Videoscape Media Suite Installation Guide

Release 5.6

Americas headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Notices

Trademark Acknowledgements

Cisco, Cisco Systems, the Cisco logo, and the Cisco Systems logo are registered trademarks or trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries.

All other trademarks shown are trademarks of their respective owners.

Publication Disclaimer

Cisco Systems, Inc. assumes no responsibility for errors or omissions that may appear in this publication. We reserve the right to change this publication at any time without notice. This document is not to be construed as conferring by implication, estoppel, or otherwise any license or right under any copyright or patent, whether or not the use of any information in this document employs an invention claimed in any existing or later issued patent.

Copyright

Table of Contents

Chapter 1

Overview

Media Suite Overview

Cisco Videoscape™ Media Suite (VMS) is designed to help service providers give consumers the unified media experience they want, anywhere, at any time, and on any screen. Media Suite is a carrier-grade, cloud-based software platform for powering comprehensive multiscreen media services. It provides all the capabilities service providers need to integrate content from multiple sources, entitle and stream content across devices, and monetize content through customizable digital bundling.

Note Previous versions of Media Suite were branded as OpenCASE, so legacy references to OpenCASE (or OC) may appear in different forms throughout this installation guide. Some of this legacy naming is embedded in code and will remain within the product. Regardless, the names OpenCASE and Media Suite should be considered identical for installation purposes.

Installer Changes

Service Providers are increasingly seeing the need to install large-scale deployments or ones that need to be centrally managed. To address this growing need, the Media Suite 5.0 installer has been completely redesigned and rewritten to provide an efficient solution for installing, scaling, and upgrading many nodes. As part of the installer evolution, the process has also been changed from using a command line approach to one that primarily utilizes a user interface.

Further, the Media Suite installer has been completely redesigned with a focus on providing

1. a mechanism to centrally manage installations;
2. an automated distributed deployment system;
3. a mechanism to efficiently scale deployments across additional nodes;
4. a mechanism to easily scale deployments of additional components;
5. a streamlined upgrade process;
6. an improved user experience that requires less user input.

While Release 4.x installers store configuration information in local property files, VMS 5.x installers store relevant data in a central database/schema, which removes the need for duplicate entry of properties. Such centralization provides numerous benefits, such as improved versioning and module identification, which enables the installer to provide correct upgrade paths, and gives it the ability to deploy VMS updates with little user intervention. Lastly, VMS 5.x provides a dedicated management node and leverages a graphical user interface to help administrators control their VMS deployments.

Guide Audience

This installation guide is intended for system administrators, database administrators, Advanced Services personnel, and system integrators who will need to size, install, upgrade, or integrate with a deployment of Media Suite and related modules.

Guide Conventions

This guide follows the conventions identified in Table 1, below.

Table 1 Guide Conventions

Convention	Description	Example
`Coded text`	Indicates user-entered content, often presented as an example.	For **JBoss home/target directory**, type a target directory for your JBoss server (for example, `/usr/local/jboss-5.x.0.0`).
Highlighted text	Indicates an interface element or an option presented for consideration.	For **Install type**, select the default **[1]**.
Hyperlinked text	A link to an online source.	To install the latest version of this software, visit www.cisco.com
In the Linux chapters, where you are required to enter content on a command line, the requirement to press the ENTER key is assumed.		

Chapter 2

Deployment Considerations

By design, Media Suite is intended to be flexible and customizable. Those characteristics allow the software to serve customers with many different use cases and hardware requirements. As such, one document cannot accurately elaborate on all supported usage scenarios. The following chapter, however, provides general information to inform a more detailed conversation. Consider the following guidelines whenever you consult with your Cisco Advanced Services representatives while planning any production-level implementation.

System Requirements

The following system requirements are enforced and must be met (at a minimum) prior to installing Media Suite 5.x:

- x86_64 Red Hat Enterprise Linux 6
- Minimum (enforced) 3000 MB RAM, as reported by `free -m`
  Recommended 8000-12000 MB RAM per virtual machine.

> **Note** Deployment strategy specifics will affect the optimal JVM requirements. Please consult with Advanced Services for the correct memory allocation recommendation for your system.

- Minimum 8000 maximum open file handles, as reported by `ulimit -n`
- Minimum 2000 maximum concurrent processes, as reported by `ulimit -u`
- A `ciscovms` user (will be created if not found)
- The $HOSTNAME variable and the hostname command must return matching, non-localhost values
- Pinging the hostname (when logged into the machine) must not result in replies from localhost or a loopback address (127...*)
- RNG daemon to speed up SecureRandom
  To avoid unexpected behavior, disable SELinux or set permissions to a minimal level. The 5.x install process will warn you if SELinux is not disabled.
- Permission to install and remove RPMs (usually as root or sudo)
- Ensure that the following commands in the path: `rpm`, `sed`, `curl`, `ping`
- All nodes, except RemoteFS, must have access to the Media Suite database/schema
- HTTP requests from any node to itself must not be blocked (proxies may cause issues; check $HTTP_PROXY)

- Multiple active NICs, especially wireless ones, can cause unpredictable JGroups behavior; ensure the hostname resolves to the IP mapped to the primary (wired is possible) NIC
- VMS must be accessed via standard port numbers (HTTP=80, HTTPS=443). A load-balancer/ proxy is required

mod_cluster

mod_cluster is an Apache module that can be used as a load balancer. For convenience, mod_cluster is recommended (but not required) for small, non-critical (i.e. non-production) test deployments. For details on setting up mod_cluster, see "Configuring Apache for VMS" on page 95.

Network Throughput

The fastest network throughput available, however, 1 Gb Ethernet should be used at a minimum.

Suggestion: 10 Gb Ethernet

Application Servers

The following general recommendations apply to each virtual machine that will be running Media Suite application servers:

CPUs: 8 virtual CPUs

RAM: 16 GB

Hard Drive: 60 GB (SSD is recommended for maximum performance)

The drives will be used to store the operating system, Media Suite installation, log, and other files.

Suggestion: Cisco's UCS Blade Servers

Note As a best practice, you should plan on establishing redundancies comprised of two instances of each module on two separate physical machines.

Load Balancers

The following diagrams depict how SSL offloading should be performed in Media Suite. This process is accomplished via port mappings that redirect incoming server requests to relevant ports for either the HTTP or AJP protocols.

Figure 1 SSL Offloading for HTTP

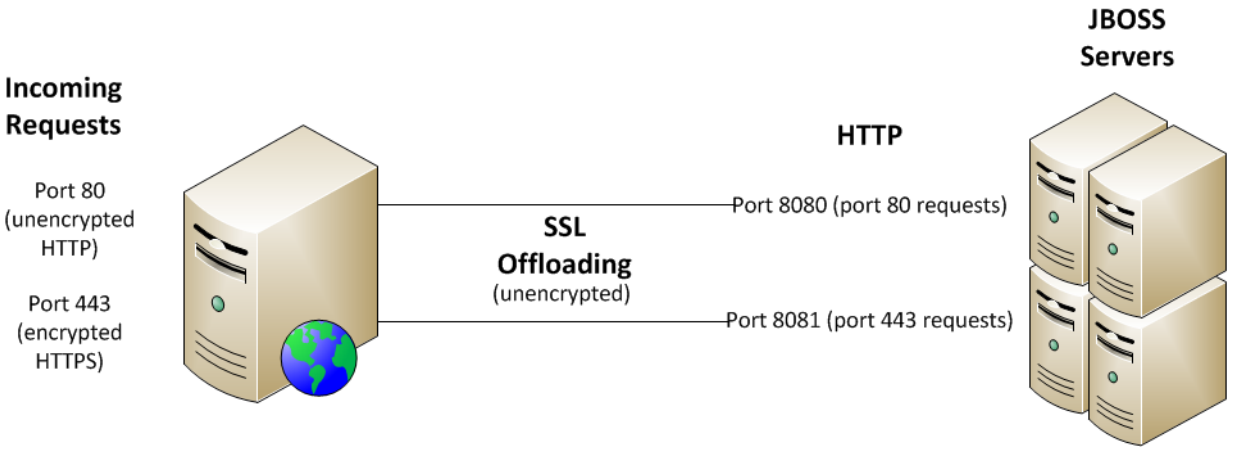
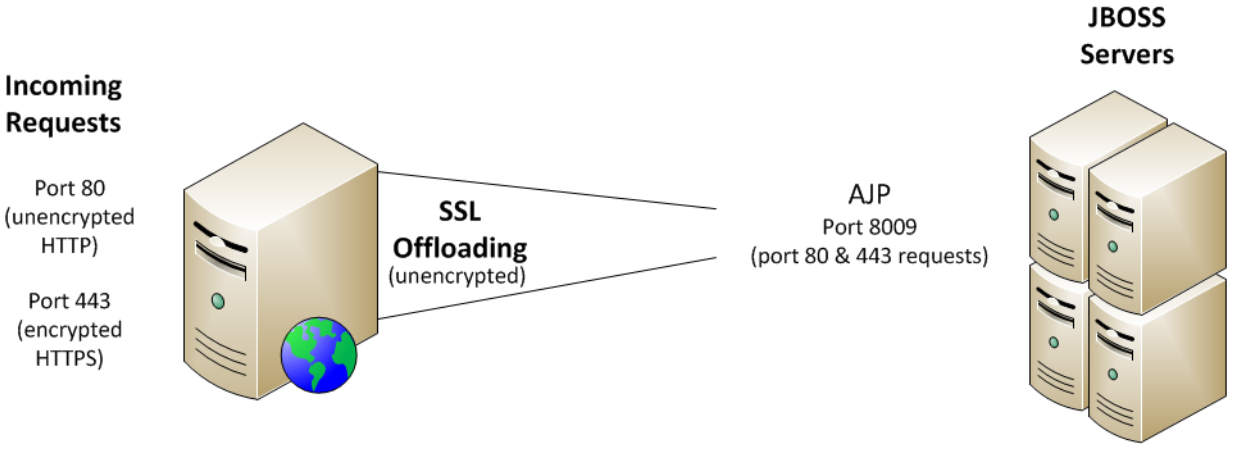


Figure 2 SSL Offloading for AJP



Database Servers

Consult with your database administrator for proper deployment sizing and redundancy considerations.

Licensing Servers

Licensing servers (whether CTM or VMS Windows services) provision licenses for DRM encrypted content. Consult with your Advanced Services representative regarding sizing and other deployment considerations for licensing servers.

Encoding/Encrypting Servers

These servers encode and/or encrypt content. For details on encoding/encrypting server requirements, refer to the Cisco Transcode Manager (CTM) installation guide.

Inter-Server Communication

Any Linux and Windows servers in your deployment must communicate with each another in order to maintain proper synchronization while performing their relevant tasks. You should ensure that all appropriate firewall channels are open, and that the correct ports are accessible. This implies that any internal Windows server communication ports should be accessible by your Linux servers and vice versa.

To enable the stated protocols and functionality, the following ports should be open between VMS and various components of the software infrastructure:

Table 2 Communication Ports

Port	Details
80	(load-balancer) HTTP
443	(load-balancer) HTTPS
1090	(JBoss) JMX connector
1098	(JBoss) RMI
1099	(JBoss) JNDI
1100 1101 1102	(JBoss) HA-JNDI
4444	(JBoss) JRMP
4445	(JBoss) Pooled Invoker
4446	(JBoss) Unified Invoker
4447	(JBoss) HA JRMP
4448	(JBoss) HA Pooled Invoker
4457	(JBoss) Messaging
4712	(JBoss) TS recovery
4713	(JBoss) TS status manager
8009	(JBoss) AJP
8080	(JBoss) HTTP
8081	(JBoss) HTTP secure
8083	(JBoss) Web service
56000 56001 56002	(FFS) Voldemort
57100 57101 57102	(JGroups) EHCache replication
57200 57201 57202	(JGroups) EHCache replication
57300 57301 57302	(JGroups) EHCache replication

Table 2 Communication Ports

Port	Details
57400 57401 57402	(JGroups) EHCache replication
57600 57601 57602	(JGroups) ESB/JMS
57700 57701 57702	(JGroups) JBM-DATA
50389	(OpenAM) Directory Server
54444	(OpenAM) Management
58989	(OpenAM) Replication

Media Suite Components

Media Suite is comprised of a number of individual components that work in conjunction with one another to implement system-wide functionality. The following section describes components and hypothetical component groupings while providing related information that should be considered when planning a deployment.

OpenAM

The OpenAM (previously called OpenSSO) component authenticates user interface and Web Service users and manages user interface session handling within Media Suite.

When planning VMS deployments, it is critical to consider redundancy for your OpenAM component. With that in mind, OpenAM is installed using one primary and one or more secondary nodes so that if a primary node fails for any reason, a secondary node will take over user authentication duties. For this reason, these nodes should reside on separate hardware instances. In addition to its normal authentication duties, the primary OpenAM node is also responsible for replicating user credentials to the secondary node.

Note Since Media Suite is stateful, any sessions being managed on the primary OpenAM node will be lost in the event of the failure of that node. This would result in users having to log back into the user interface in order to continue their work.

If hardware costs are an issue, it is possible to reduce expenses by installing OpenAM on two separate physical servers alongside CMS components. This hardware separation still provides the required redundancy for OpenAM functionality.

Content Management System

The Content Management System (CMS) is the component that handles Content Management duties.

Note Depending on how a client implements their storefront, the Content Manager may receive a large number of product search transactions. In that case, the Content Manager component should be separated out into its own cluster.

Producer

Producer is an optional module that augments the capabilities of Media Suite in the realm of metadata importation, creation, and management. As such, Producer provides a wizard that allows operators to manage specific bundle types or to edit or enter required data quickly and efficiently.

When deployed, Producer should be installed on the same nodes as Content Manager as those two modules work in conjunction with one another. For details, see "Recommended VMS Deployment Diagram" on page 12.

Workflow System

The Workflow System is a grouping of components that are collectively intended to handle workflow management services. Components included within a Workflow System cluster include:

- ESB (Enterprise Service Bus)
  Provides a framework for running the default Media Suite ESB services.
- Deployment-Specific ESB Services
  During the installation process, you will be prompted to install ESB services that are relevant to your deployment.
- Content Processor
  Provides functionality related to content processing.
- Workflow Services
  Provide functionality specific to the management and execution of workflows.

From a deployment perspective, the workflow system should consist of a minimum cluster of two distinct hardware servers. This arrangement will provide the required performance and redundancy capabilities.

Entitlement Management System

The Entitlement Manager (EM) System is a grouping of components that create, entitle, and manage digital rights. Typically this component handles the largest amount of traffic of all of the components.

An EM cluster has two components:

1. EM Linux Component
2. EM Windows Component

Whenever possible, it is advisable to split out these components so that there are:

- Two physical Linux instances (at a minimum) for redundancy and performance
- Two physical Windows instances (at a minimum) for redundancy and performance

The physical divisions above provide the flexibility to add new server instances in the future if your deployment's performance needs increase.

Note During installation, an EM database/schema will need to be created to store EM data. This database will log DRM transactions and, depending on the subscriber traffic, could grow large over time.

Administration System

The Administration System contains components that provide functionality across all of Media Suite. Those components are:

- OpenCASE-Common
  Provides system configuration, and administrator management functionality.
- SSO
  Provides user authentication functionality as described in "OpenAM" on page 7.

EPG Manager

An optional Electronic Program Guide (EPG) module is responsible for the ingestion and processing of EPG feeds. Those EPG feeds consist of a large amount of data that is ingested and indexed once per day.

