
> - *For partitioned technologies other than Solaris zones, the FlexNet Inventory Agent gathers sufficient information from the partition, and separate inventory of the host is not required for IBM sub-capacity calculations (although, of course, you may have the FlexNet Inventory Agent installed on the host to gather regular software and hardware inventory for that device). As you expect, partitions reporting inventory from the FlexNet Inventory Agent are also included in the automatic target described below.*
> - *For Solaris zones, it is critical that the FlexNet Inventory Agent is installed in the global zone as well as in the non-global zones. Only the global zone inventory reports the required number of processors and cores. If inventory from the global zone is missing, the non-global zones on this host do not contribute to the peak consumption calculations (because of the missing core and processor counts), and this represents an audit risk. When the FlexNet Inventory Agent is correctly installed in all zones, they are all automatically included in the following target.*

2. Computers running software attached to an IBM sub-capacity license (a target called `All devices consuming IBM PVU points`, which, despite the name, actually collects all devices consuming IBM sub-capacity licenses, both IBM PVU and IBM VPC where applicable) — these computers require the FlexNet Inventory Agent installed locally (called 'adoption' when you are defining targets, although you may use other methods of deployment). In production, use of this target is required for compliance with IBM's conditions.

> ***Tip:*** *The special target `All devices consuming IBM PVU points` is not visible on the* ***Discovery and Inventory Rules*** *page in the web interface. It is available only in the* ***IBM reporting and archiving settings*** *section of the* ***Inventory Settings*** *page, which is described in IBM High-Frequency Scanning, and only when you have first selected* ***Enable frequent hardware scanning for IBM sub-capacity license calculations****.*

These two automated targets rely on data already reaching the central application server. To 'seed' these automated processes, you need rules that execute the initial discovery actions on the target devices. Naturally, these rules require additional targets manually prepared:

- A target for discovery of the servers running VMware vCenter and Oracle VM Manager (do not allow adoption on these devices).
- A target for discovery and inventory of devices running software attached to an IBM PVU or IBM VPC license. In the case of (only) this target, you may also choose to configure the target for 'adoption' of devices (that is, automatically installing the FlexNet Inventory Agent locally on the target devices); or you may prefer alternative

deployment methods.

- Hyper-V hosts may be included in the previous target; or you might prefer a separate target for managing these virtual hosts. You could allow adoption of these devices; or use third-party methods to install the full FlexNet Inventory Agent locally on the Hyper-V host.

During development or testing, you may need extra targets in special circumstances, such as:

- Your inventory management team is ahead of your licensing team, and wants sign-off that they are collecting all required inventory, even though IBM sub-capacity license set up is not yet complete. The system cannot use consumption against the missing IBM sub-capacity licenses as a data source to update the list of computers that need special, more frequent hardware scans. If the inventory team knows of computers that have relevant software installed, which in due course will be linked to an IBM sub-capacity license, you can create a temporary target for those computers so that frequent hardware scans start as quickly as possible, for compliance with IBM's requirements.
- Your licensing team is ahead of your inventory team. The IBM PVU and IBM VPC licenses all exist, and a complete list of the computers to manage with frequent hardware scans could be generated; but permission for increased hardware checks on some of the target devices has not yet arrived. In this case, do not use the built-in target `All devices consuming IBM PVU points`, but instead manage your own targets that include only computers where hardware inventory gathering is permitted. This could be a politically useful temporary measure; but you should not allow it to continue, as it does not meet IBM's conditions. You can identify computers that are consuming from IBM PVU licenses, but not yet subject to high-frequency scanning, in the **IBM PVU Out-Of-Date Inventory** report – there is no equivalent report for VPC licenses.

- **Scheduled:** While the schedule for inventory gathering by the installed FlexNet Inventory Agents is determined when you enable high-frequency scanning, you can separately configure your schedule for reporting periods (which govern how often you need to archive reports for IBM), for the switch-over date for reporting, and the length of time to retain historical data.
- **Enabled:** When all else is ready, you must enable high-frequency inventory mode for IBM sub-capacity licenses.

Operation in High-Frequency Mode

Later topics give details about how to set up high-frequency inventory checking and peak consumption calculations for IBM PVU, IBM VPC, and IBM Cloud Pak licenses, allowing you to use IT Asset Management to calculate sub-capacity license consumption.

Before starting on those details, here is an outline of the normal operation of this process, starting from when setup is completed. This overview helps you understand the set-up steps in the following topics. The overview starts with the time of transition, and continues into the period when you are operating in full high-frequency mode.

1. As part of your preparation, you have deployed the FlexNet Inventory Agent to all target inventory devices where sub-capacity licensing applies. To get matching results from the two inventory systems, you need matching coverage with the two separate agents (at least during transition).

   Remember: *The FlexNet Inventory Agent must also be installed on the host for Hyper-V and in the global zone for Solaris zones.*

2. When you enable the high-frequency facility (as will be described in Turn on High-Frequency Mode), you choose

one or more targets that identify the inventory devices running software attached to IBM sub-capacity licenses. In normal operation, you require only one target, since this target is automatically maintained to include all devices known to be running this software.

3. As the facility is enabled, the device policy is updated for all those devices within your chosen target(s), and downloaded to all inventory beacons (technically, the setting `IBMPVUEnableScanning=true` is added to the `BeaconSettings.xml` file, which embodies the change to inventory beacon policy). By default, the inventory beacons are updated every 15 minutes, an interval you can modify in the **Beacon settings** section of the **Inventory Settings** page (**Data Collection > IT Assets Inventory Tasks > Inventory Settings**).
4. As each installed FlexNet Inventory Agent requests its latest policy, each inventory beacon updates any instances from which a request was received (that is, policy updates are not restricted by the subnets assigned to each inventory beacon).
5. This policy update triggers the installed FlexNet Inventory Agent to check for changes to the hardware configuration on the device where it is installed, and to report any changes in a hardware inventory file uploaded on the schedule you selected (the IBM requirement is every 30 minutes).
   - By default, an inventory file is uploaded only when there is a change in a relevant hardware property.
   - Each device also has a blocklist of hardware properties for which changes do *not* trigger an inventory upload. If need be, you can customize the list of ignored hardware changes (provided that you do not conflict with your amended license from IBM). For details, see Advanced Agent Configuration.
   - If there is changed hardware, but a transient failure causes the resulting inventory upload to fail, the archived inventory file (`.ndi.gz`) is stored locally on the inventory device. Further hardware changes in the same day overwrite this stored data with the latest inventory; but more typically, every 30 minute check finds no further hardware change, and the stored inventory archive persists until the uploader component (`ndupload`) of the FlexNet Inventory Agent attempts a scheduled catch-up overnight.

     Note: *The uploaded data is time-stamped at the date and time it is resolved into the inventory database, not at the time the inventory was collected on the target inventory device. Network up-time is a valuable asset in this mode.*
   - On Hyper-V hosts and Solaris global zones, the locally-installed FlexNet Inventory Agent uploads details of the guest VMs managed by the host.
6. The uploaded hardware inventory data travels up through the hierarchy of inventory beacons to the central application server, where it is immediately resolved into the inventory database.
7. On the same high-frequency schedule, appropriate inventory beacons remotely access your VMware virtualization hosts, and check for any changes to VM hosting. Any relevant inventory is also uploaded to the application server and resolved into the inventory database. (This inventory rule uses a separate, hidden, and automatically-maintained target of all relevant VM hosts identified from the **All Discovered Devices** listing.)
8. Once per day, the full license reconciliation process is performed:
   a. Software data and hardware data from all inventory sources are imported and merged. For accurate merging, it is critical in this mode that FlexNet inventory is registered as the primary data source.
   b. The optimal assignments of software installations to available licenses are calculated, linking the inventory device for each installation to the appropriate license.

Note: *In the special case of IBM PVU or IBM VPC licenses, the availability of unused entitlements (or excess purchases) does not affect the assessment of a target license to which to link a device. Other remaining factors which do affect the linking of inventory devices to licenses include:*

- *Any enterprise group restrictions on the license and inventory device*
- *The availability of hardware details (such as core counts) required to calculate consumption for the inventory device*
- *Whether or not the inventory device is hosted offsite by a cloud service provider – for an instance hosted in a cloud service provider, the core count is typically replaced by the vCPU count (thread count) assigned to the instance, where the vCPU count is available.*

When all appropriate factors are considered, the link between a device and a license is made, and the choice of license may (or may not) affect the outcome of the regional peak consumption.

c. For every inventory device now linked to an IBM sub-capacity license, the FlexNet process now calculates two consumption figures:

- The current full capacity consumption by that device for the license
- The current sub-capacity consumption for the device. (For stand-alone physical devices that are not eligible for sub-capacity calculations, this is a duplicate of their full capacity figure.)

Tip: *For individual devices that are not known to be* ***Eligible for sub-capacity*** *(so that this column in the* ***Consumption*** *tab on the license properties is either blank or displays No), you can manually enter an* ***Overridden consumption*** *figure. (Best practice is that you then document your reasons for this highly unusual behavior in either or both of the license properties and the inventory device properties.) If you have done this, the overridden figure is always used, regardless of any of the following logic.*

d. Consumption is then summed, taking into account the settings in the license properties sheet, on the **Use rights & rules** tab, in the **Rights on virtual machines and hosts** area:

- If the radio button **Always use full capacity license calculations** is selected, the full capacity consumption figure is used for every inventory device.
- If the radio button **Use sub-capacity license calculations where available** is selected, for every device where inventory was collected by the full (and up-to-date) FlexNet Inventory Agent locally installed, the sub-capacity consumption is included in the summing process. (If, as required, the inventory device has been assigned to a location which is a child of one of the mandatory IBM regions, its sub-capacity consumption is added into its region's subtotal.)

Tip: *If you are using a custom points rule, and that custom rule includes fractions of points (such as "15.25 points per core" for a particular host type), be aware that the fractional results of consumption on all servers are maintained throughout this summing process. Ultimately, rounding up is applied only to the final regional total, where it may be reported to IBM. For example, in an IBM region with just two VMs running software under a PVU license, and using 2 cores (30.5 points) and 5 cores (76.25 points), the regional total is 106.75 points, which is then rounded up to 107 points for the regional report. This is just less than would be the case if rounding were applied to the individual devices (108 points).*

If the inventory device is not yet assigned to an appropriate location, its consumption is included in

`Devices not assigned to any region`, and you should correct its location assignment as soon as practical, and certainly before submitting any report to IBM. In this mode, for inventory devices found in any source other than FlexNet Inventory Agent, the decision rests on the additional check box:

- If **Allow sub-capacity licensing for sources other than IT Asset Management** is clear (not selected), the figure used for inventory devices from other (non-approved) inventory sources is their current full capacity number.
- If **Allow sub-capacity licensing for sources other than IT Asset Management** is selected (checked, or ticked), the figure used for each inventory device linked to this license is its current sub-capacity number.

Important: *Use this setting with care. It is not normally an option approved by IBM. If you have due cause and approval for this setting, be sure to archive written justification that can be provided in an audit. Also be aware that historical inventory is maintained only for inventory collected by the locally-installed FlexNet Inventory Agent. Therefore, for these other inventory sources, only the figure from the most recent license compliance calculation (whether sub-capacity or full capacity) is available, and used. Normal practice is to ensure that the* ***Allow sub-capacity licensing for sources other than IT Asset Management*** *is clear (not selected).*

e. The sum of consumption across devices for the license becomes the current **Raw consumption** figure.

f. The 30-minute-interval calculations of sub-capacity points consumption are re-evaluated for the entire data retention period, taking into account all the latest retroactive data corrections.

Data corrections that are retroactive include:

- Ownership of devices by locations
- Mapping of locations to one of the three mandatory IBM regions.
- License allocations
- Changed inventory device exemptions (for example, correctly identifying testing servers)
- Manual overrides on inventory-reported core counts (edited on the **Hardware** tab of the inventory device properties)

Tip: *These hardware overrides are retroactive and permanent (until manually changed). However, in the absence of manual overrides, a change in raw incoming inventory data is not retroactive, but takes effect only from the inventory date when it was recorded. This is because the historical, time-based data for PVU consumption calculations includes processor/core changes (as well as factors like virtual hierarchy changes). For example, suppose that:*

- *Your reporting period starts on January 1*
- *Inventory device A is then reporting 2 processors, each with 8 cores (total core count is 16)*
- *On February 3, you populate two empty sockets with 2 more processors, each also with 8 cores (total core count is now 32)*
- *The reporting period ends March 31.*

The peak consumption calculations for device A in this reporting period use 16 cores between

January 1 and February 2 inclusive, and 32 cores between February 3 and March 31 inclusive. So changes in incoming inventory values are not retroactive, but take effect only from the dates when the changes occurred; and this contrasts with manual overrides of inventory values, which are retroactive and permanent.

As a result of any such retroactive data corrections, this reassessment during the full compliance calculations may result in changes to the regional peak consumption values, or the dates on which they fall. Naturally, these adjusted values are also available in the appropriate reports, ready for you to produce as required for archiving and potentially presenting to IBM:

- **IBM PVU License Consumption**
- **IBM Cloud Pak License Consumption**.

g. The **Consumption** tab of the license properties is updated with the current points consumed by each device; and the **Compliance** tab is updated with the peak results for the three IBM regions (and a fourth result for any consuming devices not yet owned by a location that is mapped to an IBM region — you should attend to these devices and locations as soon as possible, and certainly before archiving any reports for IBM). These regional peaks at their independent dates (and any unassigned peak) are added together to produce the global peak value for the IBM PVU or IBM VPC license in the reporting period, shown as **Peak consumed** in the **Compliance** tab.

Tip: *Because you are now in high frequency scanning mode, any imports from ILMT of consumed points for IBM PVU licenses that it has calculated are no longer used to calculate license consumption within IT Asset Management. However, while ILMT remains available as a secondary inventory source, the consumed points imported from ILMT are available for comparison in the* ***ILMT and FlexNet Manager License Positions*** *report (in the* ***License Compliance****,* `Compliance` *group when you are in* ***Reports*** *mode), which can compare the sub-capacity consumption calculated by ILMT with the current calculations by IT Asset Management for IBM PVU licenses. (Recall that there is no equivalent import of ILMT consumption results for IBM VPC sub-capacity licenses.)*

Note: *When using high frequency scanning mode (so that IT Asset Management is the source of truth for sub-capacity PVU points calculations) and ILMT remains as a secondary inventory source, devices marked within ILMT as deleted are never imported. In the special case where:*

- *A device has previously been imported, but only from ILMT, and therefore*
- *There is a computer record for this device within IT Asset Management, and*
- *The device is subsequently marked as deleted within ILMT, so that it is no longer included in future imports from this inventory source*

the device record within IT Asset Management is given the special status of `Archived`*. A record with this status is used only for calculations of peak consumption in the current period. Typically, at the start of the next reporting period, the device is no longer relevant, and is automatically removed from IT Asset Management.*

9. For both IBM PVU and IBM VPC licenses, the value of **Peak consumed** is always reflected on the **All Licenses** page in the **Consumed** column, being the license metric of interest to IBM for these license types.

10. As a business process, regularly check IBM PVU and IBM VPC licenses for any new devices in the **Consumption** list, and ensure that any applicable optimizations are applied promptly.

For example, brand-new inventory devices can be identified in the **Active Inventory** page, by adding the **Created** column from the column chooser, and sorting or filtering for the date(s) of interest. You can then double-click the name of any new device to open its properties. Glancing at the **Licenses** tab shows whether the most recent reconciliation linked this inventory device to any license of **License type** `IBM PVU` or `IBM VPC`.

11. On the dates configured for your reporting period roll-over (in **System Settings > Licensing** tab), the peak consumption calculations switch over to the new period (peak values are reported independently for each reporting period). On the first day of the new period, therefore, the *current* peak values (from the overnight compliance calculation) are also the *period* peak.

    Tip: *Where a region's peak consumption value is sustained over multiple days, the date reported for the peak is the last day within this reporting period when the peak applied. Therefore if consumption has not yet fallen below the peak, the date shown is when the most recent compliance calculation was run (typically yesterday or today). If a region's consumption is steady-state, the peak date may continue to show today's date throughout the entire reporting cycle.*

12. At the end of each reporting period (typically a quarter, although IBM may sometimes require monthly reporting), access and save the following report package and data views:

    - The digitally-signed IBM audit report package provided by IT Asset Management. This package can be used as a trusted source for audit as the data contained in the package can be validated for authenticity. It meets IBM's reporting requirements and is accepted by IBM. The package includes the following reports and can be downloaded from any of these report pages:
      - **IBM PVU License Consumption** report
      - **IBM VPC License Consumption** report
      - **IBM Cloud Pak Consumption** report
    - **Unlicensed Installations**(**Licenses > License Management > Unlicensed Installations**), filtered by **Publisher** = `IBM`. This data view lists installations of IBM software for which an applicable license has not been identified.

    For more information, see Reporting to IBM.

    Tip: *Remember to safely archive your reports in case of a future audit. Do this shortly after you roll over to a new reporting period, archiving the reports for the period recently closed. This is because the default data retention period is twice your reporting cycle, meaning that you can always report on the **previous** reporting period; but using the default settings, you are not able to report on the period before last (that is, two periods ago). To limit database growth, historical records are automatically deleted during the full compliance calculation after they fall outside the data retention window. Once you have archived your reports, IBM requires that you preserve them for a minimum of two years.*

Configuring Regions for IBM

To prevent 'follow-the-sun' licensing, IBM requires that you separately license target devices in each of three regions that it defines:

- Region 1: North America and South America
- Region 2: Europe and Africa

- Region 3: Asia and Australia.

For any given IBM PVU or IBM VPC sub-capacity license, IT Asset Management automatically tracks consumption within the three regions, provided that:

- All the inventory devices that are consuming from the license have been assigned to a *location* (a type of enterprise group), and
- The locations used for this purpose have been mapped to the IBM regions.

The set-up of locations and their mapping to regions is done in the **IT Asset Locations** page (**Organization > IT Asset Locations**); and inventory devices are linked to locations through the **Ownership** tab of their inventory device properties.

Tip: *By default, the location assigned to a host server is inherited by all the guest VMs running on that host (as is suitable, for example, for a departmental server and VMs). In this case, the* ***Location*** *field for the guest VM is a read-only display of the inherited value. If, instead, you want to assign VMs to locations different from their hosts (as is typical for a host in the IT server room hosting VMs for several different locations), go to the IT Asset Management Settings* ***General*** *page (****Administration > IT Asset Management Settings > General****), and in the* ***Inventory*** *tab, clear the check box for* ***Update virtual machine location to match host location****; and finally scroll down to click* ***Save****. You may then open the inventory device settings for any guest virtual machine, edit the* ***Location*** *value on its* ***Ownership*** *tab, and save the change.*

Of course, the use of locations is not limited to representing the three mandatory IBM regions. You may want to set up an entire structure of locations, where a *region* contains *countries* that contain *states* that contain *cities* that contain *offices* that are part of your enterprise. At the other extreme, if you are not using locations for any other purpose, you can minimally create just three locations, one for each of the IBM regions. The level of sophistication you implement is entirely up to you: in all structures, IT Asset Management rolls up the consumption data to provide correct regional totals for your reporting to IBM.

Note: *You cannot separately control the change-over from ILMT to IT Asset Management one region at a time. While you need to report by regions, your entire enterprise gets its points consumed results from one source or the other, and makes the change-over at the same time.*

This summary assumes that you do not already have appropriate locations specified.

To create a region-based enterprise structure:

1. Go to the **IT Asset Locations** page (**Organization > IT Asset Locations**).
2. In the **Actions** column of the `Locations` row, click the **+** icon.

   A new row appears, with editable fields for the **Name** and **Description** columns.
3. Type new values in these two fields.

   For example, if you are using locations *only* for tracking IBM regions, your first entry may be:

   - **Name**: `IBM Region 1: Americas`
   - **Description**: `Licensing roll-up for IBM PVU license management.`

   However, if you are using locations for other kinds of administration, it may be any location you wish. Choosing a high-level location is convenient, since the IBM regions by default are inherited by lower-level, 'child' locations.

In this case, you might choose an entry like `North America`.

4. When satisfied, hit `Enter` in either field, or click the blue disk (save) icon at the right end of the row.
5. With the row still selected, click **Assign IBM region**, and select one of the regions from the drop-down list to set the value for this location.
6. Repeat to create (at least) the remaining two top-level regions.
7. Optionally, select one of these regions again, and click the **+** icon to add child locations within a region. Repeat as often as needed to create your initial hierarchy of locations that roll up into the IBM regions.

This structure may now be applied to any local purchases, to the licenses you are about to create (see Configure Appropriate Licenses), and to your inventory device records (editable in the **Ownership** tab of the inventory device properties). For example, to set a single regional location value for many inventory devices at one time:

1. On a suitable inventory listing (such as the **All Inventory** page), search and filter to an appropriate set of devices.
2. Make sure that all the displayed devices are of the same type. For example, filter the **Inventory device type** column to a single value (say, `VM Host`).
3. On the left, select the devices that belong in the same location (but do not mix rows where the **Connection name** is blank with others where it is non-blank).
4. Click **Open** to display a property sheet of properties common to the selected devices.
5. Select the **Ownership** tab, and enter (or search for) the common location for these devices.
6. Click **Save**, and repeat as required for other groups of inventory devices.

Configure Appropriate Licenses

Naturally, you require appropriate license records of type `IBM PVU` or `IBM VPC` for all relevant IBM software. Less self-evident is that the license(s) must be operational (that is, have consumption recorded against them) before you can configure them to meet IBM's special requirements for high-frequency scanning. This makes it convenient to have the license(s) in place before you collect your first (or next) inventory from the computers running the relevant IBM software. In this way, the initial inventory collection and license reconciliation can immediately make your IBM sub-capacity license(s) operational, showing consumption against the linked application(s), and enabling the special configuration required for sub-capacity licensing.

For the IBM PVU license type, you may already have licenses in place, for example if they have been created automatically based on data imports from your ILMT server.

Tip: *If you deleted a license record that had been created to match imports from ILMT, this license is not recreated on subsequent imports from ILMT. If you need to 'recover' this license, create it manually.*

If you do not yet have the licenses in place, you can create them in any of the usual ways. Best practice (described here) is to drive license creation through your purchase records, since this provides an acceptable audit trail of your entitlements, and allows automation to configure many of the product use rights on the license for you. This is particularly powerful if you have the correct SKU (stock keeping unit) identifying codes for the IBM products you purchased. One example is that, when the correct SKUs are available, IT Asset Management defaults to creating full-capacity licenses for applications that are ineligible for sub-capacity licensing (whereas, when creating a license manually, you have to select the appropriate full capacity or sub-capacity template for the application, perhaps making

use of IBM's Passport Advantage Sub-capacity Licensing Eligible Product Statement).

However, creating licenses from purchase records is not mandatory, and you may proceed in other ways, such as manually creating license records and linking purchase records to them later. (To create a license manually, go to the **All Licenses** page (**Licenses > License Management > All Licenses**) and click **Create a license**, and see the *Creating a License* topic in the online help.)

Important: *Keep in mind that IBM requires separate consumption calculations and reporting in each of the major IBM-defined regions:*

- *Region 1: North America and South America*
- *Region 2: Europe and Africa*
- *Region 3: Asia and Australia.*

If your enterprise consumes IBM sub-capacity licenses across more than one of these regions, the best way to manage this is to:

1. *Ensure that you have set up enterprise groups based on location that can roll up into the appropriate IBM regions. (If you do not use locations for other purposes, it may be sufficient to set up just the top three, one per region.) For details, see Configuring Regions for IBM.*
2. *Assign the inventory devices consuming PVU points or VPC entitlements to an appropriate enterprise group (location), as noted at the end of Configuring Regions for IBM.*
3. *Best practice is to create just one IBM sub-capacity license worldwide for each product (or bundle). This allows for the correct tracking of points/entitlements across regions, and for correct totalling of the liability for the license.*

Tip: *If you have strong reasons for maintaining multiple licenses per product (such as a history of mergers and acquisitions leading to separate reporting to IBM for different entities), use a different kind of enterprise group, such as corporate units, to separate purchases and licenses into the appropriate corporate structure. Overlay the same system of locations described here so that device consumption is correctly rolled up into regions for each of your corporate divisions.*

Consumption is then automatically calculated for each of the regions reported on each license. Any consuming device that is not owned by a location suitably linked to an IBM region is flagged in the ***Consumption*** *tab of the license properties, and included in a separate subtotal for unassigned devices in the* ***Compliance*** *tab of the same license. You should attend to the ownership of these devices as soon as possible (and before reporting to IBM), since a change of regional assignment may change the peak consumption figures (and possibly also the peak dates) for the affected regions.*

One additional factor to consider is whether to create single product or multi-product licenses (the latter are used to track software bundles). If your licenses were created automatically to match imports from ILMT, bundles are (by default) automatically matched. This is also the case if your purchase includes a SKU that identifies a bundle. A multi-product license shows applications in the **Applications** tab that list multiple values in the **Product** column (which is why it is called a *multi-product* license); and the **Use rights & rules** tab is also configured differently, showing the use rights for each product, including whether each one is primary or supplementary. Supplementary products can be configured so that they do not consume license entitlements, being covered by the licensed primary product. If you are manually configuring license use rights and restrictions, a useful resource is the License Information Document (LID) for your product. You can search for these LIDs on the IBM's License Information Document Search page.

To create licenses by processing purchases (summary):

1. Do one of the following:
   - Import purchase records from your purchasing system using a business adapter (see the chapter on *The Business Adapter Studio* in the *Using FlexNet Business Adapters* PDF for details).
   - Download your detailed Order History from IBM Passport Advantage, and import it (go to the **Data Imports** page (**Data Collection > IT Assets Inventory Tasks > Data Imports**), select the **Business Data** tab, click **One-off upload**, and see *Purchase Order Upload* in the online help).
   - Create a spreadsheet of purchase records, and import the spreadsheet (similar place in IT Asset Management, and described in the same help page).
   - Create a purchase record manually (go to the **All Purchases** page (**Procurement > Purchases & Vendors > All Purchases**) and click **Create a Purchase**, and see *Creating a purchase* in the online help).

   In all cases, be sure to secure the documentary evidence of your purchases against the possibility of a future audit. (One possibility is scanning the original purchase document and attaching the scan to your purchase record using the **Documents** tab of the purchase properties.)
2. Go to the **Unprocessed Purchases** page (**Procurement > Purchases & Vendors > Unprocessed Purchases**), and select your new purchase record(s) there.
3. Click **Recalculate** to generate recommendations for processing the selected purchase(s). If you are happy with a recommendation, select that row and click **Accept**.

   If there are no recommendations (perhaps because you do not have a SKU and have not previously processed a similar purchase), select the purchase and click **Process** to step through the processing wizard instead.

   At the end of purchase processing, your purchase has been linked to an appropriate license. If your purchase record included a known SKU, most of the product use rights on the license have been configured for you. Typically, the license properties are now open.
4. In the properties sheet of your IBM sub-capacity license, select the **Use rights & rules** tab, and expand the section **Rights on virtual machines and hosts**.
5. For sub-capacity licensing, make sure that **Use sub-capacity license calculations where available** is selected.

   Tip: *Not all software is eligible for sub-capacity licensing. Check the terms of your original license agreement. (When a particular license is for an application that requires full capacity licensing, instead select* ***Always use full capacity license calculations*** *for this license.)*
6. If you plan to use any exemptions *based on device roles*, stay on the **Use rights & rules** tab, expand the section **Exemptions**, and select any device roles for which the license agreement allows a license-free installation.

   Later, when device inventory has been collected, you can assign the correct roles to appropriate inventory devices (see Set Up and Collect Inventory, and Reconcile). When the roles are correctly configured, IT Asset Management automatically exempts the appropriate devices from license consumption. (An alternative approach, adding individual exemptions directly in the **Consumption** tab of the license properties, has already been described in Additional Transition Steps.)

   Tip: *In an audit, you may be asked to substantiate these settings. A helpful audit trail may be, in the license*

*properties, to add comments in the **Notes** on the **Identification** tab, or attach a scan of the relevant part of the license agreement to the **Documents** tab.*

7. Check the **Applications** tab of the license properties to ensure that the IBM application you are expecting is linked to the license.

   This must be an application for which you expect to see an installation record after your forthcoming hardware and software inventory. The license must have a linked application for which entitlements are being consumed after the next inventory import and license consumption calculation.

8. Above the tabs in the **License Properties** page, click **Save** to store all your edits in the database.

9. Loop back to process the next available purchase record.

Whether by this process or one of the alternatives, you should end up with one IBM sub-capacity license per product (or per bundle, where appropriate), allowing automatic roll-up of consumption from devices 'owned' by each IBM region; and the license should be linked to at least one purchase that provides the initial stock of entitlements. You may link additional purchases to the license at any time, reflecting your purchases from IBM (or its dealers).

Managing IBM RVU MAPC Licenses

As you prepare to manage your IBM PVU licenses, be aware that IBM auditors typically require reports of your IBM RVU (Resource Value Unit) license consumption at the same time as you deliver your IBM PVU reports.

Because IBM RVU may measure a wide variety of resources not directly measurable in your computing estate, these licenses are not affected by imported inventory, and their consumption is not automatically updated in the overnight license reconciliation calculations.

However, there is one measure (or metric) available in IBM RVU licenses that is of particular interest to IBM auditors looking at sub-capacity license consumption: Managed Activated Processor Cores (MAPC), identified within IT Asset Management as Activated Processor Cores. Like IBM PVU licenses, this metric introduces the concept of the *highest* number of RVUs (in this case, processor cores) consumed by a licensed application – that is, management over *time* becomes important.

In summary, the generalized IBM *RVU* license type does not support automatically updating the peak number of processor cores; but this is exactly the kind of historically-aware calculation supported by the IBM *PVU* license type. As a result, you should use the following approach for automatically calculating consumption of MAPC units. The basis is to use a PVU license type to automatically track the RVU processor cores for you.

To manage consumption and reporting of MAPC in IBM RVU licenses:

1. Create an IBM PVU license (not a typo – PVU), and link this to the relevant application(s).

   ***Tip:** If you are using automatic creation of licenses from purchases, you may already have an IBM RVU license created for these applications. The easiest approach then is to change the **License type** setting on the **Identification** tab of the license properties. Thereafter, you must follow steps 2 and 3 below, and then wait for the next overnight reconciliation (or run an additional one, if you are in a role with administrator privileges) to restore correct consumption results for your modified license(s).*

   Consider giving this license a name that helps you to manage it. For example, include something like `RVU-as-PVU`, or `[RVU MAPC]`, in the license name, so it is quick and easy to identify in license lists.

2. On the **Identification** tab of the license properties:

   a. In the **Points rule set** field, enter `MAPC` and click **Search**.

   b. In the result table, select the row with **Points rule set name** showing `MAPC`, and click **Add points rule set**.

      This rule set contains just one rule, setting one point for every processor core (with all other parameters set to match any values).

   c. Click **Save** to store the updated properties.

3. From the **Applications** tab, open the properties of each of the application(s) linked to the license; and use the **Licenses** tab from the application properties to check either that only this one license is linked to each application, or else that you have dragged this license to the top of the license priorities. Remember to **Save** any changes you make to the application properties.

   Now, with each nightly reconciliation, all devices reporting installations of the application(s) linked to this special license have their reported available/assigned processor cores tracked (as 1:1 matching points). If there are changes in the number of cores available (on physical machines) or assigned (to virtual machines), this PVU license (as always) tracks the *peak* value, and when it occurred in the reporting period. As always for PVU licenses, total points are rolled up for each of the IBM regions, based on the location you assign to each device (in the **Ownership** tab of its device properties).

   When it comes time to report license consumption to IBM, you should take the following approach:

   - If your calculated consumption is below the 1:1 points tier (often, this means below 2500 cores – see the IBM Programs LID terms / Announcement letter to confirm), you may simply archive all your PVU reports (including the special `RVU-as-PVU/[RVU MAPC]` license). Be prepared to explain to IBM (or an auditor) that this special license (included in your PVU reports) is counting your RVU core points. Remember to archive your PVU reports at the end of each reporting period.
   - If your calculated consumption exceeds the 1:1 points tier (often where consumption is above 2500 cores – see the IBM Programs LID terms / Announcement letter to confirm), you will want to ensure, when reporting your consumption, that you benefit from applying IBM's license tiering to the reported RVU points. In this case, follow the additional steps below.

4. To apply tiered calculations to your RVU points consumption above 2,500 points:

   a. Create a new `IBM RVU` license (or if you have previously create one for this purpose, open its properties, and move ahead to step 4.e).

      For a new license:

      - Give it a useful name that reflect both the product being licensed and that this license is purely for "RVU tiers reporting only".
      - Do not link the license to any application, since the only function of this license is to apply tiers to the existing calculated MAPC points. Similarly, there is no need to link any devices to the license. Just having a "shell" of a license with no devices, and no linked application, avoids all risk of confusion. Best practice is to associate your purchases with this license, so that your calculated tiered consumption can be reflected against your entitlement as a reconciled license position.

      Note: *When your purchase of RVU entitlements is linked to this secondary, "RVU tiers reporting*

only" license, the original `RVU-as-PVU/[RVU MAPC]` license shows consumption with no purchased entitlements, and therefore shows as a license 'At Risk'. You should ignore this 'At Risk' status, knowing that the entitlements are attached to the secondary RVU license, which is reporting accurate RVU MAPC points consumption after the correct tier calculations are applied.

b. Switch to the **Identification** tab of this RVU tiers reporting license.

c. In the **Metric** drop-down, select `Activated Processor Cores`.

d. In the **Points rule set** control, enter some part of `Activated Processor Cores`, and click **Search**. Select the matching row from the results, and click **Add points rule set**.

This license is now configured to apply tiers to your RVU MAPC consumption. You may continue with the following at any time to report your RVU MAPC consumption, and in particular, follow these steps on the last day of each reporting period.

e. Open the license properties for your `RVU-as-PVU / [RVU MAPC]` license, select the **Compliance** tab, and copy the total number of points consumed from the **Peak consumption since** ***`periodStartDate`*** field. Save this in a document for safe-keeping.

f. Switch to the properties of your "RVU tiers reporting only" license, and select the **Identification** tab.

g. In the **Resources consumed** field, paste the total points previously copied from the **Compliance** tab of the `RVU-as-PVU / [RVU MAPC]` license.

h. Click **Save** to store your changes in the compliance database.

i. Wait for the overnight compliance calculations.

In that next calculation:

- The `RVU-as-PVU / [RVU MAPC]` license continues to calculate and report consumption on a 1:1 basis.
- If your RVU **Resources consumed** value exceeds 2,500 cores (where the first tier of 1:1 cores to points ends), the secondary RVU license you created for calculation and reporting of *tiered* license consumption applies the correct tiers from the points table to calculate the tiered RVU MAPC consumption. These tiered calculations explain the difference between the **Resources consumed** field (your input) and the **Compliance** tab **Consumed entitlements** value (the result). For example, suppose your **Resources consumed** value is 5,600 cores. The first 2,500 cores cost you one point each, and from there up to 10,000 cores cost 0.8 pt each:
  - First tier 2,500 cores x 1 pt = 2,500 pts
  - Second tier (5,600 - 2,500) cores x 0.8 pt = 2480 pts
  - Total points for 5,600 cores = 4,980 pts.

5. Archive all reports, as described in Reporting to IBM.

Tip: *The second, 'RVU tiers reporting' license can now be left idling until required in the next reporting period.*

6. Make sure that your new reporting period has started as expected, checking in the IT Asset Management Settings **General** page (**Administration > IT Asset Management Settings > General**) > **Licensing** tab, under **IBM PVU sub-capacity calculation settings**.

7. Set your process management calendar reminder ready for the last day of this reporting period, so that you can repeat the ritual and keep all your archived reports in order.

Advanced Agent Configuration

By default, the FlexNet Inventory Agent is correctly configured when you turn on high-frequency mode for calculating sub-capacity consumption of IBM PVU and IBM VPC sub-capacity licenses (for details, look ahead Turn on High-Frequency Mode, but do not omit the intervening topics before turning on).

For special circumstances, you may wish to change the default behavior around the checking of hardware inventory, as described below. To understand those preferences, it is also helpful to know the default command line used for the FlexNet Inventory Agent in high-frequency mode.

Command line

The frequent hardware scanning relies on a custom command line for the installed FlexNet Inventory Agent. This command line cannot be altered. It uses the options shown in the following example:

```
ndtrack.exe -o WMI=true
            -o Hardware=true
            -o ManageSoftPackages=false
            -o MSI=false
            -o PlatformSpecificPackages=false
            -o Software=false
            -o TrackProductKey=false
            -o IncludeRegistryKey=
            -o IncludeDirectory=
            -o EmbedFileContentDirectory=
            -o OnlyGenerateIfHardwareChanged=true
            -o PerformSymantecSFScan=false
            -o PerformIBMWebSphereMQScan=false
            -o InventorySettingsPath="
```

Enabling uploads only on hardware change

The option `OnlyGenerateIfHardwareChanged` controls whether the scanned hardware inventory is uploaded only when there are changes since the last report, or always uploaded. It is set to `true` in the command line above to minimize network traffic for additional uploads of hardware inventory. For more information, see the preference listing in *Gathering FlexNet Inventory*.For more information, see the preference listing in *Gathering FlexNet Inventory*.

Blocklisting irrelevant properties

The option `HardwareChangesClassPropertyBlocklist` identifies hardware properties that should be blocked or ignored in the check for hardware changes (that is, these values may change without triggering an upload of hardware inventory for this computer). This option has no effect if `OnlyGenerateIfHardwareChanged` is false.

The option has a string value made up of semi-colon-separated WMI classes. Each entry may be a simple class name, or a class name with trailing class property/properties (using a dot separator).

The option and its value may be stored in the Windows registry or, for UNIX-like platforms, in the `config.ini` file. It takes effect from there because it is not being overridden in the command line shown above. (For more details, see the preference listing in *Gathering FlexNet Inventory*.For more details, see the preference listing in *Gathering FlexNet Inventory*.) If the option is not specified, the following default value is automatically used (wrapped here for

readability):

```
Win32_OperatingSystem.FreePhysicalMemory.FreeVirtualMemory;
Win32_Processor.CurrentClockSpeed;
Win32_Processor.CurrentClockSpeedNonWMI;
Win32_LogicalDisk.FreeSpace;
SoftwareLicensingProduct;
MGS_OperatingSystem.LastBootUpTime.FreePhysicalMemory.FreeVirtualMemory
```

Set Up Virtual Inventory Tracking

IBM's requirements include that you must track movement of any relevant virtual machine when it moves between hosts. To do this, you also need to configure inventory gathering on virtual hosts. Three different approaches apply, depending on the methods that IT Asset Management can use to gather information about guest VMs:

- **Agent on host:** Where data about guest VMs is available to the FlexNet Inventory Agent locally installed on the host (as is the case with Microsoft Hyper-V), simply deploy the FlexNet Inventory Agent locally on the server, as you do to all other devices that may run software under IBM sub-capacity licenses. In this case, the virtual host is added to the target `All devices consuming IBM PVU points`, and is managed on the same high-frequency schedule that applies to all installations of FlexNet Inventory Agent on devices running software licensed under IBM PVU or IBM VPC licenses.
- **Agent on guest:** Where data about the VM/host relationship is available within the VM itself (as is the case for LPARs, vPARs, nPARs, containers and zones), the only requirement is to deploy the FlexNet Inventory Agent locally on the VM. This also automatically joins the `All devices consuming IBM PVU points` target.

  Tip: *For Solaris zones, only the global zone contains details about the VM/host relationship. You must ensure that the FlexNet Inventory Agent is locally installed on the global zone, as well as in any non-global zones where you want to collect inventory.*

- **Direct inventory of host:** For technologies where a locally-installed FlexNet Inventory Agent cannot access VM deployment details, but where there is an API available that exposes this data, IT Asset Management automatically maintains a separate target of such targets that require 'direct' inventory collection by an inventory beacon accessing the API. For these, you only need to configure discovery manually, since once a virtual host is discovered, it is automatically added to the (hidden) rule for high-frequency inventory gathering from these devices.

Tip: *If your virtual device is an instance hosted by a cloud service provider, make sure that the* ***Hosted in*** *property in the* ***General*** *tab of the inventory device properties is correctly set to identify the cloud service provider. This is typically automatic for AWS and Azure when inventory has been received from the FlexNet Inventory Agent installed on the instance; but for other cloud service providers, you must set the value manually (for example, this value is not returned for Google Cloud). When* ***Hosted in*** *is correctly set, the priority on properties used for calculating sub-capacity points consumption changes to the following:*

1. *If inventory returns the vCPU count (or thread count) as the number of logical processors assigned to the instance, this property is used and either:*
   - *For an IBM PVU license, multiplied by the points per core from the PVU points table for the appropriate processor type*

- *For an IBM VPC license, multiplied by the appropriate ratio (defined in the product use rights for the license):*
  - *If the IBM VPC license is configured for consumption per product, the ratio of each primary product within the Cloud Pak bundle to VPC entitlements is used*
  - *If the IBM VPC license is configured for consumption per device, the ratio of assigned/available cores to VPC entitlements is used.*

*This vCPU value is only ever used when the **Hosted in** property is populated.*

2. *If the above value is missing, inventory is checked for a core count, and if found, this is used to multiply by the appropriate points value or ratio.*
3. *If both of the above are missing, the **Consumed** count for this instance is forced to zero (since there is no audit-worthy data source); but if the processor count is available, this is used to populate the **Calculated consumption** field, as a general indication of likely consumption when the appropriate properties become available. In this case, the IBM sub-capacity license on which the incorrectly-recorded instance appears is flagged as problematic, because the consumption calculation is incorrect and likely affecting the region total consumption, possibly including its peak date and peak value (the primary factors affecting IBM sub-capacity license exposure). Keep in mind that, if you can determine and correct the vCPU or core count (perhaps using the override facility on the **Hardware** tab of the inventory device properties), this is taken as a retrospective correction and applied to the entire reporting period from the next overnight license compliance calculation.*

To set up discovery (and the resulting automatic inventory gathering) on virtual hosts:

1. For Microsoft Hyper-V virtual hosts, install the FlexNet Inventory Agent locally on the host server. (In addition, you should also install the FlexNet Inventory Agent in the primary image for the VMs, so that these also report relevant software inventory.)
2. For Solaris zones, ensure the FlexNet Inventory Agent is installed in the global zone, as well as in the non-global zones.
3. For partitioning technologies, ensure that the FlexNet Inventory Agent is installed locally on each partition. This provides both the inventory results (for software and hardware) and also the relationships between host and guests.
4. For VMware, in the web interface of IT Asset Management, ensure that every subnet containing a vCenter Server is assigned to an inventory beacon:
   a. Go to the **Beacons** page (**Data Collection > IT Assets Inventory Tasks > Beacons**).
   b. Choose the appropriate inventory beacon and click the edit icon on the right-hand end of its data row.
   c. Click the **Subnets** tab.
   d. Either search for a subnet, or enter the subnet IPv4 address in the **Find subnets to add** field.
   e. Select one or more subnets, and click **Add subnets**.
5. Set up the credentials to allow future inventory gathering from vCenter Server (managing ESX servers):
   a. On the appropriate inventory beacon, log in to FlexNet Beacon using an account with administrator privileges on the local device.
   b. Open FlexNet Beacon, and in the navigation bar, select **Password management**. In the resulting page, click **Launch Password Manager**.

c. In the FlexNet Beacon Password Manager, click **New**, and set up credentials for any vCenter hosts.

   For vCenter hosts, use an **Account Type** of `Account on VMware VirtualCenter`. When satisfied, click **Apply**.

d. Repeat for any other virtual hosts accessible from this inventory beacon (or any others). Be sure that you can reach all hosts that a relevant virtual machine can possibly move to.

e. When credentials have been established for all necessary virtual hosts, click **Exit** to close the Password Manager; and you may also exit FlexNet Beacon.

6. Back in the web interface, set up a target that covers all relevant hosts for virtual machines.

   Once the virtual hosts are discovered, IT Asset Management automatically maintains a separate target of all relevant 'direct inventory' virtual hosts, and automatically runs an inventory check on them at the same frequency that you specify for IBM PVU and VPC license peak calculations. To seed this process, you first need to ensure discovery of all VMware vCenter virtual hosts.

   a. Go to the **Discovery and Inventory Rules** page (**Data Collection > IT Assets Inventory Tasks > Discovery and Inventory Rules**). Click the **Targets** tab, and click **Create a target**.

   b. Provide a useful **Target name** and **Description** that will assist your future maintenance.

      For example, `Sub-cap Virtual Hosts` and `Initial Discovery` may help you identify this target for ongoing management. For example, since this target is used for discovery only, you may wish to leave the related rule running once a day into the future, so that it automatically picks up any new VMware vCenter hosts, and automatically integrates them into the high-frequency scanning protocol.

   c. Use the **Define machines to target** controls to specify the target virtual hosts (or the subnets in which they are connected). Add as many definition lines as required to cover all applicable virtual hosts. See the help for that page for further details.

   d. For the **Adoption options**, choose **Do not allow these targets to be adopted**.

      A locally-installed FlexNet Inventory Agent cannot access the management APIs necessary for tracking VMs. Instead, this target is used to specify *direct* inventory gathering, where an inventory beacon remotely accesses the hypervisor for data about the VMs it manages.

   e. When satisfied, click **Create** to save the target definition.

7. Create an action to trigger discovery of the targeted virtual hosts:

   a. Switch to the **Actions** tab, and click **Create an action**.

   b. Give your action a unique **Name**, and a **Description** that assists your future maintenance work.

      For example, `Virtual host discovery` and `Use this daily to capture any newly deployed virtual hosts where IBM sub-capacity licenses may be consumed.`

   c. For **Action type**, select `Discovery only`.

      After discovery, a separate (hidden) target is automatically maintained for inventory collection from discovered servers on the appropriate frequency.

   d. Under **Discovery of devices**, choose your preferred method(s) to identify the host servers.

      For most networks, you can use an ICMP echo test (ping) to specified ports by selecting **Network scan** and nominating one or more ports. If your VMware vCenter server(s) run on Microsoft Windows Server,

you might choose (either instead or as well) **Microsoft Computer Browser service**.

e. Under **Discovery actions**, expand **VMware infrastructure**, select **Discover VMware infrastructure**, and add/edit ports if required.

f. Click **Create** (scroll down to the bottom right corner of the page) to save the action definition.

8. Combine the target and action into a rule, and schedule its initial execution.

   Keep in mind that this rule exists only to seed automatic processes that take over at the right time. Specifically, you need this rule to succeed only once for each new set of target virtual hosts. This is *not* a rule you need to schedule every 30 minutes.

   a. Switch to the **Rules** tab, and click **Create a rule** to open the **Rule builder** area.

   b. Switch back to the **Actions** tab (or click an `Actions` hyperlink in the rule builder), scroll to find your saved action, and at the right-hand end of its row, click **Add to rule builder**.

      Because a rule may have only one action, all **Add to rule builder** buttons are now disabled, and a label for your chosen action appears in the rule builder.

   c. Switch to the **Targets** tab (or click a `Targets` hyperlink in the rule builder), scroll to find your saved target, and at the right-hand end of its row, click **Add to rule builder**.

      Notice that this time, you may insert more than one target into the rule.

   d. In the rule builder, click **Schedule**, and nominate your preferred schedule for this rule.

      Suggestion for the initial discovery:

      - **Frequency**: `As soon as possible`
      - **Commence within**: `10 minutes`.

      After the initial discovery, you may prefer to change this schedule to once daily at a quiet time, just as a safety net to catch additional hosts that may be deployed in future.

   e. In the rule builder, click **Save as**, and provide a **Rule name** that will remain meaningful in a long list of rules.

   f. Ensure that the next field displays `Enabled`, and click **Save**.

The following steps now occur:

- A new policy for inventory beacons is prepared and downloaded soon. By default, it is within 15 minutes. To customize this interval, go to the **Inventory Settings** page (**Data Collection > IT Assets Inventory Tasks > Inventory Settings**), and scroll down to the **Beacon settings** section.
- The policy is downloaded to all operating inventory beacons, each of which checks to see whether your rule needs to be executed by it (that is, whether there is any overlap between its assigned subnets and the target[s] declared in the rule). Unaffected inventory beacons, of course, ignore the policy change.
- Each affected inventory beacon applies your schedule, then (on schedule) triggers the action specified in the rule. Depending on subnet sizes, it may take a little while to complete the discovery process.
- Finally, the resulting discovery (`.disco`) files are uploaded to the central application server, and resolved into the inventory database. There they wait until the next full inventory import and license reconciliation, which happens by default overnight (although an operator in a role with Administrator privileges may trigger one sooner).

- After that, the discovered VMs and their hosts are visible in the **All Discovered Devices** page, and the automatic maintenance of special hidden targets and automated rules are ready to run as required (once you turn on high-frequency mode, as described in Turn on High-Frequency Mode).
- Thereafter, inventory is gathered from the target hosts automatically on the high-frequency schedule you enable.

Check Schedule and Primary Source

In high-frequency mode where IT Asset Management is responsible for your sub-capacity calculations on IBM PVU and VPC sub-capacity licenses, there are two mandatory requirements for inventory gathering:

- FlexNet inventory must be your primary inventory source. This means that other sources may be used to fill gaps where you have not installed the FlexNet Inventory Agent; but none of these other sources can ever overwrite the data collected in FlexNet inventory.
- The schedule for your full inventory import and license reconciliation must run daily. It does not matter what time of day you choose to run this import and compliance calculation (the default is to run it overnight); but it must run once each day. Among other things, this validates the links between inventory devices and IBM sub-capacity licenses, which are foundational to peak consumption calculations.

The default settings are satisfactory for high-frequency mode, but it is worth a quick check to validate that no one has modified these defaults before commencing high-frequency operations.

Tip: *If you are looking for where to set the reporting period and duration of data retention, go to the IT Asset Management Settings* ***General*** *page (****Administration > IT Asset Management Settings > General****) and select the* ***Licensing*** *tab.*

To check the daily inventory schedule and source priority:

1. In the web interface for IT Asset Management, go to the IT Asset Management Settings **General** page (**Administration > IT Asset Management Settings > General**) and select the **Inventory** tab.
2. Scroll down to the **Managing the processing queue for imports and reconciliation** section to validate the settings for inventory import and license reconciliation:
    a. Ensure that the **Frequency** setting shows `Daily`.
    b. Check that the **Schedule after** field shows the earliest time that you want to start the process. You may spin up the chosen hour, or enter a time in 12-hour format, with the am/pm identifier following after a space, such as `11:30 pm`.

       This time is local time at the central application server. The default suggested time is midnight PST (for the US cloud instance), or 7am UTC. If the system is idle at the nominated time, the process starts immediately; however, it may be delayed a little until other scheduled activities in the batch processor are completed.
    c. Ensure that the third field displays **Every** `day`.