Whenever possible, it is advisable to split out these components so that two servers are used to provide redundancy and performance scalability. When EPG functionality is required, three additional components must be installed: an Index Master Search Manager node, and Search Config Module. Those components will be discussed, in general, in the following section.

Index Master

The Index Master is an internal EPG component that communicates with Content Manager and EPG Manager to build a master index. This master index will later be copied and used by the Search Manager API to retrieve EPG information to make it available to the end consumer.

Points to consider with Index Master include:

- That only one Index Master can be active at a time. If redundancy is required, then a standby node can be manually brought online.
- Once a new Index has been created on the Index Master node, a backup should also be created. For details on creating this Standby Index Master, refer to the EPG Installation Guide.
- The index can be published as part of an EPG workflow process or it can be manually published within the Media Suite/EPG interface if required. For further details, consult the *EPG User Guide*.

Search Manager Node

The Search Manager node hosts the combination of an Index Slave and the Search Manager API. The Index Slave polls the Index Master (every 30 seconds) and is responsible for copying new indices locally so that they can be efficiently searched by the Search Manager API.

Points to consider with the Search Manager Node include:

- While there can only be one Index Master, there may be many Index Slaves.
- Each Index Slave must reside on its own node.
- Each Index Slave is paired with one Search Manager API instance.
- Additional nodes can be deployed as required for additional performance.
- On startup, the Index Slave pulls the latest index. New slave nodes can only be added to the system once an operator has confirmed that the existing slave has successfully pulled the index.

Search Config Manager

The Search Config Manager is a mandatory component that enables the configuration of Search Manager return fields and other parameters from the Media Suite user interface. For specifics, see "System Configuration" on page 105.

Timeshift TV

Timeshift TV is an optional Videoscape module that works in conjunction with Media Suite, the Electronic Program Guide (EPG), and other Cisco software and hardware components to enable, configure, and manage the recording and playback of video assets.

When deployed, Timeshift TV should be installed on the same nodes as the EPG module as those two modules work in conjunction with one another.

Peripheral Components

The following section describes components that, although a not run as part of core VMS, are still integral parts of its capabilities.

RemoteFS

The Remote File Server (RFS or RemoteFS) component can optionally be setup to act as an intermediary between VMS and filesystems in various geographic locations. The chief benefit of using RFS (over local filesystem access) is that the RFS servers can perform the work involved in copying files from one repository to another. This process bypasses the VMS servers, and thus does not impose any resource penalties when transferring files. In addition, RemoteFS does not require an agent and has no dependencies whatsoever on Media Suite or its database. For details on installing RemoteFS, see "Installing RemoteFS" on page 71.

Report Manager

The Report Manager component can generate predefined or custom reports based on details related to system content, household accounts, or provisions issued by Media Suite. The component does not require dedicated resources because its duties are not critical or time sensitive.

Points to consider when implementing Report Manager include:

- Report Manager should be installed on its own server or on a server that is not used for production purposes.

- Reports can be run offline when needed.
- Redundancy and scalability are not required for this component.
- Data used by Report Manager should be pulled from a replicated offline database in order to avoid putting unnecessary load on the live database.

Installer Automation API

Media Suite includes an installer automation service that is presented as a SOAP API. For details on the API, see “Installer Automation API” on page 97.

Virtualization

Media Suite can be deployed by utilizing one or more virtual machines per individual server. Oversubscription is not supported for those virtual memory, processor, network, or disk resources.

Recommended Deployment

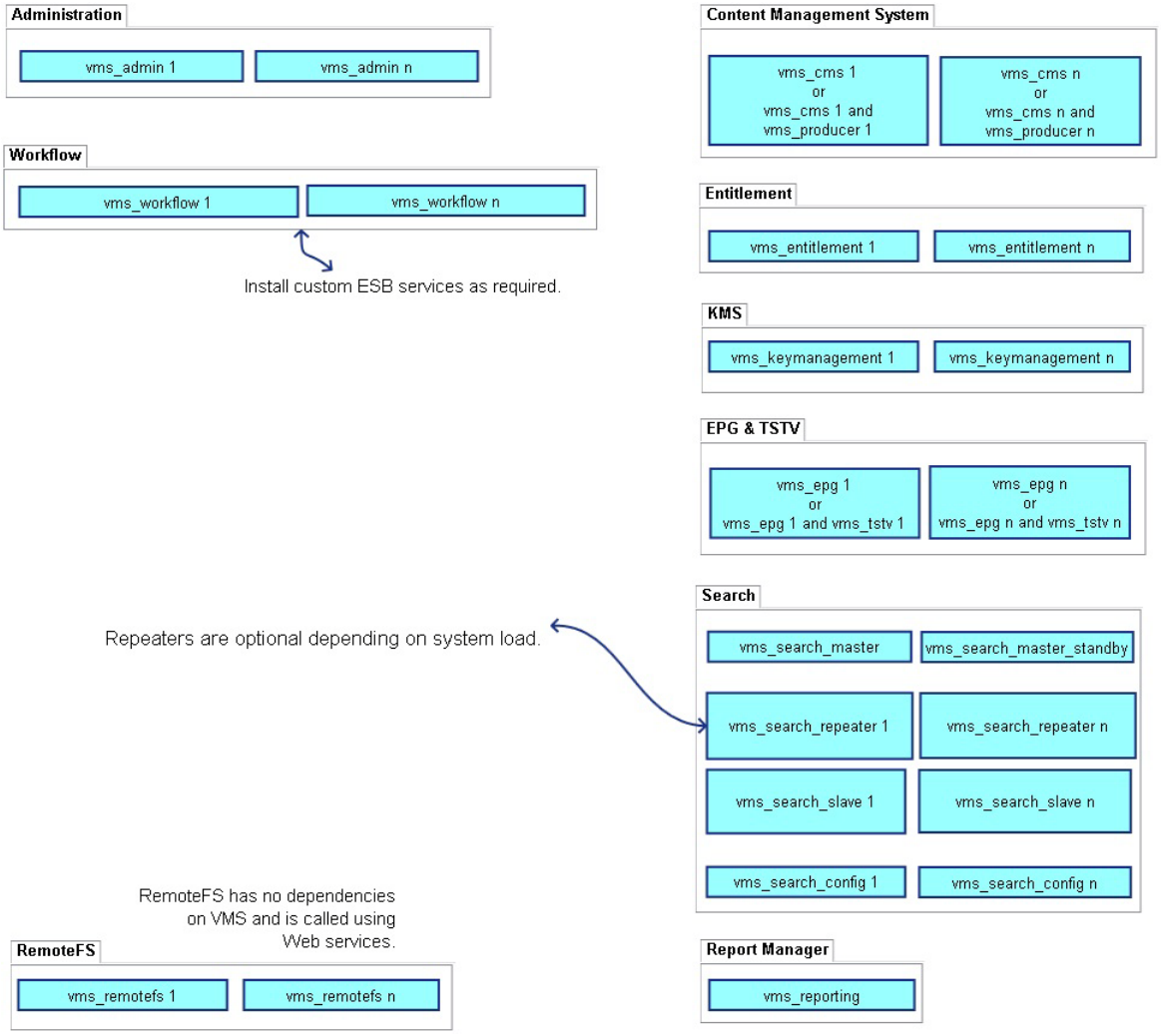
The diagram below illustrates a recommended deployment for Videoscape Media Suite Release 5.0 and later. This sample demonstrates one deployment strategy, but you should always decide a final architecture based upon the individual use cases of your deployment.

One major difference between a 4.x installation and a 5.x installation is that install components should be viewed from a perspective of packages instead of modules. For organizational purposes, those packages can be grouped into functional groupings that are defined as follows:

Administration	EPG & TSTV
Content Management System & Producer	Search
Workflow	Key Management Server
Entitlement	Report Manager
RemoteFS	

Each functional grouping includes a box that represents individual nodes in the deployment. For example, the Workflow functional grouping has “Workflow 1” to “Workflow n” nodes that provide redundancy and scalability.

Figure 3 Recommended VMS Deployment Diagram



VMS Packages

The following packages are specified in the recommended deployment.

Table 3 Module and Package Names

Module	Package Name
Administration	vms_admin
CMS	vms_cms or vms_cms and vms_producer
Entitlement	vms_entitlement
Reporting	vms_reporting
Key Management (KMS)	vms_keymanagement

Table 3 Module and Package Names

Module	Package Name
EPG	vms_epg (without TSTV) or vms_epg and vms_tstv (with TSTV)
Workflow	vms_workflow & vms_server_esb & vms_ocesb_epg (if using EPG) vms_workflow & vms_server_esb & vms_ocesb_producer (if using Producer)
Search	• vms_search_master (only one instance-wide) • vms_search_master_standby (only one instance wide, optional for HA) • vms_search_config • vms_search_slave • vms_search_repeater (optional, if repeater functionality required)

Notes:

- Review the Installer Manager packages screen, or open the `<version>.manifest` file, for a breakdown of modules and packages.
- To identify dependencies for a package, run `rpm -qp --requires /path/to/vms_package.rpm`
- To identify what a package provides, run `rpm -qp --provides /path/to/vms_package.rpm`
- Do not mix versions of packages from different releases.

Notes on the Recommended Deployment

The following notes provide further details related to the recommended deployment.

VMS Infrastructure

Except for the management node (which is always separate and contains `vms_installer`):

- Install `vms_base` on all nodes.
- Install `vms_server_standard` on all non-workflow nodes.

Workflow Nodes

- All workflow nodes require `vms_workflow` and `vms_server_esb`.
- Also install `vms_ocesb_epg` on all workflow nodes if EPG is being used. `vms_ocesb_epg` provides EPG ingest capabilities.
- Also install `vms_ocesb_producer` on all workflow nodes if Producer is being used.

Installing EPG

When installing EPG, choose both `vms_epg` and `vms_tstv` (if using TSTV).

Search Components

vms_search_master should have one vms_search_master_standby for redundancy and failover. The load balancer configuration, however, should only point to vms_search_master.

Search slaves (vms_search_slave) may optionally be connected (and made aware of) a search repeater (vms_search_repeater), which can each be connected (and made aware of) the Search Master (vms_search_master). The optional repeaters may be added depending upon system load requirements. Otherwise, search slaves are directly connected to the Search Master.

Both search slaves and repeaters are available in versions for deployments with or without Recommendations support.

RemoteFS

RemoteFS is different from all other packages. It is completely independent, does not run an agent, and has no RPM, database, or other dependencies. RemoteFS can only be installed on Linux and not Windows.

Apache mod_cluster

When using mod_cluster, you must install vms_modcluster on all nodes.

Multi-instance Deployments

Multi-instance deployments are not supported in Media Suite 5.x. A multi-instance deployment is considered one that runs multiple JBoss instances that are bound to different IP addresses on a single server.

Rolling Upgrades

To enable your system with the capability to perform rolling upgrades, it is important to plan and structure your deployment with that in mind. For further advice on planning and performing a rolling upgrade, see “Planning Rolling Upgrades” on page 47.

Chapter 3

Preparing a Database/Schema for Media Suite

This chapter provides general considerations for preparing a database for use with Media Suite. In addition, specific instructions are included on preparing either an Oracle database for a Media Suite installation.

Note Although specific recommendations are made in this section with regards to preparing a database/schema for Media Suite, you should consult with your database administrator on matters regarding general database setup and for information on configuring security settings in accordance with your company's standards.

Supported Database

The following database is supported for a Media Suite installation on Linux.

Table 4 Database Requirements

Software	Type	Version
Database	Oracle	11.2.0.1 Standard or Enterprise Editions

Details to Investigate

When researching and planning your deployment, it is advisable to discuss and answer the following questions with your Cisco Advanced Services representative and relevant IT staff:

- How many database servers will you require?
- What permissions does the database user need?
- What amount of disk space will your database require?
  500 GB is generally a recommended minimum to allocate room for growing log files.

Preparing Oracle for VMS

The following section lists considerations for preparing Oracle for a Media Suite installation. One significant difference from VMS 5.x installations is that the required Oracle database driver is now provided with your Media Suite installation package.

Note RAC style connection strings (with no line breaks) are supported by Media Suite 5.0+.

Configuring Oracle for Extended Character Sets

This section provides details on how to configure an Oracle database to support extended character sets.

To configure an Oracle database for extended character sets:

1. Login as `sysdba` to acquire the proper database permissions.
2. Execute the following commands on the Oracle SQL admin console:

```
SQL> SHUTDOWN IMMEDIATE;     -- or NORMAL
SQL> STARTUP MOUNT;
SQL> ALTER SYSTEM ENABLE RESTRICTED SESSION;
SQL> ALTER SYSTEM SET JOB_QUEUE_PROCESSES=0;
SQL> ALTER DATABASE OPEN;
SQL> ALTER DATABASE CHARACTER SET internal_use UTF8;
SQL> SHUTDOWN IMMEDIATE;     -- or NORMAL
SQL> STARTUP;
```

   Alternately, this procedure can be performed through the Oracle graphical user interface. Please consult your database administrator for further information on such procedures.

The Oracle database configuration string format is no longer parsed and can be any string that the Oracle thin driver will support. The only stipulations is that the string must be on one line (i.e. not include any line breaks). A sample Oracle connection string is shown in the following table:

Table 5 Oracle Database Connection String for Media Suite

Item	Description
Connection String	jdbc:driver:type:@DB_server_IP_or_FQDN:DB_server_listen_port:SID (for example: jdbc:oracle:thin:@1.2.3.4:1521:sampleDb)

Chapter 4

Preparing Linux for Media Suite

This chapter includes information on Linux pre-installation requirements or procedures that are needed to prepare for a Media Suite installation.

Software Requirements

The following software is required to support your Media Suite installation on Linux. The JBoss application server, JBoss ESB, OpenAM, and the installer script are provided in the installation package. All other software must be acquired separately.

Table 6 Linux Requirements - Provided Software

Software	Type	Version
Application Server	JBoss	5.1
Enterprise Service Bus	JBoss	4.11
OpenAM	Framework	10
JDK	Development Kit	1.7 update 51

Note We strongly recommend that (once any VMS migrations are complete) you remove all other JDKs entirely from your VMS servers other than the version that is provided. Media Suite is completely self-contained from a JDK perspective and any other versions are unnecessary and may confuse administrators and lead to difficulties while troubleshooting.

Table 7 Linux Requirements - Software to Obtain

Software	Type	Version	Notes
Operating System	RedHat OS	6	Enterprise Linux

Deprecated Requirements

The following items that were previously required for Media Suite 4.x no longer needs to be provided by the user for a Media Suite installation:

- JDK (of any kind)
- JAVA_HOME environment variable
- JCE (except Entitlement encryption migration – see below)
- Ant
- Oracle database drivers

Preparing your Linux Environment

Prior to installing Media Suite, your environment must be properly setup and configured with items related to your database and Java. This section explains additional considerations information on installing and configuring those items.

To configure global settings for your Linux environment:

1. Ensure that either an Oracle should be installed, configured, and running so that it can be accessed by the Media Suite Linux environment. For details, see "Preparing a Database/ Schema for Media Suite" on page 15.

Synchronizing with an NTP Server

When running Media Suite, all servers should be set to the correct time. To ensure that this is the case, Media Suite validates synchronization between the nodes and the database. If time is not synchronized, then the Agent will throw a warning message to inform administrators. Network time protocol synchronization is particularly important between Media Suite and VDS-IS servers, as it is a requirement for utilizing URL signing functionality. In Linux, server synchronization is commonly accomplished by several means through the use of an NTP server. Consult your network administrator for suitable NTP synchronization instructions for your deployment. The following section describes one approach for performing this synchronization.