       This is mandatory in high-frequency mode. Recall from Operation in High-Frequency Mode that the links from inventory devices to licenses, as well as all software inventory details, collected in the full inventory import and compliance calculations are required for retrospective checks of peak license consumption

using the latest hardware inventory updates. This baseline must be recalculated daily.

d. If you changed any values on this page, scroll down and click **Save**.

3. Go to the **Data Imports** page (**Data Collection > IT Assets Inventory Tasks > Data Imports**) once again and select the **Inventory Data** tab.

   This page lists all known inventory sources.

4. Locate the source named `FlexNet Manager Suite` (which covers FlexNet inventory), and ensure that on the right-hand end of its header row, there is a disabled button displaying `Primary`, marking it as the primary data source for inventory collection and data merging. If not, click the **Make primary** button for this row.

   You may also click in this header row to reveal more details about FlexNet inventory collection.

You may close the **Data Imports** page. Everything is now in place to use an initial inventory collection to seed the automatic processes for high-frequency mode, as described next.

Set Up and Collect Inventory, and Reconcile

The IBM approval for using IT Asset Management for sub-capacity license calculations is dependent on you using the full FlexNet Inventory Agent, locally installed on each IBM server (that contains software with IBM PVU or IBM VPC sub-capacity licensing), for high-frequency inventory gathering.

As well, operation of two hidden (and automatically maintained) targets for inventory gathering requires that discovery and inventory processes have run at least once, so as to provide 'seed' data to start the automated processes.

To help meet these requirements, this process summary covers deploying the FlexNet Inventory Agent, and triggering the first inventory upload and license compliance calculation. Also take note of the requirement for the full import and license reconciliation process to run on a daily schedule.

To initiate inventory collection:

1. Install the FlexNet Inventory Agent on each IBM server.

   You may achieve this through your preferred method, as described in *Gathering FlexNet Inventory*:

   - You may use the built-in 'adoption' process (see the topic *Automated Adoption Summary* in the above document)
   - You may use an alternative deployment technology of your choosing (for which, start with the topic *Self-Managed Deployment: Collecting the Software* in the same document)
   - You can even deploy by hand, if you so choose.

   Important: *Be sure to deploy the entire FlexNet Inventory Agent, and not just a subset of executable files.*

   In those processes, you identify (usually in a bootstrap or configuration file) the inventory beacon to which each installed FlexNet Inventory Agent should initially report.

2. Wait.

   The installed FlexNet Inventory Agent contacts its preferred inventory beacon, and downloads its default policy. This includes gathering standard hardware and software inventory (but not tracking application usage, which is

not required for IBM sub-capacity licenses). The default policy also distributes the standard schedule for inventory collection. You can see the schedule settings in the web interface of IT Asset Management by going to the **Inventory Settings** page (**Data Collection > IT Assets Inventory Tasks > Inventory Settings**) and reviewing the **Inventory agent schedule** section. The default is that the FlexNet Inventory Agent collects inventory at a random time within a one hour window, commencing at 5:45am local time on the target inventory device. As soon as the inventory collection is completed, the data is uploaded to the inventory beacon. Once successfully staged on the inventory beacon, it is uploaded to the central application server (and there is an overnight catch-up scheduled upload to recover from any temporary network problems). On the application server, a web service receives the file and resolves it into the inventory database (or, if the web service is overloaded, temporarily saves the file in the `Incoming` folder, and then imports it to the inventory database as soon as possible). Once staged in the inventory database, the data is included in the next full inventory import and license consumption calculation, which by default runs at 2am daily. In short, using the default schedules, if you install the FlexNet Inventory Agent today, you can expect to see first results of the automated processing the day after tomorrow.

If the automated processes are not fast enough for your current requirements:

a. Log into the target server using an account with local administrator privileges. (A target server is any one that may run software licensable under an IBM PVU or IBM VPC license.)

b. Start a command line window using the **Run as administrator** option.

c. Issue the following commands:

- On Microsoft Windows servers:

  ```
  ndschedag –o ScheduleType=Machine
  ```

  In the pop-up window that appears, run `Generate Inventory`.

- On Unix-like servers (such as AIX):

  a. Open the file `/var/opt/managesoft/scheduler/schedules/sched.nds` with a text editor.

  b. Find the line that defines the inventory collection event in the file, and copy the *`eventId`* associated with the event.

  c. Navigate to the `bin` sub-folder in the installation folder for FlexNet Inventory Agent.

  d. Complete the following command line, pasting the *`eventId`* on the command line inside double quotes: `ndschedag –x "`*`eventId`*`"`

d. Wait about 20-30 minutes for the inventory gathering, upload, and resolving process to complete. (This resolution is into the inventory database.)

e. As an operator in the Administrator role, in the web interface for IT Asset Management, go to the **Reconcile** page (**Data Collection > Process Data > Reconcile**).

f. Clear the check box for **Reconcile all publishers**, and use the search control that appears to select `IBM` and **Add publisher**.

g. Select the check box for **Update inventory for reconciliation** (only available to operators who are members of the `Administrator` role).

This is the switch that causes your uploaded inventory, waiting in the inventory database, to be imported for the license calculations.

h. Click **Reconcile**.

The license reconciliation process is scheduled, and runs as soon as the batch processor is free.

i. On the right-hand side, `Ctrl+click` **View more in System Tasks**.

The **System Tasks** page opens in a new tab. You can expand the entry for your process, and see separate rows for `Import inventory devices` and `Reconcile licenses`. The page refreshes automatically, or you can update the page with a manual refresh action.

When the import is completed, and license consumption has been calculated (in the reconciliation process), you can inspect the license(s) you created in Configure Appropriate Licenses and see the relevant inventory devices listed on the **Consumption** tab of the license properties.

3. Manage exemptions applicable to any inventory devices.

In Configure Appropriate Licenses, you set up the license exemptions for any device roles permitted in the license agreement. Now it is time to configure the matching device roles on any applicable devices:

a. In the **Consumption** tab of the appropriate IBM sub-capacity license, locate any inventory device that requires an exemption.

b. `Ctrl-click` the name in the **Device** column.

The inventory device properties open in another browser tab. Alternatively, you can click the **Open** button, in which case the device properties page replaces the license properties.

c. In the **General** tab of the inventory device properties, select the appropriate value from the **Device role** drop-down, and click **Save**.

From the next full compliance calculation, this inventory device will no longer consume from this license (nor from any other license that allows exemptions for the same role). It remains linked to this license (in the same way as if you had made an allocation), but its consumption is zero. If you return to, and refresh, the **Consumption** tab of the license properties, the selected reason is displayed in the **Exemption reason** column.

4. Visit each new IBM sub-capacity license (for which this is the first inventory import and license consumption calculation), and review any individual device exemptions on the **Consumption** tab, adjusting as necessary.

The license is now functional, with at least one attached application that is showing consumption after the initial inventory import. It is now possible finally to turn on high frequency scanning of devices to identify any hardware or hosting changes. The 30-minute high frequency checks feed into the daily full import and license compliance calculation, where:

- Links from inventory devices to licenses are reevaluated
- Any new devices added to IBM sub-capacity license consumption are picked up in a timely way
- The calculations of points consumed per region are reworked as part of each full license compliance calculation, retroactively covering the entire data retention period and using the latest data available (including corrections to inventory data, device exemptions, and so on).

A daily full import and reconciliation is the default schedule; and you may check and adjust your settings as follows.

5. Confirm that a full import and license compliance calculation is performed daily:

a. Go to the IT Asset Management Settings **General** page (**Administration > IT Asset Management Settings > General**).

b. Select the **Inventory** tab.

c. Scroll down to the section **Managing the processing queue for imports and reconciliation**.

d. Ensure that the following settings are in place:

- **Frequency**: `Daily`
- **Every**: `day`
- **Schedule after:** (say) `11:00 pm` (you may spin up an appropriate time, based on your other scheduled activities, such as inventory imports from third-party sources).

e. If you have made changes, click **Save** (in the bottom right corner).

Turn on High-Frequency Mode

With data flows established and populating at least one IBM sub-capacity PVU or VPC license (so that you can be sure that hidden targets are initialized), you are ready to switch into high-frequency mode for calculation of sub-capacity consumption using IT Asset Management as the source of calculations.

Because of the times required for all processes to reach production conditions (as described below this procedure), it is good practice to make this change at the start of a weekend when you expect the environment to be stable. This allows 48 hours or more for the switchover to stabilize.

Setting **Enable frequent hardware scanning for IBM sub-capacity license calculations**, as described below, has the following effects:

- Adds the default `All devices consuming IBM PVU points` target (remembering that this target also includes devices consuming from VPC sub-capacity licenses) to the list of targets available for IBM sub-capacity-related hardware scans
- Adds the default `Known vCenter or OVM Manager servers` target to the list of targets available for remote scans by appropriate inventory beacons of all known VMware vCenter and Oracle VM Manager servers
- Enables frequent hardware scanning of all target inventory devices where the FlexNet inventory agent is locally installed, as required by IBM (this setting is distributed to the installed FlexNet inventory agents through the inventory beacons, which in turn receive it through the `IBMPVUEnableScanning=true` setting in the `BeaconSettings.xml` file)
- Enables the reassessment of peak consumption values for the three IBM regions, taking into account the latest data inputs, that now occurs as part of the nightly full compliance calculations
- Prevents IBM PVU licenses incorporating any points imported from ILMT, where this remains available as an auxiliary inventory source

Tip: *When imports are still available from ILMT, and high-frequency scanning is enabled, there is a report provided that compares the current sub-capacity results imported from ILMT with those calculated by IT Asset Management.*

- Reveals additional settings in the web interface, as described below.

To configure higher frequency processing for IBM sub-capacity licenses:

1. Go to the **Inventory Settings** page (**Data Collection > IT Assets Inventory Tasks > Inventory Settings**). Scroll down to the **IBM reporting and archiving settings** section.
2. Select the **Enable frequent hardware scanning for IBM sub-capacity license calculations** check box.

   A number of additional controls appear when this check box is selected (provided that the prerequisites described in Requirements for IT Asset Management Sub-Cap have been satisfied).
3. In the **Perform extra inventory scanning every *`time-interval`*** drop-down list, choose `30 minutes`.

   This is the setting required by IBM. Other values are available for testing and development. This control sets the frequency for hardware inventory checks by the installed FlexNet inventory agent on target inventory devices, and uploads of hardware inventory where there have been any changes.
4. Select the **Collect inventory from VMware vCenter or Oracle VM Manager servers** check box.

   This is the control that turns on use of the default inbuilt rule for direct inventory gathering from these servers, as well as automatic maintenance of the hidden target, derived from the list of discovered devices, that is used for that rule.

   Note: *Although Oracle VM Manager servers are scanned for changes, all calculations for VMs on these servers are full capacity calculations. This is because currently IBM accepts Oracle VM servers only on SPARC and UltraSPARC architectures for sub-capacity consumption, and IT Asset Management supports Oracle VM servers only on x86 platforms.*
5. In the drop-down list to **Collect inventory from VMware vCenter or Oracle VM Manager servers** choose a target.

   This may be the built-in target for `Known vCenter or OVM Manager servers`. This target, automatically updated after each discovery import, lists all vCenter and OVM virtual management servers. Independently of this setting, you may choose to create a separate custom target to target sub-set of VMware vCenter or Oracle VM Manager servers that you want to scan.
6. If you want to include another target that you have defined, click the **+** button to the right of the list.

   A new line appears below your previous choice, and you can repeat the selection process for other targets you have defined.

   Tip: *When the selected target is present in the closed list, it is included in data saved when you click **Save**.*
7. In the drop-down list to **Enable frequent hardware scans by installed inventory agents within the following targets**, choose the built-in target for `All devices consuming IBM PVU points`.

   This target, automatically updated daily, lists all computers with installed software linked to either IBM PVU or IBM VPC sub-capacity licenses. If you have some special reason to do so, you may add additional targets using the same controls.

   - One possible reason for adding targets might be for 'remedial adoption'. If you include a separate target that includes the target setting **Allow these targets to be adopted**, this has the effect of discovering new devices running software linked to an IBM PVU or IBM VPC license. If the FlexNet Inventory Agent is found on these

devices, inventory gathering is left to the installed FlexNet Inventory Agent; and if not, the appropriate inventory beacon remotely installs the FlexNet Inventory Agent (that is, it 'adopts' the target device) and then leaves it to complete inventory gathering.

- However, keeping in mind that IBM high-frequency mode requires maximum performance, it would be better to attach such a target to a separate rule, run on (say) a daily schedule (especially in enterprises with large subnets). This has the benefit of returning inventory from newly-discovered devices in time for the daily automated update of the hidden target `All devices consuming IBM PVU points`.

8. When you are satisfied with your settings, click **Save** (in the bottom right of the page).
9. Configure IT Asset Management for your IBM reporting cycle and switch-over date.
   a. Go to the IT Asset Management Settings **General** page (**Administration > IT Asset Management Settings > General**). Select the **Licensing** tab, and scroll down to the **IBM reporting and archiving settings** section.

      Tip: *This section is visible only after high-frequency mode is enabled.*

   b. Select **Automatically adjust the calculation period start date**.

      This is best practice, and saves having to remember future manual work as each reporting period rolls around. See the online help about these settings for more details. Selecting this option reveals another line of controls.

   c. Insert the length of your reporting cycle to IBM (typically either one month or three months) in the **Reset the calculation period every** field.

   d. In the following date field, after **month(s) from**, insert the start date of the *current* reporting period.

      As you have just turned on high frequency processing, this is normally your current reporting period.

   e. In the **Ignore any value prior to** date field, insert the date when you are switching over to fully operational use of FlexNet Manager Suite for sub-capacity calculations.

      This is the day after you archived your last report from ILMT. As already observed, there is a ramp-up time as the processes in FlexNet Manager Suite come up to full production readiness (further details are below). Therefore, this reporting switch-over date might be three-four days after you "flick the switch" (earlier in this process). For example, if today (when you flick the switch) is Friday, with a quiet weekend to follow, you might allow Monday for checking, archive your last ILMT report on Tuesday, and set this **Ignore any value prior to** date for Wednesday. There is *no* requirement to align this date with the boundary of a reporting period, as you can submit reports from both products to jointly cover this switch-over period.

      Tip: *If there are unexpected delays in your business processes, you can adjust this date after the event, but before archiving any reports from IT Asset Management for possible submission to IBM.*

   f. In the **Keep historical data for** field, enter the number of days for which to keep historical data ready for reporting to IBM.

      Best practice is to make this at least double the length of your reporting period, as this ensures that you can update reports for the *previous* period at any time in the current period. It also caps the database space needed for storing historical data. Historical information for reassessing peak points consumption

is automatically deleted as soon as it is older than the value you set here. Notice that this is a *rolling* time window, always the specified number of days back from today's date.

g. Click **Save** (in the bottom right of the page).

10. Wait.

(It is assumed here that there has already been at least one daily full inventory import and full license reconciliation completed, as described in Set Up and Collect Inventory, and Reconcile.) All the following additional processes must now have time to complete:

- The creation of the new policy for inventory beacons. This is created within minutes, and lists all affected inventory devices.
- The collection of the new policy by all inventory beacons. By default, each inventory beacon checks for changed policy every 15 minutes, although this may have been modified in the **Beacon settings** section of the **Inventory Settings** page (**Data Collection > IT Assets Inventory Tasks > Inventory Settings**) . To check whether inventory beacons have the latest policy, go to the **Beacons** page (**Data Collection > IT Assets Inventory Tasks > Beacons**), and check the **Policy status** column, where `Up to date` means that an inventory beacon has reported having the same version of policy currently available on the central application server.
- All installed FlexNet Inventory Agents on the target list of inventory devices must individually request an update of their device policy. The policy update request happens at a randomized time once per day. Whichever inventory beacon a device contacts to request any policy updates immediately generates and downloads the updated device policy. Where applicable (that is, where the inventory device is listed in the target of devices running software licensed under IBM PVU or IBM VPC licenses), the updated policy includes the high-frequency schedule for hardware inventory checks.
- Each installed FlexNet Inventory Agent must commence executing the high-frequency hardware inventory checks. On the first occasion, and on any later occasion where there is a change to relevant hardware properties, hardware inventory is also uploaded through the hierarchy of inventory beacons. Typically the first hardware inventory collection and upload from a given device may take around ten minutes. (Keep in mind that, due to the random timing of device policy updates, some devices may not start high-frequency hardware checks until the following day.) For IBM PVU licenses only, to track devices that are not [yet] reporting as expected, go to the **IBM PVU Out-Of-Date Inventory** page (**Reporting > Inventory Reports > IBM PVU Out-Of-Date Inventory**) and run the report for the appropriate period. The report includes *all* devices linked to IBM PVU licenses, filtered by the period you choose. Two important values for the **Out-of-date reason** are:
  - `The device is not running frequent inventory scans, and has not reported recently` — these devices have been linked to an IBM PVU license through inventory gathering (from any inventory source) and after a previous full license reconciliation, but are not yet included in the internal target for high-frequency hardware scans (and it is therefore natural that they have not yet reported high-frequency hardware inventory). This may occur in the time window between turning on the high-frequency scanning schedule and the next full license reconciliation. These devices should be included in the internal target after the *next* full license reconciliation.
  - `The device has frequent inventory scans enabled, but has not reported recently` — these devices *are* included in the internal target for high-frequency scanning, so that their missing uploads may be a matter of timing: they have been targeted, but have yet to start the high-frequency scans.

- The appropriate inventory beacons must start checking virtual hosts for changes in VM details, as described in Set Up Virtual Inventory Tracking. To check operations, go to the **Activity Log** page (**Data Collection > IT Assets Inventory Status > Activity Log**), and filter the page for **Description** contains `Discovery completed on Beacon for rule: Virtual cluster server scan for IBM PVU.` (This also includes scanning for IBM VPC-licensed software.) After a full reconciliation, these devices are visible in the **All Discovered Devices** page, as well as in inventory listings as appropriate.
- Each full license reconciliation process includes the recalculation of peak values, based on:
  - Historical data through the entire data retention period — although any details for dates before the switch-over date recorded in the **Ignore any value prior to** setting (on the **Licensing** tab of the IT Asset Management Settings **General** page (**Administration > IT Asset Management Settings > General**)) are correctly ignored
  - The current settings for things like inventory corrections, license allocations, device exemptions, and the like
  - The latest imported inventory
  - The current best fit of inventory devices to IBM PVU or IBM VPC licenses.

  This process starts from the next full reconciliation (which you should leave scheduled for overnight execution) after you save the **Enable frequent hardware scanning for IBM sub-capacity license calculations** check box and associated settings on this **Inventory Settings** page. However, results should not be considered valid until all installed FlexNet Inventory Agents are reporting hardware inventory.

Note: *Even though you have now commenced high-frequency operations, data from ILMT about consumption of IBM PVU licenses (only) is still imported into staging tables to allow comparison of current PVU sub-capacity results from both ILMT and FlexNet Manager Suite. However, results calculated by ILMT are not displayed in the web interface, and you may wish to reduce system maintenance overheads by removing the ILMT agent, as described next.*

Removing ILMT as an Inventory Source

Important: *Before proceeding, collect and archive your final reports of IBM PVU and IBM VPC sub-capacity consumption from ILMT. These may be required for submission to IBM if they cover part of the current reporting period; and they may also be required for an auditor to validate the future results from IT Asset Management against the past results from ILMT.*

While you were relying on ILMT as the source of truth for sub-capacity calculations for IBM PVU and IBM VPC licenses, you probably had two inventory agents deployed to target devices:

- The ILMT agent
- The FlexNet Inventory Agent.

Now that you have switched to IT Asset Management for sub-capacity calculations on these licenses, the points calculated by ILMT are no longer visible in the web interface of IT Asset Management. You now have two choices about the ILMT agents:

- You may continue to use ILMT as an inventory source from which data is integrated into IT Asset Management, but keeping in mind that no consumed points calculated by ILMT are visible now that IT Asset Management is in high-frequency sub-capacity mode. (You may also be using these ILMT agents to continue populating ILMT with data.) As long as you have ILMT as an inventory source, IT Asset Management does provide a report (in **Reports** mode) with which you can compare the sub-capacity results calculated by the two products, at least for IBM PVU licenses.
- You may decide that you no longer want to maintain two agents, and therefore want to remove ILMT as an inventory source.

We do this by removing the inventory connection to the ILMT database. This is a two-stage process.

To remove an existing ILMT inventory connection:

1. On the inventory beacon that connects to ILMT, start FlexNet Beacon using the `Run as administrator` option, and select the **Inventory systems** page.
2. From the list of connections, select the ILMT connection, and below the listing, click **Delete**.
3. In the confirmation dialog, click **OK**.

   This removes the connection data from the inventory beacon, but there is no communication about this change from the inventory beacon to the central application server, where another manual edit is required.
4. In the web interface for IT Asset Management, go to **Data Imports** page (**Data Collection > IT Assets Inventory Tasks > Data Imports**), and select the **Inventory Data** tab.
5. Scroll down to find the row for the ILMT connection, and click that row to expand its details.
6. Just below the **Input details**, click the link to **Delete connection**, and in the confirmation dialog, click **OK**.

Details of the connection to ILMT are now removed from both the inventory beacon and the application server. Inventory data previously imported from ILMT that is *not* covered by any other inventory source is removed from the compliance database when you delete the connection details in the **Data Inputs** page of the web interface. This includes any ILMT-only inventory devices and related installation records, and (after the next compliance calculation) their contributions to license consumption. License records created to match ILMT inventory are not deleted (although, appropriately, their consumption figures may be affected).

Tip: *If you had previously linked an ILMT-only inventory device record to an asset record, the inventory device is not deleted, and instead is given a* ***Status*** *value of* `Awaiting Inventory`*. If removal was intended, the asset record and inventory device record can both be removed manually.*

You may now use your preferred company processes to remove the ILMT agent from target inventory devices.

Reporting to IBM

IT Asset Management provides a digitally-signed report package that is ready for you to submit to IBM for audits. The package includes the following reports:

- IBM Cloud Pak consumption report
- IBM PVU license consumption report
- IBM VPC license consumption report

- IBM Unlicensed Installations report

This package can be used as a trusted source for audit as the data contained in the package can be validated for authenticity. It meets IBM's reporting requirements and is accepted by IBM.

Tip: *Although some of the reports are very similar to those from ILMT, there are acceptable presentation differences between ILMT and IT Asset Management reports of sub-capacity licensing. IBM's sub-capacity reporting requirements do not require exact copies of the ILMT reports (nor is IT Asset Management intended to copy ILMT).*

Important: *Reports show data based on the latest full license reconciliation calculations. In fact, if for some reason you have not yet run a full (overnight) license reconciliation since switching to high-frequency PVU mode for IT Asset Management, reports like* ***IBM PVU License Consumption*** *are unable to display any data.*

To download the digitally-signed report package:

1. Go to any of the following pages:
   - **IBM Cloud Pak License Consumption** (**Reporting > License Reports > IBM Cloud Pak License Consumption**)
   - **IBM PVU License Consumption** (**Reporting > License Reports > IBM PVU License Consumption**)
   - **IBM VPC License Consumption** (**Reporting > License Reports > IBM VPC License Consumption**)
   - **IBM Unlicensed Installations** (**Reporting > License Reports > Unlicensed Installations**)
2. Select the reporting period.
3. Click **Download the IBM audit report**.

   The downloaded package contains the following data:
   - IBM Cloud Pak license consumption report
   - IBM PVU license consumption report
   - IBM VPC license consumption report
   - IBM Unlicensed Installations report
   - Report summary

     Note: *The report summary contains the generation date, user account that generated the report, period of the report, and statistics for the out-of-date inventory devices.*
   - Checksums
   - Privacy Enhanced Mail (PEM) file
   - RSA signature file

Beyond these typical sub-capacity reporting requirements, your enterprise might have additional reporting requirements defined by your IBM agreements, such as License Management Option (LMO) reports, which are not addressed in this topic.

More information

For more information about IBM sub-capacity licensing, the following IBM resources may be helpful:

- IBM Passport Advantage Agreement
- Passport Advantage Sub-capacity Licensing Eligible Product Statement
- Passport Advantage Virtualization (Sub-capacity) Licensing
- Virtualization Capacity License Counting Rules
- Manual Calculation of Virtualization Capacity (spreadsheet download)
- PVU Tables
- License Information Document Search.

6

Introduction to Client Access License

A Client Access License (CAL) is a software license that is required to access the services of certain server products like `Microsoft Exchange Server` or `Microsoft SharePoint Server`. A CAL is required for each accessing user or device that accesses the server product unless a processor-based or core-based license is procured for the accessed server application, or an External Connector License is procured to cover the external users' access to the server applications. A CAL provides the access rights to physical, virtual, and (sometimes) online services. Most of the Microsoft server products, including Windows Server, SQL Server, Exchange Server, SharePoint Server, and Microsoft Skype for Business Server (previously Microsoft Lync Server) require a CAL for each accessing user or device. With detailed information about CALs and CAL management, this chapter may help you in managing CALs through IT Asset Management.

CAL Types

A single (User or Device) CAL covers any number of servers for that product. For example, a user with a User CAL for Microsoft SharePoint Server can access any number of Microsoft SharePoint servers within the organization. Some Microsoft server products include client access rights in the server license itself. For products such as SQL Server, instead of buying CALs, you may choose to buy processor-based or core-based licenses for the accessed server applications instead of licensing under the server/CAL licensing model. Certain products like Microsoft SQL Server can be licensed based on the number of processors or cores in the host server.

The licensing rules for CALs differ depending on whether the accessing users are employees of the licensee organization or are external users accessing the licensed servers. Some versions of Microsoft SharePoint Server, such as SharePoint 2013 and SharePoint 2016 require CALs for employee access to those servers, but include rights for external users accessing those same servers. For example, a website powered by an external web server could be accessed by a possibly-unpredictable number of users and should be licensed via a version of SharePoint that includes external access rights or, for earlier versions, an External Connector License is required in addition to the server license. You may wish to check the usage required for servers in your environment and evaluate the various license models available for those servers before making your purchase decision.

For Microsoft servers that have underlying server system requirements, the relevant CALs are required for each server product accessed. For example, Microsoft SharePoint Server installations require both Microsoft Windows Server and Microsoft SQL Server. As such, each accessing user or device must have a Microsoft SharePoint CAL, Microsoft Windows Server CAL and Microsoft SQL Server CAL to comply with license requirements, unless access is provided by the server license itself (as is the case of servers licensed per core or covered by an External Connector License).

CAL types

This section outlines the different license types used in the context of CALs. IT Asset Management only supports Microsoft User and Microsoft Device CALs via the `FlexNet Manager for Microsoft` product.

License Type	Description
User CAL	An alternative to a Device CAL, a User CAL is required per user per server product that requires a CAL. A user with a User CAL for a server product can access any instance of that server product within the enterprise via any number of devices. For example, if a user `Peter` has a single User CAL for Microsoft SQL Server product, he can access any installation of Microsoft SQL within the enterprise.
Device CAL	An alternative to a User CAL, a Device CAL is required per device per server product that requires a CAL. An accessing device with a Device CAL for a server product can be used by any number of users to access any instance of that server product. For example, if you have a Microsoft Device CAL for `Microsoft SharePoint Server` for an inventory device, any number of users can access any instance of `Microsoft SharePoint` from this device.

Other related license types

The following related license types are not directly supported by IT Asset Management. To manage any of these licenses, creation of custom license is required.

- **Client Management License (CML)**: Required per management server (Microsoft Endpoint Configuration Manager, previously Microsoft SCCM) that is managing devices with non-server operating system (for example, Windows 10). If you procure device CALs for those devices that access the server, you do not require a CML.
- **External Connector License**: Required per server for each server product that is to be accessed by external users (non-employees) from outside the company network. For example, if some users of an external organization have been granted access to your `Microsoft Exchange Server`, that server requires an External Connector license.
- **Additive CALs**: A CAL required for a specific server, in addition to a User or Device CAL. Additive CAL is required per server when some advanced server functionality is accessed, in addition to the base server functionality. For example, `Windows Server Rights Management Services (RMS) CAL` is required in addition to Windows Server User or Device CAL.

Selecting a CAL Type

You can buy individual CALs for each accessing user or accessing device, or you may choose to buy CAL Suites (for a user or device). To make an effective decision about which CALs you should buy, you may want to know the usage details for the server products that require CALs. Organizations typically use a mixture of CAL licensing types based on their usage needs and the profiles of their user population.

Choosing Between User and Device CALs

If you choose to license servers via the Server/CAL license model, you may choose from user-based or device-based licenses. The **User CAL** approach requires you to purchase a CAL for every user that accesses a Microsoft server product (such as `Microsoft SharePoint Server`), regardless of the number of devices used to access that particular

product. This enables users to access the services from multiple client devices, including any computers from outside the organization. The User CAL also covers access via mobile devices, such as users checking their email via smartphones or tablets. One User CAL for a server product enables you to access any number of instances of that server product. For example, if you have purchased a User CAL for Windows Server for a user, that user can access multiple Windows Servers.

Alternatively, the **Device CAL** approach requires you to purchase a CAL for every device through which the services of a server product are accessed. With an appropriate Device CAL for a device, any number of users can access the server product through that accessing device. This is often a desirable option in environments such as call centers, retail outlets or manufacturing sites in which multiple users share a single PC or kiosk. The ability to mix Device and User CALs in a single environment depends on your license agreement terms, but you must assign individual CALs to either a device or a user.

To clarify this concept, take an example of an organization with 600 users accessing Windows and Exchange servers. These 600 users work in three shifts of 200 at any time. The organization may choose to purchase 600 User CALs for Windows and 600 User CALs for Microsoft Exchange Server. However, a simple analysis shows that at any time, only 200 users are using the same computers to access the servers. Therefore, the organization is likely to buy 200 Device CALS for Exchange Server and 200 Device CALs for Windows. However, if these same 200 users are also accessing servers via their smartphones, tablets or home computers, you may find the User CAL option to be more affordable.

Note: *Microsoft typically recommends you not to mix User and Device CALs.*

For external users who connect to your organization's computers (for example, guest users or business partners), you can buy additional User CALs for each accessed server or an External Connector License, if appropriate for the quantity of external users involved.

Choosing Between Individual CALs and CAL Suites

A CAL Suite is a special license prodviding CALs for several different server products. This may make a CAL Suite more cost effective than buying an individual CAL for each distinct server product. For example, if 100 users in your organization have access to Microsoft SharePoint, Microsoft Exchange, Microsoft Skype for Business (previously Microsoft Lync Server), and Microsoft System Center Servers, it is more cost effective to purchase 100 Core CAL Suite licenses instead of 100 standalone product CALs for each of these servers. Microsoft has the following two Server CAL Suite offerings:

- Microsoft Core CAL Suite
- Microsoft Enterprise CAL Suite.

Consider the following points before you purchase CAL Suites:

- You can license these suites on per user or per device basis, and cannot split between users and devices.
- CAL Suites can only be purchased with Software Assurance (maintenance) coverage. If you do not renew the Software Assurance, the CALs are locked into the current version of each CAL available at the expiry date.
- As CAL Suites contain licenses for independently-released products, CAL Suites are version-less. CAL Suites provide the right to use the most recent version of every product in the suite.

Tip: *For detailed information about Microsoft CAL Suites, see the Microsoft website.*

Virtualization and CALs

As CAL consumption is based on the access to a server product, it does not matter whether the server product has been installed on a virtual or a physical machine. You are required to buy CALs based on the number or users or devices accessing the server software. A User CAL grants access to any instance of that server product within the enterprise. Similarly, when you buy a Device CAL for a server product, any user can access any instance of the server product through that device.

How Does IT Asset Management Calculate CAL Compliance

To determine an installation of a server application on an inventory device, IT Asset Management uses evidence (any combination of installer, file, or WMI). An additional evidence type called access evidence is separately used to measure access to the server applications, such as `Microsoft SharePoint Server`.

Note: *You need the `FlexNet Manager for Microsoft` product to create and manage CALs in IT Asset Management. To gather CAL usage inventory, you must set* ***CAL inventory options*** *while creating an inventory target to* ***Allow CAL access evidence collection on these targets****. For more information, see Creating a Target in the online help (Discovery and Inventory > Discovery > Discovery and Inventory Rules > Targets > Creating a Target).*

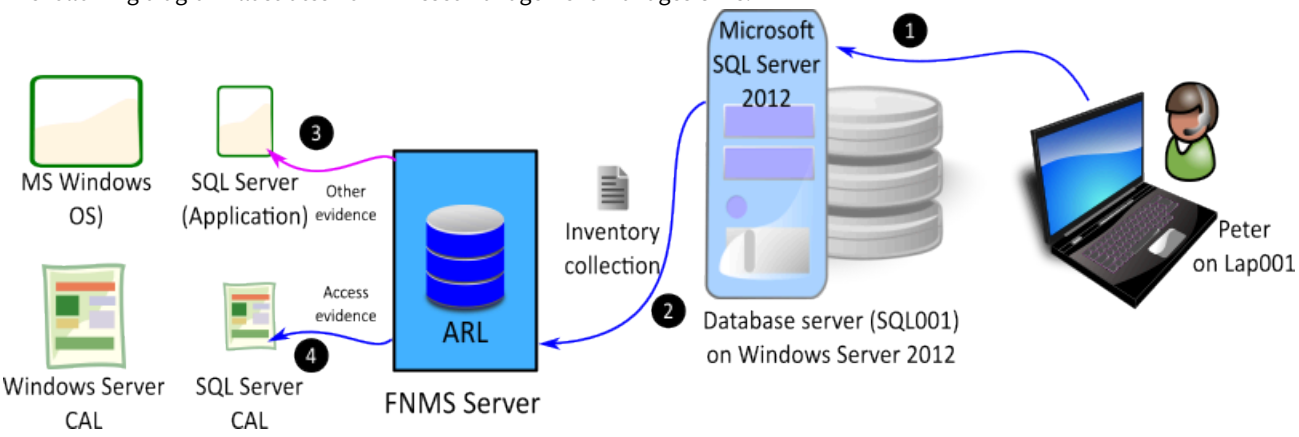
When linked to an application, the access evidence records any access event to a particular server application such as `Microsoft SharePoint Server 2013`. An access evidence record provides information about the accessed application, accessing user, and accessing device. The access evidence information is collected in a separate file (`.swacc`) as a part of the discovery and inventory process. This file is only generated for the inventory devices that have the supported server products installed on Windows Server 2012 or later and for which Microsoft's User Access Logging (UAL) capability has been enabled. Access evidence is collected through various services including User Access Logging (UAL) and PowerShell scripts for some products. IT Asset Management can collect the access evidence through any of the following methods:

- The locally-installed FlexNet inventory agent
- Zero-footprint inventory collection through inventory beacon(s)
- Inventory uploads from Microsoft Endpoint Configuration Manager (previously Microsoft SCCM) (CAL usage inventory only for Microsoft Endpoint Configuration Manager)
- Manual upload of access evidence through the **Inventory Data One-Off Upload** page.

The inventory collected from any of these sources is transferred from an inventory beacon to the central application server through regularly scheduled uploads.

For conventional device licenses, during the reconciliation process the Application Recognition Library (ARL) matches the available evidence to record an application installation. For CALs, while access evidence is linked to an application, it is not used to recognize the linked application installation; instead, it is used to identify access to that application. Like other product licenses, a CAL is also linked to the appropriate application record. However, in the case of CALs, the compliance position is generated based on the usage data (how many clients accessed the server application). The default license priorities are used to consume the appropriate license entitlement against a piece of access evidence. The collected access evidence records are visible on the **Access evidence** tab in the **All Evidence** page (**Applications & Evidence > Evidence > All Evidence**).

The following diagram illustrates how IT Asset Management manages CALs:



1. A user `Peter` accesses the `Microsoft SQL Server 2012` installation on `SQL001` database server, installed on Microsoft Windows 2012. The access event is recorded for Microsoft Windows and Microsoft SQL Server.
2. During the next inventory collection process, IT Asset Management gathers hardware, software, and access inventory for the `SQL001` inventory device.
3. During reconciliation, IT Asset Management uses evidence to report an installation of Microsoft SQL Server 2012 and Microsoft Windows 2012.
4. The collected access evidence identifies that `Peter` accessed `Microsoft Windows Server 2012` and `Microsoft SQL Server 2012` on `SQL001`.
5. Assuming that neither `Peter` nor `Lap001` has already consumed a CAL for accessing any other Windows server, IT Asset Management consumes an entitlement of Microsoft User or Device CAL for each of Microsoft SQL Server and Microsoft Windows based on the set license priorities. For more details on license priorities, see the *How Does License Consumption Order Work?* topic in the online help.

Variations to the above example

- If you have a per-processor or per-core license for Microsoft SQL Server on `SQL001`, CALs are not required.
- If `Peter` is an outside user (for example, a consultant on a customer site), an External Connector license may also be required.
- If this Microsoft Server Installation is used by both internal and external users, you need CALs and an External Connector license, or you can buy a Processor or Core-based license of `Microsoft SQL Server`.
  If the collected access evidence does not match a known server product, it is preserved as unrecognized evidence (visible on the **Unrecognized Evidence** page). By default, IT Asset Management retains the CAL usage inventory data for 90 days from the date of inventory collection, unless the value of the **Number of days to keep CAL usage inventory** control is changed. For details, see the *System Menu > System Settings > System Settings: Inventory Tab* page in the online help.

Access evidence collection methods

IT Asset Management supports multiple Microsoft Server applications for CAL management. Different underlying methods are used to collect access evidence data for different versions of these products. Here is a list of access evidence collection methods used to identify access to a server application:

- **User Access Logging (UAL):** When this access method is used, IT Asset Management collects the access evidence

through the User Access Logging (UAL) service of Windows Server, enabled by default in Windows Server 2012 or later. This service records the client device and user request events used to access Windows Server and other installed Microsoft server applications (such as `Microsoft SharePoint Server`) into a local database. IT Asset Management collects the access evidence data through a PowerShell script that uses specific Microsoft APIs for UAL. The access events are recorded for both physical and virtual clients and servers. To use this service for a particular Microsoft server application, the UAL service should be enabled on the host Windows Server. The accessed server application should be registered with the UAL service (as happens automatically during installation). For more information about enabling the UAL service, see Microsoft documentation for the `Get-Service UALSVC` PowerShell cmdlet. For more information about products registered with UAL, see Microsoft documentation for the `Get-UALOverview` PowerShell cmdlet.

The appropriate inventory beacon automatically runs specific scripts to determine which user accessed which server over the last 90 days, and through which inventory device. These scripts are run as a part of the discovery and inventory process.

- **PowerShell**: When this access method is used (for Microsoft Exchange and Microsoft Skype for Business), IT Asset Management collects access data using a PowerShell script that uses specific Microsoft APIs to track the server applications accessed by users. This data is collected as a part of the discovery and inventory process. The following details are not imported when this method is used:
  - Accessing device
  - Access date
  - Access count.

  Note: *IT Asset Management only gets the accessing user details (who accessed the server product) and not the accessing device details (the device through which the server product was accessed). These details are specific only to the date and time when inventory was collected, and not for the last 90 days. Therefore, Device CALs are not supported for this method. Also, when a supported server application (such as Microsoft Exchange, SharePoint, or Skype for Business) is installed on versions of Microsoft Windows Server prior to Microsoft Windows Server 2012 (for example, Microsoft Windows Server 2008 R2, Windows Server 2008, Windows Server 2003 and earlier), the server application will not have UAL support. In these instances, for each access to the supported server application, IT Asset Management creates an access evidence record for Microsoft Windows Server too. This is because a User or Device CAL for Microsoft Windows is also required, in addition to a User or Device CAL for the accessed server application.*

- **SCCM Adapter**: The SCCM adapter has been enhanced to collect the access evidence for Microsoft Endpoint Configuration Manager (previously Microsoft SCCM) accesses. The operator can create the corresponding core CAL license in IT Asset Management.
- **Spreadsheet imports**: You can download the CAL usage inventory template, populate it with evidence data, and upload it to create access evidence records. This method is recommended when you cannot collect access evidence. For example, when UAL service is disabled, or for servers running versions of Microsoft Windows Server prior to Microsoft Windows Server 2008 (with no UAL support). You can also use the `CAL Legacy` license type for manual management of CALs. For more information about spreadsheet imports, see the Importing Inventory Spreadsheets and CSV Files chapter of this guide.

CAL consumption mode

The consumption of CALs is calculated based on the consumption mode set in the **Use rights & roles** tab of the license properties for each CAL. You can select one of the following two options:

- **Consume entitlements based on Access**: If you set the consumption mode to `Access`, IT Asset Management consumes a User CAL for all the users listed on the **All IT Asset Users** page (**Organization > All IT Asset Users**), or a Device CAL for all the devices listed on the **All Inventory** page (**Inventory > Inventory > All Inventory**), unless any restrictions are applied through the **Restrictions** tab of the license properties. If you restrict the scope of this license to a specific enterprise group, a CAL is consumed for each user (in case of User CALs) or each device (in case of Device CALs) that are members of that enterprise group. For more information on license consumption restrictions, see the online help.

  Note: *No access evidence is required when entitlements are consumed based on access.*

- **Consume entitlements based on Usage within the time limit**: If you set the consumption mode to `Usage`, IT Asset Management consumes User or Device CAL for users or devices whose access to a server application has been reported by the appropriate access evidence. Usage tracking must be enabled for the accessed application.

Supported server products for CAL management

The following table outlines the access evidence collection methods and CAL consumption modes supported for the named Microsoft Server applications.

Table 7: Microsoft server products supported for CAL management

Microsoft Product	Product Version	User CALs Consumption		Device CALs Consumption		Inventory Data Source		Notes
		Access based	Usage based	Access based	Usage based	User CAL	Device CAL	
Windows Server	2012	Full	Full	Full	Partial	UAL	UAL	
	2012 R2	Full	Full	Full	Partial	UAL	UAL	
	2016	Full	See notes	Full	See notes			To be supported once Windows Server 2016 is released
SQL Server	2012	Full	Full	Full	Partial	UAL	UAL	Supported only on Windows Server 2012 or later
	2014	Full	Full	Full	Partial	UAL	UAL	Supported only on Windows Server 2012 or later
	2016	Full	Full	Full	Partial	UAL	UAL	Supported only on Windows Server 2012 or later

Microsoft Product	Product Version	User CALs Consumption		Device CALs Consumption		Inventory Data Source		Notes
		Access based	Usage based	Access based	Usage based	User CAL	Device CAL	
Exchange Server	2010	Full	Partial	Full	Limited	PowerShell	Spreadsheet Import	
	2013	Full	Partial	Full	Limited	PowerShell	Spreadsheet Import	
	2016	Full	Partial	Full	Limited	PowerShell	Spreadsheet Import	
SharePoint Server	2010	Full	Limited	Full	See notes	Spreadsheet Import	Spreadsheet Import	Partial support to be added later
	2013	Full	Full	Full	Partial	UAL	UAL	Supported only on Windows Server 2012 or later
	2016	Full	Full	Full	Partial	UAL	UAL	Supported only on Windows Server 2012 or later
Skype for Business (Lync)	2010	Full	Partial	Full	Limited	PowerShell	Spreadsheet Import	
	2013	Full	Partial	Full	Limited	PowerShell	Spreadsheet Import	
	2015	Full	Partial	Full	Limited	PowerShell	Spreadsheet Import	
Microsoft Endpoint Configuration Manager (previously Microsoft SCCM)	2012	Full	Full	Full	Full	Microsoft Endpoint Configuration Manager	Microsoft Endpoint Configuration Manager	Requires Microsoft Endpoint Configuration Manager inventory
	2012 R2	Full	Full	Full	Full	Microsoft Endpoint Configuration Manager	Microsoft Endpoint Configuration Manager	Requires Microsoft Endpoint Configuration Manager inventory
	2016	Full	Full	Full	Full	Microsoft Endpoint Configuration Manager	Microsoft Endpoint Configuration Manager	Requires Microsoft Endpoint Configuration Manager inventory

Where:

- Full: Full support (UAL user data or Microsoft Endpoint Configuration Manager data)
- Partial: Partial support (PowerShell script data or UAL device data)
- Limited: Limited support (requires spreadsheet import of access or usage data)

Note: *IT Asset Management collects the access evidence from the target Microsoft Windows Server, Microsoft SQL Server, and Microsoft SharePoint Server only if the UAL service of Microsoft Windows has been enabled on each accessed server. The accessed server application should be registered with the UAL service (which happens automatically during installation).*

Application usage versus CAL usage

It is important to understand the difference between application usage and CAL usage. In the above diagram, Microsoft SQL Server 2012 is an application installed on the server and recognized by IT Asset Management through imported evidence. When this application is accessed by a user (such as the database administrator) or another application installed on SQL001 device, the application usage is recorded for Microsoft SQL Server. The usage record is linked to the inventory device record of the physical server where the database has been installed.

When the service of this instance of Microsoft SQL Server is requested through a client device (for example, `Lap001`), the access event is recorded (by Microsoft UAL service) and collected for Microsoft SQL Server as CAL usage. The same process is used for Microsoft Windows 2012 on `SQL001`.

How to Manage CALs with IT Asset Management

IT Asset Management supports a number of Microsoft server applications that require CALs. This high-level process outline gives some context and guidance.

To manage CALs within IT Asset Management:

1. Ensure that the inventory import is complete, and that IT Asset Management contains evidence data (including access evidence) for the devices hosting installations of the supported Microsoft server applications that require CALs.

   You can verify this from the **All Inventory** and **All Evidence** pages.

2. To ensure that the access evidence has been collected, generate and review the **CAL Usage Inventory Report**.

3. To identify any server applications accessed without a CAL, go to the **Unlicensed CAL Usage** page (**Licenses > License Management > Unlicensed CAL Usage**).

   Available only after a compliance calculation, this page displays a list of server products that have been accessed without a valid CAL. For each accessed application, you can click the number mentioned in the **Accessing devices** or **Accessing users** column to access the **CAL Usage Inventory Report** and view the related access evidence records. It is important to note that the numbers indicate the distinct users and devices, but the report displays all access evidence records.

4. On the **Access evidence** tab of the **All Evidence** page, ensure that each record of access evidence is linked to the appropriate application.

   A null value for the **Name** column indicates that the evidence is not linked to the appropriate application. In such a case, link the evidence to the appropriate application. For more information, see the online help topics under **Evidence**.

5. Use the **All Licenses** page to check the usage of the existing CALs.

   You can set a filter for `License type= Microsoft User CAL` or `License type= Microsoft Device CAL`, and check the value of the **Consumed** and **Used** columns.

6. Use the **CAL License Summary** report to view the consumption and compliance status for licenses of the type `Microsoft User CAL` and `Microsoft Device CAL`.

   For detailed assignment information, you may check the **Consumption** tab in the license properties of a particular license. Note that some users may be accessing servers from their mobile devices.

7. For every CAL, ensure that you have selected the correct CAL consumption mode value.

   This is set with the **Consume entitlements based on** control on the **Use rights and rules** tab of the license properties. No access evidence is required if this field is set to `Access`.

8. Ensure that you have enough CALs or CAL Suite licenses to cover the usage.

   If you have some unused CALs, you can assign them to other users. You can use any of the following methods to create licenses:

   - *Processing purchases:* Use the **Unprocessed Purchases** page to recalculate and accept the recommendations to automatically create and link CAL purchases to the recommended CALs. For more information, see *Unprocessed Purchases* in the online help.
   - *Manual creation and linking:* You can create a license manually and link it to the appropriate server application. For more information, see the *Creating a License* topic in the online help.

   Note: *The CAL Legacy license type is still supported for backward compatibility. You can use this license type to manually manage CALs when the access evidence cannot be collected for any reason.*

9. If required, configure the use rights for each CAL license, as well as for every product of a CAL Suite license.

   See the **Use Rights & Rules Tab** in the online help.

10. If required, configure the required exemptions and use rights and rules on CALs.

    See **License Properties > Use rights & rules** and **Consumption** tabs in the online help.

11. If required, you may allocate users and devices to appropriate CALs.

    Consider the following example use cases:

    - For a particular access event, you want a user to consume a CAL Suite license instead of a single-product CAL.
    - You need to exempt a user from consuming a CAL. To exempt, you need to allocate first.

    See **License Properties > Consumption tab** in the online help. Note that in the case of CALs, allocations always consume entitlements.

    Tip: *In usage scenarios where many external application users use a single server account to access a server, the* ***Overridden consumption*** *column value (on the* ***Consumption*** *tab of license properties) should be adjusted to show the actual consumption. For example, if multiple external-product users are using the same Microsoft SQL Server user to access the database, the UAL data will reveal access evidence for only one user of Microsoft SQL Server. In such cases, the usage should be adjusted to actual number of users to remain license compliant. The value of the* ***Overridden consumption*** *column should be edited to reflect the actual usage.*

12. When consuming CALs in the `Access` mode, you can apply restrictions to ensure that the CALs are consumed by appropriate users or devices.

    For more information, see **License Properties > Restrictions tab** in the online help.

13. You may use the following reports to review the CAL usage in your organization:

    - **CAL License Summary**
    - **Licenses Consumed and Allocated by Enterprise Groups**.

Example Use Cases for CAL Management

This topic highlights some example scenarios for CAL management, and describes how each one of them should be managed using IT Asset Management.

Restricting CAL consumption to a specific group

You can restrict the consumption of a particular CAL to a specific enterprise group. For example, an installation of Microsoft SharePoint should only be accessed by users from the `Support` group. The following are the steps to implement this scenario using IT Asset Management:

1. Create a license of the type `Microsoft User CAL` for Microsoft SharePoint Server.
2. Ensure that all the required users have been added to the `Support` corporate group.
3. In the **Restrictions** tab of the license properties, search and add the `Support` group.
4. After the reconciliation is complete, the Microsoft SharePoint User CAL should be consumed only for the users of the added group.

Managing User and Device CALs together

Your CAL procurement strategy may include a mix of User and Device CALs. Consider the following example scenario:

- Out of the 150 users of Microsoft Exchange, 100 engineering users are full time, and work from dedicated devices.
- The remaining 50 users are in the Support department and work in two shifts, sharing 25 computers.

The following are the steps to implement this scenario using IT Asset Management:

1. Create two corporate units: `Engineering` and `Support`.
2. Create a license of the type `Microsoft User CAL` for the accessed Microsoft Exchange Server.
3. Ensure that all 100 full-time users have been added to the `Engineering` group.
4. In the **Restrictions** tab of the license properties of the `Microsoft User CAL`, search and add the `Engineering` group. The license will now be consumed only for the users of the added group.
5. Create a license of the type `Microsoft Device CAL` for the accessed Microsoft Exchange Server.
6. Ensure that all 25 support computers have been added to the `Support` group.
7. In the **Restrictions** tab of the license properties of the `Microsoft Device CAL`, search and add the `Support` group. The license will now be consumed only for the users of the `Support` group.

Note: *If the Support department users are also allowed to access the Microsoft Exchange Server via their mobile devices, home PCs, or laptops, Microsoft User CALs may be a more cost effective option to consider.*

Managing CALs for accesses from unknown external devices

Device details may not be available when a device running Microsoft Windows Server is accessed by external users (such as consultants away on customer sites). In such cases, the access evidence records show just the IP address of the accessing device. You can use the following steps to adjust the CAL consumption in this scenario:

1. Navigate to the **Unlicensed CAL Usage** page and note the number in the **Accessing devices** column for the accessed product (such as Microsoft Windows Server).
2. Go to the **All Licenses** page and search the CAL for that product.
3. Open the license properties of the license and click the **Consumption** tab.
4. Add the number mentioned in the **Accessing devices** column on the **Unlicensed CAL Usage** page to the count mentioned in the **Allocated point** for this license. This will adjust the number of unknown accesses being made to the server application.

Managing CALs for multiple indirect accesses to a server application

Multiple users of a web application or website may access a server application through the same account. For example, 100 users of an ERP solution may access Microsoft SQL Server through an internally-configured `ErpQuery` account. As only one user-account of SQL server is accessing the application, the access evidence will only indicate access by one user, instead of reporting access by 100 users. To be license compliant, you should consume 100 User CALs for Microsoft SQL Server. The following are the steps to manage CALs in this scenario:

1. Check the CAL Usage Inventory report to find a very high value in the **Access Count** column in an access record. This would provide a hint that multiple users are accessing a server application.
2. Using any method to record the number of indirect accesses, make a note of the number of users of the ERP application that are indirectly accessing the Microsoft SQL Server installation.
3. Go to the **All Licenses** page and search the appropriate User CAL license for Microsoft SQL Server.
4. Open the license properties of the license and click the **Consumption** tab.
5. Add the number noted in step 1 to the count mentioned in the **Allocated point** for this license. This will adjust the consumption for the accesses being made to the server application.

Note: *The license requirements would be different In the case of unknown external users or devices accessing an Internet-facing site that is running on Microsoft SharePoint Server. If Microsoft SharePoint version 2013 or 2016 is being used, the server license itself covers unlimited external accesses, and no additional CALs are required. For Microsoft SharePoint version 2010 and earlier, you need an External Connector Licenses to cover the anonymous access.*

Appendix A- Template Details for CAL Usage Inventory Upload

You can import CAL usage inventory through spreadsheet imports when you cannot collect access evidence for the accessed server applications. You can schedule regular inventory imports through the FlexNet Beacon, or you can use the Inventory Data One-Off Uploads feature (see *Data Inputs* in the online help) to upload CAL usage inventory spreadsheet. For more information on inventory uploads, see the *Importing Inventory Spreadsheets and CSV Files* chapter of this guide.

To upload CAL usage inventory, the spreadsheet template should be populated with at least all mandatory (**Required**) columns, and uploaded to IT Asset Management. The following table describes the columns of the CAL Usage Inventory spreadsheet template, accessible through the Inventory **Data One-Off Upload** page:

Column	Description	Mandatory
AccessCount	The number of times that the server application was accessed.	No
AccessDate	The date on which the server application was accessed.	No
AccessingDeviceComputerName	The **ComputerName** of the accessing device (through which the server application was accessed).	No, if **AccessingUser** is specified.
AccessingDeviceDomain	The domain name of the accessing device.	No
AccessingDeviceIPAddress	The IP address of the accessing device.	No, if **AccessingUser** is specified.
AccessingDeviceSerialNo	The serial number of the accessing device.	No
AccessingUser	The `DOMAIN/SAMAccountName` for the accessing user.	No, if **AccessingDeviceIPAddress** or **AccessingDeviceComputerName** is specified.
ComputerID	The ComputerID of the accessed device (where the server application has been installed). It must match the **ComputerID** specified for this device in the **Computer** spreadsheet, or the row will be ignored. Uploading the Computer spreadsheet is mandatory with the CAL inventory upload.	Yes
Edition	The edition of the accessed server application as reported by the access evidence. This data is not used for access recognition.	No

Column	Description	Mandatory
InventoryDate	Inventory date of the evidence. If not provided, defaults to the current Coordinated Universal Time (UTC).	No
ProductName	The product name of the accessed application as reported by the access evidence. For all supported server applications except Microsoft Endpoint Configuration Manager (previously Microsoft SCCM), the Version and Edition (if present) is included in the ProductName. For example, `Microsoft SharePoint Server 2013 Enterprise Preview`	Yes
Version	The version of the accessed server application as reported by the access evidence. ***Note:*** *This column should be populated only for Microsoft Endpoint Configuration Manager (previously Microsoft SCCM).*	No

7

Oracle Discovery and Inventory

Discovery and inventory information is a prerequisite for performing license compliance calculations in IT Asset Management. In addition to the general inventory collection features, IT Asset Management also offers specialized features for Oracle inventory collection. With a detailed description of each of the supported methods, this chapter may help you in deciding and implementing the appropriate inventory collection method for Oracle systems in your computing estate.

The chapter begins with introductory topics comparing the different approaches to help you choose between them.

These are followed by a section covering each approach, each of which includes these parts:

1. The first topic introducing each section lists the prerequisites required for that approach
2. Then the credentials (accounts and privileges) required for that approach are listed in detail
3. There is a topic detailing how the particular approach operates
4. Finally, troubleshooting that applies to the particular approach is covered.

The concept is that, using the introductory topics, you will choose just one approach that best fits your needs, and then read only the one related section to drill into detail, without needing to work through the variations for the other approaches. This separation should reduce possible confusion about the differences between the approaches.

To reduce cross-linking, each section *repeats* the content that is common to other approaches, so that reading only your chosen one section gives you complete coverage. You will not miss anything by skipping over the sections that are not relevant to your approach.

The chapter concludes with a number of appendices of less commonly required details that are (in general) common to all approaches.

Introduction to Oracle Discovery and Inventory

The term *Oracle discovery and inventory* refers to the process of examining a network to find and collect hardware and software inventory for every device with one or more Oracle applications installed on it. IT Asset Management performs software license compliance calculations on the collected inventory data to provide information about what software you have licensed against what software you have installed.

Oracle discovery and inventory collection, and the resulting license consumption calculations, include the following main activities:

1. **Discovery:** Identifying the devices on the network that have Oracle software installed. (Discovery is *not* needed for devices that meet one of these conditions:
   - They already have the FlexNet Inventory Agent locally installed on them
   - They are identified in a `tnsnames.ora` file used by an inventory beacon to directly gather software inventory for the Oracle database instances listed there
   - Their details of Oracle listeners and services have been manually entered in the discovered device properties
   - They have been identified by the Amazon connector as running Oracle Database in the Amazon Relational Database Service [RDS].)
2. **Inventory:** Collecting specific Oracle Database inventory, together (wherever possible) with hardware and general software inventory
3. **Compliance calculations:** Calculating of the number of license entitlements consumed, and comparing with the number of purchase licenses.

Tip: *The Oracle discovery and inventory process is the same for both physical and virtual machines.*

At the high level, there are two main ways to collect Oracle inventory in IT Asset Management:

1. You can deploy the FlexNet Inventory Agent, and in particular its core inventory collection executable `ndtrack`. This is the optimal collection method, as it not only collects Oracle inventory, but also collects hardware and other software inventory at the same time. The hardware data is important for correct calculation of consumption for Oracle license types (for example, the Oracle Processor license type has specialized calculation methods used for inventory from Solaris zones, where the core and thread counts are mandatory — in fact, the absence of core and thread counts for the VM *host* prevents consumption from Oracle Processor licenses for software running in *any* guest VM on that host). Methods of deployment are fully discussed in *Gathering FlexNet Inventory*. Relevant deployment methods for Oracle inventory include the following cases defined in that document:
   - **The Adopted case**, where the FlexNet Inventory Agent is deployed automatically through an inventory beacon directly onto the target Oracle server.
   - **The Agent third-party deployment case**, where you use a tool or process external to IT Asset Management to deploy (and perhaps also manage) the FlexNet Inventory Agent. One of these possibilities is to deploy the FlexNet Inventory Agent to a shared network folder, and set up a process to execute it on target Oracle servers.
   - **The FlexNet Inventory Scanner case**, where the lightweight FlexNet Inventory Scanner is used instead of the full agent.
   - **The Zero-footprint case**, where an inventory beacon temporarily downloads the FlexNet Inventory Agent to the target device, executes it there, and subsequently removes it again (leaving no permanent footprint).
2. You can use FlexNet Beacon on any conveniently-located inventory beacon to connect *directly* to the listener service for an Oracle database instance and collect inventory information through it. This method, called 'direct' inventory gathering, may be helpful, but it has the following limitations:
   - It does not gather hardware inventory. Hardware details are needed for calculating Oracle license positions; so gathering direct database inventory alone is not sufficient for Oracle license management. If you take this path, you must augment the direct database inventory with additional hardware inventory information (which may

come from third-party inventory tools).

Tip: *In the special case of Oracle Database running in Amazon RDS, the inventory collected by direct connection is augmented by details gathered by the Amazon connector, and the combination allows for a count of* ***Threads*** *(which Oracle takes as vCPUs) in the inventory device record, so that license consumption calculations can proceed.*

- It cannot collect any inventory from the standby database instance in an Active Data Guard configuration (although, of course, the active database instance is inventoried as usual).
- It does not gather inventory for other Oracle applications like Oracle WebLogic that may be installed on the same server; you may plan to collect this inventory with third-party tools as well.
- Direct inventory gathering combined with a network scanning discovery method cannot operate with Oracle 12c and beyond (as explained in Direct Collection of Oracle Inventory).

For all these reasons, the recommendation is to choose your preferred method to deploy the FlexNet Inventory Agent, as listed above.

While IT Asset Management can also import inventory data using `.xlsx` or `.csv` file uploads, these are not normally convenient for regular updates about Oracle software. They are available as a method of last resort when simpler approaches are not possible in your environment. For details about importing data through spreadsheets, see the Importing Inventory Spreadsheets and CSV Files chapter of this guide, or see *Managing Inventory Spreadsheet Connections* in the online help.

Tip: *When Oracle Enterprise Cloud Control is used to manage Oracle databases, be sure to collect inventory information from all database instances:*

- *Inventory from the database instance used as storage by the Oracle Cloud Management Pack (OCMP) is mandatory to provide data about the installed Oracle options, including those from OCMP itself. FlexNet inventory from this server includes a list of remote Oracle servers managed by OCMP and the database instances running on them; but it does not (and cannot) include detailed inventory for those remote devices or database instances. (The list of servers and the database instances they host is also included in the Oracle GLAS audit report available through IT Asset Management.)*
- *Inventory (including hardware inventory) from each of the managed Oracle servers and the database instances they host is needed to provide:*
  - *Hardware data needed to calculate the license consumption for the managing installation of OCMP (this license takes account of the processor cores available on each managed server)*
  - *Data on any other Oracle options installed on each server*
  - *Complete inventory information on each database instance.*

When all inventory data from all servers has been imported into the compliance database in IT Asset Management, the various inventories are merged to provide correct license consumption calculations for the database instances, the Oracle options installed on each server, and the installed options from the OCMP (taking into account the processor core counts of the managed servers). The required FlexNet inventory can be gathered by any supported method — keeping in mind that if you choose the direct inventory gathering method, you also need an auxiliary method of inventory gathering to provide the required hardware inventory. As a side benefit, the simple expedient of collecting inventory from every database instance ensures that you also correctly calculate separate license consumption for those instances that you choose to manage outside Oracle Enterprise Cloud Control (perhaps to manage the licensing

costs related to those options).

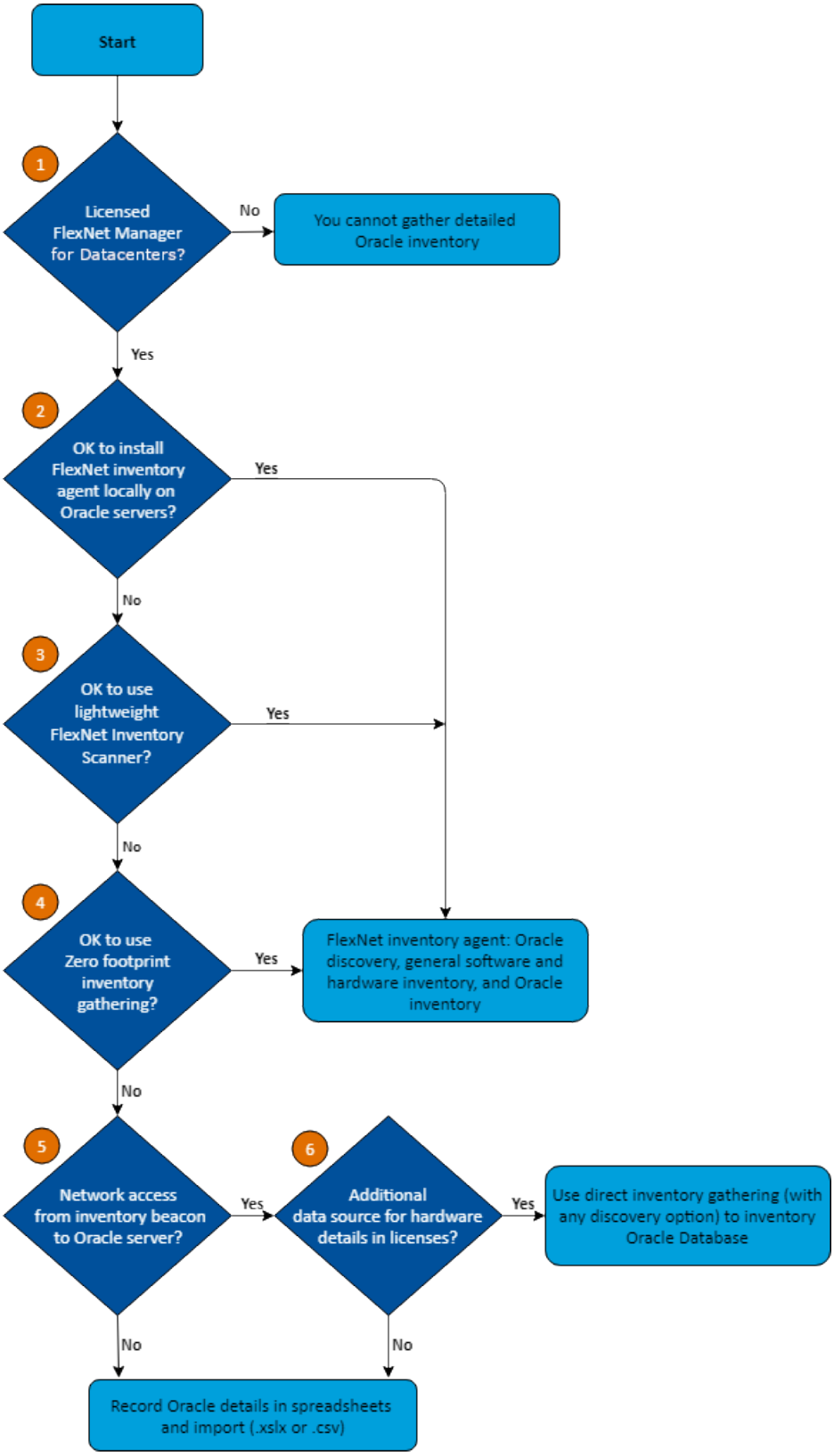
Selecting an Oracle Inventory Collection Method

The selection of a particular Oracle inventory gathering method depends on your software installation policies, and your network and database access policies determined by your network and Oracle Database administrators. For example, you may prefer to either:

- Deploy the FlexNet Inventory Agent on each of your Oracle servers
- Collect database-only inventory using one or more inventory beacons connecting through a listener to each database instance (called 'direct' inventory).

The following diagram provides a broad overview for selection of an Oracle inventory collection method. Numbers in the diagram correspond to the description below. Further details of each inventory collection method are discussed on the following pages.

Tip: *In the following discussion, the term "UNIX-like platforms" is used to group together AIX, HP-UX, all supported flavors of Linux (CentOS, Debian, Fedora, Oracle, Red Hat), OS X, and Solaris.*



1. To perform license management and compliance for Oracle-specific licenses, you need a license for the FlexNet Manager for Datacenters product. Without this product, you can only discover Oracle infrastructure and collect

basic Oracle software inventory. For more details, see Appendix F: Features Enabled in FlexNet Manager for Datacenters.

2. If you can install a FlexNet Inventory Agent on each of the target Oracle servers, this provides the most complete solution. You may use a discovery and inventory rule to 'adopt' Oracle servers (that is, automatically install the FlexNet Inventory Agent on them, and continue to manage the behavior through device policy); or you may deploy the FlexNet Inventory Agent with the tool of your choice. On a global schedule, each of the locally-installed FlexNet Inventory Agents collects the inventory for its device, and uploads the results to the appropriate inventory beacon. For more information, see How Agent-Based Collection of Oracle Inventory Works.

3. The FlexNet Inventory Scanner offers a light-weight alternative to the full FlexNet Inventory Agent, for a moderate trade-off in functionality (for example, if the first attempt at an inventory upload across the network fails for some transient reason, there is no catch-up). You also have a little more flexibility in its set-up, as you take responsibility for its scheduling as well as its deployment. For details, see How the FlexNet Inventory Scanner Collects Oracle Inventory.

4. When local installation is not an option, you may collect inventory through the Zero-footprint method. Here the FlexNet inventory core components saved on the inventory beacon are (for Windows) executed from this file share, or (for UNIX-like platforms) downloaded, temporarily installed, executed, and then removed from the target device, leaving no permanent installation footprint. The process is managed by a discovery and inventory rule configured on the central application server for IT Asset Management, and the same rule dictates the inventory collection schedule. For details, see How Zero-footprint Collection of Oracle Inventory Works. In terms of the results achieved, this is functionally identical to the locally-installed agent option.

Tip: *Any of the above three methods of using a local copy of the core inventory component (`ndtrack`) on the Oracle database server provide additional features that are not supported by the later options in the flowchart above:*

- *The local inventory component collects more detailed hardware information, to the levels required for some of the more advanced Oracle license types (for example, collecting the count of available processors, necessary for consumption calculations for Oracle Processor licenses)*
- *Provided that the installed inventory component is version 12.4 or later (that is, released with IT Asset Management 2017 R3 or later), it can also gather inventory from Oracle database instances that are in standby (that is, `MOUNTED`, but not in `READ` mode). This includes, for example, the standby database instance in an Active Data Guard configuration.*

5. If you do *not* have a direct network connection allowing an inventory beacon to communicate with the target Oracle server(s), the remaining option is to maintain Oracle Database inventory details in a spreadsheet format. You can then save the spreadsheet on an inventory beacon and have it automatically uploaded and included in Oracle license consumption calculations. However, maintaining data in this form over time is a high maintenance burden, and this approach is not recommended except as a last resort.

6. Since you *do* have network access from an inventory beacon to your Oracle server(s), you can use direct gathering of Oracle Database inventory by the inventory beacon for each Oracle database instance (called 'direct' because the inventory beacon connects directly to the Oracle database instance to gather information — although as always this remote connection is brokered by a listener). There are three different ways that the system can discover the Oracle database instances from which to gather the inventory, and all three discovery/direct inventory combinations are detailed under How Direct Collection of Oracle Inventory Works. However, direct inventory gathering *alone* has two important shortcomings, the first affecting FlexNet inventory and the second relating to Oracle GLAS data for a potential audit:

- For FlexNet inventory gathering, the direct method does not provide sufficient information to calculate consumption for all Oracle licenses (such as Oracle Named User Plus and Processor-based licenses), because these either require knowledge about the licensing of other Oracle options on the same server, or require additional details about the hardware that the database instance is running on. Therefore to use direct inventory gathering successfully for Oracle license management, you must have an additional source of inventory data both for the Oracle server hardware, and for any additional Oracle software (other than Oracle Database itself) that you want to manage. If you do not have these auxiliary data sources, you could revert to manual management of your data in spreadsheets, and importing those. (However, since spreadsheet maintenance is generally an unattractive option, you may wish to reconsider an alternative from earlier in the flowchart!)
- The Oracle Global Licensing and Advisory Services (GLAS) provides scripts to collect both hardware and software data suitable for use in an Oracle audit. These scripts, as updated from time to time by Oracle, are distributed as required within the `InventorySettings.xml` file (delivered to you through the ARL updates because you have licensed the FlexNet Manager for Datacenters product). The scripts are automatically executed during all methods of collecting Oracle inventory, providing data exclusively packaged for your delivery to Oracle if required. However, there is one very important limitation with the direct method of inventory gathering: of the two parts provided in the GLAS scripts (hardware scripts to execute on the target Oracle server, and SQL queries to execute against each Oracle database instance), *only* the software data can be gathered during the direct connection to each database instance. The hardware scripts are not downloaded to the Oracle server, and so cannot be executed. If you are determined to use the direct method of Oracle inventory collection, be prepared to use alternate methods to execute the GLAS hardware scripts on each Oracle server in order to complete the Oracle GLAS audit requirements. Or, once again, reconsider one of first three methods from the flowchart, since each of these methods collects *both* hardware and software data required for Oracle GLAS, without additional effort on your part.

One possibility is to *combine* direct gathering of Oracle inventory with local installation of either the FlexNet Inventory Agent or the FlexNet Inventory Scanner. At first glance, this may seem like obvious redundancy, but there is a corner case where this approach may be helpful: where your IT policies do not allow for OS authentication to be used with one or more of your Oracle Database installations. Since the absence of OS authentication prevents any locally installed inventory component from accessing the Oracle database instances on this server, it becomes a sensible compromise on such a server to use the locally-installed FlexNet inventory component to collect normal software and hardware inventory (including for other Oracle options on the server), *combined with* direct inventory gathering to cover the database instances on the same server.

Comparison of Inventory Collection Methods

The following table presents a summary comparison of different inventory collection methods available with IT Asset Management. In the table:

- "Y" means that the item applies to the method
- "O" means that the item is optional, or its use depends on the details of your implementation.

Feature	Agent-based	Scanner	Zero-footprint	Direct	Spreadsheet
Components Required					
FlexNet Inventory Agent	Y				
FlexNet Inventory Scanner		Y			
FlexNet Beacon installed on an inventory beacon	Y	Y	Y	Y	Y
Discovery and inventory rules	O		Y	Y	
Compatible OLEDB drivers from Oracle on inventory beacon				Y	
The `tnsnames.ora` file				O	
The OEM adapter				O	
Privileges Required					
For adoption on Windows: local or domain account (registered in Password Manager) with full access to Windows Service Control Manager on the Oracle server	Y				
For operation on Windows: • `LocalSystem` or administrator account • Registered in Password Manager • Read-only access to Windows Service Control Manager on Oracle server • Member of the Windows local security group `ora_dba` (where `LocalSystem` appears as `NT AUTHORITY\SYSTEM`).	Y		Y		
For non-privileged operation on Windows: • A non-privileged account to which you have given all access rights required for the discovery and inventory collection you require (this unsupported approach is not documented, as it is specific to your environment) • Member of the Windows local security group `ora_dba` (where `LocalSystem` appears as `NT AUTHORITY\SYSTEM`) • Has permissions to invoke the commands listed in the *Common: Child Processes on Windows Platforms* topic in *Gathering FlexNet Inventory*.		Y			
For adoption on UNIX-like platforms: • Local account with `ssh` privileges that can elevate to `root`-level privileges • Registered in Password Manager.	Y				

Feature	Agent-based	Scanner	Zero-footprint	Direct	Spreadsheet
For operation on UNIX-like platforms: The tracker runs as `root`. (If the tracker is run as any other account, Oracle inventory is not collected.) This is true for all agent-based inventory gathering: • The Adopted case • The Agent third-party deployment case • The FlexNet Inventory Scanner case • The Zero-footprint case.  ***Tip:*** *As always, it makes no difference whether you invoke the tracker directly as `root`, or whether you run as another account and use `sudo` (or similar) to elevate to `root` before invoking the tracker.*	Y	Y	Y		
An account with read-only permissions on every Oracle Database for all the tables and views needed for collecting Oracle inventory.				Y	
Collected Information					
General hardware and software inventory	Y	Y	Y		O
Oracle Database inventory	Y	Y	Y	Y	O
Inventory from standby database instances (with tracker version 12.4 or later)	Y	Y	Y		
Oracle application and engineering system inventory	Y	Y	Y		O
Oracle GLAS data for audit * The direct inventory collection method collects only the *software* data required by Oracle GLAS. For a complete audit data set, you will need another method to execute the GLAS hardware scripts on each Oracle server.	Y	Y	Y	O*	
Inventory from Oracle pluggable databases (Oracle 12c and later) — requires `InventorySettings.xml` version 27 or later.	Y	Y	Y	Y	O
User Actions Required					
Create a discovery and inventory rule. In the Adopted case (but not in the Agent third-party deployment case), ensure that the **Allow these targets to be adopted** option is selected in the target definition.	O				

Feature	Agent-based	Scanner	Zero-footprint	Direct	Spreadsheet
Create a discovery and inventory rule, and in the action definition: • For **Action type**, choose `Discovery and inventory` • Select **Network scan** as the discovery method • In the **General hardware and software inventory** section, select **Gather hardware and software inventory from all target devices**. If you wish to target exclusively Oracle servers, specify appropriate IP addresses, machine name patterns, or port details to focus the settings under **Define machines to target** in the **Targets** tab.			Y		
To use a network scan, create a discovery and inventory rule with the following options selected in the action definition: • For **Action type**, choose `Discovery and inventory` • Select **Network scan** as the discovery method • In the **Oracle database environments** section, select the **Discover Oracle database environments**, **Port scan** and/or **SNMP scan**, and **Gather Oracle database environment inventory** options.				O	
To use a `TNSNames.ora` file, create a discovery and inventory rule with the following options selected in the action definition: • For **Action type**, choose `Discovery and inventory` • Select **TNSNames file for Oracle databases** as the discovery method (you may optionally combine this, for example, with **Network scan**) • In the **Oracle database environments** section, select the **Discover Oracle database environments**, **Port scan** and/or **SNMP scan**, and **Gather Oracle database environment inventory** options.				O	
To use manually-created data to replace discovery, create discovery device records with listener and services information for all Oracle servers; and then create a discovery and inventory rule with the following options selected in the action definition: • For **Action type**, choose `Inventory only` • In the **Oracle database environments** section, select the **Gather Oracle database environment inventory** option.				O	
Separately execute (and report results for) the Oracle GLAS hardware scripts, since these cannot be executed in the direct inventory collection case.				Y	

Feature	Agent-based	Scanner	Zero-footprint	Direct	Spreadsheet
Schedule file uploads from the FlexNet Beacon.					Y

Interaction with Oracle Enterprise Manager

If you are using Oracle Enterprise Manager to manage various Oracle Database installations (together with the many database instances that these may support), there are two completely separate ways that IT Asset Management can interact with your installation(s) of Oracle Enterprise Manager, using it either:

- As a discovery tool to identify database instances
- As a management reporting tool to track Oracle options/applications being remotely managed on [some of] those database instances.

These two approaches are entirely independent, and you may use both within your enterprise, if need be.

Discovery

The goal here is to create an authoritative listing of all the database instances running on Oracle servers within your enterprise. This discovery process helps ensure that your Oracle inventory can be complete, and avoids nasty surprises if you are ever involved in an Oracle audit.

The approach is as follows:

- A unique copy of the OEM adapter, a utility from Flexera, is associated with each installation of Oracle Enterprise Manager.
- The OEM adapter extracts the identities of all the database instances known to this installation of Oracle Enterprise Manager. It structures the data into the Oracle-standard file format, `tnsnames.ora`.
- The OEM adapter is also linked to a particular inventory beacon, and automatically writes its prepared `tnsnames.ora` file into a known path on the inventory beacon. Because the OEM adapter always writes a file with the same name into the same path, the easiest configuration is to support each OEM adapter with its own distinct inventory beacon.
- In due course, the inventory beacon connects directly to each database instance identified in [all of the] `.ora` files waiting in the special directory, and collects Oracle inventory from each database instance (the "direct" inventory model). The inventory from each database instances is saved into an `.ndi` file, and uploaded to the central application server along with all other collected inventory.

You can find more insight in the topic Using tnsnames Discovery with Direct Inventory within this chapter. Details on obtaining, deploying, and configuring the OEM adapter are included in *IT Asset Management Inventory Adapters and Connectors Reference*.

Reporting on management of Oracle options

Oracle options (and management packs or applications) attached to database instances have separate licensing implications, and must be tracked. An Oracle administrator may authorize their use in either of two ways:

- She may log in directly to the database instance, and authorize use of an option locally — in inventory terms, this option is visible only in the inventory gathered locally on this Oracle server
- She may access Oracle Enterprise Manager and authorize the option from there, acting 'remotely' on the target database instance — this option is then visible both in the local inventory gathered from the remote database instance, and also in the inventory gathered from the OEM repository (the particular database instance within which Oracle Enterprise Manager stores its management data, but here named differently to reduce confusion).

Of course, these actions may be conducted by different administrators, even on the same database instance; and things may get confusing. To help you sort out where Oracle options were authorized, IT Asset Management displays the name of the Oracle Enterprise Manager server that reports management information for a given database instance (visible in the properties of the database instance). In the same properties pages, you can see every option authorized for the database instance, and for each option see whether it was authorized locally on the database instance, or remotely by someone using Oracle Enterprise Manager.

And the good news is this requires no special set-up or configuration. It does not matter which method of collecting Oracle inventory you use:

- Agent-based, with the locally installed FlexNet Inventory Agent deployed either through automated adoption or with your preferred third-party tools
- Zero-footprint, where the FlexNet Inventory Agent is remotely installed by an inventory beacon and removed again after its work is done
- The light-weight FlexNet Inventory Scanner (provided that you have also deployed `InventorySettings.xml` to the folder where the scanner executes)
- Direct inventory-gathering from the inventory beacon accessing the listener for the database instance.

In all those cases, when inventory is collected from the database instance, it automatically includes any Oracle Enterprise Manager data that is saved in that database instance (which, as then becomes obvious, is also an OEM repository). After the inventory from all database instances is uploaded and imported, IT Asset Management identifies the managing installation of Oracle Enterprise Manager against both:

- Each target database instance, and
- The individual Oracle options on it that were authorized through Oracle Enterprise Manager.

You can see the results in the web interface for IT Asset Management by opening the properties sheet for the target database instance (for example, go to the **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**), and click the chosen name in the **Instance** column). Look in the **General** tab of the properties to see the **Managing OEM**, the particular installation of Oracle Enterprise Manager that, within its *own* inventory of the OEM repository, most recently showed details about this target database instance. Switch to the **Options** tab in the same properties sheet to see the Oracle options associated with this database instance, and whether they were authorized locally (on `This instance`) or remotely (by `Oracle Enterprise Manager`, being that installation identified on the **General** tab).

Tip: *One point about configuration is to ensure you are using appropriate versions. From Oracle 12c, the OEM repository may be saved in a* ***pluggable*** *database. If you use Oracle 12c (or later), you must also use a version of the inventory component (`ndtrack`) that can collect inventory from pluggable databases. This requires version 13.0.1 of*

*the FlexNet Inventory Agent, which was distributed with IT Asset Management 2018 R1 hotfix 02. This (or a later) agent remains backward compatible with inventory beacons and the central application server for version 2016 and later, so that if need be, you can update **only** FlexNet Inventory Agent.*

Some important points to notice are these:

- Obviously, linking the two inventory sources (local database instance and OEM repository) relies on *both* inventory files being imported. Furthermore, they are matched by the hostname for the database instance recorded in both inventories. If, for example, your OEM installation reports database instances not by hostname but by IP address, then the matching cannot occur. In that case, you would see separate inventories for the two database instances involved (the target database instance and the OEM repository instance), but no connection between them.
- IT Asset Management identifies the OEM installation by the *Oracle Database server* where the management information was found. In the normal course of events where the OEM console and the OEM repository are on the same server, this goes unnoticed, and you can easily recognize the server name. However, if you have separated your OEM console (on, say, `myConsoleServer`) from the OEM repository (on, say, `fatOracleDatabaseServer`), you may not instantly recognize the latter as the OEM server name when you have in mind the different name for your OEM console.
- Your configuration of Oracle Enterprise Manager can become even more exotic. "Each administrator is associated with a specific repository... The repository tables can be installed in any database accessible to the Console. ... Also, the repositories for the administrators do not have to be in the same database." (https://docs.oracle.com/cd/A57673_01/DOC/sysman/doc/A55896_01/ch1.htm) Suppose that administrators Sam and Pat have their own *separate* OEM repositories, and these happen to be on different Oracle Database servers (`svrSamDB` and `svrPatDB`). Now, both Sam and Pat remotely access the same database instance `ODBINS47`, and both authorize some Oracle options on `ODBINS47`. You can see that when inventory is imported from `svrSamDB`, the properties of `ODBINS47` are updated to show that `svrSamDB` is its managing installation of OEM; and later, when inventory is processed from `svrPatDB`, the properties of `ODBINS47` switch over to reveal its new managing installation of OEM. In such a fluid environment, the properties of each database instance show management by the installation of OEM *most recently imported*. The behavior of linking the *latest* import from an OEM repository works well if you are actually switching management from one installation of OEM to another. The new management OEM is displayed after the very next Oracle inventory import (typically part of the overnight full inventory import and license consumption calculations). However, if you really are allowing multiple administrators working from separate OEM repositories to manage the same database instance, the assigned **Managing OEM** field reveals only the most recent import from an OEM repository. In less fluid, "normal" environments, where each database instance is managed by a single installation of OEM using an OEM repository on a *single* database server, you can rely on the stable presentation of these facts.

For more information, see the online help for the Oracle database instances properties sheet.

Special Handling of Oracle Fusion Middleware

Oracle Fusion Middleware is a large collection of software products used for the development, deployment, and management of software services. Oracle have specific conditions for tracking licensing for Oracle Fusion Middleware, and IT Asset Management is verified by Oracle as an acceptable tool to collect the appropriate inventory.

Oracle requires that a full system scan is performed to check for Oracle Fusion Middleware, and that special file evidence is collected and submitted as part of your `OracleGLASEvidence.zip` archive for reporting or as part of an audit response.

For your convenience, the standard FlexNet Inventory Agent (version 17.0.1 or later) is able to collect the required evidence, along with the additional files that Oracle requires, ready for inclusion in the `OracleGLASEvidence.zip` archive. This functionality is available in any of:

- The complete FlexNet Inventory Agent locally installed on a target inventory device
- The lightweight FlexNet Inventory Scanner, where the self-extracting executables are available on the target inventory device
- The zero footprint inventory collection method, where the necessary inventory components are downloaded from an inventory beacon, executed, and subsequently removed from the target inventory device.

(Details of these methods are available in *Gathering FlexNet Inventory*.)

However, because the full system scan on each inventory device (and resulting data uploads) can be time-consuming, this functionality is disabled by default. There are two stages of control available, both of which default to off (`false`):

- The **Enable collection of Oracle Fusion Middleware audit data** check box (located in the **Oracle Fusion Middleware scanning** section of the **Inventory Settings** page (**Data Collection > IT Assets Inventory Tasks > Inventory Settings**)) enables the inventory collection process.
- The **Include Oracle Fusion Middleware** (on the **Inventory** tab of the IT Asset Management Settings **General** page (**Administration > IT Asset Management Settings > General**)) allows the uploaded data and files to be incorporated into the `OracleGLASEvidence.zip` archive.

When the first of these, the **Enable collection of Oracle Fusion Middleware audit data** check box, is selected:

1. The change is distributed through your inventory beacons as device policy to all managed inventory devices (those receiving device policy through any inventory beacon, and subsequently automatically uploading inventory through the inventory beacon hierarchy).

   Tip: *This mechanism automatically updates locally-installed instances of FlexNet Inventory Agent, and the components on the inventory beacon used for zero footprint inventory collection. However, if you have deployed the lightweight FlexNet Inventory Scanner, ensure that you also update its associated copy of* `InventorySettings.xml`*, and if necessary update its command line with the* `-o PerformOracleFMWScan=true` *option.*

2. Each inventory device saves the downloaded preference in the `PerformOracleFMWScan` preference in the registry (or pseudo-registry file on UNIX-like platforms).

   Tip: *If you have disconnected inventory beacons, or inventory devices that are not managed automatically, you can arrange for your preferred registry tool to set this preference to* `true` *to enable this scanning by all locally-installed copies of FlexNet Inventory Agent.*

3. If the FlexNet Inventory Agent needs to scan for file evidence (as authorized by the `IncludeDirectory` preference), it checks whether `PerformOracleFMWScan` is also true, and if so, it also scans each of the visited folders for the required Oracle Fusion Middleware evidence.
4. When the FlexNet Inventory Agent completes its regular (default: daily) inventory collection, including any required

file evidence collection, it again checks whether `PerformOracleFMWScan` is true, and if so, begins a scan of the entire system for any remaining Oracle Fusion Middleware evidence. However, this is an intelligent scan that knows which folders have already been scanned as part of the normal file evidence gathering process, and so it skips all directories previously scanned. To optimize performance, no folders are scanned twice.

5. All found content related to Oracle Fusion Middleware on the inventory device is organized into a specific structure, and then archived, and included as a blob of binary data within the standard archived `.ndi` file.
6. As always, the inventory (`.ndi`) files are uploaded through your hierarchy of inventory beacons, and are eventually imported into IT Asset Management through the standard processes.
7. During import, the evidence for Oracle Fusion Middleware is processed and saved in the compliance database as *installer* evidence (not a typo – it is collected in the same manner as file evidence, but then presented as installer evidence, based on conversions found in `InventorySettings.xml`). Like all installer evidence, it can be linked to application records, which can in turn be linked to license records; and when appropriately configured, the nightly license compliance calculations include an assessment of consumption for Oracle Fusion Middleware applications. By default, this presentation of the uploaded data within IT Asset Management is the only use made of the uploaded data about Oracle Fusion Middleware applications.
8. However, once the **Include Oracle Fusion Middleware** check box is selected, the relevant uploaded data and files are also incorporated into the `OracleGLASEvidence.zip` archive (normally compiled after the nightly inventory import and license compliance calculations), ready for submission when an audit is required. This separation of control between the collection of inventory and its inclusion in the GLAS archive means you can collect the inventory, use it to correctly configure your application and license records, and correct any licensing oversights before you then save your corrected state, in case you need to submit it to Oracle in future.

Agent-Based Collection of Oracle Inventory

This section provides details about Oracle discovery and inventory using a local instance of FlexNet Inventory Agent. In this context, 'local' means that FlexNet Inventory Agent is installed and executing on the target Oracle server.

Except as noted in following topics, it does not matter how the FlexNet Inventory Agent was deployed. In the terms defined in the *Gathering FlexNet Inventory* PDF, the concept of 'local agent-based inventory collection' covers:

- The Adopted case
- The Agent third-party deployment case.

Prerequisites for local agent-based inventory collection

The following must be in place for collection of Oracle inventory by a local copy of the FlexNet Inventory Agent installed on the target Oracle server:

1. The installed Oracle Database must be release 9i or later on UNIX-like platforms; and on Windows platforms, release 10g or later is preferred.
2. You have licensed the FlexNet Manager for Datacenters product (for details, see Appendix F: Features Enabled in FlexNet Manager for Datacenters).
3. You have deployed and configured one or more inventory beacon(s) in your network, such that at least one inventory beacon can access each of your target Oracle servers. For detailed information about inventory beacons, their deployment, and configuration, see the topics under *What Is an Inventory Beacon?* in the online help. You

initiate deployment of an inventory beacon by navigating to the **Beacons** page (**Data Collection > IT Assets Inventory Tasks > Beacons**).

4. If you intend to use the capabilities of IT Asset Management to discover Oracle servers on the network and automatically install the FlexNet Inventory Agent on them (a process called 'adoption'), you also require the following:
   - Your organizational site and subnet hierarchy is recorded through the **All Subnets** page (**Inventory > Network Discovery > All Subnets**) (see *Subnets* in the online help).
   - You have assigned the defined subnets to inventory beacon(s) through the **Beacons** page (**Data Collection > IT Assets Inventory Tasks > Beacons**) (see *Assigning a Subnet to a Beacon* in the online help).
5. You have set up the accounts required for operation, and (in the Adopted case) possibly separate accounts for adoption of the target Oracle servers. Details of the accounts for different platforms are in Credentials for Local Agent-Based Inventory.
6. You have deployed the FlexNet Inventory Agent to the target devices, ensuring that for UNIX-like systems, you have authorized or installed version 13.0.1 or later of the FlexNet Inventory Agent (earlier versions of the FlexNet Inventory Agent may fail to collect Oracle inventory on UNIX-like systems where permissions prevent global access to Oracle directories or files):
   - In the Agent third-party deployment case, following the guidelines starting from the *Agent third-party deployment: Implementation* topic within the *Gathering FlexNet Inventory* PDF
   - In the Adopted case, following the guidelines in the *Adopted: Implementation* topic in the same PDF.

   Tip: *When you know the names or network locations of the Oracle servers you want to adopt, you can declare a single rule in the web interface for IT Asset Management to target and adopt these known machines. If you are unsure where all your Oracle databases may have been installed, you can use a more generalized discovery rule to find and identify those devices, and subsequently adopt them.*

Credentials for Local Agent-Based Inventory

The required accounts and their privilege levels vary across different types of operating system, and they are also different for the initial deployment (in the Adopted case) and subsequent steady-state operations. (The accounts you may need for deployment in the Agent third-party deployment case are left to your own management, and not described here.)

For the **Windows** platform:

- **Adoption** requires an account that:
  - May be either a local account on the target Windows device, or a Windows domain account
  - Has full access to the Windows Service Control Manager on that target Windows device (specifically, the account must have the `SC_MANAGER_ALL_ACCESS` access right)
  - Is registered in the secure Password Manager on the appropriate inventory beacon before running the discovery task that includes the adoption action.

  Once the FlexNet Inventory Agent is correctly installed (through the adoption process), this level of privilege is no longer required.

- **Operation** (after the FlexNet Inventory Agent is correctly installed) requires an account on the target device that:
  - Is the `LocalSystem` account on the target device (but for Oracle Database version 9i, see the following note)
  - Has read-only access to the Windows Service Control Manager (this allows discovery of Oracle services)
  - Is a member of the Windows local security group `ora_dba` (in which context, the `LocalSystem` account is displayed as `NT AUTHORITY\SYSTEM`)
  - Uses local OS authentication to take inventory; which means that the `SQLNet.AUTHENTICATION_SERVICES` property *must* be set to `(NTS)` in the `sqlnet.ora` file located in the `%ORACLE_HOME%\network\admin` directory (and be aware that, conversely, *disabling* OS authentication for your Oracle Database *prevents* the locally installed FlexNet Inventory Agent from gathering inventory from Oracle database instances). By default, Oracle disables OS authentication on Windows platforms.

Note: *Operation with Oracle Database 9i is an exceptional case. To collect Oracle 9i inventory on Windows, you* ***must*** *run `ndtrack` as a* ***non-****`LocalSystem` user account. This is only possible if you trigger the tracker with a custom command line, using your preferred scheduling tool (such as Microsoft Task Scheduler). This makes the local agent cases (whether the Adopted case or the Agent third-party deployment case) rather unsuitable for taking inventory for version 9i. In both these cases, the tracker runs under policy as `LocalSystem` (in which case it reports a failure to collect inventory from an Oracle 9i database instance); and if you run it again with a custom command line as a different account, you get inventory results. The combination of positive results gained with negative failure notifications is bound to produce confusion! For these reasons, if you have instances of Oracle Database 9i, it is better to consider the lightweight FlexNet Inventory Scanner, for which you can more easily manage your own command lines (see FlexNet Inventory Scanner Collection of Oracle Inventory); or even the Core deployment approach (for which see the Gathering FlexNet Inventory PDF).*

For **UNIX**-like platforms:

- **Adoption** requires an account that:
  - Is local on the target device
  - Has `ssh` privileges
  - Can elevate to `root`-level privileges to complete the installation
  - Is registered in the secure Password Manager on the appropriate inventory beacon before the adoption process is run (and the additional details for elevation of account privileges with your preferred tool, such as `sudo` or `priv`, are also registered there).
- **Operation** (after the FlexNet Inventory Agent is correctly installed) requires an account on the target device that:
  - Must be `root` — otherwise local Oracle inventory collection is disabled
  - May impersonate other trusted accounts with lower privilege levels — as discussed in detail in the *Common: Child Processes on UNIX-Like Platforms* topic in the *Gathering FlexNet Inventory* PDF, along with coverage of the following preferences in the `config.ini` file that may affect the choice of account to impersonate:

Tip: *With neither of the following preferences specified, the default behavior is for FlexNet Inventory Agent to impersonate the account currently running the database instance, which is assumed to be a member of the `dba` group. This is the most straight-forward configuration, with no settings needed. If, instead, you intend to specify*

the `OracleInventoryUser` preference, it must be an exact match for any Oracle user name that:

- *Is also an operating system account*
- *Has OS authentication enabled (and as well, OS authentication, which defaults to enabled for UNIX-like platforms, must not have been disabled using the `SQLNet.AUTHENTICATION_SERVICES` property in the `sqlnet.ora` file located in the `%ORACLE_HOME%/network/admin` folder)*
- *Is a member of `oinstall` (or equivalent group, granting execute permissions for `sqlplus`)*
- *Is either a current member of the `dba` group on the UNIX host server; or has adequate permissions for inventory gathering (as outlined in this table).*

OracleInventoryAsSysdba	OracleInventoryUser	Impersonation	Connection/Notes
`True` (or omitted)	Configured	The account nominated in `OracleInventoryUser` is impersonated	Database connection is made as `sysdba` (and account must be a member of the `dba` group)
`True` (or omitted)	Not configured	The account running the database instance is impersonated	Database connection is made as `sysdba`
`False`	Configured	The account nominated in `OracleInventoryUser` is impersonated	Database connection is made as that same account (which in addition to the prerequisites above, must be configured with adequate read-only privileges as detailed in Appendix C: Oracle Tables and Views for Oracle Inventory Collection)
`False`	Not configured	None	Oracle inventory collection does not proceed

Note: *On a UNIX-like platform, the tracker attempts to use `setuid` to impersonate the appropriate account to gather Oracle inventory. If you are using eTrust Access Control on this server, by default it does not permit this impersonation, and inventory gathering fails. The fix is to change the configuration of eTrust to include `ndtrack` in the `LOGINAPPL` class. For more information, see the eTrust Access Control Administation Guide (https://supportcontent.ca.com/cadocs/0/g007711e.pdf).*

- The impersonated account may need an environmental variable set within its login profile. This applies only in the case where:
  1. A target Oracle database instance is running on a UNIX platform, and
  2. This account (operating system user) was the one used to start the database instance, and
  3. The start-up specified an `ORACLE_HOME` path which included a symbolic link.

  This use of the symbolic link can hide the database instance from inventory collection by the installed tracker

(ndtrack). Either of the following workarounds may be used to ensure that the local tracker can collect inventory from this database instance (and both workarounds may be implemented together without issue):

- The account running the database instance (say *OSUser4Oracle*) may set an environment variable within its login profile specifying the ORACLE_HOME path (including the symbolic link) which was used to start the database instance. To test this setting, the following command should display the correct ORACLE_HOME path:

```
su -OSUser4Oracle -c "echo \$ORACLE_HOME"
```

  Tip: *If this environment variable is set for any account on the database server, it is applied to all database instances started by the same account on this server. Any mismatch between the (non-empty) environment variable, and the actual path used to start any of these database instances, prevents the collection of database inventory from the mismatched instance by the locally-installed inventory component (ndtrack). Conversely, you can prevent the environment variable option being used for all accounts on the target Oracle server by setting the UserDefinedOracleHome preference (details of this preference are included in Gathering FlexNet Inventory.*

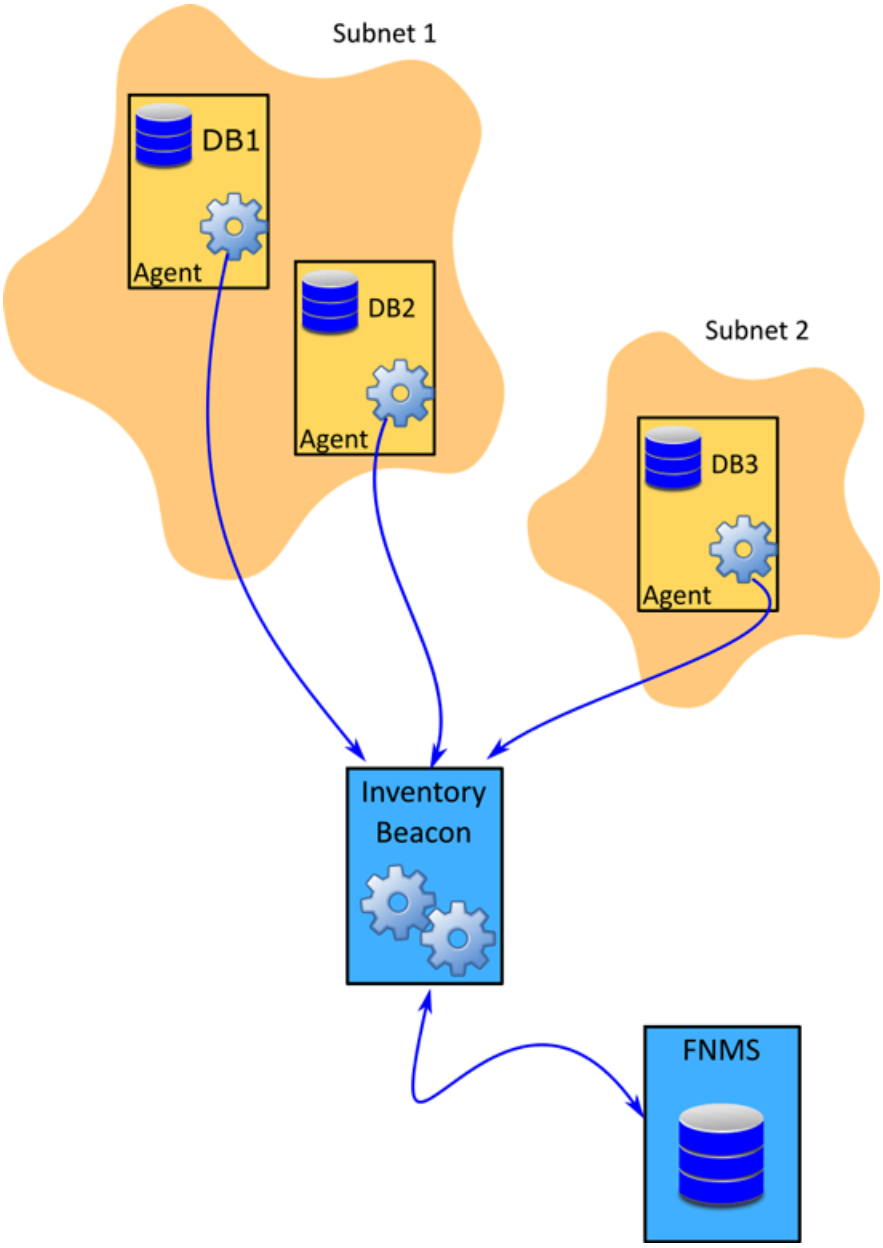
- You can ensure that the Oracle home specified in the /etc/oratab file represents the ORACLE_HOME path used to start the database instance.