To synchronize with an NTP Server:

1. Create a configuration file in the following location:
   `/etc/ntp.conf`
2. Insert the following contents into the file.

```
# Permit time synchronization with the your time source, but do not
# permit the source to query or modify the service on this system.

restrict default kod nomodify notrap nopeer noquery

server {NTP_server_1_name}
server {NTP server_2_name}
server (NTP_server_N_name}

restrict 127.0.0.1
driftfile /var/lib/ntp/drift

keys /etc/ntp/keys
```

Each time the Media Suite server is rebooted, its clock will synchronize with one of the provided NTP servers.

Adding a User

- The RPM installation process, when performed with root or sudo privileges, will create the `ciscovms` user and group
- To create a user manually: `# useradd ciscovms`

Note The `ciscovms` user and group are mandated.

Setting ulimit Values

The following section details changes that need to be made to configure ulimit values.

`/etc/security/limits.conf` (RHEL/CentOS 6.x and Fedora)

```
* soft core 0
* hard core 0
* soft nproc 2048
* hard nproc 4096
* soft nofile 8192
* hard nofile 8192
```

`/etc/security/limits.d/90-nproc.conf` (RHEL/CentOS 6.x and Fedora)

```
* soft nproc 2048
* hard nproc 4096
```

Note You will need to reboot in order for ulimit settings to take effect.

Network Structure

Load Balancer Health Checks

The load balancer should be configured to perform regular health checks on VMS modules and not direct traffic to nodes (or modules on nodes) that have failed a health check. A good health check is to generate a WSDL for a key service in a module, and ensuring that it returns a 200 (not 302, 404, 500, or anything else). The recommended WSDLs for each module are listed below:

- OpenAM (OpenSSO):
  `<server>:8080/opensso/identityservices?wsdl`
- OpenCASECommon:
  `<server>:8080/opencase/webservices/permission-service?wsdl`
- ContentManager:
  `<server>:8080/ContentManager/webservices/component-service?wsdl`
- ContentProcessor:
  `<server>:8080/ContentProcessor/webservices/contentfile-service?wsdl`

- WorkflowService:
  `<server>:8080/WorkflowService/webservices/processdefinition-service?wsdl`
- EntitlementManager:
  `<server>:8080/EntitlementManager/webservices/accountmanager-service?wsdl`
- SearchConfigManager:
  `<server>:8080/SearchConfigManager/webservices/search-config-service?wsdl`
- LinearManager:
  `<server>:8080/LinearManager/webservices/program-service?wsdl`
- OpenCASE_ESB-core:
  `<server>:8080/OpenCASE_ESB-Core/ebws/OC_ESB_INTERNAL_EVENT/EventService?wsdl`
- SearchManager:
  `<server>:8080/opencase/sm/search-manager-service/cache?wsdl`

Planning Ahead on Security

Since you will be consulting with your network administrator regarding NTP server synchronization, you may also wish to take that opportunity to discuss the process for implementing SSL data transmission. Specific URLs should be secured immediately after a Media Suite installation. For more information on this requirement, see "Securing OAuth Data Transmission" on page 41.

Note The `keytool` command referred to in the following section is available on any server with VMS packages (and consequently the JDK) installed on it.

To secure JBoss on a node to handle SSL requests:

1. Ensure that you have all required SSL certificates.
   There can be one or more certificates, and they must cover the entire certificate chain.
2. Make a client truststore in JKS format.
   To generate a truststore:
   `keytool -genkey -keystore truststore_name`

   Where:
   `-genkey` Is the command for generating the keys.

   `-keystore truststore_name` Sets the name (and optionally the path) of the keystore you are creating or referring to (if the keystore already exists).

   Note You will be prompted for the `key` and `keystore` passwords. When you are prompted for the key password, you will also be given the option of making it the same as the keystore password.

   ---

3. Importing Certificates into Client Truststore
   To list entries within a given store:
   `/opt/cisco/vms/var/jdk/bin/keytool -list -keystore /path/to/store`

To import a certificate:

```
/opt/cisco/vms/var/jdk/bin/keytool -importcert
-file /path/to/cert
-alias <choose_an_alias>
-keystore /path/to/client.truststore
```

When installing VMS, the client truststore information will be uploaded via the Installer Manager user interface. For details, see "Understanding Passwords and Security" on page 38.

Where SSL Communications Are Applicable

The process of securing JBoss to handle SSL requests is applicable in the following instances:

- where any endpoints are HTTPS
- for the Key Management Server. entitlements, or any other secure communications
- for any external services, (such as DPS, VOSM, SMRS) that need to be accessed over HTTPS
- any components that should be secured according to your deployment's security requirements

Media Suite Component Dependencies

The following section details dependencies that must be taken into consideration at various levels when installing Media Suite. Those dependencies are enforced on two levels:

1. The global level
   This occurs when selecting packages through the Installer Manager user interface. These modules within these packages have global dependencies across the deployment that will be checked when clicking **Validate Dependencies** or when saving your package selections.
2. The RPM package level
   This occurs when installing individual packages on nodes. Prior to being installed, each RPM package checks that all local dependencies have been fulfilled.

Note Version numbers are enforced for all dependencies so that if versions do not match for all packages when attempting to install, an error will be thrown and the installation process will abort.

Global Dependencies

The following table lists global dependencies that are enforced within the Installer Manager interface. These are dependencies between various Media Suite modules and are applicable instance wide and are not specific to any node. All dependencies must be version matched.

Table 8 Global Dependencies

Module Name	Dependencies
Content Manager	JBoss Application Server Administration OpenAM Single-sign-on
Entitlement Manager	JBoss Application Server Administration Content Manager OpenAM Single-sign-on
Content Processor	JBoss Application Server Administration Workflow Service Content Manager OpenAM Single-sign-on Core ESB Service JBoss Enterprise Service Bus
Workflow Service	JBoss Application Server Administration Content Processor Content Manager OpenAM Single-sign-on Core ESB Service JBoss Enterprise Service Bus
Administration	OpenAM Single-sign-on JBoss Application Server
Report Manager	JBoss Application Server Administration OpenAM Single-sign-on
Key Management	JBoss Application Server Administration OpenAM Single-sign-on
EPG	JBoss Application Server Administration Content Manager OpenAM Single-sign-on JBoss Enterprise Service Bus EPG Ingest ESB Service Workflow Service
Installation Agent	JBoss Application Server Installer
Search Configuration	JBoss Application Server Administration Search Manager OpenAM Single-sign-on

Table 8 Global Dependencies

Module Name	Dependencies
Time-Shift TV Manager	JBoss Application Server Administration OpenAM Single-sign-on EPG Manager Key Management
Recommendation Engine	JBoss Application Server Administration Entitlement Manager Search Manager OpenAM Single-sign-on
Recommendations Index Slave	Recommendations Engine
Search Index Repeater	Recommendations Engine
Search Manager	JBoss Application Server Administration OpenAM Single-sign-on Search Configuration Manager
Armada ESB Service	JBoss Application Server JBoss Enterprise Service Bus
Bind ESB Service	JBoss Application Server JBoss Enterprise Service Bus
Collator ESB Service	JBoss Application Server JBoss Enterprise Service Bus
Metadata Collator ESB Service	JBoss Application Server JBoss Enterprise Service Bus
Core ESB Service	JBoss Application Server JBoss Enterprise Service Bus
DRM Packager ESB Service	JBoss Application Server JBoss Enterprise Service Bus
Encoder ESB Service	JBoss Application Server JBoss Enterprise Service Bus
Productize ESB Service	JBoss Application Server JBoss Enterprise Service Bus
XML Reader ESB Service	JBoss Application Server JBoss Enterprise Service Bus
XML Transform ESB Service	JBoss Application Server JBoss Enterprise Service Bus
CDSTV Publisher ESB Service	JBoss Application Server JBoss Enterprise Service Bus
EPG Ingest ESB Service	JBoss Application Server JBoss Enterprise Service Bus Workflow Service EPG Manager
OpenAM Single-sign-on	JBoss Application Server Administration

Table 8 Global Dependencies

Module Name	Dependencies
JBoss Enterprise Service Bus	JBoss Application Server JBoss Enterprise Service Bus
JUDDI	JBoss Application Server JBoss Enterprise Service Bus
JBPM	JBoss Application Server JBoss Enterprise Service Bus
JBoss Application Server	Installation Agent
JBoss Enterprise Service Bus	JBoss Application Server
Search Index Master	JBoss Application Server Administration OpenAM Single-sign-on
Search Index Standby	Solr Index Master
Search Index Slave	JBoss Application Server Administration OpenAM Single-sign-on Solr Index Master
Search Index Repeater	JBoss Application Server Administration OpenAM Single-sign-on Solr Index Master
JBoss Mod-Cluster	JBoss Application Server
Producer	Content Manager Content Processor

RPM Package-Level Dependencies

The following table lists RPM package-level dependencies and conflicts. In other words, dependent packages should be installed in their order of dependency on the same node, while conflicting packages should never be installed on the same node. All dependencies must be version matched.

One example of an installation order that adheres to dependencies would be:

1. vms_base
2. vms_server_esb
3. vms_workflow
4. vms_ocesb_epg

Table 9 RPM Package-Level Dependencies

Package Name	Dependencies	Conflicts
vms_base	none	vms_installer
vms_installer	none	vms_base
vms_server_standard	vms_base	vms_server_esb
vms_server_esb	vms_base	vms_server_standard

Table 9 RPM Package-Level Dependencies

Package Name	Dependencies	Conflicts
vms_admin	vms_server_<either>	
vms_cms	vms_server_<either>	
vms_entitlement	vms_server_<either>	
vms_keymanagement	vms_server_<either>	
vms_reporting	vms_server_<either>	
vms_epg	vms_server_<either>	
vms_tstv	vms_epg	
vms_reporting	vms_server_<either>	
vms_search_config	vms_server_<either>	
vms_search_master	vms_server_<either>	vms_search_master_standby
vms_search_master_standby	vms_server_<either>	vms_search_master vms_search_repeater
vms_search_slave	vms_server_<either>	vms_search_repeater
vms_search_repeater	vms_server_<either>	vms_search_slave
vms_server_esb	vms_workflow	(These two packages must always be installed together as they are dependent upon one another.)
vms_ocesb_epg	vms_workflow	
vms_workflow	vms_server_esb	(These two packages must always be installed together as they are dependent upon one another.)
vms_modcluster	vms_server_<either>	
vms_producer	vms_cms	

Next Steps

Once you have followed the preparation steps in this chapter, your Linux environment should be ready for installing Media Suite. Proceed to the next chapter for detailed instructions.

Chapter 5

Installing Media Suite on Linux

Installation Overview

This chapter provides details for installing a new (clean install) instance of Media Suite onto your Linux environment. Before proceeding, ensure that you have completed the steps in "Preparing a Database/Schema for Media Suite" on page 15 and "Preparing Linux for Media Suite" on page 17.

The instructions provided in this chapter provide guidance on how to install Media Suite components onto a multi-server environment. If you are performing an upgrade from an earlier version of Media Suite, also consult "Upgrading Media Suite" on page 43.

Note Given the improved processes for the 5.0 installer, entire installation scripts do not need to be run against each deployment node in order to install components as they were in 4.x.

The general steps for a Media Suite installation are as follows:

1. Ensure that a clean database is available for use (clean installs only).
   For upgrades, the existing databases are used (if applicable) during the consolidation process.
2. Perform the Linux preparation steps on each node.
3. Choose an install management node.
4. Copy and install the installer RPM onto the management node.
5. Run the Installer Manager.
6. Perform the following steps within the Installer Manager user interface:
   - select the install packages that you need to install instance wide
   - configure endpoints
   - configure the database/schema
   - provide passwords for security-enabled services
   - enter module-specific settings for Media Suite
   - create or update VMS database/schema for the specific modules
7. Copy and install the dependencies and required RPMs for each deployment node.
8. Copy the `database.oci` file to the specified location on each deployment node.
9. Launch the agents on each node.
10. Return to the Installer Manager to monitor the status of the nodes as they startup and register with the system.

11 Login to Media Suite to begin your work within the application.

Understanding the Install Manager Node

The Install Management Node is critical to installing, maintaining, and troubleshooting Media Suite. As such, it should be a dedicated always-on management node that does not have VMS running on it. When troubleshooting your deployment, this is the first place you should visit as it provides useful debugging information such as installed components, node statuses, error messaging, and logs.

Deployment Recommendations

If you have not already done so, read "Planning Ahead on Security" on page 20 to ensure that you install the required components on the first node in order to avoid dependency issues during the installation process. In addition, you should review the "Recommended Deployment" on page 11, which will provide recommended component groups and an installation order for nodes. Following these recommendations should make your deployment process go smoothly.

Starting a Media Suite Install

The following section details the steps for performing a Media Suite installation via the Installer Manager user interface. Media Suite also includes an installer automation service that is presented as a SOAP API. For details on the API, see "Installer Automation API" on page 97.

To perform a Media Suite installation

1 Navigate to the installer directory.
`cd /opt/cisco/vms/var/installer`

2 Type `./install.sh`

Note If you get a permissions error when trying to execute the install script, type `chmod +x install.sh` to set execution permissions on the file for the current user.

---

3 Provide connection information for the database that you are using.

4 Type a password for the Installer Manager application. The username is always `admin`.

5 Wait for Installer Manager to start as indicated by the `Launching Installation Manager...` message. If you need to stop Installer Manager, type the command:
`sh /opt/cisco/vms/var/jboss/bin/jboss_init_vms.sh stop`

Launching Installer Manager

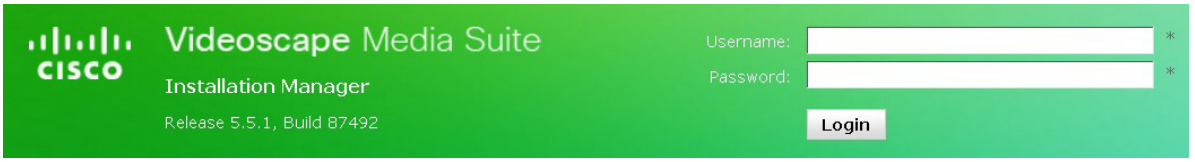
Once the installer Manager has started, you may continue with the part of the installation process that requires a user interface.