How Agent-Based Collection of Oracle Inventory Works

In local agent-based inventory collection, IT Asset Management collects Oracle inventory through the FlexNet Inventory Agent installed on each Oracle server within your network. This method is simplest in operation, as each FlexNet Inventory Agent performs discovery and inventory for its respective Oracle server, and the various processes are managed automatically.

The installation of the FlexNet Inventory Agent is detailed in the *Gathering FlexNet Inventory* PDF for either the Adopted case or the Agent third-party deployment case. This topic summarizes operation after deployment.

The following diagram shows an example scenario for a single inventory beacon.



The diagram shows three database servers, two in Subnet1 and one in Subnet2. An instance of FlexNet Inventory Agent has been installed on each of these database servers. The inventory beacon has been assigned to cover both of these subnets and can connect to every Oracle server. All the prerequisites outlined in Agent-Based Collection of Oracle Inventory are satisfied.

The process for local agent-based Oracle inventory collection runs like this:

1. By default, every 15 minutes each inventory beacon checks for any updates its own policy, from which it derives policy to share with the devices it is managing. (To adjust this download schedule, see *Inventory Settings Page* > *Beacon Settings Section* in the online help.) This policy may:
   - Update the schedule for the installed FlexNet Inventory Agents if this has been changed (see *Inventory Settings Page* > *Inventory Agent Schedule Section* in the online help)
   - Update the `InventorySettings.xml` file, if this has been changed by a recent download of the Application

Recognition Library (this file includes specialized actions for gathering Oracle inventory, including the special inventory required for Oracle Fusion Middleware).

2. When the global schedule for inventory collection by the installed FlexNet Inventory Agents triggers, each FlexNet Inventory Agent performs discovery on its local device (recorded in a `.disco` file). This process is run by the tracker component of the FlexNet Inventory Agent (`ndtrack` executable). For Oracle, the discovery process tries to identify one or more paths for `$ORACLE_HOME`, using all of these platform-dependent methods in order (and combining the resulting dataset):

UNIX-like platforms — Oracle discovery	Windows platforms — Oracle discovery
1. The tracker scans the file system for any `oracle` executables. • This scan honors the global settings for file system scans in the **File Inventory** section of the **Inventory Settings** page (**Data Collection > IT Assets Inventory Tasks > Inventory Settings**), which may limit search paths or even disable file scanning entirely. • On success, this search returns the file path (`$ORACLE_HOME`) and the executable ID.	1. The tracker looks for a registry entry under `HKLM\SOFTWARE\Wow6432Node\Oracle\` (or, on 32-bit systems, `HKLM\SOFTWARE\Oracle\`). On success, this returns the `%ORACLE_HOME%` path.
2. The tracker looks for an `oratab` file (installed in either `/etc` or `/var/opt/oracle` during database installation). On success, this provides the `$ORACLE_HOME` path, the executable ID for the `oracle` executable, and the System ID (SID) for each database instance listed in the `oratab` file. (See also note below.)	2. The tracker interrogates the Windows Service Control Manager. On success, this provides the `%ORACLE_HOME%` path, the System ID (SID) for the database instance running in that process, and the name of the related Oracle listener.
3. The tracker examines process listings for matches to `ora_smon_*`. On success, this provides the `$ORACLE_HOME` path and/or the executable ID for the `oracle` executable, as well as the SID for the running database instance and the user account running it. (See also note below.)	(No step 3 for Windows.)
4. The tracker examines process listings for `tnslsnr`. On success, this provides the `$ORACLE_HOME` path, the executable ID for the `oracle` executable, the name of the related Oracle listener, and the user account running the listener.	(No step 4 for Windows.)

Note: *If a symbolic link was used in the `$ORACLE_HOME` path to start a particular Oracle database instance on a UNIX-like platform, this can 'hide' the database instance from inventory collection by the locally-installed tracker (`ndtrack` component). To ensure inventory collection from a database instance started with a symbolic link, use either (or both) of the following workarounds:*

- *You can ensure that the Oracle home specified in the `/etc/oratab` file represents the `ORACLE_HOME` path used to start the database instance.*
- *The account running the database instance (say `OSUser4Oracle`) may set an environment variable within its login profile specifying the `ORACLE_HOME` path (including the symbolic link) which was used to start the database instance. To test this setting, the following command should display the correct `ORACLE_HOME` path:*

```
su -OSUser4Oracle -c "echo \$ORACLE_HOME"
```

***Tip:** If this environment variable is set for any account on the database server, it is applied to all database instances started by the same account on this server. Any mismatch between the (non-empty) environment variable, and the actual path used to start any of these database instances, prevents the collection of database inventory from the mismatched instance by the locally-installed inventory component (`ndtrack`). Conversely, you can prevent the environment variable option being used for all accounts on the target Oracle server by setting the `UserDefinedOracleHome` preference (details of this preference are included in Gathering FlexNet Inventory.*

3. Next, the tracker gathers general hardware and software inventory (recorded in an `.ndi` file) from the local device. This is the standard inventory gathering that the tracker gathers on every device where it is running. If the appropriate preference is set, it also includes gathering evidence for Oracle Fusion Middleware applications (seeSpecial Handling of Oracle Fusion Middleware).
4. If the tracker discovered one or more Oracle database instance(s) on the device, and all the required settings and account privileges are in place, the tracker also gathers inventory from all accessible Oracle database instances.

***Tip:** The definition of "accessible" changed at version 12.4 of the tracker (released with IT Asset Management 2017 R3):*

- *For versions 12.3 and earlier, "accessible" excluded all Oracle database instances that are in standby mode*
- *For versions 12.4 and later, the locally-installed tracker also collects inventory from Oracle database instances that are in standby (that is, `MOUNTED` but not in `RUN` mode), such as the standby instance in an Active Data Guard configuration.*

The processes for gathering Oracle inventory on different types of platform are as follows.

- For UNIX-like platforms, the tracker impersonates an appropriate account, determined by the configuration of the preferences shown below.

***Note:** On a UNIX-like platform, the tracker attempts to use `setuid` to impersonate the appropriate account to gather Oracle inventory. If you are using eTrust Access Control on this server, by default it does not permit this impersonation, and inventory gathering fails. The fix is to change the configuration of eTrust to include `ndtrack` in the `LOGINAPPL` class. For more information, see the eTrust Access Control Administration Guide (https://supportcontent.ca.com/cadocs/0/g007711e.pdf).*

Each column in this table shows a set of conditions in the first three rows, followed by the resulting behavior in the last two rows. Once a valid user account is determined, the tracker invokes the Oracle-supplied `sqlplus` utility, giving it either of the command line parameters shown in the last row as appropriate:

UNIX-like platforms — Conditions	Option 1	Option 2	Option 3	Option 4	Option 5
If the executable runs as	`root`	`root`	`root`	`root`	Any other account
and `OracleInventoryAsSysdba` =	`True`	`True`	`False`	`False`	n.a.
and `OracleInventoryUser` =	Valid user	Not set	Valid user	Not set	n.a.
Then: Impersonated account is	That user (see ***Tip*** below)	Process owner	That user (see ***Tip*** below)	Not supported	n.a.
Command line parameters for `sqlplus`	`"/ as sysdba"`	`"/ as sysdba"`	`"/ "`	*No inventory*	*No inventory*

Tip: *If you intend to specify the `OracleInventoryUser` preference, it must be an exact match for any Oracle user name that:*

- *Is also an operating system account*
- *Has OS authentication enabled*
- *Is a member of `oinstall` (or equivalent group, granting execute permissions for `sqlplus`)*
- *Is either a current member of the `dba` group on the UNIX host server (when `OracleInventoryAsSysdba=True` or is unspecified); or has adequate permissions for inventory gathering (when `OracleInventoryAsSysdba=False`).*

- For Windows platforms, the tracker normally runs as `LocalSystem`, but may be run by another account that has administrator privileges (that is, is a member of the `Administrators` security group). In either case, the same account that is running the `ndtrack` executable is used to invoke the Oracle-supplied utility `sqlplus`, using the command line parameter `"/ as sysdba"`. This remains true even if the Oracle database instance is running as a service user, as is possible from Oracle Database 12c; so that binaries controlled by the service account are now executed as `LocalSystem` (or at least with administrator privileges). It is, of course, best practice to ensure that any service account running a database instance is well secured, so that the binaries it controls are protected.

5. Where Oracle inventory is collected, the tracker also executes scripts provided by Oracle Global Licensing and Advisory Services (GLAS) to gather software and hardware information about the servers where Oracle Database is installed. (These scripts, as amended from time to time, are downloaded to the tracker from the `InventorySettings.xml` file. They are used only for the preparation of an Oracle audit report, available to operators who have appropriate access rights in the **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**), with more details available in the help for that page.)
6. The tracker records the software and hardware results of the Oracle inventory gathering in a separate `.ndi` file.

Tip: *Notice that the installed FlexNet Inventory Agent responds to settings in the **Inventory Settings** page*

*(**Data Collection > IT Assets Inventory Tasks > Inventory Settings**), and does not follow the rules or schedules in **Discovery and Inventory Rules** page (**Data Collection > IT Assets Inventory Tasks > Discovery and Inventory Rules**), which are used by inventory beacons. This means that you do not need to set a particular rule to cause the installed FlexNet Inventory Agent to perform these discovery and inventory-gathering actions, as (subject to its local settings in the Windows registry, or `config.ini` file on UNIX-like platforms) it always does these, including checking for and collecting Oracle inventory on its local device.*

7. Immediately on completion of inventory gathering, the tracker uploads the `.disco` file and one or two compressed `.ndi` files to the appropriate inventory beacon. If some temporary network problem causes this upload to fail, there is a catch-up task to try again overnight (using the `ndupload` component).
8. The inventory beacon uploads all collected discovery and inventory information to the central application server (or, if it is a member of a hierarchy of inventory beacons, uploads the data to its parent in the hierarchy, and the upload is repeated until the data reaches the application server). Data is stored initially in the inventory database.
9. In due course, the inventory import (to the compliance database) and license reconciliation process runs (typically overnight, although an operator in a role granting the `Configure inventory data and reconcile` right can also trigger a full import and reconciliation). Progress is visible on the **System Tasks** page (**Data Collection > IT Assets Inventory Status > System Tasks**).
10. The **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**) lists the database inventory for all servers discovered and inventoried up to the time of the latest reconciliation calculation.

Troubleshooting Agent-Based Collection of Oracle Inventory

Before concluding that there is a problem, ensure that you have allowed sufficient time for:

- A download (typically weekly) of the content libraries (ARL, SKU library, PURLs) with your license in place for FlexNet Manager for Datacenters, and the subsequent distribution of the `InventorySettings.xml` file to target inventory devices
- Discovery and inventory processes on the target device (check the global schedule for the FlexNet Inventory Agent)
- File upload to the inventory beacon, and from there to the central application server (typically within several minutes of the data gathering being completed)
- Inventory import from the inventory database to the main compliance database (typically scheduled overnight)
- The license reconciliation calculations (typically scheduled overnight).

In summary, allow 48 hours after the download of the content libraries. After this process has had time to complete, it's time to continue troubleshooting if either of the following is true:

- The target Oracle server does not appear in the **All Discovered Devices** page (**Inventory > Network Discovery > All Discovered Devices**).
- A known Oracle database instance is missing from the **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**).

The troubleshooting process for Oracle inventory collected by the FlexNet Inventory Agent locally installed on the Oracle Database server breaks into the following main sections. Use these links to skip directly to your suspected

problem:

- Checking prerequisites (see below)
- Validating discovery (see Issues with Discovery)
- Checking inventory collection (see Issues with Inventory Gathering)
- Ensuring uploads (see Issues with Uploads)
- Requesting more assistance (see Requesting Further Assistance).

Checking prerequisites

Prerequisites include:

- A license for the FlexNet Manager for Datacenters product (in place before the regular library updates)
- Appropriate accounts are available on the target inventory device (and, if the device is first to be adopted, in the Password Manager on the appropriate inventory beacon)
- A successful installation of the FlexNet Inventory Agent on the target inventory device (Oracle server)
- A current version of `InventorySettings.xml` correctly located with the FlexNet Inventory Agent.

To validate prerequisites, choose from the following:

1. Validate that you have licensed the FlexNet Manager for Datacenters product:
   a. Go to the **IT Assets License** page (**Administration > IT Asset Management Settings > IT Assets License**).
   b. In the **Licensed products** section, scroll down to see the card for FlexNet Manager for Datacenters.

   Detailed Oracle inventory cannot be uploaded by the locally-installed FlexNet Inventory Agent without this license term being registered on the central application server well before the process commences. If it is missing, please contact your Flexera representative for assistance.
2. Ensure that the required accounts are set up correctly (see Credentials for Local Agent-Based Inventory). In particular, for adoption make sure that the appropriate account is configured correctly in the Password Manager on the appropriate inventory beacon.

   Remember: *On UNIX-like devices, the installed FlexNet Inventory Agent can collect Oracle inventory only when it is running as root.*
3. Validate that the FlexNet Inventory Agent is installed on the target device. Step through the following until success:
   a. Go to the **All Discovered Devices** page (**Inventory > Network Discovery > All Discovered Devices**), looking for a `Yes` displayed in the **Agent installed** column for the relevant device.
      - In the Adopted case, the value is updated as soon as the automated adoption process succeeds
      - In the Agent third-party deployment case, the value is retroactively updated after an inventory upload reveals the installer package for the FlexNet Inventory Agent (which may take a little longer, and obviously requires that at least one inventory upload from the target device has succeeded).

      For adoption, when the device is listed, you can also open its properties, and check the **Status** tab. Expand

the **Adoption** section to check details.

b. If this target inventory device was intended for automatic adoption (installation of FlexNet Inventory Agent), make sure that a discovery and inventory rule that includes adoption and includes the inventory device within its target has been scheduled and run.

For details, see the topic *Adopted: Implementation* in *Gathering FlexNet Inventory*.

c. If this target inventory device was scheduled for automatic adoption, check that the subnet containing the target device is assigned to the appropriate inventory beacon: go to the **All Subnets** page (**Inventory > Network Discovery > All Subnets**), validate the identity of the inventory beacon, and check that **Status** shows `Enabled`.

d. Check for the presence of the FlexNet Inventory Agent in the following paths on the target inventory device:

- On Windows, in `C:\Program Files (x86)\ManageSoft`
- On UNIX-like platforms, in `/opt/managesoft`.

e. Again in the adoption case, check the rule execution results. Go to the **System Tasks** page (**Data Collection > IT Assets Inventory Status > System Tasks**), identify your adoption rule, and click **See details**.

f. If this target inventory device was scheduled for automatic adoption, review the following log files on the device for evidence of any problems with adoption:

- **On Windows:** `$(TempDirectory)\adoption.log`
- **On UNIX-like devices:** `/var/tmp/flexera/log/ndinstlr.log` and `ndinstlrsh.log`.

g. Check the following installation log for any errors downloading policy during the install:

- **On Windows:** `C:\Windows\Temp\ManageSoft\installation.log`
- **On UNIX-like devices:** `/var/opt/managesoft/log/installation.log`.

h. If you are unable to resolve errors specific to adoption or installation, contact Flexera Support with the log files.

4. Check access to `InventorySettings.xml`.

a. Look in the tracker log for messages like this, indicating problems accessing the file:

```
Skipping Oracle database inventory.  Failed to find inventory settings with
the Oracle inventory rules.
Invalid inventory settings file found, recognition rules and scripts will
not be executed.
```

The log file is found:

- **On Windows:** `$(TempDirectory)\ManageSoft\tracker.log`
- **On UNIX-like devices:** `/var/opt/managesoft/log/tracker.log`

b. Check for `InventorySettings.xml` in its default location:

- **On Windows:** `$(CommonAppDataFolder)\ManageSoft Corp\ManageSoft\Tracker\`

`InventorySettings\InventorySettings.xml`

- **On UNIX-like devices:** `/var/opt/managesoft/tracker/inventorysettings/InventorySettings.xml`

Tip: *If the file is present, even though (as above) the logs suggest that the inventory component (`ndtrack`) is not finding the file, review the values for both the `CacheDirectory` and `PkgCacheDirectory` preferences under `[Registry]\ManageSoft\Launcher\CurrentVersion` on the inventory device. Blank values may indicate a corrupted installation of the FlexNet Inventory Agent, in which case you could try re-installation. The default values are:*

- *`CacheDirectory` = `$(CommonAppDataFolder)/Launcher/cache`*
- *`PkgCacheDirectory` = `$(CommonAppDataFolder)/Launcher/pkgcache`*

c. Check whether the launcher component (`ndlaunch`) has saved the file in a customized location by reviewing `[Registry]\ManageSoft\Tracker\CurrentVersion\InventorySettingsPath`.

Remember that, on UNIX-like platforms, the `[Registry]` details are saved in the `config.ini` file. While this value is very unlikely to have changed, use any new value you find here as an alternative location to check for the presence of the `InventorySettings.xml` file. For more details, see the *Gathering FlexNet Inventory* PDF.

d. If the `InventorySettings.xml` file is missing, check for possible problems with policy updates to the inventory device by reviewing the `policy.log` and `installation.log` files, found with the tracker log in the platform-specific paths shown above.

With all prerequisites validated, move on to troubleshooting the remaining aspects of Oracle discovery, inventory gathering, uploading, resolving into the inventory database, and importing into the compliance database.

Issues with Discovery

Having checked the prerequisites (see Troubleshooting Agent-Based Collection of Oracle Inventory) and shown that the FlexNet Inventory Agent is correctly installed and can access the `InventorySettings.xml` file, you already know that the discovery process for *initially identifying* the target inventory device (the Oracle server) has worked.

However, there is a second kind of discovery involved in gathering Oracle inventory. Recall that the installed FlexNet Inventory Agent is policy-driven, not rule-driven (meaning that *after* device discovery, it is no longer affected by discovery or inventory rules that you set in the web interface). It has no local record of previous discoveries; it has only its policy driving it to see what is available on this inventory device. In the case of an Oracle server as the target inventory device, the installed FlexNet Inventory Agent is driven to prepare three files for upload:

- The standard hardware and software inventory (`.ndi`) file that is prepared on every inventory device

Remember: *If the `PerformOracleFMWScan` preference is true (and `InventorySettings.xml` is available and up-to-date), this `.ndi` includes any evidence found for Oracle Fusion Middleware applications.*

- A separate discovery (`.disco`) file that records the (re-)discovery of the installed Oracle Database
- A separate inventory (`.ndi`) file exclusively for the inventory of all the Oracle database instances found on this device.

Symptoms

If this stage of discovery fails, the typical symptoms are:

- The inventory device (Oracle server) is visible in the **All Discovered Devices** page (**Inventory > Network Discovery > All Discovered Devices**)
- The Oracle instance(s) are missing from the **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**).

To troubleshoot Oracle discovery:

1. To quickly determine whether the problem is in *generating* or *uploading* the discovery file, check on the inventory device for a blocked upload of the `.disco` file:

   - **On Windows**: `$(CommonAppDataFolder)\ManageSoft Corp\ManageSoft\Common\Uploads\Discovery`
   - **On UNIX**: `/var/opt/managesoft/uploads/discovery/`.

   A discovery file (`.disco`) present in the appropriate folder means discovery is working, but that the upload is failing (since, after a successful upload, the file is removed from this folder). In this case, switch to debugging uploads (see Issues with Uploads). The *absence* of a file is therefore ambiguous: it may mean either that the file creation is failing, or that the first stage of upload has succeeded and that the problem is further upstream in the path from inventory device to application server.

2. If the folder is empty, you may check the following log file for evidence of a previous successful upload:

   - **On Windows:** `$(TempDirectory)\ManageSoft\tracker.log`
   - **On UNIX-like platforms:** `/var/opt/managesoft/log/tracker.log`

   (Because uploads are attempted immediately after inventory collection, they are logged by the tracker component. If this attempt fails and the catch-up upload is attempted later, this is logged by the uploader component.) Verify the following events within the expected time-frame (by default, the FlexNet Inventory Agent collects and uploads inventory on a daily basis):

   - The event `Uploading file <filename.disco>` indicates that the *generation* of the discovery file has succeeded (upload is normally attempted shortly after creation of the discovery file, when inventory has been collected).

     Important: *Take note of the URL given in this step, which should be of the form:*

     ```
     'http://192.53.15.139/ManageSoftRL/Inventories'
     ```

     The two trailing values are normally consistent for all inventory beacons:

     - *The `ManageSoftRL` folder is the web service for receiving uploads*
     - *`Inventories` indicates the temporary storage location on the inventory beacon.*

     The IP address allows you to identify the inventory beacon that FlexNet Inventory Agent chose for this upload (keeping in mind that FlexNet Inventory Agent chooses the optimum inventory beacon for each upload, and that it may select different servers at different times, depending on networking conditions). This log entry is one of the few ways to identify the inventory beacon used for any given upload.

- The events `Upload successful` and `File <filename.disco> removed from upload directory` indicates the successful upload of the discovery file.

These events indicate that discovery is working, the first stage of upload from the inventory device to an inventory beacon has also worked, and your problem may lie further upstream. Switch to debugging uploads (see Issues with Uploads).

3. If the folder is empty and `tracker.log` gives no indication of a recent successful upload, check further in `tracker.log` for evidence of successful discovery of database instances, validating results against your expectations for this Oracle server.

   Look for entries like the following:

   ```
   Started tracking Oracle database instances using inventory recognition rules
   Executing oracle inventory rules
    ===== Discovered Oracle database instances and listeners
   ```

   This should be followed by reports of separate instances, each commencing with `+--`, and reporting either a path to the `dbhome`, an instance name discovered as a running process, or a listener name. This list typically closes with:

   ```
   Oracle database instances were discovered for inventory.
   ```

   Tip: *If this section is missing completely, the most common problem is the absence of prerequisites, including a license for FlexNet Manager for Datacenters, and the `InventorySettings.xml` file. If you have not already validated these prerequisites, refer back to Troubleshooting Agent-Based Collection of Oracle Inventory for details.*

   If the log file reports any issues with discovery, attempt to remedy those before continuing.

4. If logged results were incomplete, or you wish to examine the results intended for upload in the `.disco` file, change to the installation directory for FlexNet Inventory Agent, and use the following platform-dependent command line to generate a new discovery file:

   - **For Windows:** `ndtrack -t machine -o Upload=False`
   - **For UNIX-like devices:** `bin/sh ./ndtrack -t machine -o Upload=False`

   The `Upload=False` option prevents upload of the generated `.disco` file by the FlexNet Inventory Agent, so that it is saved in the path described in step 1. Inspect the discovery folder again:

   - If a `.disco` file is present now, continue with step 5.
   - If there is no `.disco` file created, skip to step 6.

5. Open the `.disco` file in a text editor, and check for any evidence of Oracle services.

   - If you find such evidence, the discovery process is probably working correctly. Look elsewhere for your problem, such as in uploads (see Issues with Uploads).
   - If there is no such evidence:

     a. Again look for evidence of Oracle discovery in the `tracker.log` file, as described in step 3.

b. If you can, remedy any problems highlighted in this log file.

c. If you are unable to diagnose further, generate a trace file (as in the next step) before you seek further assistance (see Requesting Further Assistance).

6. As no `.disco` file was created, and the `tracker.log` file (see step 3) does not give enough information, increase the level of tracing before running the inventory (and related discovery) command again:

a. Switch to the installation directory for FlexNet Inventory Agent.

b. Open the `etcp.trace` file from this folder in a flat text editor.

c. Uncomment (remove # from the start of) the following lines:

```
+Inventory/Oracle
+Inventory/Oracle/SDK
+Inventory/Oracle/Query
+Inventory/Oracle/Query/Substitution
+Inventory/Oracle/Query/Execution
+Inventory/Oracle/Listener
+Inventory/Oracle/Listener/Detail
+Inventory/Tracker/Environment
```

d. Also uncomment one of the lines that declare the file path and file name (pattern) for the output trace file, optionally customizing the path:

On Microsoft Windows:

```
filename=C:\ManageSoft%p_%d_%t_%u.log # filename pattern with everything!
```

On UNIX-like platforms:

```
filename=/tmp/log/ManageSoft%p_%d_%t_%u.log # filename pattern with
everything!
```

Tip: *Keep at least the %u option, so that each run gives the trace file a sequential number. With no variables in the file name, each run appends data to the same file, which can quickly become unmanageably large.*

e. Save the `etcp.trace` file, and re-run the `ndtrack` command (see step 4).

f. Examine the output `.log` file(s) in the path you declared.

g. If you can, correct any reported problems. If not, include your trace files with your log files, and seek further assistance (see Requesting Further Assistance).

Issues with Inventory Gathering

It is possible for the installed FlexNet Inventory Agent to succeed at its local discovery task (identifying the presence of Oracle Database on the inventory device), and yet to fail to deliver inventory for the associated Oracle instances on the inventory device (the Oracle server).

Be sure that you have validated all the prerequisites (see Troubleshooting Agent-Based Collection of Oracle

Inventory). In particular, remember that Oracle inventory collection on UNIX-like platforms requires that the FlexNet Inventory Agent must be running as `root`.

Symptoms

- Discovery of the Oracle server itself has succeeded, and it is therefore visible in the **All Discovered Devices** page (**Inventory > Network Discovery > All Discovered Devices**); but
- Records of Oracle database instances known to be on that server are missing from the **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**). The issue may be:
  - Inventory gathering (this topic)
  - Inventory uploads (see Issues with Uploads).

To troubleshoot Oracle inventory collection:

1. In the web interface, go to the **All Discovered Devices** page (**Inventory > Network Discovery > All Discovered Devices**) to open the properties for the target inventory device (the Oracle server), and check the **Status** tab for issues with local Oracle inventory collection.

   Providing that status uploads are running, any errors while taking inventory of an Oracle database instance are displayed here. These errors may guide discussions with your Oracle database administrator; or direct you to step 5 to check for connection problems. If the database instance isn't displayed, continue with next steps.

2. On a UNIX-like target inventory device, check whether eTrust Access Control is in use, and if so whether it has been configured to include `ndtrack` in the `LOGINAPPL` class.

   On a UNIX-like platform, the tracker attempts to use `setuid` to impersonate an appropriate user to gather Oracle inventory. By default, eTrust Access Control does not permit this impersonation, and inventory gathering fails. The fix is to change the configuration of eTrust as described. For more information, see the *eTrust Access Control Administration Guide* (https://supportcontent.ca.com/cadocs/0/g007711e.pdf).

3. On the target inventory device (of any platform type), open the `tracker.log` file in a text editor, and check for the indications of success:

   - **On Windows:** `$(TempDirectory)\ManageSoft\tracker.log`
   - **On UNIX-like platforms:** `/var/opt/managesoft/log/tracker.log`

   Look for the following events within the expected time-frame (by default, the FlexNet Inventory Agent collects and uploads inventory on a daily basis):

   - `Starting oracle inventory` and `Finished generating inventory` indicate the start and end of the Oracle inventory collection process.
   - `Uploading file <`*`filename`*` (Oracle).ndi.gz>` indicates the upload of the compressed Oracle inventory file has commenced.
   - `File <`*`filename`*` (Oracle).ndi.gz> removed from upload directory` indicates the successful upload of the inventory file.

   If these success points are all logged (or if only the last one is missing), the issue may be in *uploading* inventory files (see Issues with Uploads). If, instead, it is inventory *collection* that is failing, continue.

Tip: *The `tracker.log` file also includes details of Oracle options that are found on each database instance. For example, a database instance may produce a log entry like this:*

```
[dateTime (G, 0)] {25332} |   AdvancedCompression_ByTableCompression:
Installed = 1, Used = 0
```

This entry clarifies that the Advanced Compression option is available for the database instance, but is not (yet) known to be in use on this instance. In fact, this is one of several Oracle queries that test for use of the Advanced Compression option, such as:

- *`AdvancedCompression_ByTableCompression`*
- *`AdvancedCompression_BySecureFilesCompressionAndDeDuplication`*
- *`AdvancedCompression_ByLogArchiveCompression`*
- *`AdvancedCompression_ByFeatureDataGuard`.*

If `Used` gives a non-zero result in any of these queries, the option is considered used and licensable on the database instance. This level of detail allows you to dig deeply and identify the particular Oracle query that defines the usage.
Furthermore, if it happens that the current database instance is also the repository for an installation of Oracle Enterprise Manager, the log for each query is extended with the Oracle server name and the database instance name for each managed database instance as well:

```
[dateTime (G, 0)] {25332} |   LifecycleManagementPack_ByOEMMgmtTargets:
[dateTime (G, 0)] {25332} |      server1.example.com/orcl11g2: Installed = 1,
Used = 1
[dateTime (G, 0)] {25332} |      server1.example.com/orcl11g: Installed = 1,
Used = 1
[dateTime (G, 0)] {25332} |      server2.example.com/orcloem: Installed = 1,
Used = 1
[dateTime (G, 0)] {25332} |      server3.example.com/orcl12db: Installed = 1,
Used = 1
[dateTime (G, 0)] {25332} |      server1.example.com/orclstd: Installed = 1,
Used = 1
```

4. Ensure that the collection of inventory is correctly scheduled, using these platform-dependent commands:
   - **On Windows:** `ndschedag -t machine`
   - **On UNIX-like devices:** `/opt/managesoft/bin/ndschedag -e`

   Make sure that the `Generate Inventory` task is visible, and that the next run time is populated appropriately. If the schedule is empty, review `installation.log` to look for errors in downloading device policy.

5. Validate that the appropriate account can use `sqlplus` to connect to your test database instance:
   a. Identify the account relevant to this database instance.

      For example, on UNIX-like platforms:

      - If you are using the `OracleInventoryUser` preference on this inventory device (server), use that account

- If not, identify the account that started the database instance. For example, the following command line lists all database instances on this server in the format *oraclesid*_smon_*ORACLESID* (the placeholders are replaced with the instance's system identifier in lower- and upper-case), with each row beginning with the relevant account name for that instance:

```
ps -ef | grep smon
```

b. Validate whether the OracleInventoryAsSysdba preference (in the config.ini file for the local FlexNet Inventory Agent on UNIX) has been set to false.

This determines the command line parameters you will need when testing:

```
sqlplus "/ " # when OracleInventoryAsSysdba=false
sqlplus "/ as sysdba" # in all other cases
```

c. Still using UNIX as our example, log in as root (the FlexNet Inventory Agent runs with this level of privilege to collect Oracle Database inventory, and using this saves you providing a password with the su command), and then switch to the identified account to run sqlplus, ensuring that the environment variables are set for the Oracle home directory and the system identifier (SID) for the current database instance:

```
su accountUnderTest
export ORACLE_HOME=OracleHomeDirectory
export ORACLE_SID=OracleSID
cd $ORACLE_HOME/bin
./sqlplus "/ as sysdba" # or modified as noted above
```

If the connection is successful, sqlplus prints a Connected to: summary for the current database instance. Validate that the permissions are adequate and the database instance is fully operational with a test query such as:

```
SELECT VERSION FROM V$INSTANCE;
```

A response of the full version ID for the Oracle Database confirms that inventory collection can proceed. If connection was successful but inventory collection is still not succeeding, skip ahead to step 8. If connection did not succeed, continue here.

d. On all operating systems, check whether the preference OracleInventoryAsSysdba is set to true or is not specified (when the default is also true). If so, make sure that the account running the database instance is listed in the ORADBA group (the local ora_dba security group on Windows and the dba group on UNIX-like platforms).

By default this group exists, and the account running the database instance is listed in it; but it is possible that the group has been removed on [some of] your Oracle servers. For full details about alternative accounts and all account requirements, see the OracleInventoryUser and OracleInventoryAsSysdba preferences in *Gathering FlexNet Inventory*. If the account running the database instance is missing from the group, add it, and re-run the connection test described in step 5.

Otherwise, with the account being present but the connection test still failing, ask your Oracle DBA for assistance to resolve the problem of the account's access to sqlplus.

6. On Windows platforms, check that the OracleService*databaseName* is running.

For example:

```
C:\Users\Administrator\tasklist /svc | find "oracle"
```

Note that although the Oracle Database must attach itself to the Windows service revealed by this command, it is possible for the service to be running while the Oracle Database itself is not started. Therefore, if you found that you had to restart the Windows service, also recheck that Oracle Database (and its instances) are running. The appropriate `sqlplus` command shown in the previous step validates this.

7. If you have not already done so as part of your troubleshooting of discovery, set up for a trace file, and re-run the inventory command to check for any reported issues.

   More details are in Issues with Discovery, but the summary is:

   a. Remove the leading # from the following lines of `etcp.trace`, located in the installation folder for FlexNet Inventory Agent, and also edit the file path/name for the output log file, saving the changed file with the same name in the same location:

      ```
      +Inventory/Oracle
      +Inventory/Oracle/SDK
      +Inventory/Oracle/Query
      +Inventory/Oracle/Query/Substitution
      +Inventory/Oracle/Query/Execution
      +Inventory/Oracle/Listener
      +Inventory/Oracle/Listener/Detail
      +Inventory/Tracker/Environment
      ```

   b. Re-run the inventory command (`ndtrack -t machine`).

   c. Examine the log file you specified.

      A line like the following indicates the start of the Oracle inventory logic:

      ```
      31.8850, pid 5835 (Inventory/Tracker/Oracle/Listener): Oracle inventory is
      enabled.
      Performing Oracle inventory.
      ```

   The remaining checks are for less common corner cases that have been detected.

8. On UNIX-like systems, check whether the *`OracleHomeDirectory`*`/bin/oracle` executable may have been upgraded or replaced without restarting the database instance. If you cannot determine whether the executable has been changed, restart the database instance to test whether this clears the problem.

   Remember that the tracker examines process listings, and if the original process was not restarted, it is now referencing a file that is no longer available, because it has been updated or replaced. Of course, if the executable *has* been changed, a process restart is essential, if for no other reason than to allow the database instance to utilize the updated code.

9. Also on UNIX-like platforms, verify which version of the `ndtrack` executable is collecting inventory, and if necessary upgrade the FlexNet Inventory Agent.

   Versions of the tracker earlier than 13.0.1 may fail to gather Oracle inventory on servers where permissions have been tightened to prevent global read access to Oracle directories or files. Choose any of the following methods to verify the version of `ndtrack`:

a. If you already have a discovered device record for the server, navigate to its discovered device properties, select the **Inventory evidence** tab and the **Software** sub-tab. Find FlexNet Inventory Agent in the listing, which includes its version.

> ***Tip:*** *For UNIX-like platforms, for legacy reasons, the FlexNet Inventory Agent is listed as `ManageSoft for platform Managed Devices`.*

b. On the target inventory device (the Oracle server), read `/var/opt/managesoft/etc/config.ini` in a text editor, and validate the version of the inventory agent saved in the read-only `ETCPVersion` preference.

> ***Caution:*** *Never edit this value manually.*

c. Examine the log file for `ndtrack`, as this reports the version of the agent in use.

The default paths for logging depend on the inventory collection method:

- For the locally-installed FlexNet Inventory Agent, see `/var/opt/managesoft/log/tracker.log` (or check for a custom value saved as `ManageSoft\Tracker\CurrentVersion\LogFile` in the `/var/opt/managesoft/etc/config.ini` file)
- For the lightweight FlexNet Inventory Scanner (presented as `ndtrack.sh` on UNIX-like systems), and also for zero footprint inventory collection, see `/var/tmp/flexera/log/tracker.log`

> ***Tip:*** *If `ndtrack.sh` is executed by a non-root account `userName`, the log defaults to: `/var/tmp/flexera.userName/log/tracker.log`.*

d. On the target inventory device, review the latest `.ndi` file.

Between inventory collections, the most recent *general* inventory file is saved on the inventory device in `/var/opt/managesoft/tracker/inventories`, and the version appears at the top of the file as the `Tracker` attribute of the `Inventory` XML element.

> ***Tip:*** *The `.ndi` file for Oracle inventory is not preserved uncompressed in the same way. The Oracle inventory file is immediately compressed and an upload is attempted. But for the purpose of identifying the release of FlexNet Inventory Agent in use on this device, any `.ndi` file will suffice,*

e. If you are using `ndtrack.sh` (the lightweight inventory scanner), this command line reports the version:

```
cd directoryContainingScanner
./ndtrack.sh --version
```

If the version of `ndtrack` on this device is 13.0.0 or earlier, upgrade to a later version.

If everything appears to check out correctly but problems persist with Oracle inventory collection, please collect all details and log files and seek further assistance (see Requesting Further Assistance).

Issues with Uploads

Oracle inventory collected by the FlexNet Inventory Agent locally installed on the target inventory device (the Oracle server) must now successfully navigate a multi-part process:

- The compressed `.ndi.gz` files (and for Oracle inventory, an uncompressed `.disco` file) are uploaded from the target inventory device to an inventory beacon of its choosing. Keep in mind that this is attempted first by the tracker (`ndtrack`, the inventory component) and, if a transient problem causes a failure, it is retried overnight by the uploader (`ndupload`), each of which has its own log file.
- If there is a hierarchy of inventory beacons, the first must upload to its parent, and so on.
- The 'topmost' inventory beacon uploads to the central application server, where the receiving web service attempts to 'resolve' the uploaded files into the inventory database.
- At the next full inventory import (typically scheduled overnight, immediately preceding the license compliance calculations), the latest inventory is imported from the inventory database into the compliance database.

Symptoms

Problems with the upload process may result in these shortcomings visible in the web interface for IT Asset Management:

- Device records for your Oracle servers missing from the **All Discovered Devices** page (**Inventory > Network Discovery > All Discovered Devices**)
- Hardware inventory records (or, in the case of a problem emerging later that disrupts ongoing operations, *recent* inventory records) missing from the **All Inventory** page (**Inventory > Inventory > All Inventory**)
- Known database instances (that you have verified are still in existence and running/runnable) missing from the **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**).

Keep in mind that inventory devices that disappear from imported inventory are assumed to have been decommissioned, and after a period may be automatically removed from the compliance database and therefore the web interface. So to start with, be sure that the Oracle server and its associated database instances that you thought were in place are still operational — otherwise, their absence from the record is an accurate depiction of the conditions in your computing estate.

To troubleshoot uploads of Oracle inventory:

1. On the target inventory device (the Oracle server), quickly identify whether the first stage of upload to an inventory beacon is failing.
    a. Check the upload of `.disco` files by seeing whether the staging folder is empty (it is cleared after a successful upload):
        - On Windows, check `$(CommonAppDataFolder)\ManageSoft Corp\ManageSoft\Common\Uploads\Discovery`
        - On UNIX-like platforms, check `/var/opt/managesoft/uploads/discovery/`

        The presence of a `.disco` file here (more than a few minutes after inventory collection has run) means that upload is failing, at least for this file type (and typically then, for all file types).

b. Check the upload of `.ndi.gz` files from their staging folder:

- On Windows, check `$(CommonAppDataFolder)\ManageSoft Corp\ManageSoft\Common\Uploads\Inventories`
- On UNIX-like platforms, check `/var/opt/managesoft/uploads/Inventories`

The presence of a `.ndi.gz` file here also means that the first stage of upload is failing. For failing uploads, continue with the next step, looking for problems reported in the log file(s). On the other hand, if both the `Discovery` and `Inventories` folders are empty, this indicates that the first stage of upload (to a selected inventory beacon) has succeeded; and if there is an upload problem, it may be further upstream. You also need to review the log files on this device (details in the next step), but in this case looking to identify the inventory beacon selected for the inventory upload, so that you can follow the chain.

2. Review the appropriate log files for problems with previous uploads, or to identify the inventory beacon that was selected for the upload.

Remember that the FlexNet Inventory Agent has two ways to attempt uploads:

- The inventory tracker component (`ndtrack`) attempts an upload immediately after completing inventory collection. Any issues with this attempt are logged in `tracker.log`.
- To recover from transient issues, the uploader component attempts a catch-up overnight, and any issues here are logged in `upload.log`.

Both these log files are saved in the same platform-specific location:

- **On Windows**: `$(TempDirectory)\ManageSoft\` (provided that the tracker is running as `LocalSystem`, this expands to `C:\Windows\Temp\ManageSoft\`)
- **On UNIX-like platforms**: `/var/opt/managesoft/log/`.

a. Review at least the first and possibly both of these log files.

Look for the following markers of success (examples shown for the Oracle inventory file, but you may also find results for the general inventory, where the file name starts with `system`:

- `Uploading file '`*`deviceName`* `at `*`dateTime`* `(Oracle).ndi.gz' to 'http://192.53.15.139/ManageSoftRL/Inventories'`

Important: *Take note of the URL, which is one of the few ways to identify the inventory beacon used for any given upload (keeping in mind that FlexNet Inventory Agent chooses the optimum inventory beacon for each upload, and that it may select different servers at different times, depending on networking conditions). Identify the inventory beacon server from this IP address, as you may need to troubleshoot that server next.*

- `Upload successful`
  `File '`*`deviceName`* `at `*`dateTime`* `(Oracle).ndi.gz' removed from upload directory`

These indicate that the error is not local on this device, so that you need to troubleshoot further along the upload chain.

If the log files indicate any networking or other problems for uploading, try to remedy these; and continue with the next substep. On the other hand, if this stage has been successful, move to the next server in the upload chain (skip to step 3).

b. After making any corrections, re-run the tracker to collect updated inventory (and database discovery data), and monitor the staging folders (identified in step 1) for the creation and (after upload) removal of the files there.

The platform-specific commands are:

- **For Windows:** `ndtrack -t machine` (when run from the install directory)
- **For UNIX-like devices:** `/opt/managesoft/bin/ndtrack -t machine`

When inventory gathering is complete, repeat the review of logs for evidence of upload problems.

3. Switch to the inventory beacon identified in step 2a, and check that files aren't backed up in the incoming inventory folder *%CommonAppData%*`\Flexera Software\Incoming\Inventories`.

An empty folder is a good sign, as each inventory beacon is scheduled to check for and try uploads from this folder to its parent device every minute throughout the day.

Tip: *In the upload to an inventory beacon, your tenant ID is prepended to the file name, so that the format of the file name now becomes* `[tenantID]_deviceName at dateTime (Oracle).ndi.gz`.

4. Review `C:\Windows\Temp\ManageSoft\uploader.log` on this inventory beacon for details of any networking issues, and note the URL of the next server in the upload chain. If this is another inventory beacon, repeat these reviews there, and so on until you exhaust the hierarchy of inventory beacons.
5. If you are unable to resolve upload problems, collect your logs together to seek further help (see Requesting Further Assistance).

Requesting Further Assistance

This summary covers the preferred materials to submit with a support request, if you have been unable to solve an issue with any of the following processes:

- Adoption of an Oracle server (that is, the automatic installation of FlexNet Inventory Agent on that target inventory device)
- Discovery of Oracle Database, and database instances, by the locally-installed FlexNet Inventory Agent
- Collection of inventory from any database instance(s), including pluggable databases (from Oracle Database 12c or later)
- Oracle inventory uploads.

To prepare a support request:

1. For problems with adoption of the target Oracle server, prepare a zip archive containing:
   - A document showing:
     - The name of the target Oracle server, and its IP address

    - The name of the inventory beacon intended to undertake the adoption
  - The rule execution logs downloaded from the **System Tasks** page (**Data Collection > IT Assets Inventory Status > System Tasks**)
  - The adoption log files collected from the target inventory device:
    - For Windows: `$(TempDirectory)\adoption.log`
    - For UNIX-like devices: `/var/tmp/flexera/log/ndinstlr.log` and `ndinstlrsh.log`.

2. For other issues, prepare a zip archive containing:
  - A document showing:
    - The name of the target Oracle server, and its IP address
    - The name of the database instance(s) that should be inventoried, but are not
    - A paste of the console output resulting from a manual test with `sqlplus`
  - The installation log for policies on the target inventory device:
    - For Windows: `C:\Windows\Temp\ManageSoft\installation.log`
    - For UNIX-like devices: `/var/opt/managesoft/log/installation.log`
  - The tracker log for the activities of FlexNet Inventory Agent:
    - For Windows: `C:\Windows\Temp\ManageSoft\tracker.log`
    - For UNIX-like devices: `/var/opt/managesoft/log/tracker.log`
  - The log file output from your trace run, collected from the path you specified in the `etcp.trace` file, where the defaults are:
    - For Windows: `C:\Windows\Temp\ManageSoft\tracker.log`
    - For UNIX-like devices: `/var/opt/managesoft/log/tracker.log`
  - A copy of your current `InventorySettings.xml` file from the target inventory device:
    - For Windows: `$(CommonAppDataFolder)\ManageSoft Corp\ManageSoft\Tracker\InventorySettings\InventorySettings.xml`
    - For UNIX-like devices: `/var/opt/managesoft/tracker/inventorysettings/InventorySettings.xml`

3. Ask your registered support contact (a designated person within your enterprise who has access rights and login details) to open a new support case at https://community.flexera.com/t5/forums/postpage/board-id/@support, including a clear description of the issue. Once the case has been saved, your support contact can use the **Upload** button (in the **Attachments** section at the bottom) to attach your prepared zip archive of evidence.

FlexNet Inventory Scanner Collection of Oracle Inventory

Rather than deploying the complete FlexNet Inventory Agent, in this approach you choose to deploy the lightweight FlexNet Inventory Scanner, a reduced code set that is detailed in Gathering FlexNet Inventory. Operational details and results are very similar to the agent-based cases described earlier, but the differences are included in the topics in this section.

Prerequisites for FlexNet Inventory Scanner inventory collection

The following must be in place for FlexNet Inventory Scanner collection of Oracle inventory by an inventory beacon:

1. The installed Oracle Database must be release 9i or later on UNIX-like platforms; and on Windows platforms, release 10g or later is preferred.
2. You have licensed the FlexNet Manager for Datacenters product (for details, see Appendix F: Features Enabled in FlexNet Manager for Datacenters).
3. Each installed copy of the FlexNet Inventory Scanner must be able to access an inventory beacon to upload its `.disco` and `.net.gz` files. The FlexNet Inventory Scanner requires this path to the inventory beacon defined in its `UploadLocation` preference (normally in the command line that invokes the FlexNet Inventory Scanner). This may be one of your existing, fully-configured inventory beacons; or if it is one specifically for use by the FlexNet Inventory Scanner alone, it does not require assigned subnets (because in this case, you are managing the FlexNet Inventory Scanner, rather than letting the inventory beacon manage it).
4. You have deployed the FlexNet Inventory Scanner, following the processes in these topics from the *Gathering FlexNet Inventory* PDF:
   - *FlexNet Inventory Scanner: Implementation on Windows*
   - *FlexNet Inventory Scanner: Implementation on UNIX-like Platforms*.

   ***Tip:** For gathering Oracle inventory with the FlexNet Inventory Scanner, it is critical that you have also deployed `InventorySettings.xml` into the same folder as the FlexNet Inventory Scanner, as described in the above topics.*
5. You have configured your preferred scheduling system to invoke the FlexNet Inventory Scanner on your desired schedule, with the command-line settings needed to trigger upload of the collected inventory and Oracle discovery files to an appropriate inventory beacon. (For details, see the *Gathering FlexNet Inventory* PDF.)

Credentials for FlexNet Inventory Scanner Inventory

When using FlexNet Inventory Scanner as your inventory-gathering tool, you configure your preferred scheduling tool (such as Microsoft Task Scheduler on Windows, or `cron` on UNIX-like platforms) to invoke FlexNet Inventory Scanner with the appropriate tracker command line parameters (documented in *ndtrack Command Line* in the *Gathering FlexNet Inventory* PDF). Since this invocation is local on the target inventory device, there is no requirement to register any credentials in the Password Manager on any inventory beacon.

The credentials required on the target device vary across platforms.

On Microsoft Windows target devices

For the account to invoke FlexNet Inventory Scanner:

- The `LocalSystem` account is recommended.
- A non-`LocalSystem` account with administrator privileges is also acceptable. (This means that the account is a member of the Administrators security group in Active Directory.)

Note: *On Microsoft Windows, the tracker does not prevent invocation by an account that has lesser privileges; but you would then need to ensure that such an account had all the required access rights for the kinds of inventory you expected to gather on a target device. Since this is highly dependent on your environment, this approach is unsupported.*

- The chosen account must have read-only access to the Windows Service Control Manager (this allows discovery of Oracle services).
- It must be a member of the Windows local security group `ora_dba` (in which context, the `LocalSystem` account is displayed as `NT AUTHORITY\SYSTEM`).
- This account uses local OS authentication to take inventory; which means that the `SQLNet.AUTHENTICATION_SERVICES` property *must* be set to `(NTS)` in the `sqlnet.ora` file located in the `%ORACLE_HOME%\network\admin` directory (and be aware that, conversely, *disabling* OS authentication for your Oracle Database *prevents* FlexNet Inventory Scanner from gathering inventory from Oracle database instances). By default, Oracle disables OS authentication on Windows platforms.

Operation with Oracle Database 9i is an exceptional case. To collect Oracle 9i inventory on Windows, you **must** run `ndtrack` as a **non-**`LocalSystem` user account. To do this, ensure that the account has administrator privileges on the target device (that is, is included in the Administrators security group) so that it can collect sufficient hardware inventory information; and then set up your Windows scheduled task to include the following:

- In the **General** tab of the **Task Scheduler**, set the user account name in the field for **When running the task, use the following user account**.
- In the **Actions** tab, set the action to `Start a program`, and the **Program/script:** value to

  ```
  ndtrack.exe -t machine
  ```

  The `-t machine` option is mandatory in this scenario (in contrast, it is the default when the tracker runs as `LocalSystem`).

On UNIX-like target devices

FlexNet Inventory Scanner (`ndtrack.sh`):

- Must run as `root` to collect Oracle inventory. If it is run under any other account on UNIX-like systems, the gathering of Oracle inventory is blocked.

Tip: *As always, it makes no difference whether you invoke FlexNet Inventory Scanner (`ndtrack.sh`) directly as*

root, or whether you run as another account and use sudo (or similar) to elevate to root before invoking FlexNet Inventory Scanner.

- May impersonate other trusted accounts with lower privilege levels — as discussed in detail in the *Common: Child Processes on UNIX-Like Platforms* topic in the *Gathering FlexNet Inventory* PDF, along with coverage of the following preferences in the co-located ndtrack.ini file that affect the choice of account to impersonate:

***Tip:** With neither of the following preferences specified, the default behavior is for the FlexNet Inventory Scanner to impersonate the account currently running the database instance, which is assumed to be a member of the dba group. This is the most straight-forward configuration, with no settings needed. If, instead, you intend to specify the OracleInventoryUser preference, it must be an exact match for any Oracle user name that:*

  - *Is also an operating system account*
  - *Has OS authentication enabled (and as well, OS authentication, which defaults to enabled for UNIX-like platforms, must not have been disabled using the SQLNet.AUTHENTICATION_SERVICES property in the sqlnet.ora file located in the %ORACLE_HOME%/network/admin folder)*
  - *Is a member of oinstall (or equivalent group, granting execute permissions for sqlplus)*
  - *Is either a current member of the dba group on the UNIX host server; or has adequate permissions for inventory gathering (as outlined in this table).*

OracleInventoryAsSysdba	OracleInventoryUser	Impersonation	Connection/Notes
True (or omitted)	Configured	The account nominated in OracleInventoryUser is impersonated	Database connection is made as sysdba (and account must be a member of the dba group)
True (or omitted)	Not configured	The account running the database instance is impersonated	Database connection is made as sysdba
False	Configured	The account nominated in OracleInventoryUser is impersonated	Database connection is made as that same account (which in addition to the prerequisites above, must be configured with adequate read-only privileges as detailed in Appendix C: Oracle Tables and Views for Oracle Inventory Collection)
False	Not configured	None	Oracle inventory collection does not proceed

- The impersonated account may need an environmental variable set within its login profile. This applies only in the case where:
  1. A target Oracle database instance is running on a UNIX platform, and
  2. This account (operating system user) was the one used to start the database instance, and

3. The start-up specified an ORACLE_HOME path which included a symbolic link.

This use of the symbolic link can hide the database instance from inventory collection by the installed tracker (ndtrack). Either of the following workarounds may be used to ensure that the local tracker can collect inventory from this database instance (and both workarounds may be implemented together without issue):

- The account running the database instance (say *OSUser4Oracle*) may set an environment variable within its login profile specifying the ORACLE_HOME path (including the symbolic link) which was used to start the database instance. To test this setting, the following command should display the correct ORACLE_HOME path:

```
su -OSUser4Oracle -c "echo \$ORACLE_HOME"
```

Tip: *If this environment variable is set for any account on the database server, it is applied to all database instances started by the same account on this server. Any mismatch between the (non-empty) environment variable, and the actual path used to start any of these database instances, prevents the collection of database inventory from the mismatched instance by the locally-installed inventory component (ndtrack). Conversely, you can prevent the environment variable option being used for all accounts on the target Oracle server by setting the UserDefinedOracleHome preference (details of this preference are included in Gathering FlexNet Inventory.*

- You can ensure that the Oracle home specified in the /etc/oratab file represents the ORACLE_HOME path used to start the database instance.

How the FlexNet Inventory Scanner Collects Oracle Inventory

In this approach, you have taken responsibility for the deployment and scheduling of the lightweight FlexNet Inventory Scanner (including installing the current version of the InventorySettings.xml file in the same folder as the scanner executable).

The following description assumes that all the prerequisites listed in FlexNet Inventory Scanner Collection of Oracle Inventory have been satisfied.

1. When your chosen scheduling tool triggers the FlexNet Inventory Scanner on the target inventory device, the tracker component performs discovery on its local device (recording results in a .disco file). For Oracle, the discovery process tries to identify one or more paths for $ORACLE_HOME, using all of these platform-dependent methods in order (and combining the resulting dataset):

UNIX-like platforms — Oracle discovery	Windows platforms — Oracle discovery
1. The tracker scans the file system for any `oracle` executables. • This scan honors the global settings for file system scans in the **File Inventory** section of the **Inventory Settings** page (**Data Collection > IT Assets Inventory Tasks > Inventory Settings**), which may limit search paths or even disable file scanning entirely. • On success, this search returns the file path (`$ORACLE_HOME`) and the executable ID.	1. The tracker looks for a registry entry under `HKLM\SOFTWARE\Wow6432Node\Oracle\` (or, on 32-bit systems, `HKLM\SOFTWARE\Oracle\`). On success, this returns the `%ORACLE_HOME%` path.
2. The tracker looks for an `oratab` file (installed in either `/etc` or `/var/opt/oracle` during database installation). On success, this provides the `$ORACLE_HOME` path, the executable ID for the `oracle` executable, and the System ID (SID) for each database instance listed in the `oratab` file. (See also note below.)	2. The tracker interrogates the Windows Service Control Manager. On success, this provides the `%ORACLE_HOME%` path, the System ID (SID) for the database instance running in that process, and the name of the related Oracle listener.
3. The tracker examines process listings for matches to `ora_smon_*`. On success, this provides the `$ORACLE_HOME` path and/or the executable ID for the `oracle` executable, as well as the SID for the running database instance and the user account running it. (See also note below.)	(No step 3 for Windows.)
4. The tracker examines process listings for `tnslsnr`. On success, this provides the `$ORACLE_HOME` path, the executable ID for the `oracle` executable, the name of the related Oracle listener, and the user account running the listener.	(No step 4 for Windows.)

Note: *If a symbolic link was used in the `$ORACLE_HOME` path to start a particular Oracle database instance on a UNIX-like platform, this can 'hide' the database instance from inventory collection by the locally-installed tracker (`ndtrack` component). To ensure inventory collection from a database instance started with a symbolic link, use either (or both) of the following workarounds:*

- *You can ensure that the Oracle home specified in the `/etc/oratab` file represents the `ORACLE_HOME` path used to start the database instance.*
- *The account running the database instance (say `OSUser4Oracle`) may set an environment variable within its login profile specifying the `ORACLE_HOME` path (including the symbolic link) which was used to start the database instance. To test this setting, the following command should display the correct `ORACLE_HOME` path:*

```
su -OSUser4Oracle -c "echo \$ORACLE_HOME"
```

Tip: *If this environment variable is set for any account on the database server, it is applied to all database instances started by the same account on this server. Any mismatch between the (non-empty) environment variable, and the actual path used to start any of these database instances, prevents the collection of database inventory from the mismatched instance by the locally-installed inventory component (`ndtrack`). Conversely, you can prevent the environment variable option being used for all accounts on the target Oracle server by setting the `UserDefinedOracleHome` preference (details of this preference are included in Gathering FlexNet Inventory.*

2. Next, the tracker gathers general hardware and software inventory (recorded in an `.ndi` file) from the local device. This is the standard inventory gathering that the tracker gathers on every device where it is running. If the appropriate preference is set, it also includes gathering evidence for Oracle Fusion Middleware applications (seeSpecial Handling of Oracle Fusion Middleware).
3. If the tracker discovered one or more Oracle database instance(s) on the device, and all the required settings and account privileges are in place, the tracker also gathers inventory from all accessible Oracle database instances.

Tip: *The definition of "accessible" changed at version 12.4 of the tracker (released with IT Asset Management 2017 R3):*

- *For versions 12.3 and earlier, "accessible" excluded all Oracle database instances that are in standby mode*
- *For versions 12.4 and later, the locally-installed tracker also collects inventory from Oracle database instances that are in standby (that is, `MOUNTED` but not in `RUN` mode), such as the standby instance in an Active Data Guard configuration.*

The processes for gathering Oracle inventory on different types of platform are as follows.

- For UNIX-like platforms, the tracker impersonates an appropriate account, determined by the configuration of the preferences shown below.

Note: *On a UNIX-like platform, the tracker attempts to use `setuid` to impersonate the appropriate account to gather Oracle inventory. If you are using eTrust Access Control on this server, by default it does not permit this impersonation, and inventory gathering fails. The fix is to change the configuration of eTrust to include `ndtrack` in the `LOGINAPPL` class. For more information, see the eTrust Access Control Administration Guide (https://supportcontent.ca.com/cadocs/0/g007711e.pdf).*

Each column in this table shows a set of conditions in the first three rows, followed by the resulting behavior in the last two rows. Once a valid user account is determined, the tracker invokes the Oracle-supplied `sqlplus` utility, giving it either of the command line parameters shown in the last row as appropriate:

UNIX-like platforms — Conditions	Option 1	Option 2	Option 3	Option 4	Option 5
If the executable runs as	`root`	`root`	`root`	`root`	Any other account

UNIX-like platforms — Conditions	Option 1	Option 2	Option 3	Option 4	Option 5
and `OracleInventoryAsSysdba` =	`True`	`True`	`False`	`False`	n.a.
and `OracleInventoryUser` =	Valid user	Not set	Valid user	Not set	n.a.
Then: Impersonated account is	That user (see ***Tip*** below)	Process owner	That user (see ***Tip*** below)	Not supported	n.a.
Command line parameters for `sqlplus`	`"/ as sysdba"`	`"/ as sysdba"`	`"/ "`	*No inventory*	*No inventory*

Tip: *If you intend to specify the `OracleInventoryUser` preference, it must be an exact match for any Oracle user name that:*

- *Is also an operating system account*
- *Has OS authentication enabled*
- *Is a member of `oinstall` (or equivalent group, granting execute permissions for `sqlplus`)*
- *Is either a current member of the `dba` group on the UNIX host server (when `OracleInventoryAsSysdba=True` or is unspecified); or has adequate permissions for inventory gathering (when `OracleInventoryAsSysdba=False`).*

- For Windows platforms, the tracker normally runs as `LocalSystem`, but may be run by another account that has administrator privileges (that is, is a member of the `Administrators` security group). In either case, the same account that is running the `ndtrack` executable is used to invoke the Oracle-supplied utility `sqlplus`, using the command line parameter `"/ as sysdba"`. This remains true even if the Oracle database instance is running as a service user, as is possible from Oracle Database 12c; so that binaries controlled by the service account are now executed as `LocalSystem` (or at least with administrator privileges). It is, of course, best practice to ensure that any service account running a database instance is well secured, so that the binaries it controls are protected.

4. Where Oracle inventory is collected, the tracker also executes scripts provided by Oracle Global Licensing and Advisory Services (GLAS) to gather software and hardware information about the servers where Oracle Database is installed.

   These scripts are extracted from the `InventorySettings.xml` file that you installed in the same directory as the FlexNet Inventory Scanner. The data gathered by these scripts helps to populate the Oracle audit report available through IT Asset Management (go to the **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**), and see details in the help for that page).

5. The tracker records the software and hardware results of the Oracle inventory gathering in a separate `.ndi` file.

6. By default, the FlexNet Inventory Scanner saves collected inventory in `%temp%\FlexeraSoftware\$(UserName) on $(MachineId).ndi`, where `%temp%` is the temporary directory for the account that is running the FlexNet Inventory Scanner, with the file name showing the account and

machine ID related to the inventory run. The files remain here (for example, for your inspection in an XML editor) until over-written in the next inventory collection. Alternatively, if the command line used to invoke the FlexNet Inventory Scanner included the options `-o Upload=True -o UploadLocation=http://`*`YourBeaconServerURL`*`/ManageSoftRL`, then immediately on completion of inventory gathering, the tracker uploads the `.disco` file and one or two compressed `.ndi` files to the appropriate inventory beacon.

Tip: *If this upload should fail (for example, because of an incorrect upload parameter in the command line, or a temporary network problem), the FlexNet Inventory Scanner does not have the capacity to run a catch-up at a later time. (For this catch-up facility, install the full FlexNet Inventory Agent.)*

7. The inventory beacon uploads all collected discovery and inventory information to the central application server (or, if it is a member of a hierarchy of inventory beacons, uploads the data to its parent in the hierarchy, and the upload is repeated until the data reaches the application server). Data is stored initially in the inventory database.
8. In due course, the inventory import (to the compliance database) and license reconciliation process runs (typically overnight, although an operator in a role granting the `Configure inventory data and reconcile` right can also trigger a full import and reconciliation). Progress is visible on the **System Tasks** page (**Data Collection > IT Assets Inventory Status > System Tasks**).
9. The **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**) lists the database inventory for all servers discovered and inventoried up to the time of the latest reconciliation calculation.

Troubleshooting Oracle Inventory Using the FlexNet Inventory Scanner

If you are using the FlexNet Inventory Scanner for Oracle discovery and inventory collection, a successful installation and invocation of the FlexNet Inventory Scanner is a prerequisite. As these processes are entirely in your control, the following guide assumes that you have validated them first.

To troubleshoot Oracle discovery and inventory gathering by the FlexNet Inventory Scanner:

1. Before concluding that there is a problem, ensure that you have allowed sufficient time for:
   - Discovery and inventory processes on the target device (check the schedule you have configured for the FlexNet Inventory Scanner, and allowing just a few minutes to gather inventory, even for a large server)
   - File upload to the inventory beacon, and from there to the central application server (typically within several minutes of the data gathering being completed)
   - Inventory import from the inventory database to the main compliance database (typically scheduled overnight)
   - The license reconciliation calculations (typically scheduled overnight).

   If, after this process has completed, the target Oracle server does not appear in the **All Discovered Devices** page (**Inventory > Network Discovery > All Discovered Devices**), it is time to continue troubleshooting. (If the device does appear in this listing, but the Oracle instance is not visible in the **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**), skip forward to step 4.)
2. On the target Oracle server, examine the tracker log file in the appropriate folder from this list:

- **On Windows:** `%temp%\ManageSoft\tracker.log` (where `$(TempDirectory)` is the temporary folder for the account running the tracker), or your chosen location if you redirected the logging at the command line.
- **On UNIX-like platforms:**
  - When the executable has run as the `root` account, in `/var/tmp/flexera/log`
  - When the executable has been run by any other user account (represented as *`UserName`*), in `/var/tmp/flexera.`*`UserName`*`/log`.
  - Your chosen location if you redirected the logging at the command line or in the `ndtrack.ini` file of preferences on UNIX-like platforms.

Verify the following events:

- The event `Uploading file <`*`filename`*`.disco>` indicates the generation of the discovery file has succeeded (upload is normally attempted immediately after creation of the discovery file).
- The event `File <`*`filename`*`.disco> removed from upload directory` indicates the successful upload of the discovery file.

3. If there is no evidence of a successful upload:
   a. Check the discovery upload folder on the target Oracle server, since files should be left behind when upload fails:
      - **On Windows:** `$(TempDirectory)\FlexeraSoftware\`, where `$(TempDirectory)` is the temporary directory for the account that is running the FlexNet Inventory Scanner
      - **On UNIX-like platforms:**
        - When the executable has run as the `root` account, in `/var/tmp/flexera/uploads/Discovery`
        - When the executable has been run by any other user account (represented as *`UserName`*), in `/var/tmp/flexera.`*`UserName`*`/uploads/Discovery`.
   b. If the relevant discovery upload folder still contains the `.disco` file, discovery is working but uploads are failing:
      - Double-check that the tracker command line options passed to the FlexNet Inventory Scanner correctly identified an accessible upload location on the correct inventory beacon.
      - Ping or otherwise check network access from the target Oracle server to the inventory beacon.
      - Ensure that the `ManageSoftRL` file share remains configured on the relevant inventory beacon (it is configured as part of the installation of the inventory beacon) and accessible.
      - Can you be sure there wasn't an intermittent network problem at last inventory time?
   c. If the relevant discovery upload folder on the target Oracle server is empty, upload to the inventory beacon may have worked as the first stage, and you can check for a later problem in the upload chain:

      Switch to the relevant inventory beacon, and:

      - Check `$(CommonAppDataFolder)\Flexera Software\Incoming\Inventories`. This is the staging folder for data uploaded from target devices but not yet uploaded to the parent of this inventory beacon. If a file is still here, the upload from this inventory beacon to its parent (another

inventory beacon) has failed.

- Check logs for any issues in `$(CommonAppDataFolder)\Flexera Software\Compliance\Logging\BeaconEngine`.

4. If discovery is working but Oracle inventory is not, the target Oracle server is visible in the **All Discovered Devices** page (**Inventory > Network Discovery > All Discovered Devices**), but there is no record of the Oracle instance(s) running on that server in the **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**). To troubleshoot, return to the log file for the `ndtrack` component on the target Oracle server:

   Verify the following events:

   - `Starting oracle inventory` and `Finished generating inventory` indicate the start and end of the Oracle inventory collection process.

     Note: *On a UNIX-like platform, the tracker attempts to use `setuid` to impersonate an appropriate user to gather Oracle inventory. If you are using eTrust Access Control on this server, by default it does not permit this impersonation, and inventory gathering fails. The fix is to change the configuration of eTrust to include `ndtrack` in the `LOGINAPPL` class. For more information, see the eTrust Access Control Administration Guide (https://supportcontent.ca.com/cadocs/0/g007711e.pdf).*

   - `Uploading file <filename (Oracle).ndi.gz>` indicates the upload of the compressed Oracle inventory file.
   - `File <filename (Oracle).ndi.gz> removed from upload directory` indicates the successful upload of the inventory file.

   Troubleshooting uploads is the same as covered earlier.

5. If Oracle Database inventory gathering is failing, first ensure that the appropriate account can use `sqlplus` to connect to the database instance:

   a. Identify the account relevant to this database instance. For example, on UNIX-like platforms:

      - If you are using the `OracleInventoryUser` preference on this inventory device (server), use that account
      - If not, identify the account that started the database instance. For example, the following command line lists all database instances on this server in the format *`oraclesid`*`_smon_`*`ORACLESID`* (the placeholders are replaced with the instance's system identifier in lower- and upper-case), with each row beginning with the relevant account name for that instance:

        ```
        ps -ef | grep smon
        ```

   b. Validate whether the `OracleInventoryAsSysdba` preference (in the `config.ini` file for the local FlexNet Inventory Agent on UNIX) has been set to false. This determines the command line parameters:

      ```
      sqlplus "/ " # when OracleInventoryAsSysdba=false
      sqlplus "/ as sysdba" # in all other cases
      ```

   c. Still using UNIX as our example, log in as `root` (the FlexNet Inventory Agent runs with this level of privilege to collect Oracle Database inventory, and using this saves you providing a password with the `su` command), and then switch to the identified account to run `sqlplus`, ensuring that the environment

variables are set for the Oracle home directory and the system identifier (SID) for the current database instance:

```
su accountUnderTest
export ORACLE_HOME=OracleHomeDirectory
export ORACLE_SID=OracleSID
cd $ORACLE_HOME/bin
./sqlplus "/ as sysdba" # or modified as noted above
```

If the connection is successful, `sqlplus` prints a `Connected to:` summary for the current database instance. Validate that the permissions are adequate and the database instance is fully operational with a test query such as:

```
SELECT VERSION FROM V$INSTANCE;
```

A response of the full version ID for the Oracle Database confirms that inventory collection can proceed.

If you cannot connect successfully, ask your Oracle DBA for assistance to resolve the problem. If connection was successful, continue with the following checks.

- On UNIX-like systems, check whether the *OracleHomeDirectory*`/bin/oracle` executable may have been upgraded or replaced without restarting the database instance. Remember that the tracker examines process listings, and if the original process was not restarted, it is now referencing a file that is no longer available, because it has been updated or replaced. If you cannot determine whether the executable has been changed, restart the database instance to test whether this clears the problem. Of course, if the executable *has* been changed, a process restart is essential, if for no other reason than to allow the database instance to utilize the updated code.
- Also on UNIX-like platforms, verify which version of the `ndtrack` executable is collecting inventory (versions earlier than 13.0.1 may fail to gather Oracle inventory on servers where permissions have been tightened to prevent global read access to Oracle directories or files). Use any of the following methods to verify the version of `ndtrack`:
  - If you already have a discovered device record for the server, navigate to its discovered device properties, select the **Inventory evidence** tab and the **Software** sub-tab. Find FlexNet Inventory Agent in the listing, which includes its version. (On UNIX-like platforms, for legacy reasons, this is listed as `ManageSoft for` *`platform`* `Managed Devices`.)
  - On the target inventory device (the Oracle server), read `/var/opt/managesoft/etc/config.ini` in a text editor, and validate the version of the inventory agent saved in the read-only `ETCPVersion` preference.

    ⚠ ***Caution:*** *Never edit this value manually.*
  - Examine the log file for `ndtrack`, as this reports the version of the agent in use. The default paths for logging depend on the inventory collection method:
    - For the locally-installed FlexNet Inventory Agent, see `/var/opt/managesoft/log/tracker.log` (or check for a custom value saved as `ManageSoft\Tracker\CurrentVersion\LogFile` in the `/var/opt/managesoft/etc/config.ini` file)

- For the lightweight FlexNet Inventory Scanner (presented as `ndtrack.sh` on UNIX-like systems), and also for zero footprint inventory collection, see `/var/tmp/flexera/log/tracker.log`

  Tip: *If `ndtrack.sh` is executed by a non-root account `userName`, the log defaults to: `/var/tmp/flexera.userName/log/tracker.log`.*

- Open an `.ndi` file saved on the locally-installed FlexNet Inventory Agent, which reports the version creating the file. Between inventory collections, the most recent inventory file is saved on the inventory device in `/var/opt/managesoft/tracker/inventories`, and the version appears at the top of the file as the `Tracker` attribute of the `Inventory` XML element.
- If you are using `ndtrack.sh`, this command line reports the version:

  ```
  cd directoryContainingScanner
  ./ndtrack.sh --version
  ```

- On all operating systems, if the preference `OracleInventoryAsSysdba=true` or is not specified (when the default is also true), make sure that the account running the database instance is listed in the ORADBA group (the local `ora_dba` security group on Windows and the `dba` group on UNIX-like platforms). By default this group exists, and the account running the instance is listed in it; but it is possible that the group has been removed on [some of] your Oracle servers. For full details about alternative accounts and all account requirements, see the `OracleInventoryUser` and `OracleInventoryAsSysdba` preferences in *Gathering FlexNet Inventory*.

If the problem persists, please contact Flexera Support with the full details and the appropriate log files.

Zero-footprint Collection of Oracle Inventory

"Zero-footprint" inventory collection is a term defined in *Gathering FlexNet Inventory*. It refers to an approach where an inventory beacon initiates the inventory gathering process (using different platform-specific methods), and then removes code artifacts afterward so that there is no permanent installation footprint on the target device.

Prerequisites for Zero-footprint inventory collection

The following must be in place for Zero-footprint collection of Oracle inventory by an inventory beacon:

1. The installed Oracle Database must be release 9i or later on UNIX-like platforms; and on Windows platforms, release 10g or later is preferred.
2. You have licensed the FlexNet Manager for Datacenters product (for details, see Appendix F: Features Enabled in FlexNet Manager for Datacenters).
3. You have deployed and configured one or more inventory beacon(s) in your network, such that at least one inventory beacon can access each of your target Oracle servers. For detailed information about inventory beacons, their deployment, and configuration, see the topics under *What Is an Inventory Beacon?* in the online help. You initiate deployment of an inventory beacon by navigating to the **Beacons** page (**Data Collection > IT Assets Inventory Tasks > Beacons**).

4. You have set up the accounts required for operation, both configuring them on the target Oracle servers and recording them in the secure Password Manager on the appropriate inventory beacons. Details of the accounts for different platforms are in Credentials for Zero-footprint Inventory.

Credentials for Zero-footprint Inventory

For Zero-footprint inventory, the required accounts and their privilege levels vary across computer platforms. For each target server, two accounts may be required:

- The first initializing account allows the inventory beacon to link to the target device, either download the appropriate files (for UNIX) or run a service (for Windows), and invoke the FlexNet Inventory Agent
- The second operational account is the account under which the FlexNet Inventory Agent is run.

On Microsoft Windows target devices

- The initializing account:
  - May be either a Windows domain account, or a local account on the target device
  - Requires full access to the Windows Service Control Manager on the target device (specifically, it must have the `SC_MANAGER_ALL_ACCESS` access right)
  - Must be appropriately registered in the secure Password Manager on the inventory beacon that is responsible for collecting inventory from this target device (for details, see *IT Asset Management Help* > *Inventory Beacons* > *Password Management Page* and its child topics)
  - May conveniently be the `LocalSystem` account, since this is required for the following operational stage.
- For the operational account, FlexNet inventory core components (and in particular the `ndtrack` component) run as the `LocalSystem` account.

On UNIX-like target devices

- The initializing account:
  - Is a local account on the target inventory device
  - Has `ssh` privileges on that device
  - Must be appropriately registered in the secure Password Manager on the inventory beacon that is responsible for collecting inventory from this target device (for details, see *IT Asset Management Help* > *Inventory Beacons* > *Password Management Page* and its child topics)

  Note: *When you save the SSH account details in the Password Manager, be sure to specify the additional details for elevation of account privileges with your preferred tool (such as `sudo` or `priv`).*
- For the operational account, FlexNet inventory core components (and in particular the `ndtrack` component) run as `root`.

Tip: *As always, it makes no difference whether you invoke the tracker directly as `root`, or whether you run as*

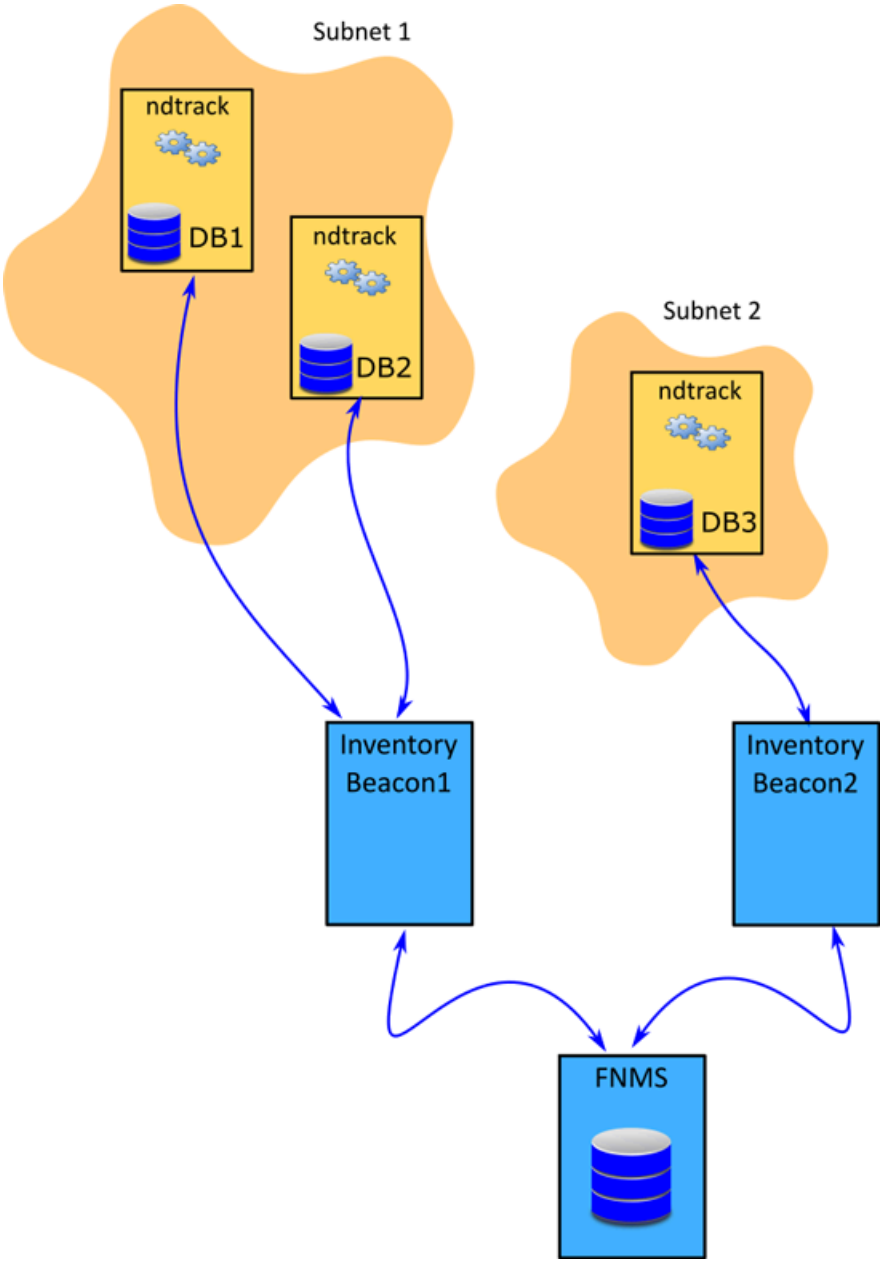
another account and use `sudo` (or similar) to elevate to `root` before invoking the tracker.

How Zero-footprint Collection of Oracle Inventory Works

In Zero-footprint inventory collection, the process is initiated by the inventory beacon. However, once the FlexNet Inventory Agent commences operation, it is running entirely in the context of the target device (the Oracle server). This means that it is functionally equivalent to the local agent-based collection of Oracle inventory, with the only differences being in the methods of initiation.

Tip: *In particular, the settings for Oracle inventory that are set in the* ***Actions*** *tab of the* ***Discovery and Inventory Rules*** *page (under* ***Oracle database environments****, where there are check boxes for* ***Discover Oracle database environments*** *and* ***Gather Oracle database environment inventory****) do not control the behavior of FlexNet Inventory Agent in this Zero-footprint case, any more than in the local agent-based collection. The Zero-footprint case is controlled only by the setting in the* ***General hardware and software inventory*** *section, and inventory collection is triggered when a rule includes the setting* ***Gather hardware and software inventory from all target devices***

The following diagram shows an example scenario for two inventory beacons.



The above diagram shows three database servers, two in Subnet1 and one in Subnet2. The Subnet1 is assigned to Inventory Beacon1 and Subnet2 is assigned to Inventory Beacon2. Each inventory beacon can connect to the Oracle server(s) in its assigned subnet. All the prerequisites outlined in Zero-footprint Collection of Oracle Inventory are satisfied.

The process for Zero-footprint collection of Oracle inventory runs like this:

1. To control the behavior of the inventory beacon(s), an operator sets an appropriate rule in the **Discovery and Inventory Rules** page (**Data Collection > IT Assets Inventory Tasks > Discovery and Inventory Rules**):
   - **Target**: One or more target(s) to identify all Oracle servers in your network.
   - **Action**: These settings:
     - For **Action type**, choose `Discovery and inventory`.

- Under **Discovery of devices**, select **Network scan** (and ensure the appropriate ports for your target devices are listed)
- In the **General hardware and software inventory** section, select **Gather hardware and software inventory from all target devices**.

- **Schedule**: Set your preferred schedule for execution of the completed rule.

2. By default, every 15 minutes each inventory beacon checks for any updates to its policy, which also transfers any changed discovery and inventory rules. (To adjust this download schedule, see *Inventory Settings Page > Beacon Settings Section* in the online help.) Each inventory beacon exercises only those rules that apply to its assigned subnet(s). The only action setting relevant to the Zero-footprint case is **Gather hardware and software inventory from all target devices**.
3. When the related schedule triggers an applicable rule, the inventory beacon initiates the process in ways appropriate to the target platform. Full details are available in the *Gathering FlexNet Inventory* PDF; but in summary:
   - **For Windows:** The FlexNet Beacon engine logs into the target server, and creates a Windows service that executes `ndtrack.exe` from the `mgsRET$` file share on the inventory beacon. Note that `InventorySettings.xml` is available to the FlexNet Inventory Agent so that advanced Oracle inventory capabilities are available; but there is no way to pass device-specific preferences to `ndtrack.exe`.
   - **For UNIX:** The FlexNet Beacon engine logs into the target server, elevates its privileges to root level, creates a secure link back to the inventory beacon, and copies `ndtrack.sh` to the target server and executes it. This script identifies the platform, unzips the appropriate `ndtrack` executable, and runs it. Note that the `ndtrack.ini` (which contains any agent preference settings you may have configured there) and `InventorySettings.xml` are delivered to the target device as well, allowing the functionality described below. However, since these are stored on the inventory beacon, the same values are delivered to each target device managed by an individual inventory beacon.

Tip: *The remaining steps in the process are almost identical to those for local agent-based inventory collection (with the exception of upload retries). The descriptions are repeated here for your convenience.*

4. The tracker (`ndtrack` executable) runs discovery on the target device, writing results to a `.disco` file. For Oracle, the discovery process tries to identify one or more paths for `$ORACLE_HOME`, using all of these platform-dependent methods in order (and combining the resulting dataset):

UNIX-like platforms — Oracle discovery	Windows platforms — Oracle discovery
1. The tracker scans the file system for any `oracle` executables. • This scan honors the global settings for file system scans in the **File Inventory** section of the **Inventory Settings** page (**Data Collection > IT Assets Inventory Tasks > Inventory Settings**), which may limit search paths or even disable file scanning entirely. • On success, this search returns the file path (`$ORACLE_HOME`) and the executable ID.	1. The tracker looks for a registry entry under `HKLM\SOFTWARE\Wow6432Node\Oracle\` (or, on 32-bit systems, `HKLM\SOFTWARE\Oracle\`). On success, this returns the `%ORACLE_HOME%` path.
2. The tracker looks for an `oratab` file (installed in either `/etc` or `/var/opt/oracle` during database installation). On success, this provides the `$ORACLE_HOME` path, the executable ID for the `oracle` executable, and the System ID (SID) for each database instance listed in the `oratab` file. (See also note below.)	2. The tracker interrogates the Windows Service Control Manager. On success, this provides the `%ORACLE_HOME%` path, the System ID (SID) for the database instance running in that process, and the name of the related Oracle listener.
3. The tracker examines process listings for matches to `ora_smon_*`. On success, this provides the `$ORACLE_HOME` path and/or the executable ID for the `oracle` executable, as well as the SID for the running database instance and the user account running it. (See also note below.)	(No step 3 for Windows.)
4. The tracker examines process listings for `tnslsnr`. On success, this provides the `$ORACLE_HOME` path, the executable ID for the `oracle` executable, the name of the related Oracle listener, and the user account running the listener.	(No step 4 for Windows.)

Note: *If a symbolic link was used in the `$ORACLE_HOME` path to start a particular Oracle database instance on a UNIX-like platform, this can 'hide' the database instance from inventory collection by the locally-installed tracker (`ndtrack` component). To ensure inventory collection from a database instance started with a symbolic link, use either (or both) of the following workarounds:*

- *You can ensure that the Oracle home specified in the `/etc/oratab` file represents the `ORACLE_HOME` path used to start the database instance.*
- *The account running the database instance (say `OSUser4Oracle`) may set an environment variable within its login profile specifying the `ORACLE_HOME` path (including the symbolic link) which was used to start the database instance. To test this setting, the following command should display the correct `ORACLE_HOME` path:*

```
su -OSUser4Oracle -c "echo \$ORACLE_HOME"
```

Tip: *If this environment variable is set for any account on the database server, it is applied to all database instances started by the same account on this server. Any mismatch between the (non-empty) environment variable, and the actual path used to start any of these database instances, prevents the collection of database inventory from the mismatched instance by the locally-installed inventory component (`ndtrack`). Conversely, you can prevent the environment variable option being used for all accounts on the target Oracle server by setting the `UserDefinedOracleHome` preference (details of this preference are included in Gathering FlexNet Inventory.*

5. Next, the tracker gathers general hardware and software inventory (recorded in an `.ndi` file) from the local device. This is the standard inventory gathering that the tracker gathers on every device where it is running. If the appropriate preference is set, it also includes gathering evidence for Oracle Fusion Middleware applications (seeSpecial Handling of Oracle Fusion Middleware).
6. If the tracker discovered one or more Oracle database instance(s) on the device, and all the required settings and account privileges are in place, the tracker also gathers inventory from all accessible Oracle database instances.

Tip: *The definition of "accessible" changed at version 12.4 of the tracker (released with IT Asset Management 2017 R3):*

- *For versions 12.3 and earlier, "accessible" excluded all Oracle database instances that are in standby mode*
- *For versions 12.4 and later, the locally-installed tracker also collects inventory from Oracle database instances that are in standby (that is, `MOUNTED` but not in `RUN` mode), such as the standby instance in an Active Data Guard configuration.*

The processes for gathering Oracle inventory on different types of platform are as follows.

- For UNIX-like platforms, the tracker impersonates an appropriate account, determined by the configuration of the preferences shown below.

Note: *On a UNIX-like platform, the tracker attempts to use `setuid` to impersonate the appropriate account to gather Oracle inventory. If you are using eTrust Access Control on this server, by default it does not permit this impersonation, and inventory gathering fails. The fix is to change the configuration of eTrust to include `ndtrack` in the `LOGINAPPL` class. For more information, see the eTrust Access Control Administration Guide (https://supportcontent.ca.com/cadocs/0/g007711e.pdf).*

Each column in this table shows a set of conditions in the first three rows, followed by the resulting behavior in the last two rows. Once a valid user account is determined, the tracker invokes the Oracle-supplied `sqlplus` utility, giving it either of the command line parameters shown in the last row as appropriate:

UNIX-like platforms — Conditions	Option 1	Option 2	Option 3	Option 4	Option 5
If the executable runs as	`root`	`root`	`root`	`root`	Any other account

UNIX-like platforms — Conditions	Option 1	Option 2	Option 3	Option 4	Option 5
and `OracleInventoryAsSysdba` =	`True`	`True`	`False`	`False`	n.a.
and `OracleInventoryUser` =	Valid user	Not set	Valid user	Not set	n.a.
Then: Impersonated account is	That user (see ***Tip*** below)	Process owner	That user (see ***Tip*** below)	Not supported	n.a.
Command line parameters for `sqlplus`	`"/ as sysdba"`	`"/ as sysdba"`	`"/ "`	*No inventory*	*No inventory*

Tip: *If you intend to specify the `OracleInventoryUser` preference, it must be an exact match for any Oracle user name that:*

- *Is also an operating system account*
- *Has OS authentication enabled*
- *Is a member of `oinstall` (or equivalent group, granting execute permissions for `sqlplus`)*
- *Is either a current member of the `dba` group on the UNIX host server (when `OracleInventoryAsSysdba=True` or is unspecified); or has adequate permissions for inventory gathering (when `OracleInventoryAsSysdba=False`).*

- For Windows platforms, the tracker normally runs as `LocalSystem`, but may be run by another account that has administrator privileges (that is, is a member of the `Administrators` security group). In either case, the same account that is running the `ndtrack` executable is used to invoke the Oracle-supplied utility `sqlplus`, using the command line parameter `"/ as sysdba"`. This remains true even if the Oracle database instance is running as a service user, as is possible from Oracle Database 12c; so that binaries controlled by the service account are now executed as `LocalSystem` (or at least with administrator privileges). It is, of course, best practice to ensure that any service account running a database instance is well secured, so that the binaries it controls are protected.

7. Where Oracle inventory is collected, the tracker also executes scripts provided by Oracle Global Licensing and Advisory Services (GLAS) to gather software and hardware information about the servers where Oracle Database is installed. (These scripts, as amended from time to time, are downloaded to the tracker from the `InventorySettings.xml` file. They are used only for the preparation of an Oracle audit report, available to operators who have appropriate access rights in the **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**), with more details available in the help for that page.)
8. The tracker records the software and hardware results of the Oracle inventory gathering in a separate `.ndi` file.
9. Immediately on completion of inventory gathering, the tracker uploads the `.disco` file and one or two compressed `.ndi` files to the appropriate inventory beacon.

Tip: *If this upload fails (say, because of a temporary network issue), there is no catch-up retry in the Zero-*

footprint case.

10. The inventory beacon uploads all collected discovery and inventory information to the central application server (or, if it is a member of a hierarchy of inventory beacons, uploads the data to its parent in the hierarchy, and the upload is repeated until the data reaches the application server). Data is stored initially in the inventory database.
11. In due course, the inventory import (to the compliance database) and license reconciliation process runs (typically overnight, although an operator in a role granting the `Configure inventory data and reconcile` right can also trigger a full import and reconciliation). Progress is visible on the **System Tasks** page (**Data Collection > IT Assets Inventory Status > System Tasks**).
12. The **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**) lists the database inventory for all servers discovered and inventoried up to the time of the latest reconciliation calculation.

Troubleshooting Zero-footprint Collection of Oracle Inventory

This case is very similar to the local agent-based collection of Oracle inventory, so if the following notes don't provide the answers you need, you can also check Troubleshooting Agent-Based Collection of Oracle Inventory.

Most of the troubleshooting information for this method is displayed on the **Rules** page in the web interface of IT Asset Management.

Note: *Inventory can be collected by running Oracle inventory using the `BeaconEngine.exe` process, or the `FlxRemoteExecutionLauncher.exe` process. When attempting to gather Oracle inventory from a large number of databases (10K+), there is the possibility of the `BeaconEgine.exe` process crashing due to memory issues. If the `BeaconEgine.exe` process crashes, see* ***step 11****.*

Note: *Additional troubleshooting steps unique to running Oracle inventory using the `FlxRemoteExecutionLauncher.exe` process will be added from* ***step 11*** *onwards when required.*

To troubleshoot Zero-footprint collection of Oracle inventory:

1. In the web interface for IT Asset Management, go to the **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**) to identify the database instances that have been discovered successfully. You may wish to focus on instances you expected to find that are not present on this page. (It is helpful to know the device name of an Oracle server you wish to examine, as you can use the name in searches in various lists.)
2. Another quick check is to look at the **All Discovered Devices** page (**Inventory > Network Discovery > All Discovered Devices**) and validate that the Oracle server is listed there. If the device itself cannot be discovered, no services on it can be discovered either. Also check for Oracle discovered devices listing problems:
    a. Click the filter icon above the right end of the list, and in the quick filter row, select **Oracle** = `Yes`.
    b. Next to the filter icon, click the notification icon to reduce the list to only those Oracle servers reporting problems.
    c. In the list, click the **Name** of a discovered device to open its properties, and select the **Status** tab.
    d. Expand the **Oracle Database inventory** section, which displays any error message returned by the Oracle listener on this discovered device. Seek the help of your Oracle database administrator to resolve any issues.

Here are some common errors with suggested things to check:

Error Message	Notes
ORA-12170: TNS: Connect timeout occurred	• Ensure that no firewall is blocking connection to the database. • Ensure that the database is online and running on correct IP.
ORA-12541: TNS: no listener	• The database is shutdown. Ask your database administrator to start the database. • Ensure that no firewall is blocking connection to the database. • Ensure that the listener is not password protected. If it is, add the listener password to the Password Manager on the appropriate inventory beacon.
ORA-00942: table or view does not exist	Ensure that the table listed in the error message exists. Ask your database administrator to create the table if it does not exist.
ORA-12514: TNS: listener does not currently know of service requested in connect descriptor	TNS names file is not correct. This is common when a database is set to listen for only “SID” or only “Service_Name”.
ORA-01017: invalid username/password; logon denied	Indicates incorrect permissions. Ask your Oracle Database administrator to rerun the Flexera 'audit user' script (for details, see Credentials for Direct Collection of Oracle Inventory).
ORA-01034: ORACLE not available ORA-27101: shared memory realm does not exist Linux-x86_64 Error: 2: No such file or directory	These errors typically mean that a discovered Oracle database instance was not running at inventory time. Ask your Oracle Database administrator whether the instance can be made active for the next inventory gathering process.
ORA-01033: ORACLE initialization or shutdown in progress	Oracle is stuck in a reboot process. Ask your database administrator to shutdown and restart the database.
ORA-12518: TNS: listener could not hand off client connection	Indicates a network problem. Ensure that the appropriate inventory beacon can access the Oracle server.

Error Message	Notes
ORA-00604: error occurred at recursive SQL level 1	Indicates incorrect permissions. Ask your Oracle Database administrator to rerun the Flexera 'audit user' script (for details, see Credentials for Direct Collection of Oracle Inventory). ***Note:*** *On a UNIX-like platform, the tracker attempts to use* `setuid` *to impersonate the appropriate account to gather Oracle inventory. If you are using eTrust Access Control on this server, by default it does not permit this impersonation, and inventory gathering fails. The fix is to change the configuration of eTrust to include* `ndtrack` *in the* `LOGINAPPL` *class. For more information, see the eTrust Access Control Administration Guide (https://supportcontent.ca.com/cadocs/0/g007711e.pdf).*
ORA-00257: archiver error. Connect internal only, until freed	Indicates an archive error. Ask your Oracle Database administrator to check the `archiver.log` file for the error.

3. Go to the **Discovery and Inventory Rules** page (**Data Collection > IT Assets Inventory Tasks > Discovery and Inventory Rules**), select the **Rules** tab and:

   a. Find the appropriate rule, and check that its **Rule status** value shows `Enabled`. (A disabled rule never executes.)

   b. Click the title of the relevant rule and note the **Action** name and **Targets** name for this rule for use shortly. Also validate that the **Schedule** details are as expected.

   c. Look in **Current run** or **Last completed run** to see the number of database instances discovered, inventoried, skipped, and failed. Here are some notes to assist your analysis of these figures:

Column	Notes
Service discovered	The number of Oracle database instances discovered in the last execution of the rule. This number should match (or possibly exceed) your expectations for Oracle servers within the targeted subnets. However, database discovery *cannot* succeed when: • The target device is powered off at the time of rule execution (to recover, re-run the rule when the server is operational). • The listener on the device is not operating when the rule executes (to recover, re-run the rule when the listener is running). • The password for the listener is not available in the Password Manager on the inventory beacon attempting discovery. • The listener returns an error, or does not identify any services (database instances).

Column	Notes
Inventory completed	The count of those instances from which database inventory has been collected. In an ideal world, this count of successful inventories may match the discovery result in the previous column. Differences should be tracked in the next two columns.
Inventory skipped	The count of database instances where collection of database inventory was not attempted. If a server is within scope for the inventory beacon, but no listener is identified on that server, it is counted as skipped.
Inventory failed	The count of database instances where inventory collection was attempted but failed. This may be because: • The credentials for the database instance were not configured in the Password Manager on the inventory beacon attempting to collect database inventory. • The instance was not running at inventory time. • The instance was running, but was not active (that is, was in the idle state) at inventory time.

d. Look to the bottom of the same expanded panel, and click **Show/hide task status and history**. By default this lists the last five executions of the rule. You can expand one (such as the most recent) to see all the inventory beacons that received the discovery and inventory rule (it still shows as `Scheduled` on inventory beacons that do not manage the appropriate subnets). You should find the intended inventory beacon in this list, where you can check the value in the **Status** column.

e. Use the **+** icon to expand more details for the chosen inventory beacon, and you see entries such as `Performing discovery` and `Gathering Oracle database inventory`. In the right-hand column are links for **Download log**. Download these, unzip the archive, and examine in a text file editor for clues to the problem.

4. In the area summarizing the run results, the **Devices failed to be inventoried** field indicates the number of devices for which general hardware and software inventory collection has failed. You can click the link to view the details in the **Rule Execution Details** page. For more information, see the **Rule Execution Details** page in the online help.

5. Using the noted **Action** name for this rule, switch to the **Actions** tab, locate the relevant action, and click the editing (pencil) icon on the right-hand side to show the details. In the **Discovery of devices** section, where **Network scan** is selected, verify the correct ports are included for all Oracle servers where database instances are running (especially focusing on instances that have not yet been discovered). If you are not sure about the correct port settings for each Oracle server, ask your system administrator. Recall that a probe on the ports listed here must get a response from the server itself before further discovery or inventory work is attempted.

6. In the **General hardware and software inventory** section, ensure that the **Gather hardware and software inventory from all target devices** option is selected.

7. If it appears that connection has been made and inventory collected, check for problems with file upload from the inventory beacon to the central application server. On the inventory beacon, examine `C:\Windows\Temp\ManageSoft\uploader.log` for any issues with uploads.

8. Still on the inventory beacon, check for inventory-specific errors in `$(CommonAppDataFolder)\Flexera Software\Compliance\Logging\InventoryRule\DeviceInventory.log` and `OracleDBInventory.log` files.

9. To enable richer logging on the inventory beacon, enable tracing by removing the hash character (#) from the following lines of the `etdp.trace` file present in the `%Program Files (x86)%\Flexera Software\Inventory Beacon\` folder:

```
+Inventory/Oracle
+Inventory/Oracle/SDK
+Inventory/Oracle/Query
+Inventory/Oracle/Query/Substitution
+Inventory/Oracle/Query/Execution
+Inventory/Oracle/Listener
+Inventory/Oracle/Listener/Detail
```

   Important: *After making changes to the `etdp.trace` file, you must use the Windows Service Controller on the inventory beacon to stop and restart the `beaconengine` service.*

10. Rerun the rule to collect Oracle discovery and inventory information, so that additional logging is created. See whether these richer logs assist in identifying the problem. Wait until after the next inventory import, and then look for both the discovered device (**All Discovered Devices**) and its Oracle instance (**Oracle Instances**). (Rather than waiting, an operator in an Administrator role can navigate to the **Reconcile** page (**Data Collection > Process Data > Reconcile**), ensure that the **Update inventory for reconciliation** option is selected, and click **Reconcile**.)

11. When attempting to gather Oracle inventory from a large number of databases (10K+), there is the possibility of the `BeaconEgine.exe` process crashing due to memory issues. To address this issue, you can execute the Oracle inventory process externally using `FlxRemoteExecutionLauncher.exe`, separate from the `BeaconEngine.exe` process. This approach allows memory to be returned to the system after the completion of each Oracle inventory task. To enable this option, add the `ExternalProcessExecution` registry entry with the value, `Oracle`, for the following registry key: `HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\ManageSoft Corp\ManageSoft\RemoteExecution\CurrentVersion`. If the `ExternalProcessExecution` registry entry is not present, the Oracle inventory process will continue to be executed within `BeaconEngine.exe`, as previously established.

12. If the problem persists, contact Flexera support with detailed information and log files.

Direct Collection of Oracle Inventory

The phrase 'direct collection of inventory' refers to techniques where an inventory beacon connects directly to a data source (possibly through an API) to collect information. In the case of Oracle, the method requires that FlexNet Beacon (the code engine on an inventory beacon) connects to the Oracle Database (using an OLEDB client library) through the Oracle Net Listener, and collects the available information as database inventory.

Important: *The direct collection inventory method only gathers information about database instances. It does not, for example, collect any hardware inventory data nor software inventory about any other products. This means that*

*the direct collection inventory method **alone** is insufficient to correctly determine license consumption for Oracle products:*

- *For licensing of database instances and options, hardware information (such as the count of cores) is missing, making this inventory method (used alone) unsuitable for managing, for example, Oracle Processor licenses*
- *For other products such as Oracle Fusion Middleware, there is no software inventory available from the Oracle Database (although inventory for Oracle Fusion Middleware can be collected separately by the FlexNet Inventory Agent, as outlined in Special Handling of Oracle Fusion Middleware).*
- *No inventory of database instances that are in standby mode is possible.*

This means that, if you choose to use direct collection of Oracle inventory (and want to manage Oracle licenses), you must augment the approach with other ways of gathering hardware and general software inventory. You could achieve this, for example, through third-party products, from which you import inventory into IT Asset Management to allow for data integration and license consumption calculations. Alternatively, reconsider local Agent-Based Collection of Oracle Inventory.