To launch the Installer Manager:

1 Navigate to: `http://{hostname}:8080/InstallerManager`

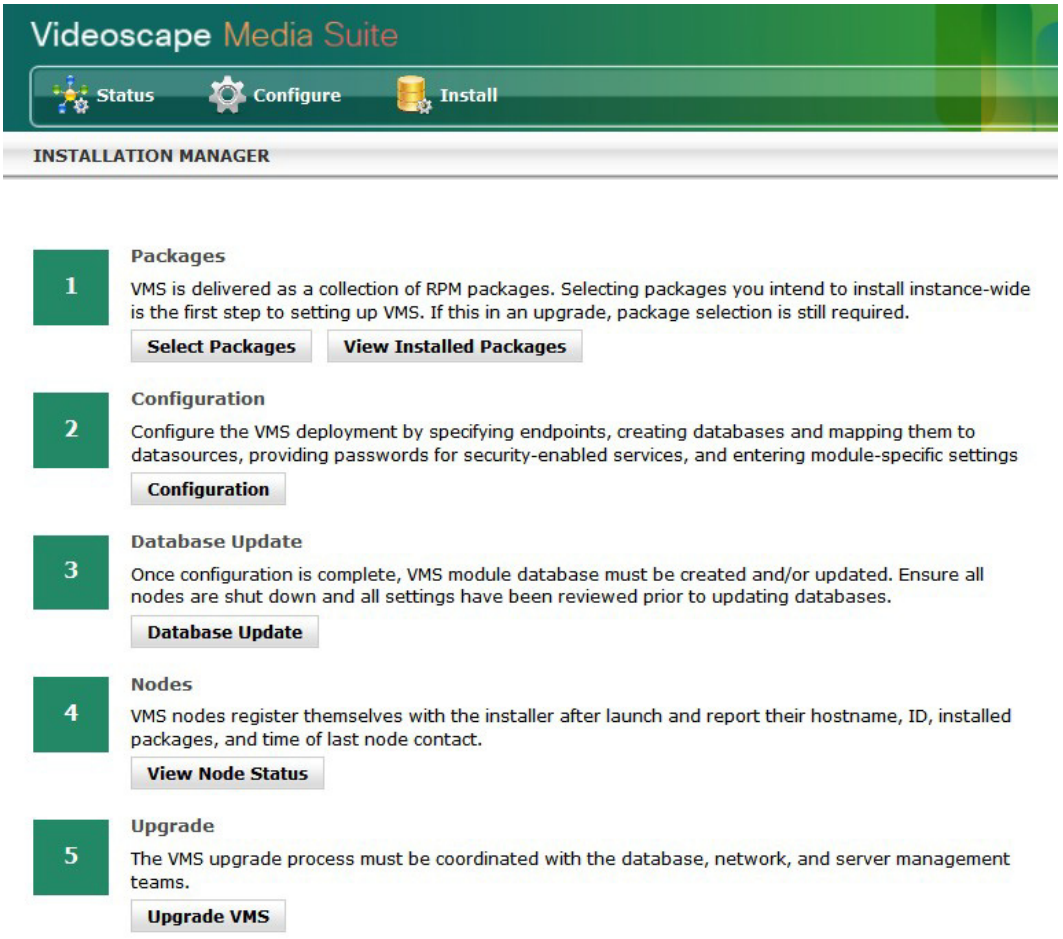
2 Login to the interface. The username is `admin` and the password is whatever you entered when you launched the installer.

Figure 4 Media Suite Login Screen



3 The home page for Installer Manager shows the major steps for the installation process that should be performed within the user interface. We will explain those steps within this section.

Figure 5 Installer Manager home page



Selecting Packages

Prior to performing this action, the agents (and all JBoss servers) must be stopped on all nodes and out of contact with the Installer Manager for at least 90 seconds.

To select packages for installation instance wide:

1 Navigate to **Configure > Select Packages**.
On this page you will see a list of package names (with module name labels and version numbers) that are available to the installer.

2 Select the checkboxes to the left of the modules/packages that you require. The provided packages are defined as follows:

Table 10 Packages and Descriptions

Packages	Description
vms_admin	Provides secure administrator sign on and system configuration functionality.
vms_base	Must be installed onto all nodes. Contains the install agent that resides on each node and that communicates with the Installer Manager to provide updates.
vms_cms	Provides Content Manager functionality.
vms_entitlement	Provides Entitlement module functionality.
vms_epg	Provides EPG module functionality.
vms_keymanagement	Provides Key Management Server functionality.
vms_modcluster	Should be installed on all nodes when using Apache mod_cluster. mod_cluster is the recommended load balancer for easy setup for non-production deployments.
vms_ocesb_epg	Should be installed on all workflow nodes if EPG is being used.
vms_ocesb_producer	Provides Producer functionality.
vms_reporting	Provides Reporting module functionality.
vms_search_config	Provides VOD and EPG search config manager functionality which allows you to customize feed output.
vms_search_master	Provides VOD, EPG, and TSTV search master functionality.
vms_search_master_standby	Provides VOD, EPG, and TSTV search master standby functionality that replicates the master index for redundancy.
vms_search_repeater	An optional component that provides additional scalability for VOD, EPG, and TSTV feeds.
vms_search_slave	Provides VOD, EPG, and TSTV search slave functionality that replicates the master index for robust distribution of feeds.
vms_server_esb	Should be installed on workflow nodes. Provide infrastructure for ESB functionality.
vms_server_standard	Should be installed on all non-workflow nodes.
vms_tstv	Provides TSTV module functionality.
vms_workflow	Should be installed on all workflow nodes.

The following rules should be adhered to when selecting packages:

- Only select packages that will be installed instance wide.
- Do not select packages that will not be installed.
- Do not select a subset of your intended choices, but select all required packages at once.

Validating Dependencies

Once you have selected all required packages:

1. Click **Validate Dependencies**.
   Dependencies will be checked for all selected packages and, if necessary, messages will be shown detailing any missing packages that you need to select to proceed with the install process.
2. Click **Save**. The Select Packages page will continue to show any unselected packages.

Note Packages cannot be removed once they have been selected and that choice has been saved.

Selecting Packages

To select packages:

1. After all required packages and dependencies have been chosen, click **Save** to select the packages.

Viewing Selected Packages

To view all selected packages:

1. Navigate to **Status > Installed Packages**.
   This page will show you general information on all of the packages that you have selected.

Configuring Endpoints

The following section will explain how to specify load-balanced network endpoints that will be used for Media Suite inter-module communication. The endpoints that are displayed will depend upon the packages that you have selected earlier on in the install process.

To configure Endpoints for all imported packages:

1. Navigate to **Configure > Endpoints**.
   This page allows you to customize hostname Endpoint options for the packages that you have imported.
2. Change the hostname values for each endpoint.
3. Click **Save**.

Additional Information for Setting Endpoints

A Bulk Edit Settings section (with a related text box) allows you to change the hostname values for all fields at once. Click **Apply to All** to commit the entered value across all Endpoints.

The SSO Domain is the `.domain` name where any cookies will be stored. All Web UI endpoints must live under this domain. For example, `.cisco.com`

Hovering your mouse over a label will show a description for the relevant setting at right. In short, however, the settings may be described as follows:

- **Web UI** - Directs menu link requests for pages in the module.
- **SOAP Webservices** - Directs SOAP calls to this module from other VMS modules.
- **REST Webservices** - Directs REST calls to this module from other VMS modules.
- **Published REST Webservices** - REST Webservice URLs in this module that are placed by VMS in various published documents.

Table 11 List of Endpoints

Category	Available Endpoints	Context Root
Administration	Web UI SOAP Webservices REST Webservices	/opencase /opencase/webservices /opencase/resource/rest
Content Manager	Web UI SOAP Webservices REST Webservices Published Webservices	/opencase/ContentManager /ContentManager/webservices /opencase/ContentManager/resource/rest /opencase/ContentManager/resource/rest
Content Processor	Web UI SOAP Webservices REST Webservices	/opencase/ContentProcessor /ContentProcessor/webservices /opencase/ContentProcessor/resource/rest
Core ESB Service	SOAP Webservices	/OpenCASE_ESB-Core
EPG Manager	Web UI SOAP Webservices REST Webservices	/opencase/LinearManager /LinearManager/webservices /opencase/LinearManager/resource/rest
Entitlement Manager	Web UI SOAP Webservices REST Webservices Published Webservices	/opencase/EntitlementManager /EntitlementManager/webservices /opencase/EntitlementManager/resource/rest /opencase/EntitlementManager/resource/rest
Key Management	Web UI REST Webservices	/opencase/KeyManagementServer /opencase/kms/rest
OpenAM Single-sign-on	Web UI SOAP Webservices	/opensso/UI /opensso
Recommendations Engine	REST Webservices	/ffs
Report Manager	Web UI	/opencase/ReportManager
Search Configuration Manager	Web UI SOAP Webservices	/opencase/SearchConfigManager /SearchConfigManager/webservices
Search Index Master	Web Non-UI	/solr.master
Search Manager	REST Webservices	/opencase/sm/resource/rest
Workflow Service	Web UI SOAP Webservices REST Webservices	/opencase/WorkflowService /WorkflowService/webservices /opencase/WorkflowService/resource/rest

For all endpoints, selecting the HTTPS protocol does not activate the JBoss SSL connector, but HTTPS will be used for inter-module traffic (in an SSL offload arrangement).

The following restrictions exist when entering endpoint information:

- Port numbers are mandated and are therefore not prompted for. You must use an HTTP proxy on either port 80 (for unsecure communications) or port 443 (for secure communications).
- A proxy/load balancer must be used to access Media Suite. Accessing the server directly via ports 8080/8081 is not supported.

Configuring the Database/Schema

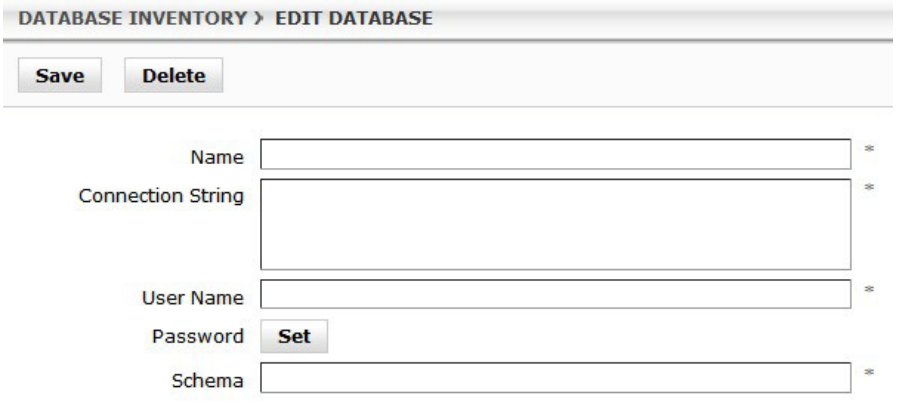
The following section describes how to manage database/schema within Installer Manager.

To add a database to your deployment:

1. Navigate to **Configure > Databases**.
   This page allows you to manage all databases that are used by Media Suite. When you first launched Installer Manager within the Linux command line, you were prompted for database information that will be used for installation purposes. On a new deployment, only that information will appear on this page to begin with. Afterwards, you can configure the offline Reporting database, if needed.
2. Once all the standard database information has been entered, click **Save.** At this point, the system will attempt to validate and connect to the database. If a connection cannot be made, error messages will be displayed and the database information will not be saved. If you cannot connect to the database for whatever reason, select any menu item to exit the page.

Note Oracle connection strings should not contain any line breaks. For details on proper database connection strings, see "Preparing a Database/Schema for Media Suite" on page 15.

Figure 6 Add New Database Page



To download and use the database.oci file:

1. Navigate to **Configure > Databases**.
2. Click **Download database.oci** and save the file. This file allows all the agents to contact Installer Manager.

Note The `database.oci` file can be copied at any point after the Installer Manager install script has been initially run. That script requests the information for the database/schema that will be used for the install.

---

3 Copy the `database.oci` file into `/opt/cisco/vms/state` after you have installed the `vms_base` package. The `database.oci` file will automatically be used once the agent is started.

Mapping Datasources

This user interface page has been removed as of Media Suite Release 5.5 as all datasources now map to only one database. Consequently, no datasource mapping is required. For further details, see "Database/Schema Consolidation" on page 44.

Configuring Module-Specific Settings

This page displays configuration settings that are specific to each module. The provided sections will depend upon the packages that you selected earlier on the **Configure > Select Packages** page.

To configure module-specific settings:

1 Navigate to **Configure > Module Specific Settings**.
Possible options that you may see for the various sections on the page are as follows:

Language Configuration

When performing an upgrade, if the Content Manager or Linear (i.e. EPG) modules contain data that refers to locales outside of the selected locale list, that data will be erased during the upgrade process. The system will, however, check for any such data, provide a warning, and give users the opportunity to expand their locale selection to retain that data.

Table 12 Language Configuration Options

Option	Description
Used Languages	Choose only the locales in which your content data will be provided. **Warning** Do not select all available languages because the nodes will not be able to handle that workload and the deployment will fail. A more reasonable scenario would be to select one (or a few) required languages.
Default Admin Language	Sets the language of the Media Suite user interface.
Default System Language	Sets the Search Manager's default search language.
Localization Package	Provide the localized property files for the administration interface for any required languages. For usage details, see "Localizing Media Suite" on page 42.

EPG Manager

Table 13 EPG Manager Configuration Option

Option	Description
Lineup switch hour	Sets the hour at which new channel lineups will be used by the EPG module.

JBoss Application Server

Table 14 JBoss Application Server Configuration Options

Option	Description
Log level for VMS modules	Sets log level verbosity for the system logs. Log levels can be one of: DEBUG, INFO, WARN, or ERROR with DEBUG being the most descriptive and Error being the least.
Initial heap size as % of RAM	The percentage of RAM on the machine to be allocated to the JBoss initial heap size.
Initial heap size limit (MB)	The MB limit on the JBoss initial heap size.
Max heap size as % of RAM	The percentage of RAM on the machine to be allocated to the JBoss maximum heap size.
Max heap size limit (MB)	The MB limit on the JBoss maximum heap size.

Note Although the user-specified values are respected with regards to memory settings, heuristics are employed by the Agent to ensure that JBoss has enough memory to launch and the Operating System and then has sufficient memory left to perform its work.

JBoss Mod-Cluster

Mod-Cluster is an Apache software load balancer that has been pre-integrated into Media Suite. Arguably mod cluster is the single most convenient and fast way of setting up a load balancer for non-production uses. If you would like to use Mod-Cluster, make sure to select the `vms_modcluster` package on the **Configure > Select Packages** page.

Table 15 JBoss Mod-Cluster Configuration Options

Option	Description
Proxy host: port	Set the hostname and port number (6666) of the server side that runs Apache mod_cluster.

For details on setting up mod_cluster, see “Configuring Apache for VMS” on page 95.

OpenAM Single-sign-on

The OpenAM SSO component manages authentication of Administrators within VMS.

Table 16 OpenAM Single-sign-on Configuration Options

Option	Description
Configuration State	This represents the system's view of the SSO primary node. The three possible states are: 1. Packages Installed - this is the first stage. The Primary SSO node has not yet been selected or configured. The Agent on the Primary node will advance the state at the appropriate time, and not the user. 2. Primary Node Selected - the agent will advance the state to this level once SSO nodes are launched and the system selects a node as the Primary. This selection process is arbitrary and performed at random. 3. Primary Node Ready - after the agent performs the administration and configuration processes, it will push the state to Primary Node Ready. This is important because the Agents will not autostart other JBoss nodes until the Primary SSO is ready. That staggered startup procedure greatly reduces any startup problems that might occur with the system if users were to try to logon while SSO was not ready.
Primary Node	If you want to switch the current SSO Primary Node while the system is running, you may select another Primary Node from the dropdown.
Migration File	A migration tool is run against an existing VMS SSO instance (either 4.x or 5.x) and the tool produces an XML file that you should upload into this field. For details on performing the SSO migration, see "Administrator Account (SSO) Migration" on page 46.

Custom File Deployment

The Custom File Deployment page may optionally be used to deploy custom components, patches, or custom ESB services. To facilitate this process, the Installer Manager provides users with a convenient mechanism to perform such deployments centrally.