***Tip:** Using a locally-installed FlexNet Inventory Agent solves all of the above issues. For example, it can return inventory of Oracle Fusion Middleware and report this in the standard archive for handing off to Oracle GLAS at audit time. IT Asset Management is verified by Oracle for collecting inventory of Oracle Fusion Middleware.*

The direct method for direct collection of Oracle inventory does not inherently include any process for *discovery* of Oracle installations. Details of an Oracle database instance must be already available to allow the direct connection to proceed. Therefore, before using direct collection of Oracle inventory, you must decide on, and implement, a discovery process.

Four distinct methods of providing discovery details are available:

- **Network discovery:** You can set rules so that the FlexNet Beacon engine probes the network to discover appropriate servers. It tests those servers for the presence of one or more Oracle listeners, from which it collects access details for the database instances known to each listener. The inventory beacon then accesses the discovered servers immediately for inventory gathering; and it also uploads the discovery results along with the inventory results. This approach produces discovered device records in IT Asset Management.

***Note:** This discovery approach is supported only for earlier versions of Oracle Database:*

  - *For version 9i and below, it is recommended that you have set a listener password. If you have done so, you need to provide this password as part of the credentials needed for this approach to work.*
  - *For versions 10g and 11g, it was best practice not to set a listener password, since the default behavior was changed to disallow remote connections and rely on operating system authentication for local access. If you wish to use direct collection of inventory with these versions, you must supply a listener password to turn on support for remote connections.*
  - *Making a remote administration connection to the listener (using the listener's administration password) to discover the services known to the listener is not possible from Oracle 12c onwards. Therefore, **for Oracle Database 12c and beyond, you cannot use network discovery in conjunction with direct collection of Oracle Database inventory.** (This does not affect the remaining discovery methods, below, which already provide the service name for each Oracle database instance; and, together with the credentials saved in the inventory beacon's Password Manager, these are the only details required for the listener to broker remote connection to the database instance to allow for inventory gathering.)*

- **Using `tnsnames.ora`:** This is a standard Oracle file that identifies database instances and connection details. You may take a copy from your Oracle server and save it on the inventory beacon, or you may use the OEM adapter to create one. When you first trigger direct inventory collection using this method of discovery, there are no matching discovered devices visible in the web interface of IT Asset Management; but as the first inventory upload occurs, discovery information is also uploaded, and matching discovered device and inventory device records are created in the same import process.
- **Manually-created records:** You can manually enter the listener and services information for each Oracle server through the web interface of IT Asset Management.
- **Amazon connector:** This connector does the work of discovering installations of Oracle Database running in Amazon Relational Database Service (RDS). The data imported from the connector automatically creates records for both a discovered device and an inventory device. The discovered device can then be used as a target for direct collection of Oracle Database inventory, with all the usual prerequisites (such as recording credentials for each database in the password store on the inventory beacon that accesses the database, and also using the **Cloud regions** tab of the **Beacon Properties** page to narrow the management range of each inventory beacon).

More information about each discovery method is available in the appropriate topics:

- Using Network Discovery with Direct Inventory
- Using tnsnames Discovery with Direct Inventory
- Using Manual Discovery with Direct Inventory.

When discovered device records are available from a previously-completed discovery process, you may also choose to conduct direct inventory collection using those existing records. In this case, the existing discovery information is downloaded from the central application server to the inventory beacon, and used for direct inventory gathering. In operation, this is identical to the case with manually-created records, so see Using Manual Discovery with Direct Inventory.

Prerequisites for direct collection of Oracle inventory

The following must be in place for collection of Oracle inventory by direct connection from an inventory beacon to the Oracle Database:

1. You have licensed the FlexNet Manager for Datacenters product (for details, see Appendix F: Features Enabled in FlexNet Manager for Datacenters).
2. You have deployed and configured one or more inventory beacon(s) in your network, such that at least one inventory beacon can access each of your target Oracle servers. For detailed information about inventory beacons, their deployment, and configuration, see the topics under *What Is an Inventory Beacon?* in the online help. You initiate deployment of an inventory beacon by navigating to the **Beacons** page (**Data Collection > IT Assets Inventory Tasks > Beacons**).
3. You have set up the accounts required for operation and recorded these in the Password Manager on the appropriate inventory beacon (for more information, see Credentials for Direct Collection of Oracle Inventory).
4. You have downloaded and installed on each inventory beacon the appropriate version of the 32-bit Oracle Provider for OLEDB used by the FlexNet Beacon engine to connect the Oracle Database. This driver is included the Oracle Data Access Components (ODAC), and must be the version that is compatible with the target Oracle Database accessed by that inventory beacon.
   - To determine the appropriate driver for your target version of Oracle Database, use your Oracle support account

to access the OLEDB driver compatibility matrix from `http://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=207303.1`. Another version of the compatibility matrix (maintained by a third-party consulting organization) is available at `http://www.dba-oracle.com/t_oracle_client_versions_higher_lower_database_release.htm`. The ODAC versions are available for download from Oracle.

- Each inventory beacon may have exactly one version of the OLEDB drivers installed. As a result, if you have two (or more) different Oracle Database servers that require different versions of the OLEDB drivers, each *must* be managed by a separate inventory beacon. Since inventory beacons are assigned distinct subnets to manage, this means that if the incompatible Oracle Database products are installed within the same subnet, you must split the subnet to separate those Oracle servers, and then assign each new subnet to a distinct inventory beacon, each of which has the appropriate OLEDB driver matching the target Oracle Database version.

Credentials for Direct Collection of Oracle Inventory

The accounts and privileges required for direct collection of inventory data from an Oracle database instance are relatively straight-forward. In addition to inventory collection, the entire process must allow for the initial *discovery* of devices and their database instances to query, and some discovery methods add their own requirements for credentials.

Using network discovery

This method of discovery requires two sets of credentials for discovery.

The first is an account that can be used to log into the target device and probe specified ports to test for the presence of an Oracle Net Listener. This account must be registered in the Password Manager on the inventory beacon responsible for discovery (and subsequent inventory gathering).

Once a device has been discovered, and the Oracle Net Listener is known to exist on that device, the listener must accept a remote connection and status request (that is, remote administration). This request identifies the database instances (services) that the listener knows about. (The listener password is required only for the case of network discovery, since this is the only case that requires a remote administration connection to the listener; in the other cases, the database instances are already identified without this request.) This password must be registered in the secure Password Manager on the inventory beacon that is to complete the discovery and collect the database inventory (only the password is required, and no account name is needed with this listener password.)

Tip: *If there are multiple listeners on the Oracle server and these have multiple passwords, each password for listeners you will access must be recorded in the Password Manager. These are not differentiated by any listener identification: the FlexNet Beacon engine simply steps through each listener password in turn until one works.*

Note: *This method of remote administration of the listener to discover the available Oracle database instances has been barred from Oracle Database version 12c onwards. To use direct inventory gathering with later versions of the database, you must use one of the other discovery methods; and in those alternative discovery methods, the listener's administrative password is not required. (Of course, this is independent of the credentials for inventory gathering, described below, that are required after discovery has been completed.)*

Using tnsnames.ora

There are no special credentials required, other than the normal ones to get the `tnsnames.ora` in place on the inventory beacon (for details see Using tnsnames Discovery with Direct Inventory):

- If you are manually transferring a `tnsnames.ora` file from your Oracle server, you need to log in to the inventory beacon with sufficient privileges to access the file path (typically, with administrator privileges)
- If you are using the OEM adapter, this writes the `tnsnames.ora` file into the correct location, and the credentials needed for the adapter are covered in *IT Asset Management Inventory Adapters and Connectors Reference*.

Using manually-created discovery device records

No additional credentials are required — an operator in a Role with sufficient privileges to create the records does the data input, and these discovered device records are then utilized automatically when required.

Using Amazon connector

There are no special requirements when using the Amazon connector to discover Oracle Databases running in Amazon Relational Database Service (RDS). For details about running the connector, see either of the following:

- The online help under **FlexNet Manager Suite Help > Inventory Beacons > Inventory Systems Page > Connecting to External Inventory Systems > Managing PowerShell Connections > Managing AWS Connections** and child topics
- The section on the AWS connector in *IT Asset Management Inventory Adapters and Connectors Reference*.

Credentials for inventory gathering

Regardless of the discovery method you use, direct inventory collection proceeds by having the inventory beacon connect to the listener requesting access to each service/database instance (no listener password is required for this request). These connection requests require the service name, the service user account, and the service password for each Oracle database instance. The service user account:

- Is a member of the OS-specific ORADBA group (the local `ora_dba` security group on Windows platforms and the `dba` group on UNIX-like platforms)
- Has at least read-only permissions for all the tables and views needed for collecting Oracle inventory (listed in Appendix C: Oracle Tables and Views for Oracle Inventory Collection)
- Has the user name and password registered in the secure Password Manager on the inventory beacon that is to collect inventory from each database instance.

One potentially helpful practice is to use the same set of credentials on all target Oracle servers as a special "audit account". This makes it easier to register a single set of credentials in the Password Managers on all applicable inventory beacons, and to script creation of the account consistently across all your Oracle servers. If you choose to use a common audit account across servers, Flexera provides a script to create and configure this database user. To get this script, log into the Flexera Community Knowledge Base at .

Note: *The sole purpose of creating this audit user is to collect Oracle inventory. However, IT Asset Management counts it as a named user while calculating license compliance for Oracle licenses. You can adjust the license consumption for this user to avoid consuming license entitlements. Navigate to the* ***Oracle Instance Properties >***

Oracle users *page for each affected database instance and set the consumption for this user to zero. For more information, see the associated online help.*

How Direct Collection of Oracle Inventory Works

In the direct inventory collection method, the FlexNet Beacon engine on an inventory beacon connects directly to an Oracle server that lies within its assigned subnet(s) (or, in the special case of Amazon RDS, in its assigned cloud region), and collects only Oracle Database inventory. While the direct connection step in itself is relatively straight-forward, it must be preceded by your choice of discovery method used to identify the target Oracle server. For completeness, each of these discovery methods is covered in the following topics, and you may focus only on your chosen discovery method together with the subsequent direct inventory collection. The four discovery methods are:

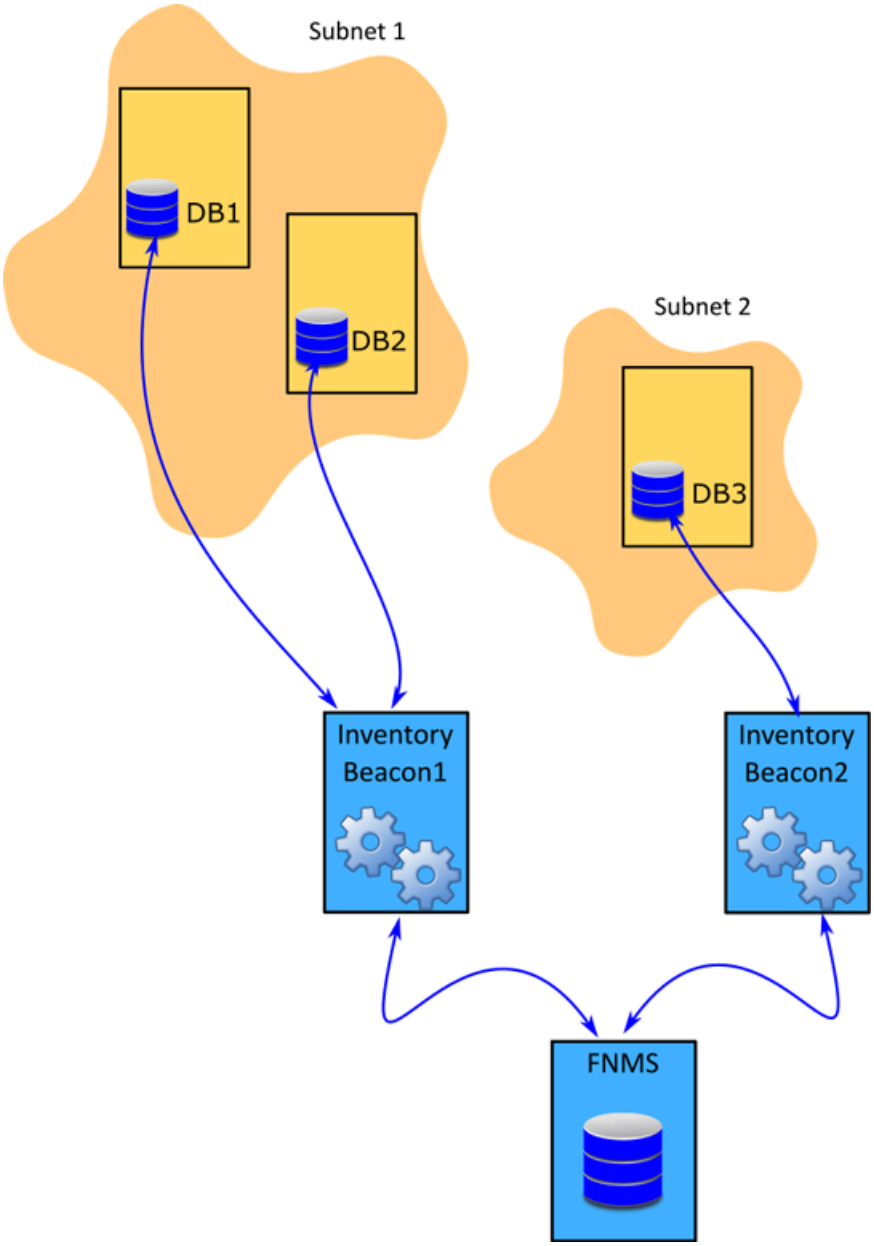
- Network scanning (see Using Network Discovery with Direct Inventory)
- Processing `tnsnames.ora` files, whether these be copied from your Oracle server(s), created with the OEM adapter, or even edited manually (see Using tnsnames Discovery with Direct Inventory)
- Manually creating discovered device records including the necessary connection information (see Using Manual Discovery with Direct Inventory)
- Using the Amazon connector to discover installations of Oracle Database in Amazon RDS and create discovered device and inventory device records for each case; and then using the discovered device records to target for direct inventory collection.

Using Network Discovery with Direct Inventory

In this approach, each inventory beacon takes responsibility for both discovery of the Oracle servers and taking inventory of the database instances on them.

Remember: *From Oracle Database 12c onwards, password access direct to the listener to query it for the available instances has been blocked, so that this method of discovery is available only for earlier versions of Oracle Database.*

The following diagram shows an example scenario:



The above diagram shows three database servers, two on Subnet1 and one on Subnet2. The Subnet1 is assigned to Inventory Beacon1 and Subnet2 is assigned to Inventory Beacon2. Note that in this diagram, DB1 and DB2 are running versions of Oracle Database compatible with the same OLEDB driver, as discussed in Direct Collection of Oracle Inventory; separately, DB3 may also be compatible, or may require a different version, which may be the reason that it is managed by a separate inventory beacon.

The following description assumes that all the prerequisites listed in Direct Collection of Oracle Inventory have been satisfied, including the installation of an appropriate version of the 32-bit OLEDB driver (one version per inventory beacon) for each target Oracle Database.

1. It is likely that the Oracle Net Listener(s) on each target Oracle server has/have been configured with a password for administrative use, as when collecting status and the list of known services (as discussed in Direct Collection of Oracle Inventory, which also lists the versions of Oracle Database for which this approach is supported). If so, ensure that the password is recorded in the secure Password Manager on each appropriate inventory beacon. No

user name is required when you record the listener password.

2. On each target Oracle server, set up the special "audit account" for collecting database inventory (see Credentials for Direct Collection of Oracle Inventory).
3. Record the credentials for the special audit account in the secure Password Manager on each applicable inventory beacon.
4. Create a rule that combines network discovery with direct inventory. Go to the **Discovery and Inventory Rules** page (**Data Collection > IT Assets Inventory Tasks > Discovery and Inventory Rules**) and use these settings:
   - **Target**: Create a target to identify all Oracle servers (up to and including Oracle Database 11g) in your network. You can use subnet name, IP address, or pattern matching on the device name to identify devices in the target definition.
   - **Action**: Create an action that includes these settings:
     - For **Action type**, choose `Discovery and inventory`.
     - In the **Discovery of devices** section, select **Network scan**, and ensure that the appropriate ports are listed. These are the ports tested for the existence of the server itself (typically, for example, 22 for a UNIX server and 139 for a Windows server). Customize this list if your environment requires it. The hardware (server) must respond before further discovery of Oracle details is attempted.
     - Expand the **Oracle database environments** section, and select **Discover Oracle database environments**. In the additional controls for **Discovery methods**, select **Port scan**, and if necessary update the list of ports to include all listener ports used in your environment. When each inventory beacon executes this action, it tests each port in turn on each target server, seeking a response from Oracle Net Listener. (Less commonly, if you have SNMP configured on your Oracle servers, you may select the **SNMP** option as well, or instead.)
     - In the same section, select **Gather Oracle database environment inventory**. This is the switch to turn on direct inventory gathering for Oracle database instances.
     - **Schedule**: Specify the running schedule for this rule.

   This completes the initial configuration for network discovery with direct inventory gathering for Oracle. The remainder of these steps cover normal operation.
5. By default, every 15 minutes each inventory beacon checks for any updates to its policy, which also transfers any changed discovery and inventory rules. (To adjust this download schedule, see *Inventory Settings Page > Beacon Settings Section* in the online help.) Each inventory beacon exercises only those rules that apply to its assigned subnet(s).
6. When the related schedule triggers an applicable rule:
   a. The inventory beacon reports to the central application server that the task is commencing. You can go to the **Discovery and Inventory Rules** page (**Data Collection > IT Assets Inventory Tasks > Discovery and Inventory Rules**) and click the rule name to view its status. Wait (while the remainder of the process takes place) until the **Status** field shows `Completed`. This process may take some time to complete and you may have to revisit or refresh the page from time to time.
   b. The inventory beacon scans each of its assigned subnets, recording every device that responds to probing on the specified device ports. (Each of these discovered devices will be reported to the central application server in the uploaded `.disco` file, regardless of what else is found on each device.)

c. The inventory beacon next probes each discovered device for a response on each of the specified listener ports (or, less commonly, collects SNMP data from each discovered device and checks for the presence of Oracle database instances).

d. From devices where a listener responds, the inventory beacon extracts the list of known Oracle services (database instances).

All this information is included in the uploaded `.disco` file. For direct inventory gathering (where the discovery files may be larger than with the locally-installed FlexNet Inventory Agent), `.disco` files are compressed for upload.

7. The FlexNet Beacon engine (on each inventory beacon) uses the discovered services information to connect to each database instance, and collect Oracle Database inventory data. The results are saved into an `.ndi` inventory file (specific to Oracle inventory only) for each server.

Tip: *Because listeners may know about database instances on several distinct servers, one inventory beacon may return database inventory from multiple servers, and (depending on the configuration of your network and listeners) not necessarily restricted to the subnet(s) assigned to that inventory beacon. The FlexNet Beacon engine uses information retrieved from each database instance to identify the servers, and ensure that it returns one* `.ndi` *inventory file per server.*

8. Where Oracle inventory is collected, the FlexNet Beacon engine also executes the *software* scripts provided by Oracle Global Licensing and Advisory Services (GLAS) to gather software information about the Oracle Database installation. It is important to realize two things:

- These are the same GLAS scripts, as amended from time to time, that are downloaded to the inventory beacon in the `InventorySettings.xml` file (which is saved in `Program Files (x86)\Flexera Software\Inventory Beacon\Remote Execution\Public\Inventory`). As always, they do not contribute to the collection of FlexNet inventory, but are used only for the preparation of an Oracle audit report (available to operators who have appropriate access rights in the **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**), with more details available in the help for that page.)
- However, the Oracle GLAS scripts include two separate parts: SQL queries to gather information from Oracle database instance(s); and scripts to execute on the target server and gather the required hardware information. With this direct connection method of inventory collection, there is no file transfer of any kind to the target server, so that the GLAS hardware scripts *cannot be executed* on the Oracle server by this method. In contrast, the GLAS SQL queries *are* executed during the direct connection to each database instance, and the resulting GLAS software data is uploaded to the central application server; but in the normal course of events, the absence of hardware data is unlikely to satisfy an Oracle audit. Therefore, if your strategy is to use only the direct method of inventory collection, you should also plan another method to execute the GLAS hardware scripts on each Oracle server. Alternatively, reconsider the local installation of either the full FlexNet Inventory Agent, or the lightweight FlexNet Inventory Scanner; or re-evaluate the Zero-footprint method of inventory collection. All of these methods conveniently collect, upload, and create audit-ready packages for *all* the software and hardware data required for database licensing by Oracle GLAS, and without additional effort on your part.

9. Immediately after inventory gathering, the inventory beacon compresses the `.ndi` file(s) and `.disco` files, and uploads them to the central application server (or, if it is a member of a hierarchy of inventory beacons, uploads the data to its parent in the hierarchy, and the upload is repeated until the data reaches the application server). It is initially stored in the inventory database.

10. In due course, the inventory import (to the compliance database) and license reconciliation process runs (typically overnight, although an operator in a role granting the `Configure inventory data and reconcile` right can also trigger a full import and reconciliation). Progress is visible on the **System Tasks** page (**Data Collection > IT Assets Inventory Status > System Tasks**).
11. The **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**) lists the database inventory for all servers discovered and inventoried up to the time of the latest reconciliation calculation.

Troubleshooting Direct Inventory Using Network Scan

In direct gathering of Oracle inventory, the FlexNet Beacon engine connects directly to Oracle databases using database port information. If you select network scan as the method of discovery, the FlexNet Beacon engine first tries to discover the required port information by finding and querying the appropriate Oracle listener services.

To troubleshoot Oracle inventory gathering with discovery by network scanning:

1. In the web interface for IT Asset Management, go to the **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**) to identify the database instances that have been discovered successfully. You may wish to focus on instances you expected to find that are not present on this page. (It is helpful to know the device name of an Oracle server you wish to examine, as you can use the name in searches in various lists.)
2. Another quick check is to look at the **All Discovered Devices** page (**Inventory > Network Discovery > All Discovered Devices**) and validate that the Oracle server is listed there. If the device itself cannot be discovered, no services on it can be discovered either. Also check for Oracle discovered devices listing problems:
    a. Click the filter icon above the right end of the list, and in the quick filter row, select **Oracle** = `Yes`.
    b. Next to the filter icon, click the notification icon to reduce the list to only those Oracle servers reporting problems.
    c. In the list, click the **Name** of a discovered device to open its properties, and select the **Status** tab.
    d. Expand the **Oracle Database inventory** section, which displays any error message returned by the Oracle listener on this discovered device. Seek the help of your Oracle database administrator to resolve any issues.

    Here are some common errors with suggested things to check:

Error Message	Notes
ORA-12170: TNS: Connect timeout occurred	• Ensure that no firewall is blocking connection to the database. • Ensure that the database is online and running on correct IP.

Error Message	Notes
ORA-12541: TNS: no listener	• The database is shutdown. Ask your database administrator to start the database. • Ensure that no firewall is blocking connection to the database. • Ensure that the listener is not password protected. If it is, add the listener password to the Password Manager on the appropriate inventory beacon.
ORA-00942: table or view does not exist	Ensure that the table listed in the error message exists. Ask your database administrator to create the table if it does not exist.
ORA-12514: TNS: listener does not currently know of service requested in connect descriptor	TNS names file is not correct. This is common when a database is set to listen for only “SID” or only “Service_Name”.
ORA-01017: invalid username/password; logon denied	Indicates incorrect permissions. Ask your Oracle Database administrator to rerun the Flexera 'audit user' script (for details, see Credentials for Direct Collection of Oracle Inventory).
ORA-01034: ORACLE not available ORA-27101: shared memory realm does not exist Linux-x86_64 Error: 2: No such file or directory	These errors typically mean that a discovered Oracle database instance was not running at inventory time. Ask your Oracle Database administrator whether the instance can be made active for the next inventory gathering process.
ORA-01033: ORACLE initialization or shutdown in progress	Oracle is stuck in a reboot process. Ask your database administrator to shutdown and restart the database.
ORA-12518: TNS: listener could not hand off client connection	Indicates a network problem. Ensure that the appropriate inventory beacon can access the Oracle server.

Error Message	Notes
ORA-00604: error occurred at recursive SQL level 1	Indicates incorrect permissions. Ask your Oracle Database administrator to rerun the Flexera 'audit user' script (for details, see Credentials for Direct Collection of Oracle Inventory). ***Note:*** *On a UNIX-like platform, the tracker attempts to use* `setuid` *to impersonate the appropriate account to gather Oracle inventory. If you are using eTrust Access Control on this server, by default it does not permit this impersonation, and inventory gathering fails. The fix is to change the configuration of eTrust to include* `ndtrack` *in the* `LOGINAPPL` *class. For more information, see the eTrust Access Control Administration Guide (https://supportcontent.ca.com/cadocs/0/g007711e.pdf).*
ORA-00257: archiver error. Connect internal only, until freed	Indicates an archive error. Ask your Oracle Database administrator to check the `archiver.log` file for the error.

3. Go to the **Discovery and Inventory Rules** page (**Data Collection > IT Assets Inventory Tasks > Discovery and Inventory Rules**), select the **Rules** tab and:

   a. Find the appropriate rule, and check that its **Rule status** value shows `Enabled`. (A disabled rule never executes.)

   b. Click the title of the relevant rule and note the **Action** name and **Targets** name for this rule for use shortly. Also validate that the **Schedule** details are as expected.

   c. Look in **Current run** or **Last completed run** to see the number of database instances discovered, inventoried, skipped, and failed. Here are some notes to assist your analysis of these figures:

Column	Notes
Service discovered	The number of Oracle database instances discovered in the last execution of the rule. This number should match (or possibly exceed) your expectations for Oracle servers within the targeted subnets. However, database discovery *cannot* succeed when: • The target device is powered off at the time of rule execution (to recover, re-run the rule when the server is operational). • The listener on the device is not operating when the rule executes (to recover, re-run the rule when the listener is running). • The password for the listener is not available in the Password Manager on the inventory beacon attempting discovery. • The listener returns an error, or does not identify any services (database instances).

Column	Notes
Inventory completed	The count of those instances from which database inventory has been collected. In an ideal world, this count of successful inventories may match the discovery result in the previous column. Differences should be tracked in the next two columns.
Inventory skipped	The count of database instances where collection of database inventory was not attempted. If a server is within scope for the inventory beacon, but no listener is identified on that server, it is counted as skipped.
Inventory failed	The count of database instances where inventory collection was attempted but failed. This may be because: • The credentials for the database instance were not configured in the Password Manager on the inventory beacon attempting to collect database inventory. • The instance was not running at inventory time. • The instance was running, but was not active (that is, was in the idle state) at inventory time.

d. Look to the bottom of the same expanded panel, and click **Show/hide task status and history**. By default this lists the last five executions of the rule. You can expand one (such as the most recent) to see all the inventory beacons that received the discovery and inventory rule (it still shows as `Scheduled` on inventory beacons that do not manage the appropriate subnets). You should find the intended inventory beacon in this list, where you can check the value in the **Status** column.

e. Use the **+** icon to expand more details for the chosen inventory beacon, and you see entries such as `Performing discovery` and `Gathering Oracle database inventory`. In the right-hand column are links for **Download log**. Download these, unzip the archive, and examine in a text file editor for clues to the problem.

4. Using the noted **Action** name for this rule, switch to the **Actions** tab, locate the relevant action, and click the editing (pencil) icon on the right-hand side to show the details. In the **Discovery of devices** section, where **Network scan** is selected, verify the correct ports are included for all Oracle servers where database instances are running (especially focusing on instances that have not yet been discovered). If you are not sure about the correct port settings for each Oracle server, ask your system administrator. Recall that a probe on the ports listed here must get a response from the server itself before further discovery or inventory work is attempted.

5. Verify the listener ports settings specified in the **Oracle database environments** section of the action definition. If you are not sure about the correct port settings, ask your database administrator. For more information, see Using Network Discovery with Direct Inventory.

6. Switch to the **Targets** tab, locate the target whose name you noted earlier, and click the editing (pencil) icon on the right-hand end to examine the specified target. Validate that the setting for **Define machines to target** includes the Oracle server you are searching for. If you are suspicious that your target is not working correctly, you can also expand each of your other targets looking for an `Exclude` condition in **Define machines to target**. Any overlapping exclude condition in any target, even though that target is not linked in this particular rule, always overrules any `Include` condition. Be sure that your target Oracle server is not accidentally excluded.

7. Check that the appropriate inventory beacon can access the Oracle server:

    a. If necessary, identify the subnet that contains the undiscovered Oracle server, and find this subnet in the **All Subnets** page (**Inventory > Network Discovery > All Subnets**) to identify the inventory beacon managing the subnet. (If the subnet has not yet been assigned to an inventory beacon, fix that first by clicking the edit icon, and in the **Beacon name** column, selecting an inventory beacon from the drop-down list.)

    b. Validate that the correct 32-bit OLEDB drivers are installed on the appropriate inventory beacon (as described in Direct Collection of Oracle Inventory).

    c. On the appropriate inventory beacon, run the FlexNet Beacon software as administrator, and examine the local Password Manager.

    d. If (as is likely) the Oracle listener for the undiscovered database instance on the Oracle server is protected by an administrative password, ensure that the password (only, no account name required) is saved in the Password Manager.

    e. Ensure that the special account to access the database instance has been recorded in the secure Password Manager. For details, see Credentials for Direct Collection of Oracle Inventory, and for required database permissions see Appendix C: Oracle Tables and Views for Oracle Inventory Collection. Ensure that the credentials set up for the database instance are identically matched in the Password Manager.

8. If it appears that connection has been made and inventory collected, check for problems with file upload from the inventory beacon to the central application server. On the inventory beacon, examine `C:\Windows\Temp\ManageSoft\uploader.log` for any issues with uploads.

9. For inventory-specific errors, check the `$(CommonAppDataFolder)\Flexera Software\Compliance\Logging\InventoryRule\DeviceInventory.log` and `OracleDBInventory.log` files on the inventory beacon.

10. To enable richer logging on the inventory beacon, enable tracing by removing the hash character (#) from the following lines of the `etdp.trace` file present in the `%Program Files (x86)%\Flexera Software\Inventory Beacon\` folder:

```
+Inventory/Oracle
+Inventory/Oracle/SDK
+Inventory/Oracle/Query
+Inventory/Oracle/Query/Substitution
+Inventory/Oracle/Query/Execution
+Inventory/Oracle/Listener
+Inventory/Oracle/Listener/Detail
```

    Important: *After making changes to the `etdp.trace` file, you must use the Windows Service Controller on the inventory beacon to stop and restart the `beaconengine` service.*

11. Rerun the rule to collect Oracle discovery and inventory information, so that additional logging is created. See whether these richer logs assist in identifying the problem. Wait until after the next inventory import, and then look for both the discovered device (**All Discovered Devices**) and its Oracle instance (**Oracle Instances**). (Rather than waiting, an operator in an Administrator role can navigate to the **Reconcile** page (**Data Collection > Process Data > Reconcile**), ensure that the **Update inventory for reconciliation** option is selected, and click

Reconcile.)

12. If the problem persists, contact Flexera support with detailed information and log files.

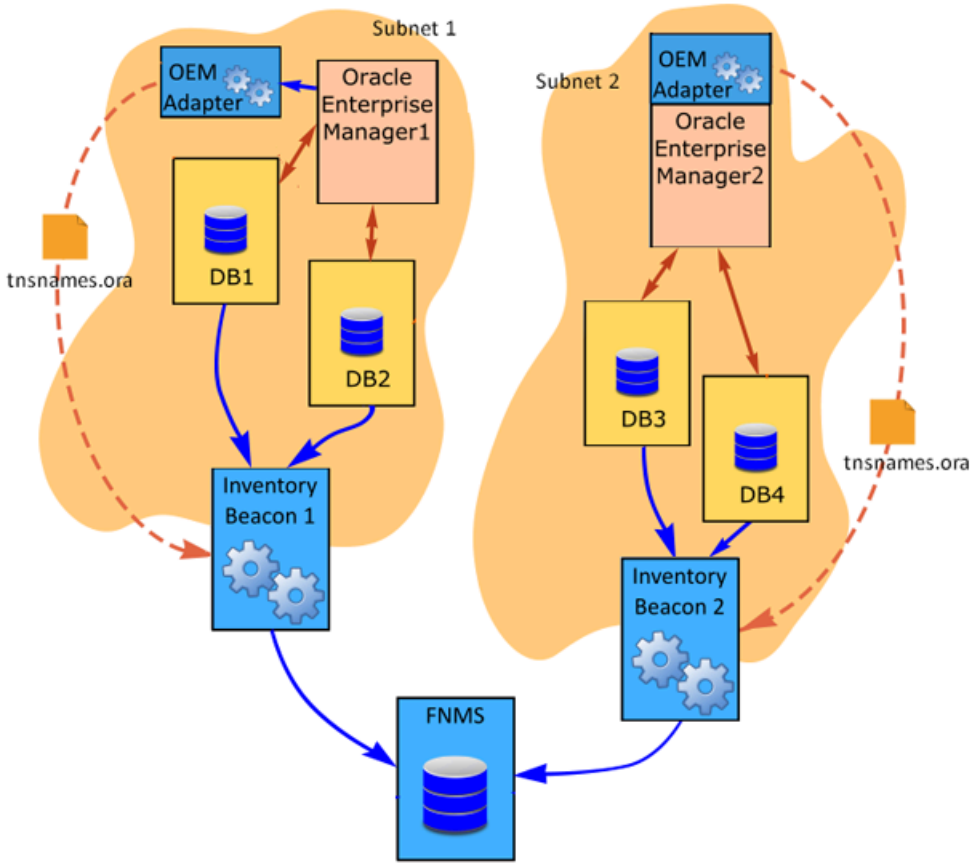
Using tnsnames Discovery with Direct Inventory

In this approach, the direct inventory part of the process is unchanged, but there is a different method to discover the Oracle database instances from which to gather inventory. This approach uses the content of a `tnsnames.ora` file to identify the target instances.

This Oracle-standard file can be created in any of three ways:

- You can copy it from your Oracle server and save it to the folder (identified below) on the inventory beacon where it will automatically be actioned.
- You can use the OEM adapter to create an `tnsnames.ora` file in the standard format, and save it to the correct folder on the appropriate inventory beacon that will perform the direct inventory collection. The OEM adapter is documented in *IT Asset Management Inventory Adapters and Connectors Reference*.
- You could create one manually (although this would represent a considerable maintenance burden, and is not a recommended path).

The following diagram shows an example scenario of the most interesting case, using the OEM adapter. If you are intervening manually, be sure to place your copy of the `tnsnames.ora` file in the correct folder as described below.



The above diagram shows Subnet 1 and Subnet 2. Subnet 1 is assigned to Inventory Beacon 1 and Subnet 2 is

assigned to Inventory Beacon 2. Each subnet contains:

- Two Oracle Database servers
- Oracle Enterprise Manager
- The OEM adapter (from Flexera), which may be co-located with Oracle Enterprise Manager or installed in another convenient location with fast network access to its related installation of Oracle Enterprise Manager (there is a 1:1 relationship, with one adapter required for each instance of Oracle Enterprise Manager).

Note that in this diagram, DB1 and DB2 are running versions of Oracle Database compatible with the same OLEDB driver, as discussed in Direct Collection of Oracle Inventory; similarly, DB3 and DB4 are running versions using the same OLEDB driver, although this may be a different driver than used in Subnet 1, since this subnet is already managed by a separate inventory beacon.

The following description assumes that all the prerequisites listed in Direct Collection of Oracle Inventory have been satisfied, including the installation of an appropriate version of the 32-bit OLEDB driver (one version per inventory beacon) for each target Oracle Database.

1. On each target Oracle server, set up the special "audit account" for collecting database inventory (see Credentials for Direct Collection of Oracle Inventory).
2. Record the credentials for the special audit account in the secure Password Manager on each applicable inventory beacon.
3. Create a rule that combines the use of `tnsnames.ora` for discovery with direct inventory. Go to the **Discovery and Inventory Rules** page (**Data Collection > IT Assets Inventory Tasks > Discovery and Inventory Rules**) and use these settings:
   - **Target**: Create a target to identify all Oracle servers in your network. You can use subnet name, IP address, or pattern matching on the device name to identify devices in the target definition.

     Tip: *The limitation on remote administration of the Oracle listener, disallowed from Oracle Database 12c, is not relevant when you are using discovery through* `tnsnames.ora`*. No administrative password for the listener is required with this discovery method.*
   - **Action**: Create an action that includes these settings:
     - For **Action type**, choose `Discovery and inventory`.
     - In the **Discovery of devices** section, select **TNSNames file for Oracle databases** (only).
     - Expand the **Oracle database environments** section, and ensure that **Discover Oracle database environments** is clear (not selected).
     - In the same section, select **Gather Oracle database environment inventory**. This is the switch to turn on direct inventory gathering for Oracle database instances.
   - **Schedule**: Specify the running schedule for this rule.

   This completes the initial configuration for using `tnsnames.ora` for discovery with direct inventory gathering for Oracle. The remainder of these steps cover normal operation.
4. Each instance of the OEM adapter connects to its Oracle Enterprise Manager, and collects the connection information for all Oracle Database servers managed by that instance of Oracle Enterprise Manager.

5. The OEM adapter formats the information into a `tnsnames.ora` file.
6. The OEM adapter connects to the inventory beacon it has registered, and saves the `tnsnames.ora` file in the special path *%CommonAppData%*`\Flexera Software\Repository\TNSNames`.

Tip: *If you are copying a `tnsnames.ora` file from your Oracle server, be sure to save it to this same path on your inventory beacon: `%CommonAppData%\Flexera Software\Repository\TNSNames`. Every `.ora` file in this folder is automatically used for targeting for direct inventory collection from the Oracle database instances that it identifies. This means that you can even mix files generated by the OEM adapter (always named `tnsnames.ora`) with files copied from another Oracle server, or one you have created manually. In these latter cases, be sure to rename the files you place manually (for example, `manualtnsnames.ora` or `tnsnamesServer33.ora`), because the next run of the OEM adapter overwrites the `tnsnames.ora` file in the TNSNames repository.*

7. By default, every 15 minutes each inventory beacon checks for any updates to its policy, which also transfers any changed discovery and inventory rules. (To adjust this download schedule, see *Inventory Settings Page > Beacon Settings Section* in the online help.) Each inventory beacon exercises only those rules that apply to its assigned subnet(s).
8. When the related schedule triggers an applicable rule:
   a. The inventory beacon reports to the central application server that the task is commencing. You can go to the **Discovery and Inventory Rules** page (**Data Collection > IT Assets Inventory Tasks > Discovery and Inventory Rules**) and click the rule name to view its status. Wait (while the remainder of the process takes place) until the **Status** field shows `Completed`. This process may take some time to complete and you may have to revisit or refresh the page from time to time.
   b. The inventory beacon opens each `.ora` file it finds in *%CommonAppData%*`\Flexera Software\Repository\TNSNames`, and converts each database instance there to a discovered device record. Each identified database instance is then tested against the current scope of the inventory beacon, where 'scope' means the intersection of its assigned subnet and the target declared as part of the rule it is currently running. Database instances that are both within the assigned subnet and within the current target are added to an *xxxx*`(InScope).disco` file. Database instances that fail either of these tests are added to an *xxxx*`(OutScope).disco` file.

   All this information is included in the two `.disco` files when they are uploaded to the central application server. For direct inventory gathering (where the discovery files may be larger than with the locally-installed FlexNet Inventory Agent), `.disco` files are compressed for upload.
9. The FlexNet Beacon engine (on each inventory beacon) uses the discovered services information to connect to each database instance, and collect Oracle Database inventory data. The results are saved into an `.ndi` inventory file (specific to Oracle inventory only) for each server.
10. Where Oracle inventory is collected, the FlexNet Beacon engine also executes the *software* scripts provided by Oracle Global Licensing and Advisory Services (GLAS) to gather software information about the Oracle Database installation. It is important to realize two things:
    - These are the same GLAS scripts, as amended from time to time, that are downloaded to the inventory beacon in the `InventorySettings.xml` file (which is saved in `Program Files (x86)\Flexera Software\Inventory Beacon\Remote Execution\Public\Inventory`). As always, they do not contribute to the collection of FlexNet inventory, but are used only for the preparation of an Oracle audit

report (available to operators who have appropriate access rights in the **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**), with more details available in the help for that page.)

- However, the Oracle GLAS scripts include two separate parts: SQL queries to gather information from Oracle database instance(s); and scripts to execute on the target server and gather the required hardware information. With this direct connection method of inventory collection, there is no file transfer of any kind to the target server, so that the GLAS hardware scripts *cannot be executed* on the Oracle server by this method. In contrast, the GLAS SQL queries *are* executed during the direct connection to each database instance, and the resulting GLAS software data is uploaded to the central application server; but in the normal course of events, the absence of hardware data is unlikely to satisfy an Oracle audit. Therefore, if your strategy is to use only the direct method of inventory collection, you should also plan another method to execute the GLAS hardware scripts on each Oracle server. Alternatively, reconsider the local installation of either the full FlexNet Inventory Agent, or the lightweight FlexNet Inventory Scanner; or re-evaluate the Zero-footprint method of inventory collection. All of these methods conveniently collect, upload, and create audit-ready packages for *all* the software and hardware data required for database licensing by Oracle GLAS, and without additional effort on your part.

11. Immediately after inventory gathering, the inventory beacon compresses the `.ndi` file(s) and `.disco` files, and uploads them to the central application server (or, if it is a member of a hierarchy of inventory beacons, uploads the data to its parent in the hierarchy, and the upload is repeated until the data reaches the application server). It is initially stored in the inventory database.
12. In due course, the inventory import (to the compliance database) and license reconciliation process runs (typically overnight, although an operator in a role granting the `Configure inventory data and reconcile` right can also trigger a full import and reconciliation). Progress is visible on the **System Tasks** page (**Data Collection > IT Assets Inventory Status > System Tasks**).
13. The **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**) lists the database inventory for all servers discovered and inventoried up to the time of the latest reconciliation calculation.

Troubleshooting Direct Inventory Using tnsnames.ora

In direct gathering of Oracle inventory, the FlexNet Beacon engine connects directly to Oracle databases using database port information. If you select the `tnsnames.ora` file as the method of discovery, the FlexNet Beacon engine uses the port information contained in that file.

To troubleshoot Oracle discovery and inventory using `tnsnames.ora`:

1. In the web interface for IT Asset Management, go to the **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**) to identify the database instances that have been discovered successfully. You may wish to focus on instances you expected to find that are not present on this page. (It is helpful to know the device name of an Oracle server you wish to examine, as you can use the name in searches in various lists.)
2. Check the **All Discovered Devices** page (**Inventory > Network Discovery > All Discovered Devices**) and validate that the Oracle server is listed there. If the `tnsnames.ora` file is resolved correctly and the data from it is uploaded, a discovered device record is created (after inventory import). (If the device is present, can you check whether the Oracle service was running at the time of last inventory gathering? The service must be running for Oracle discovery and inventory to succeed.) You can also use this listing to check for Oracle servers that are discovered, but reporting problems:

a. Click the filter icon above the right end of the list, and in the quick filter row, select **Oracle** = `Yes`.

b. Next to the filter icon, click the notification icon to reduce the list to only those Oracle servers reporting problems.

c. In the list, click the **Name** of a discovered device to open its properties, and select the **Status** tab.

d. Expand the **Oracle Database inventory** section, which displays any error message returned by the Oracle listener on this discovered device. Seek the help of your Oracle database administrator to resolve any issues.

Here are some common errors with suggested things to check:

Error Message	Notes
ORA-12170: TNS: Connect timeout occurred	• Ensure that no firewall is blocking connection to the database. • Ensure that the database is online and running on correct IP.
ORA-12541: TNS: no listener	• The database is shutdown. Ask your database administrator to start the database. • Ensure that no firewall is blocking connection to the database. • Ensure that the listener is not password protected. If it is, add the listener password to the Password Manager on the appropriate inventory beacon.
ORA-00942: table or view does not exist	Ensure that the table listed in the error message exists. Ask your database administrator to create the table if it does not exist.
ORA-12514: TNS: listener does not currently know of service requested in connect descriptor	TNS names file is not correct. This is common when a database is set to listen for only “SID” or only “Service_Name”.
ORA-01017: invalid username/password; logon denied	Indicates incorrect permissions. Ask your Oracle Database administrator to rerun the Flexera 'audit user' script (for details, see Credentials for Direct Collection of Oracle Inventory).
ORA-01034: ORACLE not available ORA-27101: shared memory realm does not exist Linux-x86_64 Error: 2: No such file or directory	These errors typically mean that a discovered Oracle database instance was not running at inventory time. Ask your Oracle Database administrator whether the instance can be made active for the next inventory gathering process.

Error Message	Notes
ORA-01033: ORACLE initialization or shutdown in progress	Oracle is stuck in a reboot process. Ask your database administrator to shutdown and restart the database.
ORA-12518: TNS: listener could not hand off client connection	Indicates a network problem. Ensure that the appropriate inventory beacon can access the Oracle server.
ORA-00604: error occurred at recursive SQL level 1	Indicates incorrect permissions. Ask your Oracle Database administrator to rerun the Flexera 'audit user' script (for details, see Credentials for Direct Collection of Oracle Inventory). ***Note:*** *On a UNIX-like platform, the tracker attempts to use* `setuid` *to impersonate the appropriate account to gather Oracle inventory. If you are using eTrust Access Control on this server, by default it does not permit this impersonation, and inventory gathering fails. The fix is to change the configuration of eTrust to include* `ndtrack` *in the* `LOGINAPPL` *class. For more information, see the eTrust Access Control Administration Guide (https://supportcontent.ca.com/cadocs/0/g007711e.pdf).*
ORA-00257: archiver error. Connect internal only, until freed	Indicates an archive error. Ask your Oracle Database administrator to check the `archiver.log` file for the error.

3. Go to the **Discovery and Inventory Rules** page (**Data Collection > IT Assets Inventory Tasks > Discovery and Inventory Rules**), select the **Rules** tab and:

   a. Find the appropriate rule, and check that its **Rule status** value shows `Enabled`. (A disabled rule never executes.)

   b. Click the title of the relevant rule and note the **Action** name and **Targets** name for this rule for use shortly. Also validate that the **Schedule** details are as expected.

   c. Look in **Current run** or **Last completed run** to see the number of database instances discovered, inventoried, skipped, and failed. Here are some notes to assist your analysis of these figures:

Column	Notes
Service discovered	The number of Oracle database instances discovered in the last execution of the rule (that is, the total number identified in all the `.ora` files saved on every inventory beacon executing the rule). This number should match (or possibly exceed) your expectations for Oracle servers within the targeted subnets, as identified in the `.ora` files. However, database discovery *cannot* succeed when: • The target device is powered off at the time of rule execution (to recover, re-run the rule when the server is operational). • The listener on the device is not operating when the rule executes (to recover, re-run the rule when the listener is running). • The listener returns an error, or does not identify any services (database instances). In addition, the `tnsnames.ora` file saved on a particular inventory beacon may identify database instances that are out of scope for that inventory beacon (either outside its assigned subnets, or outside the target[s] for the current rule). Any out-of-scope database instances are included in this count of **Service discovered**; but the out-of-scope instances are skipped for inventory gathering by this inventory beacon.
Inventory completed	The count of those instances from which database inventory has been collected. In an ideal world, this count of successful inventories may match the discovery result in the previous column. Differences should be tracked in the next two columns.

Column	Notes
Inventory skipped	The count of database instances where collection of database inventory was not attempted. Inventory gathering cannot succeed in the following cases, which are logged in `$(CommonAppDataFolder)\Flexera Software\Compliance\Logging\InventoryRule\`*`RuleID`*`\OracleDBInventory.log` on the inventory beacon: • A database instance listed in the `.ora` files on an inventory beacon falls outside the scope of the inventory beacon (where the 'scope' is the intersection of the subnets assigned to the inventory beacon and the target for the rule that the inventory beacon is executing). In this case the `OracleDBInventory.log` file includes an entry like `Skipped device '`*`deviceName`*`' because it was not within the authorised ranges for this beacon and rule`. In the case where the database instance is in the correct subnet assigned to the inventory beacon, but not listed in the targets incorporated in the currently-executing rule, the log entry is like `Skipped device '`*`deviceIP`*`' because it has no targets that are active (part of the current rule)`. • A database instance identified in an `.ora` file is not active at inventory time. In this case the `OracleDBInventory.log` file includes an entry like `No inventory gathered for device '`*`deviceIP`*`' because discovery did not find the service on the device`. (Direct inventory gathering cannot access any standby database instances to collect inventory, since these instances are also 'dark' to the listener. To collect inventory covering standby database instances, use a locally-installed tracker of version 12.4 or later, through one of Agent-Based Collection of Oracle Inventory, FlexNet Inventory Scanner Collection of Oracle Inventory, or Zero-footprint Collection of Oracle Inventory.)
Inventory failed	The count of database instances where inventory collection was attempted but failed. This may be because: • The credentials for the database instance were not configured in the Password Manager on the inventory beacon attempting to collect database inventory. • The instance was not running at inventory time. • The instance was running, but was not active (that is, was in the idle state) at inventory time. (Direct inventory gathering cannot access any standby database instances to collect inventory, since these instances are also 'dark' to the listener — as noted above.)

d. Look to the bottom of the same expanded panel, and click **Show/hide task status and history**. By default this lists the last five executions of the rule. You can expand one (such as the most recent) to see

all the inventory beacons that received the discovery and inventory rule (it still shows as `Scheduled` on inventory beacons that do not manage the appropriate subnets). You should find the intended inventory beacon in this list, where you can check the value in the **Status** column.

e. Use the **+** icon to expand more details for the chosen inventory beacon, and you see entries such as `Performing discovery` and `Gathering Oracle database inventory`. In the right-hand column are links for **Download log**. Download these, unzip the archive, and examine in a text file editor for clues to the problem.

4. Using the noted **Action** name for this rule, switch to the **Actions** tab, locate the relevant action, and click the editing (pencil) icon on the right-hand side to show the details. In the **Discovery of devices** section, ensure that **TNSNames file for Oracle databases** is selected.

5. Check the set-up of the inventory beacon responsible for directly gathering the Oracle inventory:

   a. If necessary, identify the subnet that contains the undiscovered Oracle server, and find this subnet in the **All Subnets** page (**Inventory > Network Discovery > All Subnets**) to identify the inventory beacon managing the subnet. (If the subnet has not yet been assigned to an inventory beacon, fix that first by clicking the edit icon, and in the **Beacon name** column, selecting an inventory beacon from the drop-down list.)

   b. Validate that the correct 32-bit OLEDB drivers are installed on the appropriate inventory beacon (as described in Direct Collection of Oracle Inventory).

   c. On the appropriate inventory beacon, run the FlexNet Beacon software as administrator, and examine the local Password Manager.

   d. Ensure that the special account to access the database instance has been recorded in the secure Password Manager. For details, see Credentials for Direct Collection of Oracle Inventory, and for required database permissions see Appendix C: Oracle Tables and Views for Oracle Inventory Collection. Ensure that the credentials set up for the database instance are identically matched in the Password Manager.

   e. Verify the existence of one or more `.ora` file(s) in the TNSNames repository folder (`$(CommonAppDataFolder)\Flexera Software\Repository\TNSNames`) on the inventory beacon. If none is present, trace back the methods that should be updating the file(s) here (as discussed in Using tnsnames Discovery with Direct Inventory).

6. If it appears that connection has been made and inventory collected, check for problems with file upload from the inventory beacon to the central application server. On the inventory beacon, examine `C:\Windows\Temp\ManageSoft\uploader.log` for any issues with uploads.

7. For inventory-specific errors, check the `$(CommonAppDataFolder)\Flexera Software\Compliance\Logging\InventoryRule\DeviceInventory.log` and `OracleDBInventory.log` files on the inventory beacon.

8. To enable richer logging on the inventory beacon, enable tracing by removing the hash character (#) from the following lines of the `etdp.trace` file present in the `%Program Files (x86)%\Flexera Software\Inventory Beacon\` folder:

```
+Inventory/Oracle
+Inventory/Oracle/SDK
+Inventory/Oracle/Query
```

```
+Inventory/Oracle/Query/Substitution
+Inventory/Oracle/Query/Execution
+Inventory/Oracle/Listener
+Inventory/Oracle/Listener/Detail
```

Important: *After making changes to the `etdp.trace` file, you must use the Windows Service Controller on the inventory beacon to stop and restart the `beaconengine` service.*

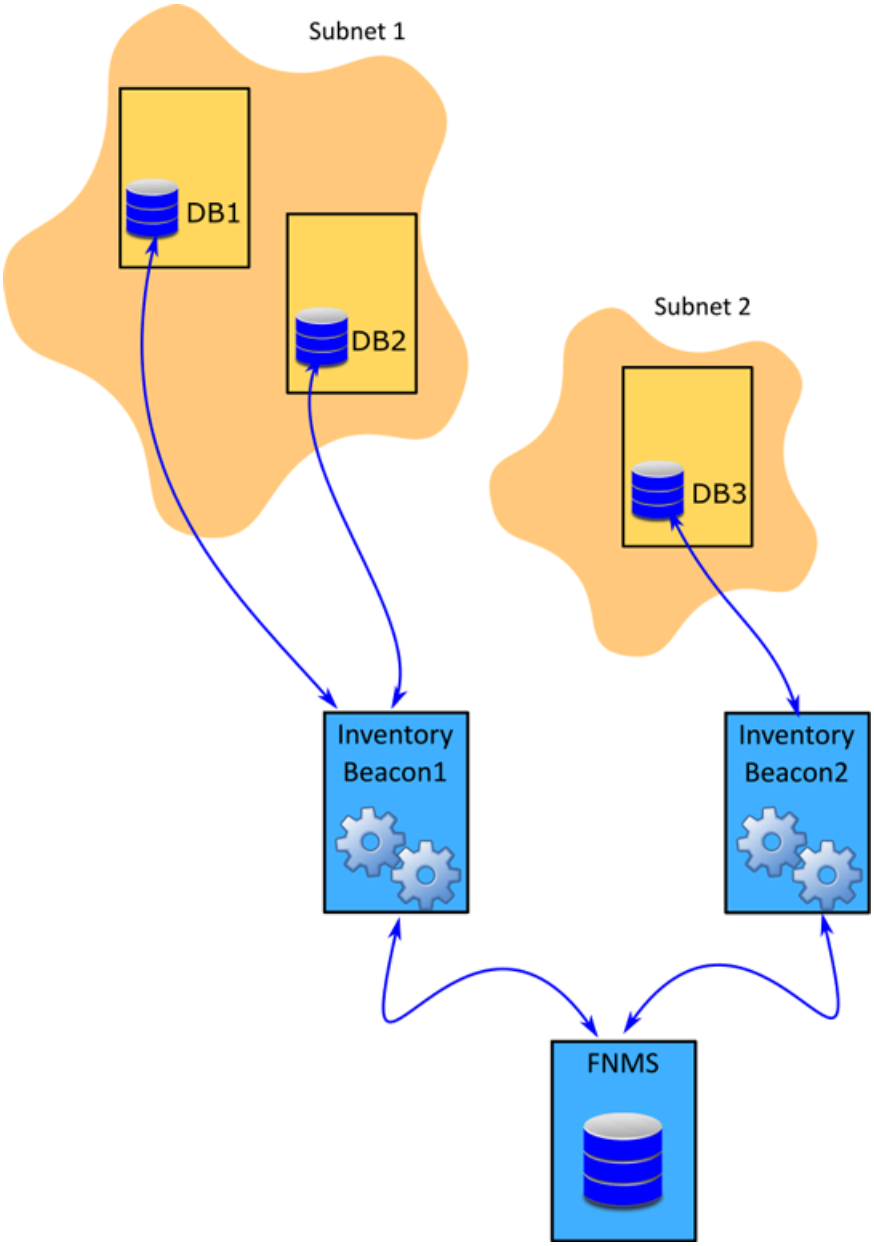
9. Rerun the rule to collect Oracle discovery and inventory information, so that additional logging is created. See whether these richer logs assist in identifying the problem. Wait until after the next inventory import, and then look for both the discovered device (**All Discovered Devices**) and its Oracle instance (**Oracle Instances**). (Rather than waiting, an operator in an Administrator role can navigate to the **Reconcile** page (**Data Collection > Process Data > Reconcile**), ensure that the **Update inventory for reconciliation** option is selected, and click **Reconcile**.)
10. If the problem persists, contact Flexera support with detailed information and log files.

Using Manual Discovery with Direct Inventory

In this approach, the direct inventory part of the process is unchanged, but the discovery device records are created manually ahead of the inventory-gathering process. You may resort to manual creation of discovered device records for testing purposes, or because a target device is temporarily unavailable.

Tip: *You may also use this method when you want to re-use discovered device records created in earlier passes of inventory gathering, so that you by-pass discovery for this occasion.*

The concepts in this scenario are straight-forward, and the direct inventory gathering itself is identical; but for consistency here is the scenario diagram, followed by the complete description.



The above diagram shows three database servers, two on Subnet1 and one on Subnet2. The Subnet1 is assigned to Inventory Beacon1 and Subnet2 is assigned to Inventory Beacon2. Note that in this diagram, DB1 and DB2 are running versions of Oracle Database compatible with the same OLEDB driver, as discussed in Direct Collection of Oracle Inventory; separately, DB3 may also be compatible, or may require a different version, which may be the reason that it is managed by a separate inventory beacon.

The following description assumes that all the prerequisites listed in Direct Collection of Oracle Inventory have been satisfied, including the installation of an appropriate version of the 32-bit OLEDB driver (one version per inventory beacon) for each target Oracle Database.

1. On each target Oracle server, set up the special "audit account" for collecting database inventory (see Credentials for Direct Collection of Oracle Inventory).
2. Record the credentials for the special audit account in the secure Password Manager on each applicable inventory beacon.

3. If a necessary discovered device record (one for each target Oracle server) does not yet exist, create it:

   a. For example, go to the **All Discovered Devices** page (**Inventory > Network Discovery > All Discovered Devices**) and click **Create a Discovered Device**.

   b. Complete the identification data on the **General** tab of the discovered device properties.

      The details you supply are used later to attempt connection to this device, so (as always) accuracy is paramount.

   c. Switch to the **Databases** tab, select **This device is running Oracle software**, and create details for the listener(s) on this device, and as children of each listener, the service(s) it manages. (For more information, see *Creating Listeners and Services* in the online help.) When data entry is complete, remember to click **Create** to save the discovered device record.

      Tip: *Because you are already identifying the services by name, you do not need to provide an administrative password for the listeners to request service names; and therefore, unlike the network scan method of discovery, this approach works with all versions of Oracle. (Of course, this service must accept the service user account and password that you have saved on the inventory beacon.)*

      Repeat this until all the required discovered device records are available.

4. Create a rule that combines the use existing discovered device records with direct inventory. Go to the **Discovery and Inventory Rules** page (**Data Collection > IT Assets Inventory Tasks > Discovery and Inventory Rules**) and use these settings:

   - **Target**: Create a target to identify all Oracle servers in your network. You can use subnet name, IP address, or pattern matching on the device name to identify devices in the target definition. In this case, be sure that all your manually-created (or previously existing) discovered devices are covered by the target.
   - **Action**: Create an action that includes these settings:
     - For **Action type**, choose `Discovery and inventory`.
     - In the **Discovery of devices** section, select **Use previously discovered devices**. (Although it is possible to mix discovery methods in the one action and rule, for simplicity the remainder of this description assumes that this is the only discovery selection you make).

       Tip: *A shortcut instead of these two settings is to choose* `Inventory only` *for* ***Action type****.*

     - Expand the **Oracle database environments** section, and ensure that **Discover Oracle database environments** is clear (not selected).
     - In the same section, select **Gather Oracle database environment inventory**. This is the switch to turn on direct inventory gathering for Oracle database instances.
   - **Schedule**: Specify the running schedule for this rule.

   This completes the initial configuration for using existing discovery records with direct inventory gathering for Oracle. The remainder of these steps cover normal operation.

5. Every 30 minutes, each inventory beacon asks the central application server whether there is any change in the discovered device records it holds. Where the records have been updated, the entire set is downloaded in a series of compressed `.disco.gz` files, so that each inventory beacon knows about all discovered devices, and

can operate on those that match both its assigned subnets and the targets in any rule being executed.

6. By default, every 15 minutes each inventory beacon checks for any updates to its policy, which also transfers any changed discovery and inventory rules. (To adjust this download schedule, see *Inventory Settings Page > Beacon Settings Section* in the online help.) Each inventory beacon exercises only those rules that apply to its assigned subnet(s).

7. When the related schedule triggers an applicable rule:

   a. The inventory beacon reports to the central application server that the task is commencing. You can go to the **Discovery and Inventory Rules** page (**Data Collection > IT Assets Inventory Tasks > Discovery and Inventory Rules**) and click the rule name to view its status. Wait (while the remainder of the process takes place) until the **Status** field shows `Completed`. This process may take some time to complete and you may have to revisit or refresh the page from time to time.

   b. Because this rule specifies re-use of existing discovery records (only), the inventory beacon selects from the existing, downloaded records of discovered devices only those that match both its assigned subnet(s) and the rule targets.

   (In due course, the discovered device records are updated with new status results, and are therefore uploaded again as usual when the processes are complete.)

8. For each discovered device record within the scope of the inventory beacon, the FlexNet Beacon engine uses each listener port and service name to connect to each database instance, and collect Oracle Database inventory data. The results are saved into an `.ndi` inventory file (specific to Oracle inventory only) for each server.

9. Where Oracle inventory is collected, the FlexNet Beacon engine also executes the *software* scripts provided by Oracle Global Licensing and Advisory Services (GLAS) to gather software information about the Oracle Database installation. It is important to realize two things:

   - These are the same GLAS scripts, as amended from time to time, that are downloaded to the inventory beacon in the `InventorySettings.xml` file (which is saved in `Program Files (x86)\Flexera Software\Inventory Beacon\Remote Execution\Public\Inventory`). As always, they do not contribute to the collection of FlexNet inventory, but are used only for the preparation of an Oracle audit report (available to operators who have appropriate access rights in the **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**), with more details available in the help for that page.)

   - However, the Oracle GLAS scripts include two separate parts: SQL queries to gather information from Oracle database instance(s); and scripts to execute on the target server and gather the required hardware information. With this direct connection method of inventory collection, there is no file transfer of any kind to the target server, so that the GLAS hardware scripts *cannot be executed* on the Oracle server by this method. In contrast, the GLAS SQL queries *are* executed during the direct connection to each database instance, and the resulting GLAS software data is uploaded to the central application server; but in the normal course of events, the absence of hardware data is unlikely to satisfy an Oracle audit. Therefore, if your strategy is to use only the direct method of inventory collection, you should also plan another method to execute the GLAS hardware scripts on each Oracle server. Alternatively, reconsider the local installation of either the full FlexNet Inventory Agent, or the lightweight FlexNet Inventory Scanner; or re-evaluate the Zero-footprint method of inventory collection. All of these methods conveniently collect, upload, and create audit-ready packages for *all* the software and hardware data required for database licensing by Oracle GLAS, and without additional effort on your part.

10. Immediately after inventory gathering, the inventory beacon compresses the `.ndi` file(s) and `.disco` files, and

uploads them to the central application server (or, if it is a member of a hierarchy of inventory beacons, uploads the data to its parent in the hierarchy, and the upload is repeated until the data reaches the application server). It is initially stored in the inventory database.

11. In due course, the inventory import (to the compliance database) and license reconciliation process runs (typically overnight, although an operator in a role granting the `Configure inventory data and reconcile` right can also trigger a full import and reconciliation). Progress is visible on the **System Tasks** page (**Data Collection > IT Assets Inventory Status > System Tasks**).
12. The **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**) lists the database inventory for all servers discovered and inventoried up to the time of the latest reconciliation calculation.

Troubleshooting Direct Inventory Using Manual Discovery Records

In direct gathering of Oracle inventory, the FlexNet Beacon engine connects directly to Oracle databases using database port information. If you use previously-existing records of discovered devices (potentially including records created manually), the FlexNet Beacon engine uses the port information contained in your existing records. Obviously, it is critical that these records contain correct data for each discovered device, so start by validating those properties against any discovered device that is not reporting correctly.

To troubleshoot Oracle discovery and inventory using existing/manually-created discovery device records:

1. In the web interface for IT Asset Management, go to the **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**) to identify the database instances that have been discovered successfully. You may wish to focus on instances you expected to find that are not present on this page. (It is helpful to know the device name of an Oracle server you wish to examine, as you can use the name in searches in various lists.)
2. Check the **All Discovered Devices** page (**Inventory > Network Discovery > All Discovered Devices**) and validate that the Oracle server is listed there. (If the device is present, can you check whether the Oracle service was running at the time of last inventory gathering? The service must be running for Oracle discovery and inventory to succeed.) You can also use this listing to check for Oracle servers that are discovered, but reporting problems:
    a. Click the filter icon above the right end of the list, and in the quick filter row, select **Oracle** = `Yes`.
    b. Next to the filter icon, click the notification icon to reduce the list to only those Oracle servers reporting problems.
    c. In the list, click the **Name** of a discovered device to open its properties, and select the **Status** tab.
    d. Expand the **Oracle Database inventory** section, which displays any error message returned by the Oracle listener on this discovered device. Seek the help of your Oracle database administrator to resolve any issues.

    Here are some common errors with suggested things to check:

Error Message	Notes
ORA-12170: TNS: Connect timeout occurred	• Ensure that no firewall is blocking connection to the database. • Ensure that the database is online and running on correct IP.

Error Message	Notes
ORA-12541: TNS: no listener	• The database is shutdown. Ask your database administrator to start the database. • Ensure that no firewall is blocking connection to the database. • Ensure that the listener is not password protected. If it is, add the listener password to the Password Manager on the appropriate inventory beacon.
ORA-00942: table or view does not exist	Ensure that the table listed in the error message exists. Ask your database administrator to create the table if it does not exist.
ORA-12514: TNS: listener does not currently know of service requested in connect descriptor	TNS names file is not correct. This is common when a database is set to listen for only “SID” or only “Service_Name”.
ORA-01017: invalid username/password; logon denied	Indicates incorrect permissions. Ask your Oracle Database administrator to rerun the Flexera 'audit user' script (for details, see Credentials for Direct Collection of Oracle Inventory).
ORA-01034: ORACLE not available ORA-27101: shared memory realm does not exist Linux-x86_64 Error: 2: No such file or directory	These errors typically mean that a discovered Oracle database instance was not running at inventory time. Ask your Oracle Database administrator whether the instance can be made active for the next inventory gathering process.
ORA-01033: ORACLE initialization or shutdown in progress	Oracle is stuck in a reboot process. Ask your database administrator to shutdown and restart the database.
ORA-12518: TNS: listener could not hand off client connection	Indicates a network problem. Ensure that the appropriate inventory beacon can access the Oracle server.

Error Message	Notes
ORA-00604: error occurred at recursive SQL level 1	Indicates incorrect permissions. Ask your Oracle Database administrator to rerun the Flexera 'audit user' script (for details, see Credentials for Direct Collection of Oracle Inventory).  ***Note:*** *On a UNIX-like platform, the tracker attempts to use* `setuid` *to impersonate the appropriate account to gather Oracle inventory. If you are using eTrust Access Control on this server, by default it does not permit this impersonation, and inventory gathering fails. The fix is to change the configuration of eTrust to include* `ndtrack` *in the* `LOGINAPPL` *class. For more information, see the eTrust Access Control Administration Guide (https://supportcontent.ca.com/cadocs/0/g007711e.pdf).*
ORA-00257: archiver error. Connect internal only, until freed	Indicates an archive error. Ask your Oracle Database administrator to check the `archiver.log` file for the error.

3. Go to the **Discovery and Inventory Rules** page (**Data Collection > IT Assets Inventory Tasks > Discovery and Inventory Rules**), select the **Rules** tab and:

   a. Find the appropriate rule, and check that its **Rule status** value shows `Enabled`. (A disabled rule never executes.)

   b. Click the title of the relevant rule and note the **Action** name and **Targets** name for this rule for use shortly. Also validate that the **Schedule** details are as expected.

   c. Look in **Current run** or **Last completed run** to see the number of database instances discovered, inventoried, skipped, and failed. Here are some notes to assist your analysis of these figures:

Column	Notes
Service discovered	Because you are using only existing discovered device records, no new discovery is attempted in the current rule, so that this figure is 0.
Inventory completed	The count of those instances from which database inventory has been collected.

Column	Notes
Inventory skipped	The count of database instances where collection of database inventory was not attempted. Inventory gathering cannot succeed in the following cases, which are logged in `$(CommonAppDataFolder)\Flexera Software\Compliance\Logging\InventoryRule\`*`RuleID`*`\OracleDBInventory.log` on the inventory beacon: • An existing discovered device record falls outside the scope of the inventory beacon (where the 'scope' is the intersection of the subnets assigned to the inventory beacon and the target for the rule that the inventory beacon is executing). In this case the `OracleDBInventory.log` file includes an entry like `Skipped device '`*`deviceName`*`' because it was not within the authorised ranges for this beacon and rule`. In the case where the database instance is in the correct subnet assigned to the inventory beacon, but not listed in the targets incorporated in the currently-executing rule, the log entry is like `Skipped device '`*`deviceIP`*`' because it has no targets that are active (part of the current rule).` • A database instance identified in a discovered device record is not active at inventory time. In this case the `OracleDBInventory.log` file includes an entry like `No inventory gathered for device '`*`deviceIP`*`' because discovery did not find the service on the device`. (Direct inventory gathering cannot access any standby database instances to collect inventory, since these instances are also 'dark' to the listener. To collect inventory covering standby database instances, use a locally-installed tracker of version 12.4 or later, through one of Agent-Based Collection of Oracle Inventory, FlexNet Inventory Scanner Collection of Oracle Inventory, or Zero-footprint Collection of Oracle Inventory.)
Inventory failed	The count of database instances where inventory collection was attempted but failed. This may be because: • The credentials for the database instance were not configured in the Password Manager on the inventory beacon attempting to collect database inventory. • The instance was not running at inventory time. • The instance was running, but was not active (that is, was in the idle state) at inventory time. (Direct inventory gathering cannot access any standby database instances to collect inventory, since these instances are also 'dark' to the listener — as noted above.)

d. Look to the bottom of the same expanded panel, and click **Show/hide task status and history**. By

default this lists the last five executions of the rule. You can expand one (such as the most recent) to see all the inventory beacons that received the discovery and inventory rule (it still shows as `Scheduled` on inventory beacons that do not manage the appropriate subnets). You should find the intended inventory beacon in this list, where you can check the value in the **Status** column.

e. Use the **+** icon to expand more details for the chosen inventory beacon, and you see entries such as `Performing discovery` and `Gathering Oracle database inventory`. In the right-hand column are links for **Download log**. Download these, unzip the archive, and examine in a text file editor for clues to the problem.

4. Using the noted **Action** name for this rule, switch to the **Actions** tab, locate the relevant action, and click the editing (pencil) icon on the right-hand side to show the details. In the **Discovery of devices** section, ensure that **Use previously discovered devices** is selected.

5. Check the set-up of the inventory beacon responsible for directly gathering the Oracle inventory:

   a. If necessary, identify the subnet that contains the undiscovered Oracle server, and find this subnet in the **All Subnets** page (**Inventory > Network Discovery > All Subnets**) to identify the inventory beacon managing the subnet. (If the subnet has not yet been assigned to an inventory beacon, fix that first by clicking the edit icon, and in the **Beacon name** column, selecting an inventory beacon from the drop-down list.)

   b. Validate that the correct 32-bit OLEDB drivers are installed on the appropriate inventory beacon (as described in Direct Collection of Oracle Inventory).

   c. On the appropriate inventory beacon, run the FlexNet Beacon software as administrator, and examine the local Password Manager.

   d. Ensure that the special account to access the database instance has been recorded in the secure Password Manager. For details, see Credentials for Direct Collection of Oracle Inventory, and for required database permissions see Appendix C: Oracle Tables and Views for Oracle Inventory Collection. Ensure that the credentials set up for the database instance are identically matched in the Password Manager.

6. If it appears that connection has been made and inventory collected, check for problems with file upload from the inventory beacon to the central application server. On the inventory beacon, examine `C:\Windows\Temp\ManageSoft\uploader.log` for any issues with uploads.

7. For inventory-specific errors, check the `$(CommonAppDataFolder)\Flexera Software\Compliance\Logging\InventoryRule\DeviceInventory.log` and `OracleDBInventory.log` files on the inventory beacon.

8. To enable richer logging on the inventory beacon, enable tracing by removing the hash character (#) from the following lines of the `etdp.trace` file present in the `%Program Files (x86)%\Flexera Software\Inventory Beacon\` folder:

```
+Inventory/Oracle
+Inventory/Oracle/SDK
+Inventory/Oracle/Query
+Inventory/Oracle/Query/Substitution
+Inventory/Oracle/Query/Execution
+Inventory/Oracle/Listener
+Inventory/Oracle/Listener/Detail
```

Important: *After making changes to the* `etdp.trace` *file, you must use the Windows Service Controller on the inventory beacon to stop and restart the* `beaconengine` *service.*

9. Rerun the rule to collect Oracle discovery and inventory information, so that additional logging is created. See whether these richer logs assist in identifying the problem. Wait until after the next inventory import, and then look for both the discovered device (**All Discovered Devices**) and its Oracle instance (**Oracle Instances**). (Rather than waiting, an operator in an Administrator role can navigate to the **Reconcile** page (**Data Collection > Process Data > Reconcile**), ensure that the **Update inventory for reconciliation** option is selected, and click **Reconcile**.)
10. If the problem persists, contact Flexera support with detailed information and log files.

Appendix A: Pseudo-SKUs for Oracle Bundles

SKUs are key to many aspects of automation in IT Asset Management, including the ability to propose meaningful license structures for an application.

Most SKUs are determined by the publisher of the software, and you may have access to their SKUs through your purchase order records.

However, the following SKUs are not in that group. These are codes created by Flexera (rather than the publisher of the software) as a convenience for identifying certain software bundles. They are of no relevance outside IT Asset Management, for which they have been created. In the cases below, they identify software bundles published by Oracle. We make these pseudo-SKUs available as an aid for data input, as a shorthand way of identifying the bundles.

SKU	Description	Product family
FLX-ORCL-100067	WebLogic Server Enterprise Edition Named User Plus Perpetual License	WebLogic
FLX-ORCL-100071	WebLogic Server Standard Edition Named User Plus Perpetual License	WebLogic
FLX-ORCL-100075	WebLogic Suite Named User Plus Perpetual License	WebLogic
FLX-ORCL-100068	WebLogic Server Enterprise Edition Processor Perpetual License	WebLogic
FLX-ORCL-100072	WebLogic Server Standard Edition Processor Perpetual License	WebLogic
FLX-ORCL-100076	WebLogic Suite Processor Perpetual License	WebLogic
FLX-ORCL-100108	Business Intelligence Standard Edition Named User Plus Perpetual License	Business Intelligence

SKU	Description	Product family
FLX-ORCL-100109	Business Intelligence Standard Edition Processor Perpetual License	Business Intelligence
FLX-ORCL-100111	Business Intelligence Suite Enterprise Edition Plus Named User Plus Perpetual License	Business Intelligence
FLX-ORCL-100113	Business Intelligence Suite Enterprise Edition Plus Upgrade Only Named User Plus Perpetual License	Business Intelligence

Appendix B: Components for Oracle Inventory Collection

Each of the different inventory collection methods described in this chapter may involve different software components within IT Asset Management. The number and types of software components involved in Oracle inventory collection depends on the selected inventory collection method. This section provides a brief overview of each of the software components involved in inventory collection.

FlexNet Inventory Agent

A software agent that may be installed on different kinds of computers to collect software and hardware inventory information for the device on which it is installed. Its core components are also installed on each inventory beacon, allowing Zero-footprint collection of inventory. The essential component, called the tracker, is a 32-bit executable that transforms the inventory information into an XML formatted (`.ndi`) file and uploads it to an inventory beacon. All locally-installed FlexNet Inventory Agents collect inventory according to the global agent inventory collection schedule defined in the web interface for IT Asset Management (go to the **Inventory Settings** page (**Data Collection > IT Assets Inventory Tasks > Inventory Settings**)).

Where the FlexNet Inventory Agent has been locally installed on the target device, it collects discovery information, hardware and general software inventory along with specifically Oracle inventory, and uploads collected data to an inventory beacon. However, it is significant to note that the FlexNet Inventory Agent is autonomous, and may make its own choice of inventory beacon to which it uploads, which may not be the same one assigned to manage the subnet containing the inventory device (for example, if another inventory beacon responds faster, or based on other logic).

FlexNet Inventory Scanner

The FlexNet Inventory Scanner provides an alternative to the full FlexNet Inventory Agent that is somewhat lighter to deploy and manage. It includes only the FlexNet inventory core components, so that it lacks functionality such as usage tracking, or upload retries after transient network failures. However, it packages these components as self-extracting executables, known as:

- On Microsoft Windows, `FlexeraInventoryScanner.exe`
- On UNIX-like platforms, `ndtrack.sh`.

As part of the functionality available in this case, the operational executables are removed after each execution, although the FlexNet Inventory Scanner package itself is not automatically removed.

In this configuration, and given appropriate command lines, the FlexNet inventory core components are capable of collecting hardware and software inventory from a target device, and uploading it to a location specified in the command line. For more details about the FlexNet Inventory Scanner, see *Gathering FlexNet Inventory*.

FlexNet Beacon

You can install FlexNet Beacon on supported versions of Microsoft Windows Server to make it operate as an inventory beacon, where it acts as a collection point for inventory and business information within the enterprise (the supported versions of Windows Server are listed in the *IT Asset Management System Requirements and Compatibility* PDF for each version of IT Asset Management). One or more network subnets are assigned to an inventory beacon, and the inventory beacon applies rules (potentially including device adoption) within the assigned subnets. The rules may include collect data remotely through Zero-footprint inventory gathering. When the inventory beacon collects Zero-footprint inventory, it follows the schedule set in the applicable rule (in contrast to the locally-installed FlexNet Inventory Agent, which, as described above, does not).

Any inventory beacon also accepts uploads from any installed instance of FlexNet Inventory Agent or FlexNet Inventory Scanner that contacts it. Uploads are not filtered by the assigned subnets.

Each inventory beacon then uploads collected data to its parent in a hierarchy. The parent may be another inventory beacon acting as a concentrator, or it may be the central application server, where all uploads eventually arrive. Here it is processed to update the database records representing the assets, both software and hardware, located within the enterprise's computing environment. For more information about inventory beacons, see *What is an Inventory Beacon?* in the online help.

Oracle Enterprise Manager Adapter

The Oracle Enterprise Manager (OEM) adapter is a software component that provides an alternative or additional method of discovering Oracle databases in your computing estate. IT Asset Management requires discovery information before collecting inventory. Discovery (for Oracle databases) includes the collection of connection details to allow inventory gathering. The OEM adapter gathers the database connection details for all the databases managed by an instance of Oracle Enterprise Manager and formats them into a standard `tnsnames.ora` file. Each installation of OEM adapter can connect to only one instance of Oracle Enterprise Manager, saving the resulting `tnsnames.ora` file to a fixed location on the inventory beacon (`$(CommonAppDataFolder)\Flexera Software\Repository\TNSNames`). Therefore, you need an inventory beacon and one installation of the OEM adapter for each instance of Oracle Enterprise Manager.

The `tnsnames.ora` file generated by the OEM adapter is used by the FlexNet Beacon engine as one possible discovery method for use with direct inventory gathering method to collect Oracle Database inventory. If you decide to use direct inventory gathering but not to deploy the OEM adapter, you can also copy the standard `tnsnames.ora` manually from the Oracle server to the TNSNames repository on the appropriate inventory beacon.

tnsnames.ora

The `tnsnames.ora` file is an Oracle-standard configuration file that contains connection descriptors for the services running on an Oracle Database. The connection descriptors contain the host name, protocol, service name, and port for each of the services running on an Oracle Database. The `tnsnames.ora` file may be created in at least two ways:

- By default, each time the services configuration is updated, Oracle automatically saves updated connection information in a `tnsnames.ora` file stored on the Oracle server. If this file is copied to the appropriate inventory beacon and saved in `$(CommonAppDataFolder)\Flexera Software\Repository\TNSNames`, the inventory

beacon can use this standard Oracle file as its discovery process before taking direct inventory of the Oracle database instances identified in the file.

- A separate `tnsnames.ora` file may be generated by the OEM adapter, as described above, for use in direct inventory gathering.

Use of one or more `tnsnames.ora` files must be configured in a discovery and inventory rule in the web interface of IT Asset Management (for details, see Using tnsnames Discovery with Direct Inventory).

Discovery and Inventory Rules

You can use a discovery and inventory rule to configure discovery and inventory processes that you choose to run from an inventory beacon. You also need a rule if you choose to install FlexNet Inventory Agent automatically on each of the discovered Oracle servers (called "adoption" of the device). A rule is a combination of one or more targets, an action, and a schedule. The action properties determine the actions to perform and the targets' properties determine the devices on which to perform the action. The target adoption settings can be used to adopt the identified devices (that is, to install FlexNet Inventory Agent locally on the devices). When a discovery and inventory rule executes, it identifies devices based on the target definition(s) and performs the action specified in the rule on those devices. For more information about discovery and inventory rules, see *Discovery and Inventory Rules* in the online help.

Appendix C: Oracle Tables and Views for Oracle Inventory Collection

In general, Oracle recommends collecting database data using any user account that has the following privileges:

- `CREATE SESSION`
- `SELECT ANY TABLE`
- For database version 9.1 and higher: `SELECT ANY DICTIONARY`.

Suggested system-supplied user accounts depend on whether Oracle Database Vault is in use:

- When Oracle Database Vault is *not* in use: `SYS` or `SYSTEM` are good choices
- When Oracle Database Vault is in use:
  - Use `PARTICIPANT` or `OWNER` authorization on 'Oracle Data Dictionary' realm
  - Use `DV_SECANALYST` role for querying views supplied by Oracle Database Vault.

If you do not want to collect inventory data as either the `SYS` or the `SYSDBA` user, you may need to create a user account to collect Oracle inventory (for details, see the *Credentials for...* topic for your preferred method of Oracle inventory collection, earlier in this chapter). The database user must have at least read-only access to the following tables and views on each inventoried database instance.

Important: *The following list may not be valid for all versions of Oracle Database. For example, several of the following tables/views do not exist in versions prior to Oracle Database 11. If you attempt to grant privileges to a table/view that does not exist in your version of Oracle Database, the result will contain an error message, which can be ignored.*

Tables/views accessible for all users (PUBLIC)

- GV$INSTANCE
- GV$PARAMETER
- V$ARCHIVE_DEST_STATUS
- V$BLOCK_CHANGE_TRACKING
- V$CONTAINERS
- V$DATABASE
- V$DATAGUARD_CONFIG
- V$INSTANCE
- V$LICENSE
- V$OPTION
- V$PARAMETER
- V$SESSION.

Tables/views accessible for SYS, or with SYSDBA privileges

- DBA_ADVISOR_TASKS
- DBA_AUDIT_TRAIL
- DBA_AWS
- DBA_CPU_USAGE_STATISTICS
- DBA_CUBES
- DBA_DV_REALM
- DBA_ENCRYPTED_COLUMNS
- DBA_FEATURE_USAGE_STATISTICS
- DBA_FLASHBACK_ARCHIVE_TABLES
- DBA_FLASHBACK_ARCHIVE_TS
- DBA_FLASHBACK_ARCHIVE
- DBA_INDEXES
- DBA_LOB_PARTITIONS
- DBA_LOB_SUBPARTITIONS
- DBA_LOBS

- DBA_MINING_MODELS
- DBA_OBJECT_TABLES
- DBA_OBJECTS
- DBA_RECYCLEBIN
- DBA_REGISTRY
- DBA_SEGMENTS
- DBA_SQL_PROFILES
- DBA_SQLSET_REFERENCES
- DBA_SQLSET
- DBA_TAB_PARTITIONS
- DBA_TAB_SUBPARTITIONS
- DBA_TABLESPACES
- DBA_TABLES
- DBA_USERS
- MODEL$
- REGISTRY$HISTORY
- ROLE_SYS_PRIVS
- USER_ROLE_PRIVS
- USER_SYS_PRIVS.

In addition, the following procedure needs EXECUTE privileges:

- UTL_INADDR.

Tables/views accessible in the {OEMOWNER} schema, typically using the account SYSMAN

- MGMT$DB_DBNINSTANCEINFO
- MGMT$TARGET
- MGMT$TARGET_MEMBERS
- MGMT$TARGET_PROPERTIES
- MGMT_ADMIN_LICENSES
- MGMT_FU_LICENSE_MAP
- MGMT_FU_REGISTRATIONS

- MGMT_FU_STATISTICS
- MGMT_INV_COMPONENT
- MGMT_INV_CONTAINER
- MGMT_LICENSE_CONFIRMATION
- MGMT_LICENSE_DEFINITIONS
- MGMT_LICENSED_TARGETS
- MGMT_LICENSES
- MGMT_TARGET_TYPES
- MGMT_TARGETS
- MGMT_VERSIONS.

Tables/views in special schemata, typically using the SYS account (and addressing the correct schema name)

- 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
- CONTENT.ODM_DOCUMENT
- DMSYS.DM$MODEL
- DMSYS.DM$OBJECT
- DMSYS.DM$P_MODEL
- GSMADMIN_INTERNAL.SHA_DATABASES
- LBACSYS.LBAC$POLT
- LBACSYS.OLS$POLT
- MDSYS.SDO_GEOM_METADATA_TABLE
- ODM.ODM_MINING_MODEL
- ODM.ODM_RECORD

- `OLAPSYS.DBA$OLAP_CUBES`.

Tip: *In general, Flexera does not recommend creating a special user account just for collecting inventory, because of the implicit high maintenance effort. This is because a user can only be granted access to tables/views that exist* ***at the moment of granting rights****. This means:*

- *Continuous efforts/monitoring to ensure that the special "scanning user" account has been created on all database instances*
- *Repeated effort to keep the grants accurate on all existing database instances*
- *Repeated investigations into whether a missing table access is due to a non-existent table or due to missing grants of rights to the scanning user.*

Remember: *If you choose to create a custom user for database inventory collection, the grant of access to tables/views should be repeated regularly to cope with any database changes that may have occurred since the previous grant. This is particularly the case, for example, around any database patches or upgrades.*

Appendix D: Deploying Inventory Tools to a Shared Location

In agent-based inventory collection, all copies of the FlexNet Inventory Agent locally installed on target inventory devices run according to a single global inventory collection schedule. This single schedule may not provide sufficient flexibility for your environment.

It is possible to deploy the key executables to a shared location where they can be accessed by target inventory devices, and then to configure different devices to trigger inventory collection on different schedules. These key executables are collectively called the FlexNet inventory core components.

If you decide to do this, it is best practice to ensure that only one Oracle server accesses the shared deployment of the FlexNet inventory core components at a time. As a result, the total number of shared deployments that may be required depends on the network structure and access constraints. For example, if an Oracle server is not allowed to access the shared deployment of the FlexNet inventory core components present in a different subnet, you may have to deploy another shared deployment in the subnet of the target Oracle server.

For more information about use of the FlexNet inventory core components, see *Core deployment: Details* in *Gathering FlexNet Inventory* PDF. In particular, for deployment instructions, see the *Core deployment: Implementation* topic in that chapter.

Appendix E: Oracle Standard Users Exempted From Consuming Licenses

When calculating consumption of license entitlements for Oracle database instances, IT Asset Management automatically exempts the following standard named Oracle user accounts so that they do not consume entitlements:

- ABM
- AD
- AD_MONITOR
- AHL
- AHM
- AK
- ALR
- AME
- AMF
- AMS
- AMV
- AMW
- AN
- ANONYMOUS
- AP
- APPLSYS
- APPLSYSPUB
- APPQOSSYS
- APPS
- APPS_MRC
- AR
- AS
- ASF
- ASG
- ASL
- ASN
- ASO
- ASP
- GCS
- GHR
- GL
- GMA
- GMD
- GME
- GMF
- GMI
- GML
- GMO
- GMP
- GMS
- GMW
- GNI
- GR
- HCA
- HCC
- HCN
- HCP
- HCT
- HR
- HRI
- HXC
- HXT
- IA
- IAM
- IBA
- IBC
- ORASSO_PUBLIC
- ORDPLUGINS
- ORDSYS
- OSE$HTTP$ADMIN
- OSM
- OTA
- OUC
- OUTLN
- OJVMSYS
- OWA_MGR
- OWAPUB
- OWF_MGR
- OZF
- OZP
- OZS
- PA
- PAY
- PBR
- PER
- PERFSTAT
- PFT
- PJI
- PJM
- PM
- PMI
- PN
- PO
- POA

- AST
- AU
- AURORAJISUTILITY$
- AURORAORBUNAUTHENTICATED
- AX
- AZ
- BEN
- BIC
- BIE
- BIL
- BIM
- BIN
- BIS
- BIV
- BIX
- BIY
- BLC
- BLEWIS
- BNE
- BOM
- BSC
- CCT
- CDOUGLAS
- CDR
- CE
- CHV
- CLA
- CLE
- IBE
- IBP
- IBT
- IBU
- IBW
- IBY
- ICX
- IEB
- IEC
- IEM
- IEO
- IEP
- IES
- IET
- IEU
- IEV
- IEX
- IGC
- IGF
- IGI
- IGS
- IGW
- IMC
- IMT
- INTERNET_APPSERVER_REGISTRY
- INV
- IP
- IPA
- POM
- PON
- PORTAL
- PORTAL_APP
- PORTAL_DEMO
- PORTAL_PUBLIC
- PORTAL30
- PORTAL30_DEMO
- PORTAL30_PUBLIC
- PORTAL30_SSO
- PORTAL30_SSO_PS
- PORTAL30_SSO_PUBLIC
- POS
- PQH
- PQP
- PRP
- PSA
- PSB
- PSP
- PSR
- PTE
- PTG
- PTX
- PV
- QA
- QOT
- QP
- QRM

- CLJ
- CLKRT
- CLL
- CLN
- CN
- CPGC
- CRP
- CS
- CSC
- CSD
- CSE
- CSF
- CSI
- CSL
- CSM
- CSN
- CSP
- CSR
- CSS
- CST
- CTB
- CTSYS
- CUA
- CUC
- CUE
- CUF
- CUG
- CUI
- IPD
- IPM
- IRC
- ISC
- ISX
- IT_PERF
- ITA
- ITG
- IZU
- JA
- JE
- JG
- JL
- JMF
- JTF
- JTI
- JTM
- JTR
- JTS
- JUNK_PS
- KWALKER
- LNS
- MDDATA
- MDSYS
- ME
- MFG
- MGMT_VIEW
- MIA
- QS
- QS_ADM
- QS_CB
- QS_CBADM
- QS_CS
- QS_ES
- QS_OS
- QS_WS
- RCM
- REPADMIN
- RG
- RHX
- RLA
- RLM
- RMG
- RRC
- RRS
- SCOTT
- SH
- SHT
- SI_INFORMTN_SCHEMA
- SPIERSON
- SQLAP
- SQLGL
- SSOSDK
- SSP
- SYS
- SYSADMIN

- CUN
- CUP
- CUR
- CUS
- CZ
- DBSNMP
- DCM
- DDD
- DEM01
- DISCOVERER5
- DIP
- DMSYS
- DNA
- DOM
- DSGATEWAY
- DSSYS
- DT
- DUMMY_GMO
- EAA
- EAM
- EC
- ECX
- EDR
- EGO
- EMS
- ENG
- ENI
- EVM
- MIV
- MQA
- MRP
- MSC
- MSD
- MSO
- MSR
- MST
- MWA
- OAM
- OCA
- ODM
- ODM_MTR
- ODQ
- ODS
- ODSCOMMON
- OE
- OFA
- OKB
- OKC
- OKC_REP_TXT_INDEX_OPTIMIZE
- OKC_REP_TXT_INDEX_SYNC
- OKE
- OKI
- OKL
- OKO
- OKP
- OKR
- SYSDG
- SYSKM
- SYSTEM
- TEST
- TRACESVR
- UDDISYS
- VEA
- VEH
- WCRSYS
- WFADMIN
- WH
- WIP
- WIRELESS
- WK_PROXY
- WK_TEST
- WKPROXY
- WKSYS
- WMA
- WMS
- WMSYS
- WPS
- WSH
- WSM
- XDB
- XDO
- XDP
- XLA
- XLE

- EXFSYS
- FA
- FEM
- FF
- FII
- FLM
- FND
- FPA
- FPT
- FRM
- FTE
- FTP
- FUN
- FV
- OKS
- OKT
- OKX
- OLAPDBA
- OLAPSVR
- OLAPSYS
- ONT
- OPI
- ORACLE_OCM
- ORAOCA_PUBLIC
- ORASSO
- ORASSO_DS
- ORASSO_PA
- ORASSO_PS
- XNA
- XNB
- XNC
- XNI
- XNM
- XNP
- XNS
- XNT
- XTR
- ZFA
- ZPB
- ZSA
- ZX

Appendix F: Features Enabled in FlexNet Manager for Datacenters

The basic license for IT Asset Management includes some functionality related to Oracle. With only the basic license, you can:

- Discover Oracle databases (but not collect inventory from them)
- Collect installer evidence data (basic software inventory of other products).

However, to enable collection of detailed Oracle inventory, you must have licensed the FlexNet Manager for Datacenters product for IT Asset Management. This additional license adds many more features:

- License management and compliance for Oracle Processor, Oracle Named User Plus, Oracle Application User, and other Oracle-specific licenses
- Detailed Oracle inventory of database options and Oracle E-Business Suite
- Inventory of specialized systems like Exadata, SuperCluster, and so on
- Inventory and license reconciliation for the Oracle license types stated above.

To check whether you have licensed FlexNet Manager for Datacenters:

1. Go to the **IT Assets License** page (**Administration > IT Asset Management Settings > IT Assets License**).
2. Scroll down to the section on **Licensed products**, and identify the card for FlexNet Manager for Datacenters.
3. If the card is missing or grayed out, detailed Oracle inventory cannot be processed, and you should discuss your requirements with your Flexera consultant.

Appendix G: Version Identification for Inventory and GLAS scripts

Collection of Oracle inventory by the FlexNet Inventory Agent (or its core inventory tracker component) relies on the correct deployment of the `InventorySettings.xml` file. The required location for this file varies across the inventory collection methods, and is described in the preceding sections that cover each of the methods.

If you are using your own (third party) method of deploying [components of] FlexNet Inventory Agent, you have also assumed responsibility for keeping the related copies of the `InventorySettings.xml` file up to date. Since updates to the `InventorySettings.xml` file are delivered as part of updates to the Application Recognition Library, your manual process may require you to keep track of the most recently downloaded version of `InventorySettings.xml`, and to manually identify when changes occur.

Tip: *Not every update to `InventorySettings.xml` brings a change to the scripts provided by the Oracle Global Licensing and Advisory Services (GLAS). `InventorySettings.xml` contains other specialized inventory functionality unrelated to Oracle Database, and each change naturally triggers a version change in `InventorySettings.xml`, whether the change relates to Oracle inventory or not.*

To accommodate this need, the process below allows you to identify versions of `InventorySettings.xml`.

Note: *Scripts in `InventorySettings.xml` are kept backward compatible, such that earlier versions of the inventory component(s) are able to use later versions of `InventorySettings.xml` without danger. Rarely, an enhancement in the Oracle GLAS scripts may require a matching enhancement to the functionality within FlexNet Inventory Agent (or its core components); or even to FlexNet Beacon, the code entity on each inventory beacon. In these cases, legacy versions of the FlexNet Inventory Agent or FlexNet Beacon deployed in your environment simply ignore the new (and for them, unrecognized) functionality, and use earlier, compatible functionality. Two examples (one for each case) of this are:*

- *Affecting the installed FlexNet Inventory Agent, an example case is the collection of inventory from standby Oracle database instances, which requires version 12.4 of the tracker (shipped with IT Asset Management 2017 R3) or later. Versions of the tracker prior to that can run later GLAS scripts, but cannot collect inventory from standby database instances. To ensure the availability of this functionality, check that the versions of FlexNet Inventory Agent installed on your Oracle servers is version 12.4 or later. To assist with cases like this one, the process below includes identifying the running version of an installed FlexNet Inventory Agent.*
- *Affecting the inventory beacon, an example case is the automatic updates of `InventorySettings.xml` for use on the inventory beacon (as distinct from the update process that refreshes the same file for installed copies of the FlexNet Inventory Agent, which is a quite separate process and was in place far earlier than the updates for*

inventory gathering driven by the inventory beacon). This functionality to update `InventorySettings.xml` for use by the inventory beacon was introduced with IT Asset Management version 2016 R1, prior to which the same file was installed as part of inventory beacon installation, and could be updated only by a replacement installation of FlexNet Beacon on the same server. This means that any legacy inventory beacons in your enterprise that have not been updated since before version 2016 R1 are locked into the then-current version of `InventorySettings.xml`. The only remedy is to allow the automatic update of the FlexNet Beacon code (or, in cases where you are manually managing disconnected inventory beacons, to manually update the installed version of FlexNet Beacon). To assist in troubleshooting the versions of inventory beacons in use, the process below includes identifying the running version of FlexNet Beacon.

Normally, it is not necessary for you to go further and track the version of the Oracle GLAS scripts, since these are included in an updated `InventorySettings.xml` file as soon as they are received from Oracle GLAS. However, if you have some special reason for checking the version of the GLAS scripts in use, the following points are relevant:

- There is no direct mapping from the version of `InventorySettings.xml` to the embedded version of the GLAS scripts.
- Since the Oracle queries are encrypted within `InventorySettings.xml`, you cannot inspect the version of the GLAS scripts by examining the delivery file.
- Operations within your enterprise may result in multiple versions of `InventorySettings.xml`, and more particularly multiple versions of the GLAS scripts, being in use simultaneously. For example, if you have a mixture of installations, some where FlexNet Inventory Agent is updated automatically by policy (reporting to an inventory beacon), and others where you rely on manual updates to `InventorySettings.xml`, these may get out of sync. For this reason, the question of which version of the GLAS scripts collected audit data for a particular database instance can only be answered authoritatively at the level of the individual instance, as described below.

To identify versions of `InventorySettings.xml`, the GLAS scripts, FlexNet Inventory Agent, and FlexNet Beacon in use:

1. Watch for the emails that announce updates to the Application Recognition Library, as these may announce an update to the version of `InventorySettings.xml` that is included in the download.

   In the absence of an announcement, or to validate a particular copy of `InventorySettings.xml`, continue with this process.

2. Open your current version of `InventorySettings.xml` in a flat text editor.

   Copies of `InventorySettings.xml` are available:

   - On managed inventory devices (where the FlexNet Inventory Agent is locally installed), in the folder identified in the `InventorySettingsPath` preference. This preference is saved in `[Registry]\ManageSoft\Tracker\CurrentVersion`, and the default values are:
     - On Windows platforms: `$(CommonAppDataFolder)\ManageSoft Corp\ManageSoft\Tracker\InventorySettings\`
     - On UNIX-like platforms: `/var/opt/managesoft/tracker/inventorysettings`.
   - On an inventory beacon, in the folder `$(CommonAppDataFolder)\Flexera Software\Beacon\InventorySettings`. This is the file that is packaged and distributed to installed copies of FlexNet Inventory

Agent. (The file used for direct inventory collection by the inventory beacon from Oracle database instances is separate, and is saved in `Program Files (x86)\Flexera Software\Inventory Beacon\Remote Execution\Public\Inventory`.)

3. Inspect the first line of the file.

```
<InventorySettings RevisionNumber="nn" ...
```

The value of the `RevisionNumber` attribute identifies the version of `InventorySettings.xml`.

4. To identify the revision of the Oracle GLAS scripts used to report an Oracle database instance, inspect its properties:
   a. Go to the **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**).
   b. If necessary, use filtering to find your chosen database instance, and validate that the **Audit evidence** column for this instance displays `Yes`.

      This indicates that the Oracle GLAS scripts have been run on your chosen database instance, and the results uploaded.
   c. In the **Instance** column, click (or Ctrl-click) the name of your database instance to open its properties (in a new tab).
   d. Select the **Attributes** tab, and scroll down to the `LMS_DETAIL_RL_SCRIPT_VERSION` attribute.

      The associated value is the version of the Oracle GLAS scripts used to report on this Oracle database instance.

Tip: *You can also find this information by downloading and exploring the GLAS archive files, or by capturing and reading the `.ndi` file generated for Oracle inventory. However, both of these methods involve much more effort than just looking at the attributes reported for the Oracle database instance.*

Remember: *If you used the direct collection method for your Oracle inventory, only the SQL queries for database inventory from the Oracle GLAS scripts have been executed automatically. Since GLAS hardware scripts cannot be run by this method of inventory collection, you may need other methods to run the hardware GLAS scripts and integrate the results with your GLAS audit data. For more details, see the topic matching your preferred discovery method used with direct inventory collection, under Direct Collection of Oracle Inventory.*

5. To verify the version of the FlexNet Inventory Agent in use:

Tip: *This relies on a completed installation process for FlexNet Inventory Agent. If you have used other methods to distribute only the core components (such as the tracker, `ndtrack`), this individual file is not traceable in the following way. Only completed installations (on either Windows platforms or UNIX-like platforms) produce the installer evidence records visible as described here.*

   a. Go to the **Installed Applications** page (**Applications & Evidence > Applications > Installed Applications**), filter the application **Name** column for names that contain `Inventory Manager Agent`.