To deploy a custom file:

1 Navigate to **Configure > Custom File Deployment.**

2 Click **Add New**.

3 Fill in the required information as per the following table.

Table 17 Custom File Deployment Options

Option	Description
Name	Provide a descriptive name for the file to be deployed.
Target Path	Specify the node location within the JBoss deploy directory where you would like this file installed.
Target Module	This file will only be deployed on nodes where this module is installed.
Place inside module directory	If you want to deploy the file inside of an existing module directory, select this option and the Installer will append the correct folder location to the Target Path field. In order to use this properly, you need to know the underlying directory structure and you should be careful not to specify a directory that already exists because the agents will erase directories. One big advantage of this feature, overall is that you only ever need to upload the file once. Later, if you reinstall a file, the agents will redeploy. Also, if you upload the file a second time (and the checksum is different) the agents will redeploy as well. If no file extraction is required, then any existing files with the same name at the target location will be overwritten.
Extract as archive	By default, files are deployed as is - be they a jar file, a zip file, or a class, for example. However, if you have a zip-compatible file (such as a zip, jar, or ear), this option will extract the archive. If an archive is extracted, however, then the base directory will be erased based upon the path. As a best practice, consult with your Advanced Services representative at the planning stage so that no existing data is erased in this instance.
Deploy only when server is stopped	If you are deploying files that do not require a server restart, then leave this option unchecked.
File Size	Once a file uploaded, the file size will be shown.
File	Allows you to browse and select the file or archive that you would like to deploy.

Custom Log4j Appenders

Media Suite logs standard system activity by way of the Log4j logging utility. In the event that you would like to use custom system events, you may also wish to enable custom logging to monitor that usage. The following section explains how you can upload and use custom Log4j appenders to implement custom logging within your deployment.

To implement custom logging in Media Suite:

1 Within Installer Manager, navigate to **Configure > Custom Log4j Appenders**.

2 Click **Add New**.

3 Type a name for the custom appender.

4 Click **Browse** to select and open a JAR file containing the custom logging functionality that you require.

5 Copy and paste the Log4j.xml fragment that specifies parameters for the custom logging.

Figure 7 Sample Log4j.xml Fragment

```
<!-- Used by the customer appender  -->
<appender name="CustomerAppender"
class="com.cisco.videoscape.customer.logging.CustomerAppender">
                                <param name="rulesFilepath"
value="CustomerStateLoggingConfig.xml" />
</appender>

<appender name="Customer-ASYNC" class="org.apache.log4j.AsyncAppender">
                                <param name="BufferSize" value="256" />
                                <appender-ref ref="CustomerAppender" />
                                <appender-ref ref="FILE" />
</appender>

<category name="com.cisco.videoscape.customer">
                                <priority value="TRACE"/>
</category>

<category name="VMS_STATE" additivity="false">
                                <priority value="INFO" />
                                <appender-ref ref="Customer-ASYNC" />
</category>
```

6 Click **Save**.

7 Restart Media Suite. When the system restarts, the selected JAR file will automatically be copied into the `/opt/cisco/vms/var/jboss/server/vms/lib` folder of all nodes in the deployment.

Note Although the code is automatically deployed, it cannot be automatically removed. That would need to be performed manually on a node-by-node basis. If you, however, update the fragment to discontinue custom logging usage, the existing logging code will not be invoked.

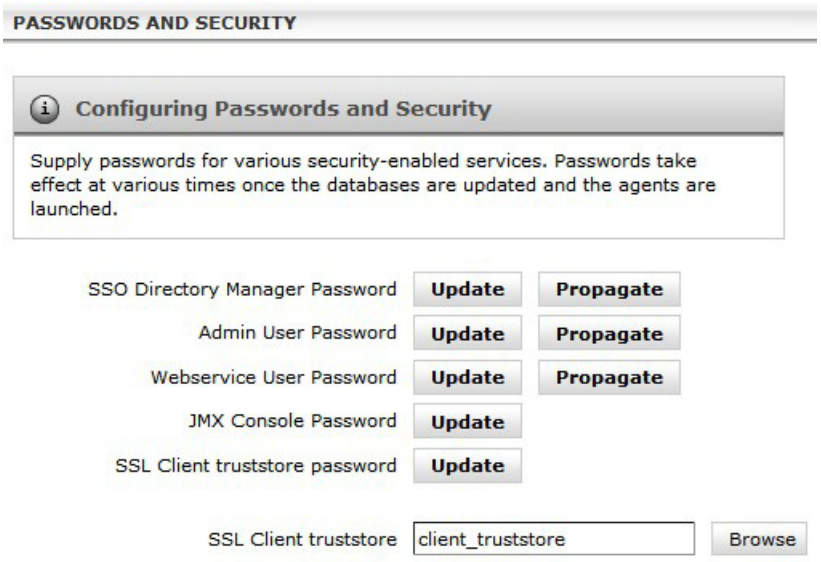
Understanding Passwords and Security

The Passwords and Security page manages passwords for various security enabled services. In addition, functionality is provided to streamline the process of propagating passwords when required. Passwords will take effect at various times once the database/schema is updated and the agents are launched.

The types of passwords that may be managed on this page include:

- **SSO Directory Manager Password**, for the AM Admin user
- **Admin User Password**, for the Media Suite default administrator.
- **Webservice User Password**, for the VMS Web service user.
- **JMX Console Password**, for the JMX Console and JMX connector.
- **SSL Client Truststore Password**, for HTTPS authentication of Media Suite endpoints.

Figure 8 Configuring Passwords and Security



Setting or Updating Passwords

To set or update a password:

1. Navigate to **Configure > Passwords and Security**.
2. Click **Set** to establish an initial password or click **Update** for any existing password that you need to change.
3. Type and confirm the new password. The password must be 8 characters or longer.
4. Click **Save**.
5. If your organization has a requirement to rotate its passwords at a specified interval, click **Propagate**, which will push out any updated passwords to the database and to SSO nodes.

Note JMX passwords are set per node and cannot be propagated.

SSL Client Truststore Password

If any Media Suite endpoints utilize HTTPS, administrators will be required to provide a truststore for SSL purposes. A field is provided to browse and upload the SSL Client Truststore. The provided client truststore must contain all required certificates to cover the certificate chain served by the load balancers. After uploading the truststore and clicking **Upload**, the truststore will be applied to all JBoss nodes in the instance.

Updating the Database/Schema

After you have completed configuring the install (and verified your work), you will need to update the database/schema to take all of the configuration parameters into account.

Note If you have any non default SSO logins or entitlements that need to be migrated, then those processes must be performed prior to updating the database/schema. You will be warned if you have not performed a migration on an existing system. For details on the migration processes, see "Monitor the **Node Status** page to be aware of any warnings or errors so that you can review them." on page 46 and "Administrator Account (SSO) Migration" on page 46.

To update the Media Suite database/schema:

1. Navigate to **Install > Update Databases**.
   The page will show all modules, the target datasource, and the database status of each module.
2. Click **Execute**.
   The status will refresh and all items will transition from a "Ready" state to "In Progress" to the final "Complete" state. On an empty database, this process will take a few minutes.
   If there are any errors, the status column will show the Error state. Afterwards, you can click the "Logs" column to download and examine the log for details. In addition, this log can be submitted as a part of your support request if required.

   Once complete, the update process may log relevant

   information or warnings that should be known to administrators. If any are logged, they will be visible under the Messages column. Click the icon to view those logs.

Deploying RPM Packages onto Nodes

Once all database/schema has been updated and all modules are in a Complete state, you should deploy the appropriate RPM packages onto each node.

To deploy RPM packages onto nodes:

1. Copy the RPMs onto each node.
2. Install them with the `rpm -Uvf /path/to/vms_package_name.rpm` command. For details on the Linux RPM installation and other related commands, see "Linux RPM Procedures" on page 87.
3. Place the `database.oci` into `/opt/cisco/vms/state` after you have installed the `vms_base` package. The `database.oci` file will automatically be used once the agent is started.
4. Launch the agent via the Linux command line:
   `sh /opt/cisco/vms/var/agent/agent.sh start -fc -nc -au`
   For additional agent details, see "Managing Install Agents" on page 91.
5. Within a few seconds, the node will appear on the node status page (located at **Status > Nodes** on the Installer Manager user interface). The JBoss status, JGroups Members, installed packages, node name and any error or status messages will be displayed and updated every 30 seconds.
   - Hover over the "JGroups Members" to view the membership list of IP addresses. That list has been made available to make debugging easier.

Clicking the node name will open a page that displays Node details and messaging.

- Click the underlined number under Status and Messages to view verbose node detail information with messages and timestamps.
- If you launch all of the agents simultaneously, they will all wait until SSO is configured and then the other JBosses will be launched.
- To stop JBoss instances click **Stop** under the Status and Messages column.

If there are deployment errors in JBoss when it starts. Those errors should be caught and handled as soon as possible. The Installer Manager will catch all errors and Stop the node so that debugging can be performed. Once you have fixed the problem, click **clear stop command** and the JBoss instance will attempt to startup again.

Note Prior to logging in, ensure that your SSO nodes are up and running.

Startup and Shutdown Considerations

The following section lists best practices for starting up and shutting down Media Suite nodes.

1. Start the agent via the Linux command line.
   For details on managing nodes, see "Managing Install Agents" on page 91.
2. Start and stop JBoss only via the Installer Manager user interface. Do not use the shell. This task is performed on the **Node Status** page.

Note The agent only starts JBoss when -fc -nc options are both used.

Securing OAuth Data Transmission

During the operation of Media Suite, there are a few OAuth URLs that must be protected with SSL so that sensitive data is not transmitted in plain text over the network. Consult with your network administrator for the correct manner in which to secure data transmission for your deployment.

The following URL must be protected:

- /opencase/EntitlementManager/oauth/authenticate
  This URL is used for user authentication for OAuth. Sensitive credentials, consisting of a username and password, are transmitted during this request.

The following URLs should be protected:

- /opencase/EntitlementManager/oauth/request_token
  This URL is used by OAuth clients to obtain a request token. No passwords are sent in this request, but the response contains a token key and secret, which should be protected.
- /opencase/EntitlementManager/oauth/access_token
  This URL is used by OAuth clients to obtain an access token. No passwords are sent in this request, but the response contains a token key and secret, which should be protected.

Note In addition to the previously mentioned URLs, you should secure any sensitive data so that it is not broadcast unencrypted across unsecure channels. Given the extensible nature of Media Suite, we cannot obviously address specific scenarios in this section.

Localizing Media Suite

Media Suite ships with a localization package (`vms_localization-<build_no>.zip`) that is used to translate the administration interface and other components into various languages.

To localize Media Suite:

1. Unzip the included `vms_localization-<build_no>.zip` file.
2. Duplicate the "en_US" directory and change the locale name to your required target language and country, for example, "de_De" for German. Consult the user interface tooltip for a list of valid locales.

   Note Localized labels for a given locale will only be used if that locale is part of the administrator's selected locale set on the **Configure > Module Specific Settings** page.

3. Edit each property file under the new directory to translate the label values into your target language.
4. Re-zip the package and upload the zip archive using the provided "Localization Package" field on the **Configure > Module Specific Settings** page.

Next Steps

At this point, you are ready to configure Windows 2008 and then to install the Windows Service components. To begin that process, see Chapter 7, "Preparing Windows Server 2008 for Media Suite".

Chapter 6

Upgrading Media Suite

This chapter provides information related to performing an upgrade to a 5.x version of Media Suite.

Note We strongly recommend that (once any migrations are complete) you remove all other JDKs entirely from your VMS servers other than the version that is provided. Media Suite is completely self-contained from a JDK perspective and any other versions are unnecessary and may confuse administrators and lead to difficulties while troubleshooting.

Upgrading from Media Suite 4.x

The following section addresses details related to upgrading from earlier versions of Media Suite to 5.x. Review all of these considerations prior to performing an upgrade:

- Media Suite 4.1.2.2 (or later) is required for a direct upgrade to 5.x. Therefore, earlier releases must first by upgraded to 4.1.2.2 (or later), and then to 5.x.
- Media Suite 4.1.5 (and later) will need to be upgraded to 5.0.1 and not 5.0).
- All Linux filesystem-level artifacts (everything that is not in the database) must be discarded. That includes compass indexes, which will need to be reindexed once the upgrade is complete.
- Upgrades must be performed only after stopping all workflow activities, B2B calls, and waiting for relevant queues to clear (i.e. `OCME_PRODUCT_CHANGE_QUEUE` and `OCME_NOTIFICATION` must be empty prior to shutting down the 4.x instance).
- Multiple (pre-5.5) database/schemas must be consolidated into one. For details, see “Database/Schema Consolidation” on page 44.

Note Consult with your Cisco Advanced Services representative prior to performing this upgrade as special procedures may be required in your particular instance.

Understanding the Upgrade Process

The following process will allow you to upgrade VMS with a minimal amount of disruption to the service.

Note Many of the following steps are identical to those in a standard install. See “Installing Media Suite on Linux” on page 27 for details.

The upgrade process should be performed in the following order:

1. Create a deployment map for the upgraded deployment based on your requirements.
2. Run the SSO Migration tool to export administrator credentials to an XML file. Once this action has been performed, no new SSO credentials can be created or altered on the old system.
3. Keep your Media Suite 4.x system running. Install the required Media Suite 5.x RPMs onto their designated nodes. To prevent conflicts, make sure that your previous VMS instance does not reside at: `/opt/cisco/vms`

Database/Schema Consolidation

Prior to VMS 5.5, deployments could have multiple database/schemas. In this release, we have mandated that only one database/schema can be used. Consequently, when upgrading to Release 5.5 or greater, you will need to consolidate several database/schemas into one. This provides significant ongoing benefits when upgrading and maintaining your deployment.

Database/schemas may be consolidated in a number of ways and we advise you to consult with your database administrator for the best approach for your situation.

One possible approach, for example, would be to utilize Oracle's DataPump tool to automate the consolidation process.

Note Prior to performing this procedure, ensure that you have made the appropriate snapshots and backups in case you need to revert any part of the process.

A database/schema consolidation would be performed in the following manner:

1. Create your new single (unified) schema.
2. Create your new single (unified) tablespace.
3. Export your multi-schema database into one intermediate dump file using DataPump's export command. For example:
   ```
   expdp system/{password} SCHEMAS=SCHEMA_1, SCHEMA_2, SCHEMA_N
   directory=DATA_PUMP_DIR dumpfile={SCHEMA_DUMP_FILE.dmp}
   logfile={SCHEMA_dpEXPORT.log}
   ```
4. Import the multi-schema database contained in `{SCHEMA_DUMP_FILE.dmp}` using DataPump's import command. For example:
   ```
   impdp system/{system_user_password} REMAP_SCHEMA=SCHEMA_1:New_Single_Schema,
   SCHEMA_2:New_Single_Schema, SCHEMA_N:New_Single_Schema TRANSFORM=oid:n
   DIRECTORY=DATA_PUMP_DIR DUMPFILE={SCHEMA_DUMP_FILE.dmp}LOGFILE=
   {SCHEMA_dpIMPORT.log} REMAP_TABLESPACE=Tablespace_1:NewSingleTablespace,
   Tablespace_2:NewSingleTablespace,Tablespace_n:NewSingleTablespace
   table_exists_action=append
   ```
5. A single database/schema has now been created. You will now work with this new database/schema as you proceed with the remainder of the upgrade process within Installer Manager.

Note For all Media Suite 5.x to 5.5+ upgrades, you must clear the existing pre-5.5 Installer Manager installation. This can be performed by removing the `vms_installer` package and then deleting the `/opt/cisco/vms` folder.

Upgrading VMS with Installer Manager

At this stage, you proceed by using Installer Manager to configure and complete the remainder of the upgrade process.