      This listing shows all installed versions that are in use within your enterprise, together with the installation count for each version.
   b. Click the hyperlinked value of **Name** for the release you want to investigate.

The **Application Properties** page opens.

c. Select the **Evidence** tab, and ensure the **Installer** evidence type is selected at the top of this page.

One row covers the `FlexNet Inventory Agent` (in very old legacy versions, known as `ManageSoft for managed devices`), and the **Version** column provides the "inner" version of the FlexNet Inventory Agent. For example, if you navigated to the properties for `Inventory Manager Agent 2017 R3`, the associated installer evidence record shows `FlexNet Inventory Agent` version `12.40.%`, a wild-card match for all minor updates to the 12.4 release of FlexNet Inventory Agent.

An alternative path when you are interested in the version of FlexNet Inventory Agent running on a specific inventory device is:

a. Find the target inventory device (for example, in the **All Inventory** or **Active Inventory** listings).

b. Click the device **Name** to open its property sheet.

c. Select the **Applications** tab to find the entry for `Inventory Manager Agent`.

d. Click that **Product** name to open the application properties.

e. Choose the **Evidence** tab to identify the version of FlexNet Inventory Agent.

6. To identify the version of FlexNet Beacon running on an individual inventory beacon:

a. On the inventory beacon, start the Registry Editor.

Tip: *If you are not already logged in as an administrator, you may be prompted for privilege escalation at this point.*

b. Navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\ManageSoft Corp\ManageSoft`.

c. Examine the `ETDPVersion` key, which provides a *`major.minor.build`* version for the installed FlexNet Beacon software.

Any version number earlier than `12.1` requires an update to allow for automatic updates of `InventorySettings.xml` for use by the inventory beacon.

Appendix H: Adjustments to settings for Oracle GLAS information

FlexNet Manager Suite contains data about Oracle installations. Some of this data can be extracted and provided to people from Oracle Global Licensing and Advisory Services (Oracle GLAS), who can analyse that data for you to assist with the status of your Oracle license consumption.

This section is intended to provide information on how to override the following Oracle GLAS information:

- Database edition
- License metrics (when multiple Oracle license metrics are in use within the enterprise)
- Oracle installation environment usage (when other than production).

Adjusting the Oracle Database Edition

Note: *This refers to the database server edition used for licensing the server (not to be confused with the installed edition); for example, a DB SE installation covered by DB SE1 license.*

Exported Oracle GLAS ORCL_OVERVIEW data from FlexNet Manager Suite includes the installed Oracle Database edition. Where an installed Oracle Database edition is different from the licensed Oracle edition used within the FlexNet Manager Suite, Flexera recommends updating the Oracle Database edition value within GLAS ORCL_OVERVIEW. You should update the value of the Oracle Database edition that is installed on the server, with the edition recorded in the license manually. (Oracle's former License Management Service [LMS} is now renamed Global Licensing and Advisory Services [GLAS].)

To update the licensed Oracle Database edition:

1. Extract the exported Oracle GLAS data from FlexNet Manager Suite. (Refer to the *Oracle Instances* topic in the online help for FlexNet Manager Suite for instructions.)
2. Find the record for the particular Oracle Database server and in the ORCL_OVERVIEW.csv, update DATABASE EDITION with the appropriate licensed edition, such as:
    a. Oracle Database Standard Edition One (SE1)
    b. Oracle Database Standard Edition (SE)
    c. Oracle Database Enterprise Edition (EE)

Adjusting the License Metric Setting

The license metric value within an exported GLAS ORCL_OVERVIEW file comes from the license type used within FlexNet Manager Suite to license an Oracle Database installation.

Where multiple license types (for example, `Oracle Named User Plus` and `Oracle Processor`) are used to license Oracle database installations within FlexNet Manager Suite, Flexera recommends using allocations or restrictions with such Oracle Database installations.

To apply allocations to Oracle Database licenses:

1. Open the license properties.
2. On the **Consumption** tab, search for, and select, the required installation of Oracle Database.
3. Use the **Allocate** button to choose the kind of allocation to make for the selected installation.

    Similarly, restrictions may be applied in the **Restrictions** tab of the license properties; or you may prefer to use the **Group assignment** tab to manage license priorities.

Adjusting the Environment Usage Setting

By default, all inventory devices are listed for production use. Where Oracle is installed in an environment other than

production (for example, “Test”), Flexera recommends updating the inventoried device’s role, which automatically adjusts the license consumption within FlexNet Manager Suite, as well as updating the files exported for delivery to Oracle License Management Services.

To adjust the inventory device role:

1. Go to the **Oracle Instances** page (**Inventory > Inventory > Oracle Instances**).
2. Select the Oracle server to open the inventory device properties.
3. On the **General** tab, select the appropriate device role value (for example, **Test**) to update the default Production device role.
4. Save the updated properties.
5. In the **License Compliance** menu, click **Reconcile**. (Either wait for an overnight reconcile, or if you have Administrator access, reconcile immediately. Restrict the reconcile to Oracle only by deselecting Reconcile all publishers.)
6. Perform an Oracle GLAS data export. (Refer to the *Oracle Instances* topic in the online help for FlexNet Manager Suite for instructions.)

   ORCL_OVERVIEW will now reflect the updated environment usage for the Oracle Database installation on that server.

8

Flexera Analytics

Flexera Analytics, powered by Cognos, allows you to build custom reports based on weekly snapshots of your licensing and installation data.

Improved security

From release 11.0.11 of IBM Cognos (shipped with IT Asset Management 2018 R2), Flexera Analytics supports the use of HTTPS protocols for direct access from your web browser. You can also use the Transport Layer Security (TLS) 1.2 protocol to improve the security of data transfers, and this is the recommended security level for HTTPS communications. For more information, please see:

- *Configuring IIS to Use SSL/TLS Encryption*
- *Reconfigure Cognos Analytics to Use Third-Party SSL Certificates*.

These topics are both available in both of *Installing FlexNet Manager Suite On-Premises* and *Upgrading FlexNet Manager Suite On-Premises*.

More About Flexera Analytics

Data for reports is automatically copied from the compliance database to the data warehouse database once a week at 6am (central server time) on Sunday mornings.

The system preserves the 12 most recent weekly snapshots. As well, the last snapshot taken within a month is preserved as a monthly snapshot, and the 36 most recent monthly snapshots are preserved. (The other weekly snapshots taken earlier each month are automatically culled as they are outside the 12 most recent.)

Tip: *Even though Flexera Analytics was only made available in cloud at the 2017 R2 release of IT Asset Management, snapshots have been collected on the above schedule for about the previous two years, or since you started using the cloud implementation (whichever is the most recent). This historical data is now available for your use in reports.*

The cloud implementation of Flexera Analytics has the following limitations:

- The `Analytics Administrator` role is not available.
- Cognos log files are not available.

- You cannot include any custom SQL in reports. If you attempt to do so, you will see an error `You do not have permission to create or edit SQL/MDX.`

If the default number of operators in each Cognos role is not sufficient for your needs, please contact your Flexera Consultant to request additional quantities at no additional cost.

Before attempting to access Flexera Analytics, make sure that you have added the operator accounts to the appropriate role:

1. Go to the **IT Asset Accounts** page (**Administration > IT Asset Management Settings > IT Asset Accounts**).
2. In the **Roles** tab, expand the **Business reporting portal** section.
3. Select the appropriate privileges from the drop-down list (such as `Analytics User`).
4. Give the role an appropriate name and description, and click **Create**.
5. Switch to the **All Accounts** tab, and create or select your account, and assign it to the new role.

Data Models for Flexera Analytics

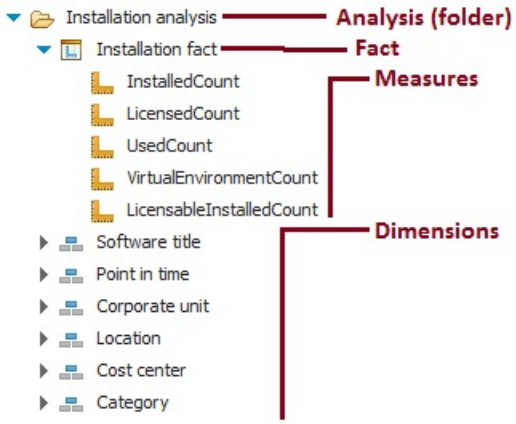
Two separate packages are supplied with IT Asset Management for use in Flexera Analytics:

- **FlexNet Manager Platform Data Warehouse (analysis)** contains a dimensional model that allows considerable flexibility for online analytical processing (OLAP). This data comes from the data warehouse database (suggested name: FNMSDataWarehouse), and some report-time specialized queries that calculate trending data sets and the like.

The model consists of the following elements:

- *Folder* — In the dimensional model, each folder is called an *analysis*, including **Installation analysis** for reports about software inventory within your enterprise; and **Consumption analysis** for reports about license positions.
- *Fact* — In the dimensional model, the first child of each analysis folder is the central fact around which the related data dimensions and measures are organized. For example, the **Installation analysis** contains the **Installation fact**; and similarly, the **Consumption analysis** starts with the **Consumption fact**.
- *Dimension* — In the dimensional model, each dimension gives a related data set that you can use to analyze the central fact. For example, it's obvious that when you report on your **Installation fact**, you need to analyze in terms of individual software applications; so that there is a **Software title** dimension available.
- *Measure* — Each measure contains a specific value derived (in general) from one or more fields in one of the source databases. In the dimensional model, some measures are attached directly to the fact. For example, the **Installation fact** has an **Installed (max qty)** measure that you can include in your reports. Measures also relate to other dimensions, but here they are referred to as *attributes*, and you generally choose from individual *values* of the attribute, rather than selecting the entire attribute. For example, the **Software title** dimension expands into a hierarchy of the **Publisher**, **Product**, and **Application** measures; but for creating reports, you generally select one of the publisher's names, and so on.

Figure 3: Elements in the dimensional data model



The dimensional data model is covered in the following sections.

Data Warehouse (Analysis) Model

FlexNet Manager Platform Data Warehouse (analysis) dimensional model centers around two related facts for analysis:

- **Installation fact**
- **Consumption fact**.

Because these facts are so closely related, they share several dimensions in common. These shared dimensions are grouped separately in this document, and described only once.

Installation Analysis

The **Installation analysis** is a folder which contains:

- The **Installation fact** (see Installation Fact: Measures)
- A **Software title** dimension that is unique to the **Installation analysis** (see Software Title Dimension)
- Several common dimensions shared with the **Consumption analysis** (see Point in Time Dimension and Enterprise Group Dimensions).

Together these dimensions and associated measures/attributes allow you to analyze software installation numbers over time in your enterprise.

Installation Fact: Measures

The **Installation fact** allows tracking of software installations over time. The **Installation fact** supports the following measures. These general comments apply to all measures:

- For each measure, the figure shown is the maximum value found in all data snapshots included in your reporting period.

- All values are filtered by the access rights of the current operator (or report user). Access rights are managed through roles to which the operator is assigned, using enterprise groups of various kinds as filters. For example, if operator Sam is denied access to data from your North American location (and its children), then a report entry for license consumption in your Chicago office always shows zero when viewed by Sam, regardless of how many licenses are really consumed in Chicago.
- Since you may build a report using any combination of enterprise groups, the totals described for the following measures may be segmented across those groups, as expected. This subtotalling and segmentation effect is omitted from the following descriptions for simplicity.

Table 8: Measures for the Installation fact (alphabetical listing)

Measure	Notes
Installed (max qty)	The total number of application installations, as filtered by your chosen dimensions and identified in the most recent compliance calculation before the snapshot. *Related management view:* **Installed Applications** (**Applications & Evidence > Applications > Installed Applications**).
Installed on VMs (max qty)	The subset of the total application installations that are on devices identified as virtual machines. This is not relevant to license consumption, since these installations may be covered by special rights, or by exemptions, or in other ways so that they do not consume from a license. *Related management view:* No direct equivalent.
Licensable installations (max qty)	The subset of the total application installations that appear to be licensable (regardless of whether or not they are currently consuming from a license). *Related management view:* In application properties, on the **Devices** tab, you can add the **Licensable** column, and filter for `Yes`. The results returned gives the equivalent count for the chosen application.
Licensed (max qty)	The subset of the total application installations that have been covered by a license (and are consuming license entitlements) in the most recent compliance calculation before the snapshot. *Related management view:* **Installed Applications** (**Applications & Evidence > Applications > Installed Applications**), filtered by application name and **Licensed** = `Yes`.
Used (max qty)	The subset of the total application installations that appear to be in use as at the most recent compliance calculation before the snapshot. *Related management view:* **Installed Applications** (**Applications & Evidence > Applications > Installed Applications**), filtered by application name and with **Usage** column displayed (this gives the count).

Software Title Dimension

As part of the **Installation analysis**, the **Software title** dimension allows you to drill down to an individual application

(or 'software title') as part of analyzing software installed throughout your enterprise.

The **Software title** dimension expands into a three-level hierarchy that gives you access to (in order):

- **Publisher**
- **Product**
- **Software Title Name** (or, as seen within IT Asset Management, the application name).

You may report on any level. For example, if you choose only a publisher, your report includes all products and applications from that publisher.

Table 9: Attributes for the Software title dimension, Publisher level

Attribute	Notes
Publisher	The name of the company that publishes this application (and product). *Related management view:* For each application, the publisher is linked through the **General** tab of the application properties. However, the publisher must exist first for linking, and these records are maintained in the **All Vendors** page (**Procurement > Purchases & Vendors > All Vendors**). The publisher for each application is widely available in application listings, such as the **All Applications** page (**Applications & Evidence > Applications > All Applications**).

Table 10: Attributes for the Software title dimension, Product level

Attribute	Notes
Product	The common name for a family of applications, independent of version or edition details. This must be unique for any given publisher. *Related management view:* For each application, the **Product** name is displayed in the **General** tab of application properties (and when you manually create an application record, you can link the product value there). All applications that share a common value in that **Product** field are deemed to be distinct versions/editions of the same product. The product name is widely available in application listings, such as the **All Applications** page (**Applications & Evidence > Applications > All Applications**) (although sometimes you must add it to the listing from the column chooser).

Table 11: Attributes for the Software title dimension, Software Title (or application) level (alphabetical listing)

Attributes	Notes
Application	The name of the application. *Related management view:* For each application, the **Name** is displayed in the **General** tab of application properties (and when you manually create an application record, you can edit the application name there). The application name appears in all application listings, such as the **All Applications** page (**Applications & Evidence > Applications > All Applications**).

Attributes	Notes
Application status	Shows the current status for the application, from values such as `Authorized`, `Ignored`, `Deferred`. and others. *Related management view:* For each application, the **Status** is displayed in the **General** tab of application properties (and when you manually create an application record, you can edit the status there). The application **Status** is widely available in application listings, such as the **All Applications** page (**Applications & Evidence > Applications > All Applications**).
Classification	The classification applied to this application, such as `Commercial` or `Malware`. *Related management view:* For each application, the **Classification** is displayed in the **General** tab of application properties (and when you manually create an application record, you can edit the classification there). The application classification is widely available in application listings, such as the **All Applications** page (**Applications & Evidence > Applications > All Applications**).
Edition	The edition of this application. Editions describe different levels of product functionality, such as Standard and Pro. *Related management view:* For each application, the **Edition** is displayed in the **General** tab of application properties (and when you manually create an application record, you can edit the edition there). The application edition is widely available in application listings, such as the **All Applications** page (**Applications & Evidence > Applications > All Applications**).
Is licensed	A Boolean that indicates whether the application is linked to any license. *Related management view:* For each application, the **Licenses** tab of application properties lists the license(s) to which this application is linked. You can also attach or detach licenses to/from the application on that tab. The **Licensed** column is widely available in application listings, such as the **All Applications** page (**Applications & Evidence > Applications > All Applications**).
Version	The version (or release number) of this application. *Related management view:* For each application, the **Version** is displayed in the **General** tab of application properties (and when you manually create an application record, you can edit the version there). The application version is widely available in application listings, such as the **All Applications** page (**Applications & Evidence > Applications > All Applications**).

Consumption Analysis

The **Consumption analysis** is a folder which contains:

- The **Consumption fact** (see Consumption Fact: Measures)
- A **Software license** dimension that is unique to the **Consumption analysis** (see Software License Dimension)
- Several common dimensions shared with the **Installation analysis** (see Point in Time Dimension and Enterprise Group Dimensions).

Together these dimensions and associated measures/attributes allow you to analyze license consumption numbers over time in your enterprise.

Consumption Fact: Measures

The **Consumption fact** allows tracking the results of license consumption calculations over time. It supports the measures listed below. These general comments apply to all measures:

- For each measure, the figure shown is the maximum value found in all data snapshots included in your reporting period.
- All values are filtered by the access rights of the current operator (or report user). Access rights are managed through roles to which the operator is assigned, using enterprise groups of various kinds as filters. For example, if operator Sam is denied access to data from your North American location (and its children), then a report entry for license consumption in your Chicago office always shows zero when viewed by Sam, regardless of how many licenses are really consumed in Chicago.
- Since you may build a report using any combination of enterprise groups, the totals described for the following measures may be segmented across those groups, as expected. This sub-totalling and segmentation effect is omitted from the following descriptions for simplicity.

Table 12: Measures for the Consumption fact (alphabetical listing)

Measure	Notes
Allocated (max qty)	The number of installations (or users, depending on license type) that have specifically received allocations from the license in question. Allocations are essentially a manual linking of the installation/user with the license, to the exclusion of all other licenses, and may be made individually on the **Consumption** tab of the license properties. While they remain a 1:1 association, they can also be processed in bulk on the **Apply Allocations and Exemptions** page. *Related management view:* See either the **Consumption** tab of the license properties, or the **Apply Allocations and Exemptions** page (**Licenses > License Management > Apply Allocations and Exemptions**).
Consumed (max qty)	The consumption count for each license, as filtered by your chosen dimensions and identified in the last exported compliance calculation for each point in time. This is the final consumption calculation for each license, taking into account all beneficial aspects like items covered by other product use rights, device exemptions and the like. *Related management view:* The **Compliance** tab of license properties displays the **Consumed entitlements** count. Also visible in many license listings, such as the **All Licenses** page (**Licenses > License Management > All Licenses**).

Measure	Notes
Covered by downgrade right (max qty)	The number of license entitlements consumed under a downgrade right (that is, the consumption occurred against a license for a later version or higher edition, but which included a downgrade right). *Related management view:* This collective total is not available in management views.
Covered by rights on VMs and hosts (max qty)	The number of virtual machines that are linked to the selected license, but that are not consuming any entitlements because they are covered by special product use rights. *Related management view:* For an individual license, look at the **Consumption** tab of license properties. Filter for **Inventory device type** of `Virtual machine`, and check the **Consumed** value (shows zero for any virtual device linked to this license but not consuming at the last license compliance calculation), and check for an **Exemption reason** such as `Covered by virtual application access`.
Covered by second use right (max qty)	The number of license entitlements that were *not* consumed because the related consumption was covered by the second use right on the license. *Related management view:* This collective total is not available in management views. For an individual license, open the license properties, and review the **Consumption** tab. A device contributing to this count shows `0` in the **Consumed** column, and an **Exemption reason** of `Second use`.
Exempted (max qty)	The number of installations linked to the license that do not consume a license entitlement because they have an exemption. Exemptions may take either of two forms: • An exemption for an individual inventory device linked to the license can be made on the **Consumption** tab of the license properties (these manual exemptions may also be made in bulk using the **Apply Allocations and Exemptions** page) • An exemption may be made automatically during license consumption calculations when the inventory device has been assigned a role (such as `Test`) that matches an exemption reason declared in the **Use rights & rules** tab of the license properties. *Related management view:* See either the **Consumption** tab of the license properties, or the **Apply Allocations and Exemptions** page (**Licenses > License Management > Apply Allocations and Exemptions**).

Measure	Notes
Installed (max qty)	The number of installed software records that are linked to the license in question. This is unlikely to be the total number of installations of the same application. *Related management view:* For an individual license, look at the **Applications** tab of the license properties, and add the **Installed** column from the column chooser. There is no management view that shows this value for many licenses at a time. (Keep in mind that this is *not* the **Installed** count on an applications listing such as **Installed Applications**, because that figure is not filtered by license.)
Last purchase date	The most recent purchase date of all the purchases linked to the selected license. *Related management view:* For an individual license, you can check the result on the **Purchases** tab of the license properties, adding the **Purchase date** to the table from the column chooser, and sorting on the dates to see the latest one.
Licensed cores (max)	The number of processor cores that are covered by a license, totaled for all inventory devices linked to this license at the appropriate license consumption calculation. *Related management view:* In the properties for an individual license, check the **Consumption** tab and add the **Cores** column from the column chooser. This shows the number of cores licensed for each device attached to this license. (Remember that the core count for individual devices may be missing from inventory, depending on the inventory tool used; and that you can manually correct the value in the **Hardware** tab of the inventory device properties.) No management view of multiple licenses shows the total cores licensed.
Linked VMs (max qty)	The number of devices linked to this license that are virtual machines. This includes virtual machines which are not consuming entitlements because they are covered by other product use rights, exemptions, and the like. *Related management view:* This collective total is not available in management views.
Purchased (max qty)	The total number of license entitlements owned by the enterprise (or enterprise group) for the given license, being the sum of the **Licensed from PO** and **Extra entitlements** values stored in the properties of the license. This is the number of entitlements you are entitled to consume for this license. (Where individual purchases have been associated with particular enterprise groups, these facts are reflected in the subtotals for relevant groups.) *Related management view:* See the **Compliance** tab of the individual license properties, or review a license listing such as the **All Licenses** page (**Licenses > License Management > All Licenses**).

Measure	Notes
Shortfall/Availability (max)	The calculated difference between **Total entitlements** and **Consumed entitlements** (visible in the **Compliance** tab of the license properties). Positive values show entitlements still available, and negative values show that consumption exceeds purchases (a purchasing shortfall). *Related management view:* For each license, see the **Shortfall/Availability** figure in the **Compliance** tab of the license properties. Also available in license listings such as the **All Licenses** page (**Licenses > License Management > All Licenses**).
Total purchase price (max)	The total of all recorded costs for all the purchases linked to the particular license. (This may be unrelated to the costs of licenses consumed.) *Related management view:* The total price for each individual purchase is visible in the **All Purchases** page (and in the **Financial** tab of the properties of each purchase, where details can be recorded). There is no management view that shows the rolled-up total of all the purchases linked to a specific license.
Used (max qty)	Out of the installation records linked to the particular license, this is the subset for which there are usage records that meet the current criteria for software usage (by default, an application must have been used at least once in the last 3 months, with the settings adjustable on the **Usage** tab of the application properties). *Related management view:* You can inspect which devices/users are known to have used the licensed software on the **Consumption** tab of the license properties. As well, license views such as the **All Licenses** page (**Licenses > License Management > All Licenses**) offer the **Used** count for the installations linked to each license.

Software License Dimension

As part of the **Consumption analysis**, the **Software license** dimension allows you to drill down to an individual license for a specific product, as part of analyzing license consumption throughout your enterprise.

The **Software license** dimension expands into a four-level hierarchy that gives you access to (in order):

- **Publisher**
- **License Type**
- **Product Name**
- **License Name**.

Table 13: Attributes for the Software license dimension, Publisher level

Attribute	Notes
Publisher	The name of the company that publishes this application (and product). *Related management view:* For each application, the publisher is linked through the **General** tab of the application properties. However, the publisher must exist first for linking, and these records are maintained in the **All Vendors** page (**Procurement > Purchases & Vendors > All Vendors**). The publisher cited in each license is widely available in license listings, such as the **All Licenses** page (**Licenses > License Management > All Licenses**).

Table 14: Attributes for the Software license dimension, License type level

Attribute	Notes
License type	The type of each license (as defined in IT Asset Management). Used here as a grouping mechanism, so that you must first choose the license type to drill down any further. *Related management view:* For each license, the **License type** appears in the **Identification** tab of the license properties. **License type** is also widely available in license listings, such as the **All Licenses** page (**Licenses > License Management > All Licenses**).

Table 15: Attributes for the Software license dimension, Product name level

Attribute	Notes
Product	The common name for a family of applications, independent of version or edition details. This must be unique for any given publisher. *Related management view:* For each license, the **Product** name is displayed in the **Applications** tab of the license properties (and you can link other applications there, each of which displays its own **Product** value). Keep in mind, too, that a license may be linked to applications from more than one product. The **Product** name is widely available in license listings, such as the **All Licenses** page (**Licenses > License Management > All Licenses**) (although sometimes you must add it to the listing from the column chooser).

Table 16: Attributes for the Software license dimension, License name level (alphabetical listing)

Attributes	Notes
Compliance status	Indicates whether or not use of software under this license complies with the license terms and conditions (and some related values). *Related management view:* For individual licenses, the **Compliance status** value is displayed in the **Compliance** tab of the license properties. The **Compliance status** column is widely available in license listings, such as the **All Licenses** page (**Licenses > License Management > All Licenses**).

Attributes	Notes
Duration	The localized form of the possible license duration values, for which the English defaults are: • `Perpetual` (a license that is not time-limited) • `Subscription` (a license that must be renewed by regular payments, usually annually) • `Time limited` (a license that will expire, and typically cannot be renewed, such as a time-limited evaluation license). *Related management view:* For individual licenses, the **Duration** value is displayed in the **Identification** tab of the license properties. The **Duration** column is widely available in license listings, such as the **All Licenses** page (**Licenses > License Management > All Licenses**) (although sometimes you must add it to the listing from the column chooser).
Estimated unit price	An approximate cost for a single entitlement purchased for this license. If the **Override unit price** has been set on the **Purchases** tab of the license properties, this is the value used. Otherwise, the unit price (**Total price** divided by **Effective quantity**) is taken from the most recent purchase linked to the license. *Related management view:* For an individual purchase, the unit price is listed in the **Financial** tab of the purchase properties, as **Quantity X at unit price:**. The **Unit price (*currency*)** column is widely available in listings of purchases (although you sometimes you must add it to the listing from the column chooser).
Expiry date	For subscription (or other time-limited) licenses, this is the date when the current license expires. Any value is ignored when `SoftwareLicenseDurationID = 3` (that is, for a perpetual license.) A null value in this field also means that the license is perpetual, since it does not have an expiry date. *Related management view:* For individual licenses, the **Expiry date** value is displayed in the **Identification** tab of the license properties. The **Expiry date** column is widely available in license listings, such as the **All Licenses** page (**Licenses > License Management > All Licenses**) (although sometimes you must add it to the listing from the column chooser).

Attributes	Notes
Grants downgrade right	A Boolean that indicates whether this license offers downgrade rights. ***Tip:*** *There is no corresponding reporting on the upgrade right. This is because when the upgrade right is available, IT Asset Management automatically updates the primary application on the license to the most recent version. This means, in effect, that every license with the upgrade right has already been upgraded to the max.* *Related management view:* For license types that support downgrade rights, there is a **Downgrade rights** section in the **Use rights & rules** tab of the license properties that gives details. There is no equivalent of this Boolean in management views within IT Asset Management.
Grants rights on VMs and hosts	A Boolean that indicates whether the license provides specific rights for virtual machines or their hosts. *Related management view:* For individual licenses, when the **Use rights & rules** tab includes a section on **Right of second use**, the setting **No special virtualization rights** sets this Boolean to `0` (false). The remaining three settings on that section set this Boolean to `1` (true).
Grants second use right	A Boolean that indicates whether this license offers second use rights. Only supported for license types where **License type supports second use right** is `1` (true). *Related management view:* For license types that support second use rights, there is a **Right of second use** section in the **Use rights & rules** tab of the license properties that gives details. There is no equivalent of this Boolean in management views within IT Asset Management.
License	The full name of this license. *Related management view:* For each license, the **Name** is displayed (end editable) in the **Identification** tab of license properties. The **Name** is displayed in every license listing, such as the **All Licenses** page (**Licenses > License Management > All Licenses**).
License edition	The edition of this license. While this is strictly an attribute of the license, a common convention is to allow automatic updating to reflect the edition of the latest application version linked to the license. *Related management view:* For each license, the **Edition** is displayed (end editable) in the **Identification** tab of the license properties. The **Edition** is widely available in license listings, such as the **All Licenses** page (**Licenses > License Management > All Licenses**) (although sometimes you must add it to the listing from the column chooser).

Attributes	Notes
License status	Where this license was (at snapshot time) in the license life-cycle: whether `Purchased`, `Received`, `In stock`, `Active`, or `Retired`. ***Tip:** Do not confuse this value with **Compliance status**.* *Related management view:* For each license, there is a **Status** drop-down list available in the **Identification** tab of the license properties. The **Status** column is widely available in license listings, such as the **All Licenses** page (**Licenses > License Management > All Licenses**) (although sometimes you must add it to the listing from the column chooser).
License type supports second use right	A Boolean that indicates whether the current license type is capable of supporting the right of second use. *Related management view:* For individual licenses, the **Use rights & rules** tab includes a section on **Right of second use** when this value is true, and otherwise omits the section entirely. There is no equivalent available in management views.
License version	The version of this license. Note that this version number applies strictly to the license, and so may have been used in custom ways in your enterprise; but a common convention is to make the license version reflect the version of the application initially licensed here (although these versions may change through upgrade and downgrade rights). Another convention is to allow automatic updating to reflect the latest application version linked to the license. *Related management view:* For each license, the **Version** is displayed (end editable) in the **Identification** tab of the license properties. The **Version** is widely available in license listings, such as the **All Licenses** page (**Licenses > License Management > All Licenses**) (although sometimes you must add it to the listing from the column chooser).
Subject to true-up	A Boolean that indicates whether this is a true up license. A true up license will not be displayed as **At risk**, with any over-consumption in a period being adjusted through the regular (usually annual) true up process. *Related management view:* For each license, there is a **Subject to true up** check box available in the **Identification** tab of the license properties. The **Subject to true up** column is widely available in license listings, such as the **All Licenses** page (**Licenses > License Management > All Licenses**) (although sometimes you must add it to the listing from the column chooser).

Common Dimensions

This section documents dimensions that are common to both the **Installation** analysis and the **Consumption** analysis. These include:

- The **Point in time** dimension, which sets the reporting period for your custom reports, and determines how many saved data snapshots are included in them

- Four dimensions covering the four types of enterprise group (**Corporate unit**, **Location**, **Cost center** and **Category**) that allow you to segment the reported figures across your enterprise, using the corporate structure you have established in IT Asset Management
- Four near-duplicate dimensions (**Corporate unit (Software license)**, **Location (Software license)**, **Cost center (Software license)** and **Category (Software license)**), which, although they are unique to the **Consumption** analysis, are included in this section because their attributes and values exactly duplicate the previous four. The distinct purpose of these additional dimensions is to allow reporting on the *ownership* of software licenses, as distinct from the *consumption* of software licenses using the previously-mentioned four dimensions. Thus you could, for example, use custom reports to track down licenses assigned to (owned by) the Marketing department, but accidentally being consumed by the Sales team.

Point in Time Dimension

The **Point in time** dimension expands into a tree hierarchy with three levels:

- **Year**
- **Month** within a year
- **Day**, which is really the full date/time value when the snapshot was recorded in the data warehouse database.

You can, of course, pick any period (such as a year) to prepare your report, and then within the published report drill through to smaller time intervals, down to the individual data snapshot. Snapshots are recorded weekly.

This dimension has no equivalent in the management views within IT Asset Management, and does not reflect data stored in the compliance database.

Table 17: Attributes for the Point in time dimension

Attributes	Notes
Year	The year in which a snapshot was captured.
Month	The month in which a snapshot was captured.
Month-year	A convenience caption for reporting that combines the numeric month and year values. For example, August 2016 is represented as `8 - 2016`.
Day	The day (number within the month) on which the snapshot was captured.
Date	The full date/time value when the snapshot was stored (for example, `2016-09-03 14:40:00.000`). The date is represented in the extended format defined in ISO 8601:2004 (that is, `YYY-MM-DD`) and the time is in 24-hour format listing hours, minutes, seconds, and milliseconds. The time zone is not included, but all timestamps are recorded using the time setting on your central application server.

Enterprise Group Dimensions

There are four kinds of enterprise group in IT Asset Management:

- **Corporate unit**
- **Location**
- **Cost center**
- **Category**.

Each of these four kinds is available as a dimension for your custom reports, present in common in both the **Installation analysis** and the **Consumption analysis**. This allows you to segment either application installation numbers or license consumption numbers across various kinds of enterprise groups, as best fits your corporate approach to group management.

Tip: *Keep in mind that the numbers quoted for each enterprise group are "rolled up totals": that is, they are the total for all the children of this group, plus any local value for the group itself. For example, consider this simplified location hierarchy (each row being the only child of the one above) with the local installation counts of a particular application as shown. Each location then shows the rolled-up total installation values given in the third column:*

cation	*Local installations*	*Rolled-up total shown*
North America HQ	*12*	*55*
North-west Region Office	*10*	*43*
Chicago Office	*33*	*33*

In reality, the rolling up of totals is more complex, since any high-level enterprise group is likely to have many peer children, each of which has many peer children, and so on down through the tree.

In the case of the **Consumption analysis** (only), as well as the basic enterprise groups that you may use to analyze consumption across different groups, there is a second set of the same groups with a slightly different naming convention:

- **Corporate unit (from license)**
- **Location (from license)**
- **Cost center (from license)**
- **Category (from license)**.

These identify the same kinds of groups, but in this case as an attribute of the license itself, identifying any relationship between the license and enterprise groups established on the **Ownership** tab of the license properties.

The presence of enterprises groups in these two ways, both tracking the *ownership* of the license and separately segmenting the *consumption* of the same license, allow you to probe scenarios such as licenses that you thought were assigned to one group being consumed elsewhere (which means the license was *over*-assigned to the first group, and has spare capacity).

Each of the enterprise group dimensions expands to show a tree hierarchy with maximum depth of 10 levels of child groups of the same kind. Each level is identified in the group's label, such as **Location - level 3**.

At each level, each corporate group has only a single attribute available:

Table 18: Attribute for the enterprise group dimensions, at each of the 10 levels

Attribute	Notes
`Enterprise group` **name**	The name of each Corporate unit, Location, Cost center, or Category, as displayed in the label (in place of the *`Enterprise group`* placeholder). *Related management view:* The name of each enterprise group is available wherever groups are included in management views within IT Asset Management. The details are maintained in the listing for each enterprise group, where you may expand the hierarchy to any focus point, and either click the **+** icon to add a new child, or click the edit (pencil) icon to modify the name of an existing group.

9

Authentication

User authentication is a critical component of the security structure in your organization. Many security-conscious organizations are moving beyond traditional user name and password authentication within each tool they use, to a standard "single sign-on" approach. This chapter describes the authentication technologies and configuration requirements that can be used with IT Asset Management to integrate with a single sign-on infrastructure.

Single Sign-On Support with SAML

To enable single sign-on using an identity provider, IT Asset Management includes support for Security Assertion Markup Language (SAML) 2.0 technology, and will integrate with any identity providers that are compliant with SAML 2.0.

Tip: *The terminology for SAML describes the two sides of the relationship with the following terms:*

- *The system that controls operator login for authentication is called an "identity provider". Any identity provider that complies with SAML 2.0 is supported. Examples include:*
  - *Okta (http://www.okta.com)*
  - *G Suite (http://gsuite.google.com)*
  - *SalesForce (http://www.salesforce.com).*
- *The software that the operator can access after login (in this case, IT Asset Management) is called a "service provider".*

Tip: *A limitation of the underlying library (`Sustainsys.Saml2`) means that SAML authentication for IT Asset Management cannot support Federal Information Processing Standards (FIPS).*

Using Single Sign-on

When single sign-on has been configured appropriately, an attempt to log in to IT Asset Management will be redirected to the identity provider (IdP), where the login is supported. You may also log in to IT Asset Management directly from the identity provider, provided that this has been configured with the appropriate link to IT Asset Management.

When logging out, you can choose to close the IT Asset Management session, without affecting the session on the

identity provider (or any other service provider).

Configuring Single Sign-on (Overview)

Your SaaS implementation of IT Asset Management includes a fixed set of values needed by your identity provider. In turn, your service provider also needs details from your identity provider. The major configuration steps are as follows:

1. Implement your chosen identity provider, configuring it for your preferred strategy.

   Note: *As a service provider, IT Asset Management supports both:*

   - *Sign-on initiated by the identity provider (that is, an operator logs into the identity provider directly, and then selects IT Asset Management)*
   - *Sign-on initiated by the service provider (that is, an operator navigates to your customized tenant URL, and the login is redirected through your identity provider).*

   In either method, the operator is granted access to the web interface of IT Asset Management, and can access all functionality authorized by the various roles to which the operator's account is assigned. As well, IT Asset Management supports digital signing (using certificates) of all communications ("assertions") between the identity provider and service provider. However, it does not support additional encryption of assertions (other than the encryption provided by the HTTPS protocol); nor does log out from the identity provider automatically log the operator out of IT Asset Management (that is, single sign-out is not supported).

   Tip: *As part of the ordering and implementation process, new customers should advise Flexera of your preferred tenant subdomain name. If you have not done this, the subdomain is identified by your tenant GUID, which is a meaningless, hard-to-type jumble of letters and numbers. If this describes what you see as the first part your tenant URL, please advise your Flexera support contact of your preferred tenant subdomain name so that the URL can be updated for you.*

2. Provide the information about the service provider (your cloud instance of IT Asset Management) to the identity provider, copying the information provided and if necessary handing off to the administrator responsible for the identity provider. For details, see Configuring IT Asset Management for Single Sign-On Integration.

3. Receive from the administrator of your identity provider either the URL to download the metadata file for the identity provider, or a copy of the metadata file itself; and enter the appropriate details to configure IT Asset Management to validate and make use of the metadata file. Details are also included in Configuring IT Asset Management for Single Sign-On Integration.

4. If your identity provider requires use of a digital certificate to sign assertions it sends to IT Asset Management, the certificate (signature public key) is included in the metadata file. When either you upload a copy of the metadata file (if that was separately supplied), or enter the URL for the metadata file (which is downloaded and validated by IT Asset Management), certificate details are automatically stored in the central database, and validated against all future assertions.

5. As you integrate your identity provider with your cloud instance of IT Asset Management, you may from time to time change the balance of authentication responsibilities between the two systems to best suit your implementation progress. You may, for example:

   - Start with authentication fully managed by IT Asset Management, with accounts created in Flexera Account Management. This is the default or starting position when your tenant is first configured within IT Asset Management.

- Keep your default or main authentication management with IT Asset Management, but start switching individual identities or accounts over to your identity provider, as a pilot project to validate that all is well.
- Make your identity provider the default way of logging in, but allow operators to also log in directly through IT Asset Management when required.
- Enforce your single sign-on solution as the only path through which operators may log into IT Asset Management.

This choice of operating modes is made in the same configuration page as the other integration settings are registered. The current choice of mode is applied equally to all operators within your tenant in IT Asset Management. The fastest and best controlled authentication experience comes when each operator bookmarks a customized, tenant-specific URL that includes your own subdomain (for example, `https://exampleTenant.flexnetmanager.com/Suite`). When an operator who is not currently logged in navigates to this URL, the login is redirected to the appropriate service based on the mode setting you have currently selected.

Configuring IT Asset Management for Single Sign-On Integration

A single location provides all the controls to configure integration between your identity provider and this service provider (IT Asset Management), as well as to turn on/off single sign-on operation in controlled steps reflecting your implementation progress.

Tip: *A limitation of the underlying library (`Sustainsys.Saml2`) means that SAML authentication for IT Asset Management cannot support Federal Information Processing Standards (FIPS).*

To configure IT Asset Management as an SSO service provider:

1. Log in to IT Asset Management as an operator that is a member of the `Administrator` role.

   The necessary settings are available only to an operator with administrator privileges.

2. Go to the IT Asset Management Settings **General** page (**Administration > IT Asset Management Settings > General**).
3. Select the **Security** tab, and scroll down to the **Authentication** section.
4. Copy the read-only values from the **General information** part, label them, and send them to the person who administers your identity provider (or, if you are authorized to do so, add those values required by your identity provider, so that you configure it to recognize IT Asset Management as a service provider).

   Tip: *In your identity provider, you must set a **NameID** that the identity provider uses to uniquely identify the person logging in, when it asserts this identity back to the service provider. In IT Asset Management 2023 R2.4, this must be the employee's email address. Within IT Asset Management, this value is available in the following equivalent places:*

   - *In the web interface, in the **Account Properties** page as the **Account** field (where the value is specified during account creation). Once an account has been saved in IT Asset Management, this value is no longer*

editable (nor can the account be deleted, as it may relate to historical activity; but you may disable an account when appropriate).

- *As a read-only value in the **Login** column of the **IT Asset Accounts** page (**Administration > IT Asset Management Settings > IT Asset Accounts**).*
- *In the `OperatorLogin` column in the `ComplianceOperator` table of the compliance database, which is the database column underlying the two previously-mentioned display places. Only an operator whose value asserted by the identity provider is matched in this table is granted access to a tenant within IT Asset Management.*

5. To communicate with the identity provider, IT Asset Management needs an XML file of metadata including URLs of endpoints, information about supported bindings, identifiers, and public keys. (For more information, see the SAML 2.0 metadata schema available at http://docs.oasis-open.org/security/saml/v2.0/saml-schema-metadata-2.0.xsd.) Receive back from your administrator colleague (or extract the data from the identity provider yourself, if authorized) either of the following:

   - A URL for download of the metadata document from the identity provider. Enter this value in the **URL** field in the **SAML identity provider metadata** part of the page.
   - A copy of the metadata XML file at an accessible file location on your network. Use the **Metadata file** controls to browse to and upload this file to IT Asset Management.

   ***Tip:** You may freely choose either of these methods, but you cannot save the page if you have completed both fields. When you save your setting, IT Asset Management downloads the metadata file (if necessary), and validates its contents. If there are any problems, an alert helps you to remedy those, until your service provider is correctly configured to interact with your identity provider.*

6. Choose the operating mode for authentication, deciding whether to have one or both identity providers active (and if both, which one should have priority). Notice that this setting affects all operators within your tenant (you cannot have different settings for different operators' accounts):

   - **Flexera Account Management** — This default case requires that you create accounts using the Flexera Account Management page. Operators provide their credentials (account name and password) on the Flexera login screen. Anyone navigating directly to your tenant-specific URL is redirected through the Flexera login screen.
   - **Flexera (default) with SAML identity provider (pilot)** — Intended as a transition state, this prompts for credentials at the Flexera login screen; but operators (provided that their single sign-on credentials have been configured in your identity provider) can choose a **Log in with single sign-on** button.
   - **SAML identity provider (default) with Flexera option** — Operators navigating to your customized tenant-specific URL (such as `https://exampleTenant.flexnetmanager.com/Suite`) are redirected to the login page from your SAML identity provider (for example, Okta). However, those operators needing to use Flexera credentials have the option of navigating to a non-tenanted URL (such as `https://www.flexnetmanager.com/Suite`), where the Flexera login screen is presented. Through this screen, while in this mode, logging in with Flexera credentials succeeds.

     ***Tip:** It is a good safety net for at least one administrator to preserve a Flexera account that can be used to login through the non-tenanted URL in case emergency configuration changes are needed.*

- **SAML identity provider only** — Operators navigating to your custom subdomain (such as `https://exampleTenant.flexnetmanager.com/Suite`) are redirected to log into your single sign-on solution. Operators with only single sign-on credentials who attempt to log into the non-tenanted URL (https://www.flexnetmanager.com/Suite) fail, and do not gain access. An operator with both kinds of credentials may use the Flexera credentials with the Flexera login screen; but when Flexera Account Management passes back authorization for this operator to access her registered tenant (which is now configured for SAML only), access is refused because the incorrect kind of credentials have been used. The login attempt is redirected to your single sign-on solution, and the operator must log in again.

  Note: *In this SAML-only mode, there is a risk that subsequent incorrect configuration changes could lock out all access to IT Asset Management. If this happens, contact Flexera Support, asking for your tenant authentication mode to be switched back to* ***SAML identity provider (default) with Flexera option****. Once this is done, an administrator who has preserved a Flexera account can log in through the non-tenanted URL, and repair the configuration. To mitigate this (perhaps relatively small) risk of total lock-out, the SAML-only mode is recommended only where there are strict security requirements that prevent normal operations with the previous mode, which keeps the Flexera option enabled. At the very least, consider switching back to allow the Flexera option before making major configuration changes.*

  Tip: *You may revisit this screen at any time to change the mode setting, based on progress through your single sign-on implementation plan.*

7. Click **Save** (bottom right).

   If you used the URL option for the metadata XML file, the file is downloaded and checked. Your settings are validated, and you have an opportunity to fix any problems. When all is well, the configuration details are saved to the central compliance database for the current tenant.

For more information about these controls in the IT Asset Management Settings **General** page, see the online help.

Managing Operators

There are two main aspects of managing operators:

1. Creating credentials (or identities) for the operators in both IT Asset Management and your current identity provider, whether that be your SAML single sign-on solution or Flexera Account Management (or possibly both if you are in a mixed mode).
2. Assigning each operator to the appropriate role. In IT Asset Management, access and privileges are controlled by the **Role**(s) assigned to an operator. Without a role, an operator cannot view any pages of IT Asset Management, even though a valid identity may be used. Role assignment can only be performed by an administrator (that is, an operator who is already assigned to the `Administrator` role).

Another minor point may be to manage the expectations of operators using Flexera Account Management if you modify the **Timeout period** setting in the **Security** tab of the IT Asset Management Settings **General** page.

Creating an operator

Operator identities may be created manually. You may create the identity first in your SAML-compliant tool, after which there are two ways you can create the matching identity within IT Asset Management:

1. As an existing administrator in IT Asset Management, you can create the local account manually. In the **Account** field on the **Create an Account** page in IT Asset Management, enter the operator's email address. (This differs from the use of Windows Authentication, where you can select the account name from a drop-down list, imported from Active Directory.)

   - If you are using Flexera Account Management, this is the value that the operator uses to log in.
   - If you are using a SAML-compliant single sign-on identity provider (such as Okta), this is the account identity that is passed from your identity provider to IT Asset Management (the service provider) in the identity assertion. This is independent of the user name with which the operator logs into the identity provider.

   For more information, see the online help for the **Create an Account** page.

   Tip: *If you are planning to migrate identity management from Flexera Account Management to your single sign-on solution, it is very helpful if you can use the same* ***Account*** *value for both identity providers (which must be the operator's email address). This makes it possible to link either identity provider to the same account within IT Asset Management. As a result, you avoid a build-up of disabled accounts that can never be deleted from IT Asset Management.*

2. You can have the new operator try to log in. When the first attempt is made to log into IT Asset Management with an identity newly-registered in the single sign-on solution, the matching local account is automatically created in IT Asset Management.

   Important: *While the local account is automatically created, no roles are assigned to it. As a result, the operator receives a* `Sign In Failure` *message on this first login attempt (a secure outcome). To permit access, an administrator needs to add the appropriate role(s) for each new operator. For this reason, if you as administrator want to use this labor-saving approach, it is best done in collaboration with each of the new operators, so that they are not confused by the deliberate failure that, for security reasons, persists until roles are assigned.*

Ensuring administrator access

We have seen that operators must be assigned to roles *before* having access to IT Asset Management; and we have also noted that role assignment can only be performed by an administrator. When you are using your SAML-compliant, single sign-on solution, this could produce a chicken-and-egg situation, where no one can log in to make anyone an administrator.

The solution is that an administrator account (an operator who is assigned to the `Administrator` role) can be automatically created by an assertion from the SAML identity provider. The `Administrator` role is the only role that can be automatically assigned as a result of assertion by the identity provider.

To create an administrator automatically, arrange for your identity provider to include, in the appropriate identity assertion, a custom property called `FnmsAdmin`. This custom assertion needs to return a Boolean value (either `true` or `false`) to indicate whether the user is to be assigned to the `Administrator` role in IT Asset Management.

You may be able to set this property manually (for each identity), or programmatically. For example, your identity provider may support creating a group for all identities which are to have administrator properties. You may then be able to test whether the current identity is a member of that group, in order to return the `true` or `false` Boolean result. For example, when using the identity provider Okta, the values used are:

Attribute	Value
`Name`	`FnmsAdmin`
`Name format` (optional)	`Basic`
`Value`	`isMemberOfGroupName("Administrators")` ***Tip:*** *Function name is case-sensitive.*

Once the `FnmsAdmin` property is configured in your identity provider, it is passed to IT Asset Management, including on the first login attempt for a new identity. As seen in the previous section above, the first login attempt with the new identity creates the matching local account in IT Asset Management. When the assertion says `FnmsAdmin` is true, the assignment to the `Administrator` role is made automatically, and the initial login attempt succeeds. (Contrast this with previous comments, that *non*-administrator operators see a sign in failure until they have been assigned to one or more roles.)

Note: *In the account properties for this operator, the* ***Role*** *is set to `Administrator`, and because of the way this was asserted, it cannot be removed. If you wish to remove the administrator permission, you must first change the setting from the identity provider, and thereafter update the role assignment within IT Asset Management.*

Impact of session timeout

The session timeout setting (**Timeout period** on the **Security** tab of the IT Asset Management Settings **General** page) only affects those operators who *either*:

- Log in using Flexera Account Management accounts
- Log in using a SAML-2.0-compliant single sign-on solution (such as Okta), but that identity provider does not return the optional `SessionNotOnOrAfter` attribute within its assertion (that is, the identity provider does not return any timeout information).

When an operator first logs in, the identity provider usually sends information about the projected session expiry as part of its identity assertion for the authorized operator (and when the identity provider *does* do this, the value it provides controls the expiry, and the **Timeout period** value in the web interface is ignored).

Whether set by IT Asset Management (through your web interface settings) or by the identity provider, the timeout countdown is recorded in a cookie in the operator's browser, and a non-zero countdown is refreshed after each action that interacts with the central application server (that is, the timeout restarts after each relevant action, such as saving data, moving to a new page, or searching). If the operator "goes quiet" and the countdown expires, the next attempt at any relevant activity causes IT Asset Management to request a new authorization from the identity provider. This means, of course, that the login screen reappears when, after a 'quiet time', the operator attempts some relevant action in the web interface for IT Asset Management. To minimize disruption, after logging in again, the operator is returned to the page they were looking at when the timeout occurred.

Tip: *Configuration of your SAML-compliant identity provider may include setting* ***SAML Offset Minutes*** *to allow for clock variations between the identity provider and the service provider. Typically this value is set for 5 minutes, and this is added to the session expiry time. This means that if you choose a* ***Timeout period*** *of (say) 30 minutes, the session expires only after* ***35*** *minutes of inactivity.*