To continue the upgrade from Installer Manager:

1. Choose a dedicated node for Installer Manager that is separate from any existing nodes.
2. Launch Installer Manager.
3. Provide the database connection information that is requested by the Installer Manager script.
4. Once the script has completed, the `database.oci` file may be downloaded. Place a copy of the `database.oci` file into `/opt/cisco/vms/state` onto each node at any point after the `vms_base` RPM package has been installed.
5. Within the Installer Manager user interface, perform the following steps:
   - select packages for installation
   - validate dependencies
   - configure endpoints
   - configure the database/schema
   - configure module-specific settings
     At this stage, you will import the SSO migration file when configuring the "OpenAM Single-sign on". For details, see "Administrator Account (SSO) Migration" on page 46.
   - update passwords
     Any changes that you make to these default passwords will override any defaults that are ported over by the SSO migration tool. That should not cause any problems with the upgrade.
   - DO NOT perform the **Update Database** process that you would normally perform during a clean install. The database update will be done at a later stage.
6. Shut down the old service.
7. Perform the customer account migration process.
   For details, see "Monitor the **Node Status** page to be aware of any warnings or errors so that you can review them." on page 46.
8. Execute the **Update Databases** process within the Installer Manager user interface.
9. Launch all agents directly using the command line. Startup order is unimportant and agents may be started simultaneously.
10. Navigate to the **Node Status** page to view the status of each node as it becomes recognized by the Installer Manager.

Best Practices for Upgrading

As a failsafe, it is important to:

- Backup the filesystem, JBoss, SSO, and any other Media Suite application directories.
- Take a snapshot of the original Media Suite database.
- Monitor the **Update Databases** page for any warning or error messages that might have resulted from the process. A **Reset Error State** button is available to clear the error state and attempt the update again/ That button should only be used after examining the logs and identifying the cause of any failures.
- Monitor the **Node Status** page to be aware of any warnings or errors so that you can review them.

Customer Account Migration

The following section describes the process of migrating customer account credentials (within the same database) by using a supplied script. This process is applicable if you have a 4.x version of Media Suite installed and if you have customer account logins that need to be migrated to a 5.x instance.

- Perform the following migration step after launching the Media Suite 5.x Installer Manager, but prior to invoking the database updaters.
- Copy `vms_entitlement_encryption_migration.zip` to a Linux machine with a JDK and JCE Unlimited. This can be any machine, and does not need to be the one running Media Suite or the installation manager.
- Extract the archive.
- Execute the script with either:
  `sh encryption_migration.sh /path/to/database.oci`
  or
  `sh encryption_migration.sh http://host:8080/InstallerManager/database.oci`

Administrator Account (SSO) Migration

The following section describes the process of migrating administrator account credentials by using a supplied script. This process is applicable if you have a 4.x version of Media Suite installed and if you have non-default account logins that need to be migrated to a 5.x instance. In addition, the script can also be used for any 5.x to 5.y account migrations.

Perform the following migration step prior to launching the installation manager and shutting down the existing (4.x) OpenSSO primary server:

1. Copy `vms_sso_migration.zip` to a Linux machine. This tool can be run against 4.x and 5.x Media Suite instances.
2. Extract the archive.
3. Invoke the SSO migration script and point it to the existing (4.x) OpenSSO primary server.

4. Execute `sh sso_migration.sh` with the following arguments:
   - `<ldap url>` of the form `ldap://<host>:<port>`, where the port number is the directory server port, which is usually 50389
   - `<output file.xml>`
5. You will be prompted for the `<amAdmin password>` from the existing (4.x) installation.
6. Save the resulting output file and upload it to the Installer Manager in the SSO section of the module-specific settings page.

Database Updater Behavior

Unlike the legacy 4.x installer, the 5.x installer never drops tables or any other database entities. Therefore, in a clean install, the user must drop and re-create the database prior to starting the installation.

Note For clean installs, no VMS entities should exist before you install VMS 5.x.

Planning Rolling Upgrades

The rolling upgrade process may vary per deployment type, and will depend upon the required components and use cases. One approach, for example, might be to separate your deployment into A/B groups so that every module has a node presence in each group. Consult with your Cisco Advanced Services representative when planning to implement rolling upgrade capabilities in your deployment. They will help you work through a solution that is specific to your situation.

Chapter 7

Preparing Windows Server 2008 for Media Suite

Overview

The Windows servers in Media Suite can provide functionality related to transcoding, encryption, licensing, and distribution. The following chapter explains how to prepare a Windows Server 2008 environment for the installation of the Media Suite Windows service components.

Note Although Media Suite theoretically supports both Windows 2003 and Windows 2008 environments, a Windows 2008 installation is recommended for optimal compatibility with the newest transcoding and DRM technologies.

The steps for preparing Windows Server 2008 for Media Suite are:

Preparing Your Windows Environment

The following sections cover the processes for preparing a Windows environment for Media Suite.

Software Requirements

The following software and dependencies are required to support Media Suite components in a Windows Server 2008 environment.

Table 18 Windows Software Requirements

Software	Type	Version	Notes
Operating System	Windows Server	2008 R2	Different versions of Windows Server 2008 may be used. This guide documents an installation using Windows Server 2008 R2 Standard Edition. Other editions may require that some features be manually added. Those specific features will be listed in this chapter for your convenience.
Framework	Microsoft .NET	3.5.x and 4.x	Installed as per instructions in this chapter.
Web Server	Internet Information Server	7.x	Installed as per instructions in this chapter.
PlayReady SDK	VMS 4.0.x and later VMS 4.1.1 and later VMS 4.1.3.2 and later	SDK 1.5.2 SDK 2.0 SDK 2.1	Must be obtained from Microsoft. This SDK must be installed prior to installing the PlayReady License Services.

Installing IIS

Internet Information Services (IIS) is installed by adding the Web Site (IIS) role to Windows Server 2008. Afterwards, you will need to add Server Roles to the Web server to include additional required functionality.

To install IIS:

1. Click the Server Manager icon to the right of the **Start** menu.
2. Click **Roles** under the Server Manager tree.
3. Click **Add Roles**.
   The **Add Roles Wizard** opens with the **Before You Begin** window.
4. Click **Next**.
   The **Select Server Roles** window opens.
5. Select the **Web Server (IIS)** role and click **Next**.
   The **Introduction to Web Server (IIS)** windows opens.
6. Click **Next**.
   The **Select Role Services** window opens.

7 Select the following role services for IIS:

Table 19 Web Server Roles to Select for a Media Suite Installation

Web Server		
	Common HTTP Features	
		Static Content Default Document Directory Browsing HTTP Errors HTTP Redirection
	Application Development	
		ASP.NET .NET Extensibility ASP CGI ISAPI Extensions ISAPI Filters Server Side Includes
	Health and Diagnostics	
		HTTP Logging
	Security	
		Basic Authentication Windows Authentication Digest Authentication Request Filtering
	Performance	
		Static Content Compression
	Management Tools	
		IIS Management Console
	IIS 6 Management Compatibility	
		IIS 6 Metabase Compatibility IIS 6 WMI Compatibility IIS 6 Scripting Tools IIS 6 Management Tools

8 Click **Next**.
The **Confirm Installation Selections** window opens.

9 Click **Install**.
The role services will install and the **Installation Results** window opens.

10 Click **Close**.

Adding Server Features

After installing the server roles you will need to install server features.

To add features to your web server:

1. Click the Server Manager icon to the right of the **Start** menu.
2. Click the **Features** node under the Server Manager tree.
3. Click **Add Features**.
   The **Add Features Wizard** opens. Once you begin selecting the features below, a dialog will open that requests the installation of other services. Click **Add Required Features** to install those dependencies.
4. Select the following features:
   - .NET Framework 3.5.1 Features
   - Desktop Experience
   - Message Queuing
     Selecting the Message Queuing node will only select the Message Queuing Server child node. This is the correct selection.
   - Windows Process Activation Service
5. Click **Next**.
   The **Confirm Installation Selections** window opens.
6. Click **Install**.
   The installation will proceed and the **Installation Results** window opens.
7. Click **Close** and restart the server as requested.

Installing .NET Framework 4.x

To install the .NET Framework, we recommend performing an online search for the 64-bit Microsoft .NET Framework 4 (Web Installer). The installer filename is `dotNetFx40_Full_setup.exe`. As of the writing of this document, the link to this file (subject to change) is:

http://www.microsoft.com/downloads/en/details.aspx?FamilyID=9cfb2d51-5ff4-4491-b0e5-b386f32c0992&displaylang=en

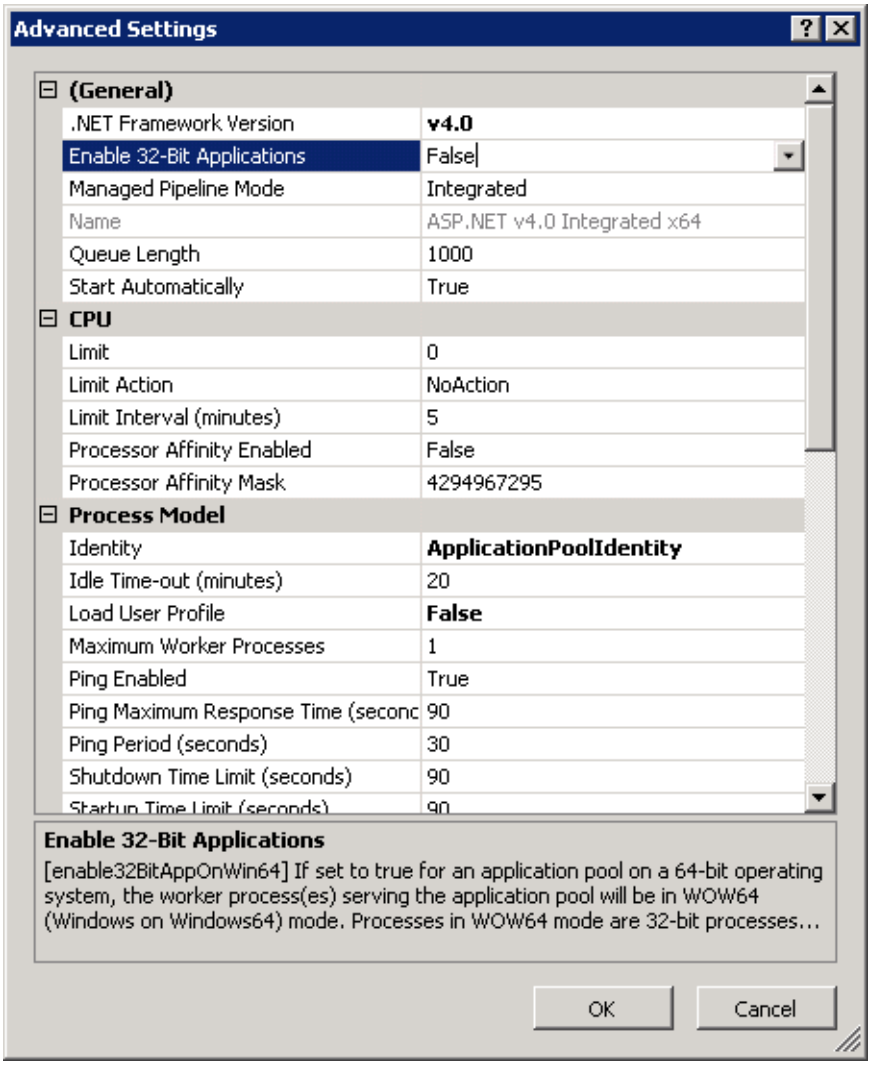
Creating ASP.NET v4.0 Integrated x64 Application Pools

To create 64-bit application pools:

1. In **Server Manager**, navigate to **Roles** > **Web Server (IIS)** > **Internet Information Services (IIS) Manager**.
   The **Internet Information Services (IIS) Manager** window opens.
2. On the **Internet Information Services (IIS) Manager** window, expand the node with your server name**,** and select the **Applications Pools** node.
3. Click **Add Application Pool...** on the **Actions** pane on the right.
4. In the **Add Application Pool** dialog, for **Name**, type ASP.NET v4.0 Integrated x64.
5. Select the following settings:
   - .NET Framework v4.0.30319 (for the .NET Framework version)
   - Integrated (for the Managed pipeline mode)
6. Click **OK**.

7 In the **Actions** pane, click the **Advanced Settings** link.
The **Advanced Settings** window opens.

Figure 9 Advanced Settings for ASP.NET v4.0 Integrated x64 Application Pools



8 On the **Advanced Settings** window, set the option **Enable 32-bit Applications** to False (64-bit applications only).

9 Click **OK**.

Configuring ISAPI and CGI Restrictions

Next, you must configure both the ISAPI and CGI restrictions settings to accept your web services. This configuration must be done prior to installing Media Suite; or the web services will not install.

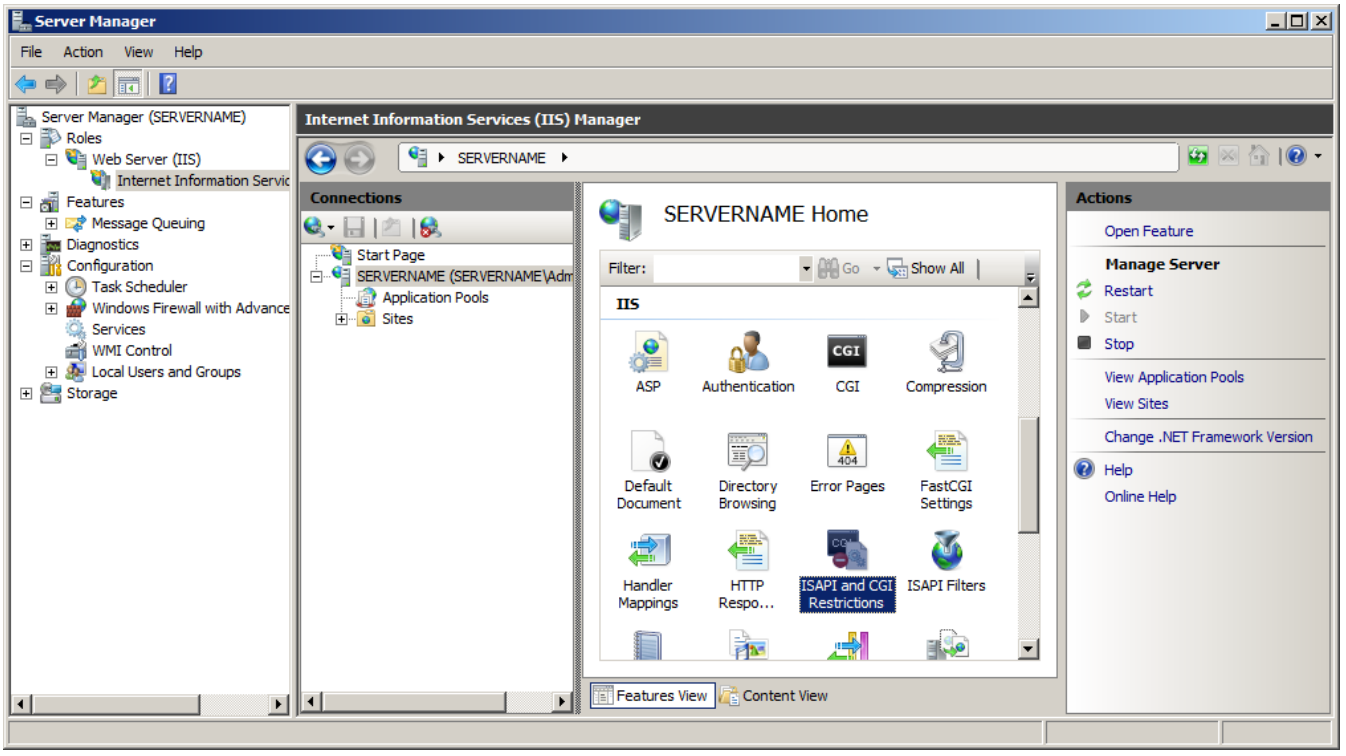
To configure ISAPI and CGI restrictions:

1 In **Server Manager**, expand **Roles > Web Server (IIS) > Internet Information Services (IIS) Manager**.
The Internet Information Services (IIS) Manager pane opens, in **Features View**.

2 In the **Internet Information Services (IIS) Manager** pane, scroll down to the IIS section and double-click the option **ISAPI and CGI Restrictions** icon.

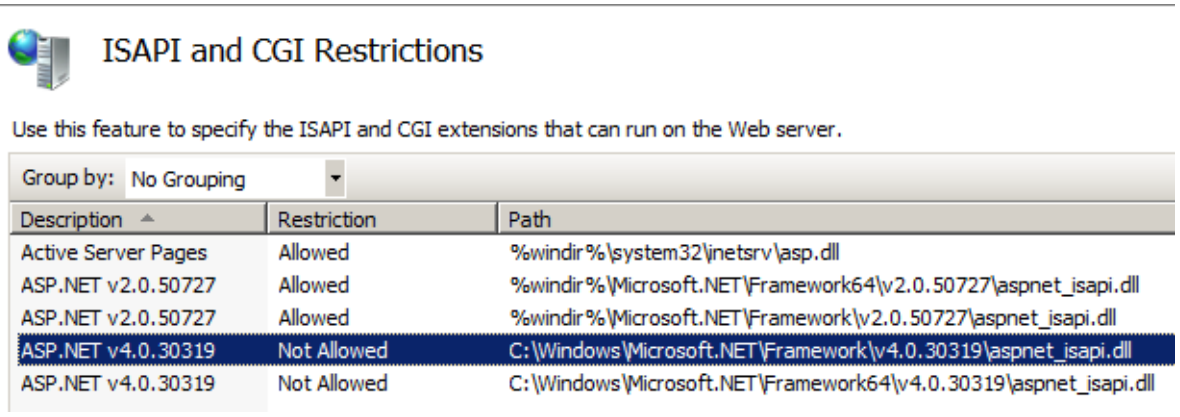
Note If the ISAPI and CGI Restrictions icon is not visible, then the Application Development roles services have not been installed on your operating system. Add them by selecting **Roles > Web Server (IIS)** and clicking **Add Role Services**. A dialog will appear with the Application Development Roles. Add all of the roles.

Figure 10 Internet Information Services (IIS) Manager Pane



The ISAPI and CGI Restrictions pane opens, in **Features View**.

Figure 11 ISAPI and CGI Restrictions Pane



3 On the **Actions** pane, click the **Open Feature** link.

4 In the **ISAPI and CGI Restrictions** pane, there will be two versions of **ASP .NET v4.0.30319**. One is 32 bit, and the other is 64 bit. Select each version and click **Allow** in the actions pane.

Figure 12 ISAPI and CGI Restrictions Allowed

ASP.NET v4.0.30319	Allowed	C:\Windows\Microsoft.NET\Framework\v4.0.30319\aspnet_isapi.dll
ASP.NET v4.0.30319	Allowed	C:\Windows\Microsoft.NET\Framework64\v4.0.30319\aspnet_isapi.dll

Creating the Web Service User

Before installing the service components for Media Suite, you must create a dedicated local user to manage the services. Some of the services require you to authenticate against this user before the installation of that component can be completed.

To create a user:

1 From the **Start** menu, right-click **Computer** and select **Manage**.

2 Expand the Configuration node then **Local Users and Groups**.

3 Right-click the **Users** folder and select **New User** from the menu.
The **New User** dialog opens.

4 For **User name**, type WS_User.

5 For the **Full name**, type Web Service User.

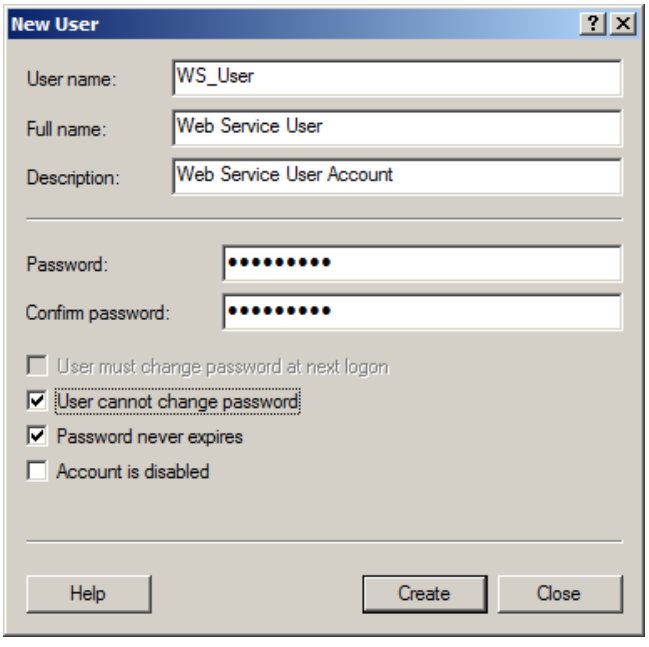
6 For **Description**, type Web Service User Account.

7 For **Password and Confirm Password**, enter a strong password for the account and confirm the password.

8 Clear the **User must change password at next logon** check box.

9 Select the **User cannot change password** and **Password never expires** options.

Figure 13 New User Dialog



10 Click **Create**.

Configuring WS_USER to Run as a Service

The WS_USER user and directory permissions are used for the Media Suite Web Services. If your system is configured as a domain, before you can begin, your system administrator must create a Group Policy Object in order to configure permissions for WS_USER. If that step is not accomplished before you attempt to configure user permissions, the option to add a user or group will not be available to you.

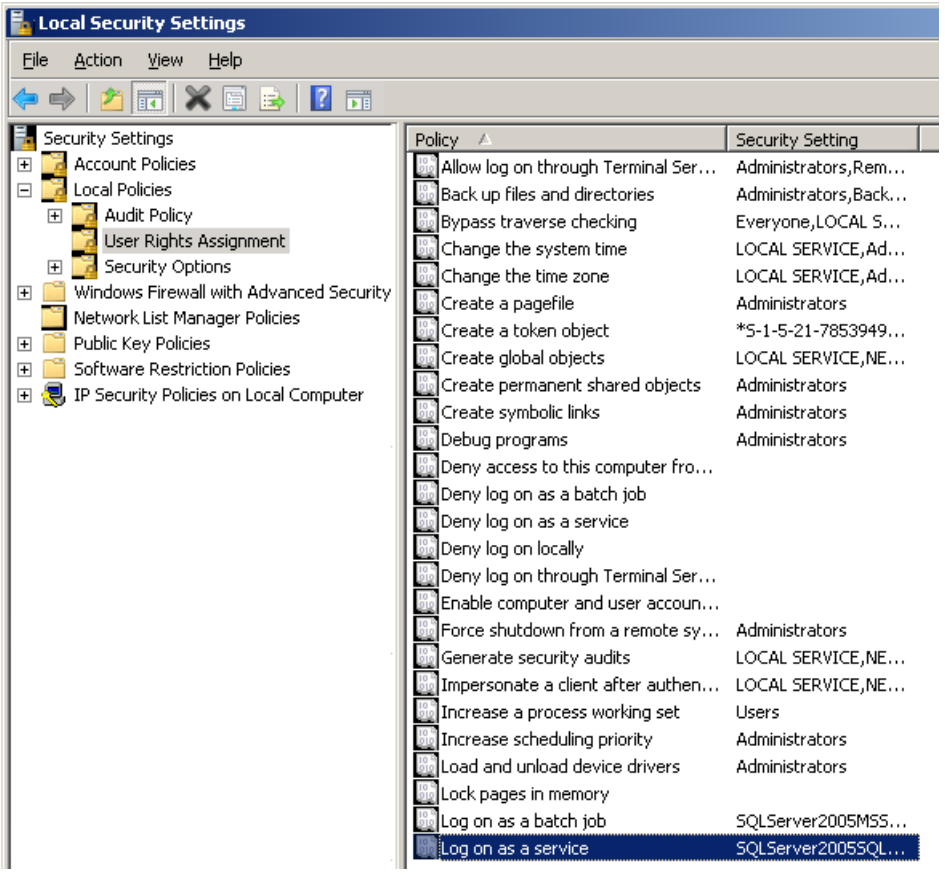
To configure user permissions for WS_USER:

1 From your **Start** menu, right-click **Administrative Tools** > **Local Security Policy** and choose the **Run as administrator** option.
The **Local Security Settings** window opens.

2 Expand the **Local Policies** folder and select the **User Rights Assignment** folder.

3 Double-click the **Log on as a service** policy.

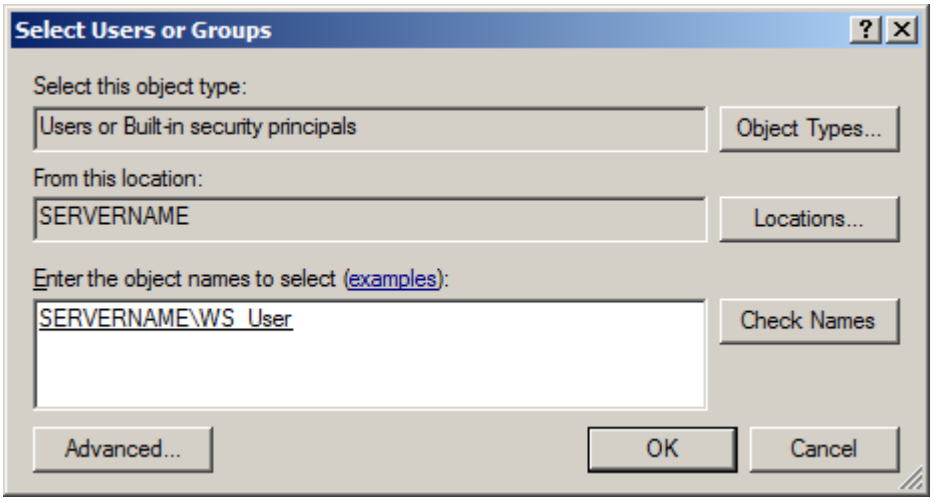
Figure 14 Local Security Settings Window



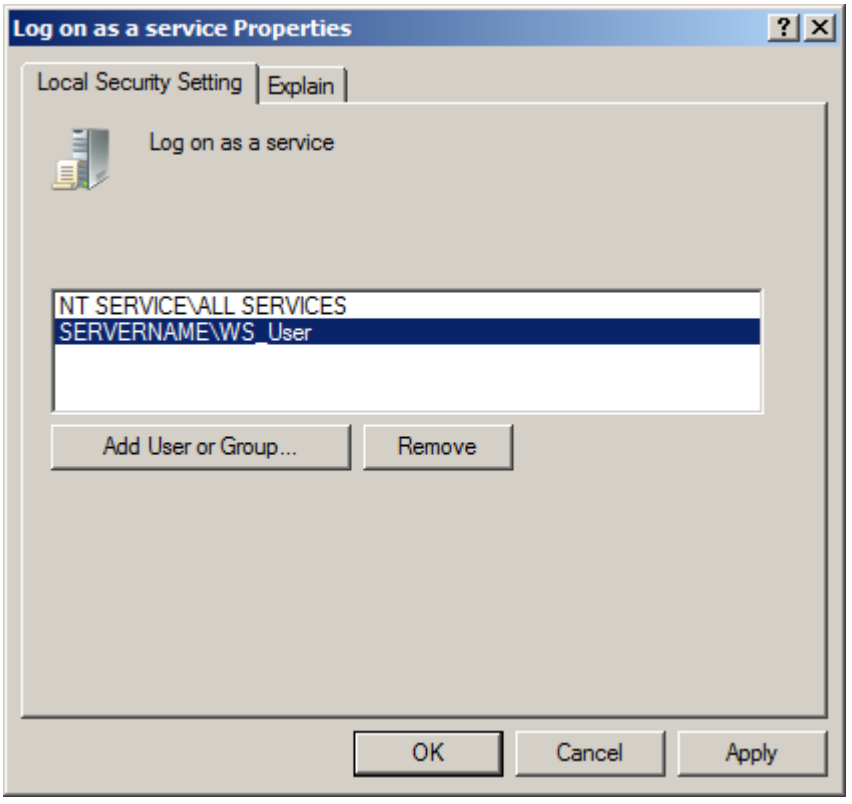
4 On the **Properties** window, click the **Add User or Group** button.
The **Select Users or Groups** window opens.

5 In the **Enter the object names to select** field, type `WS_USER` and click **Check Names**. This will verify that the name was spelled correctly, and that the user can be found on the requested server.

Figure 15 Select Users or Groups Window



6 Click **OK**.
The WS_USER user now appears in the **Local Security Setting** list on the **Properties** window.



7 Click **OK**.

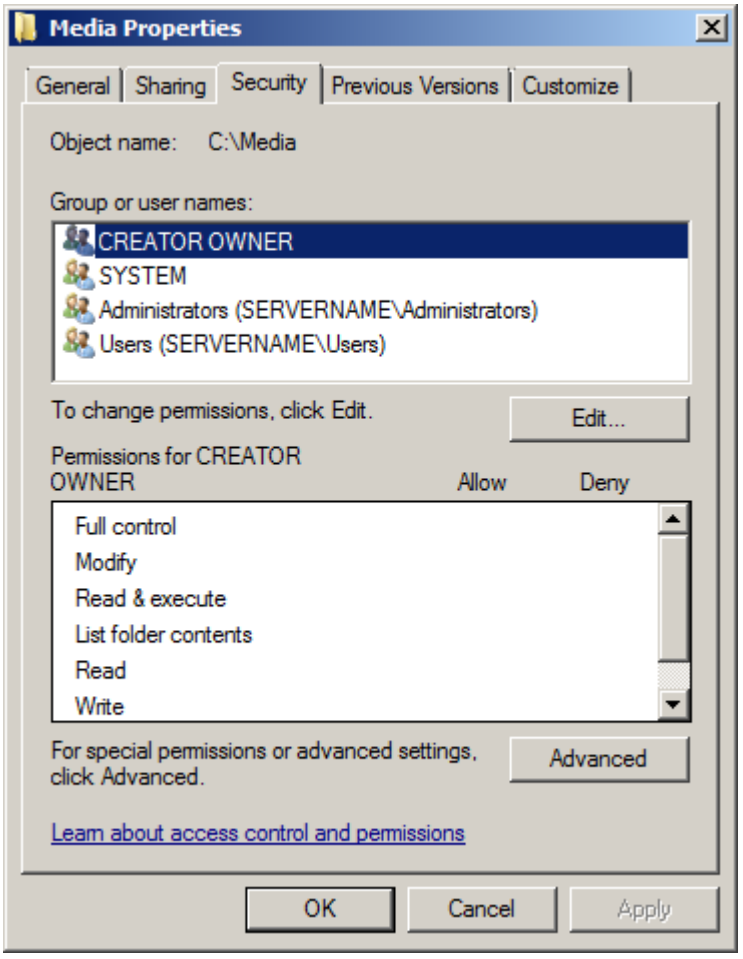
Configuring Directory Permissions

In this section we will create two folders: one to store media, and a second for storing Media Suite Web Services for IIS. Next, we will configure permissions for those folders.

To create and configure the folders:

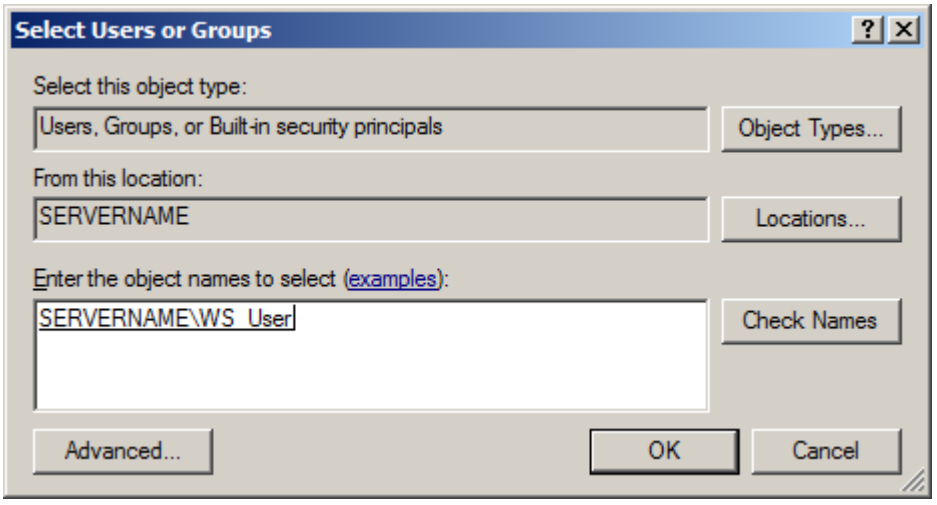
1. Create a directory for storing your media (for example, `c:\Media`)
2. Create a second directory for storing Media Suite Web Services for IIS (for example, `c:\ServiceWebSite`)
3. On both these folders, you will need to setup permissions for WS_USER. We will start by setting permissions for the `c:\Media` folder.
4. Right-clicking on the folder and select properties.
5. Select Security tab.

Figure 16 Folder Permissions, Security Tab



6. Click **Edit**.
7. Click **Add**.

Figure 17 Select Folder Users or Groups

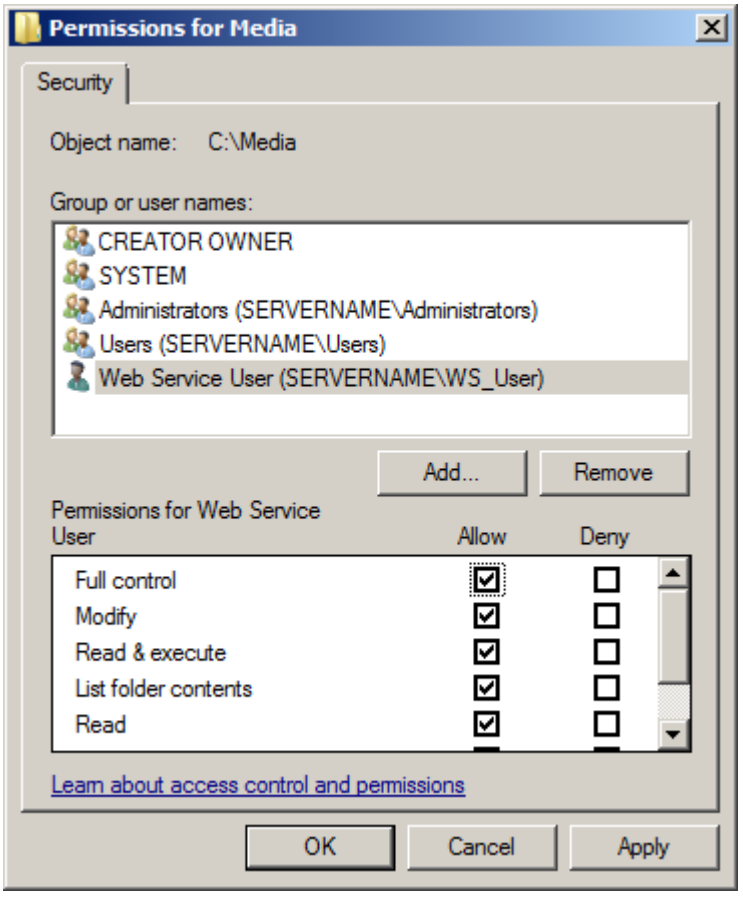


8 Type `WS_User` and click **Check Names**.

9 Click **OK**.

10 Grant WS_USER full **Full Control** access to the folder.

Figure 18 Granting Full User Control to Folder



11 Click **OK** twice to complete the process.

12 Repeat the previous steps to grant full access to WS_USER for the `ServiceWebSite` folder.

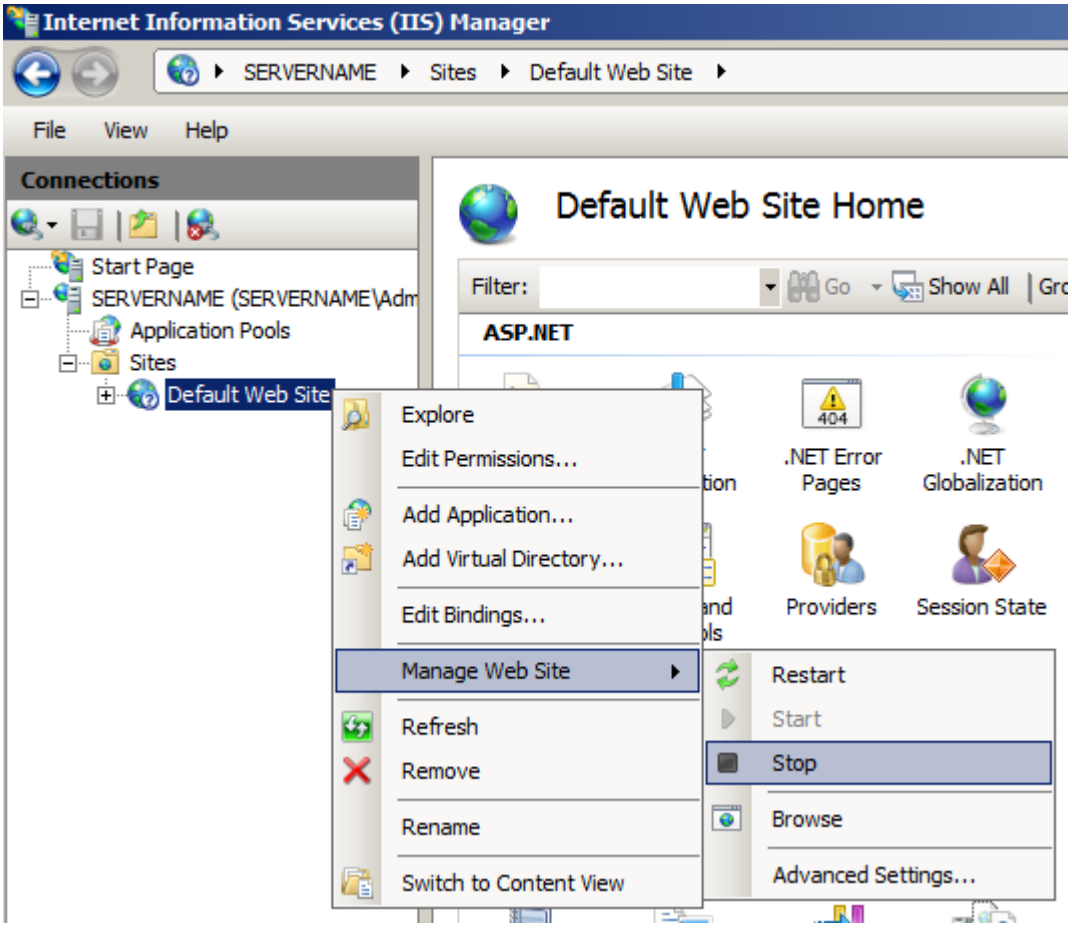
Creating a Web Site

Now that you have created folders with the appropriate access permissions, you will need to create a web site to be used by the Media Suite services.

To create a web site for your Media Suite services:

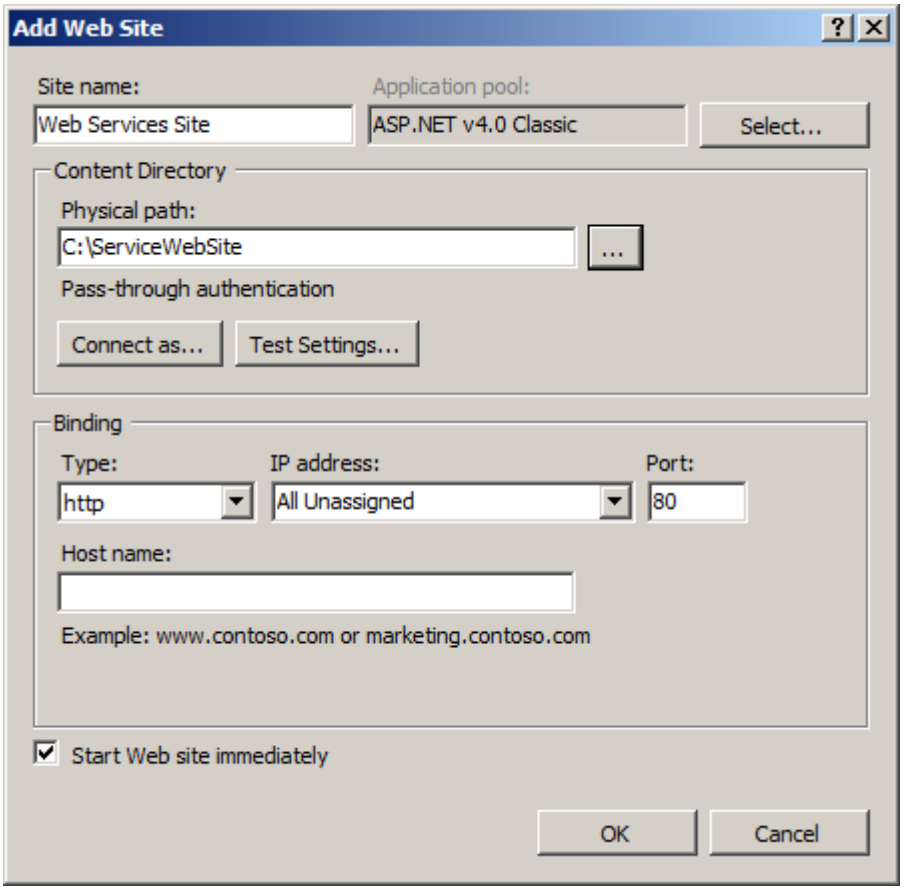
1. From your **Start** menu, click **Administrative Tools** > **Internet Information Services (IIS) Manager**.
   The **Internet Information Services (IIS) Manager** window opens.
2. Expand the tree for your local computer; expand **Sites**.
3. For security purposes, on the **Default Web Site**, click **Manage Web Site** > **Stop**.

Figure 19 Stopping the Default Web Site



4. Right-click the **Sites** directory and select **Add Web Site**.
   The **Add Web Site** dialog appears.

Figure 20 Add Web Site Window



5 In the **Site name** field, type `Web Services Site`.

6 For **Application pool**, use the **Select...** button to browse to the **ASP.NET v4.0 Classic** option and click **OK**.

7 In the **Physical path** field, browse to the web site directory you created in earlier (for example, `c:\ServiceWebSite`).

8 For **IP Address**, select the desired address.

9 For **Port**, enter 80.

Note If you are using any of these settings for another web site, Windows server will warn you about the potential conflict, but will allow you to continue.

---

10 Click **OK**.

Your new web site will appear in the list and should be running.

Chapter 8

Installing Services on Windows Server 2008

Design improvements in Media Suite Release 5.5 and later have led to a streamlining of the installation process for Windows components. All Windows services have been removed, also removing the need to install, start, and otherwise manage those components. Instead, only Web services are now used. The following chapter details instructions for installing the Media Suite Windows Web service components. Prior to installing these components, all the Windows pre-installation steps should be performed on the operating system. If you have not yet prepared your Windows environment for installing these Web services, return to Chapter 7, “Preparing Windows Server 2008 for Media Suite”.

Installing Web Service Components

The installation instructions for this chapter cover the following topics:

- "Verifying the PlayReady License Service Version", as shown below
- “Installing PlayReady License Service” on page 64

The final section describes how to update PlayReady license keys prior to their expiry:

- “Updating PlayReady License Service Keys” on page 68

Installing PlayReady SDK (x64)

Prior to performing the PlayReady License Setup, you must install the appropriate PlayReady x64 SDK for your deployment as shown on “Preparing Your Windows Environment” on page 49. PlayReady SDKs can be obtained from Microsoft. The following installation process will reference the contents of the PlayReady SDK folder.

Note You cannot simply copy the PlayReady SDK folder onto new servers that are used to generate licenses. You need to run the installer individually on each server in order to properly perform the SDK installation.

Verifying the PlayReady License Service Version

If you need to verify the PlayReady License Service version that is installed on your Windows server, there are a couple of ways to perform this action.

To view the PlayReady License Server version using Windows Explorer:

1 Navigate to where the PlayReady License Service is installed.

2 Open the **bin** folder.

3 Right-click `opencase.drm.license.playready.dll` and select **Properties**.

4 The **Details** tab will display the product version of the license server.

To view the PlayReady License Server version in "Programs and Features":

1 In Windows, navigate to **Control Panel > Programs > Programs and Features**.

2 Scroll down to "Media Suite 5 PlayReady License Service". The version number will be displayed in the list.

Installing PlayReady License Service

The file `PlayReadyLicenseService.5.5` contains the web component of the PlayReady License Service.

Note Prior to running this installer, the PlayReady SDK must already be installed and you must have your PlayReady License keys available.

To install the PlayReady License Service:

1 Use the Windows Control Panel to uninstall any previous version of PlayReady License Service from your system.

2 The PlayReady License Service Installer is contained within a compressed archive. Open the archive and browse to the **WindowsServices.5.x-{BUILD_DATE}/ PlayReadyLicenseService.5.x** directory.

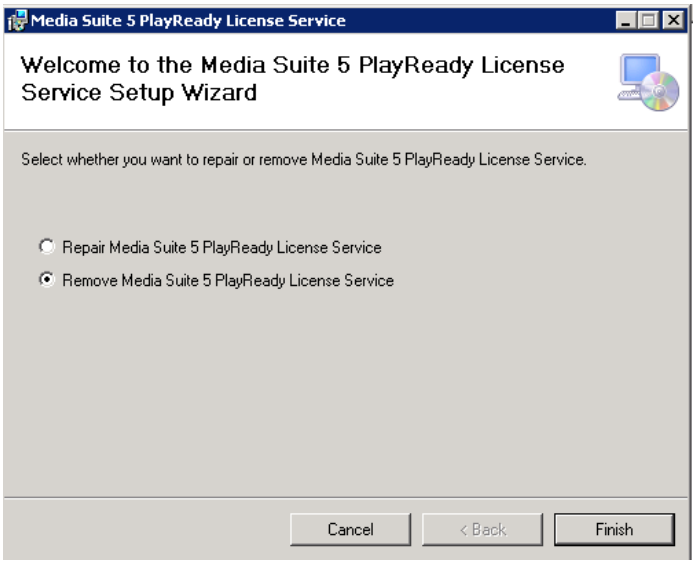
Note The PlayReady License Service Installer can be decompressed using software such as 7-Zip or WinRAR.

---

3 Double-click **Setup.exe** to launch the installer.

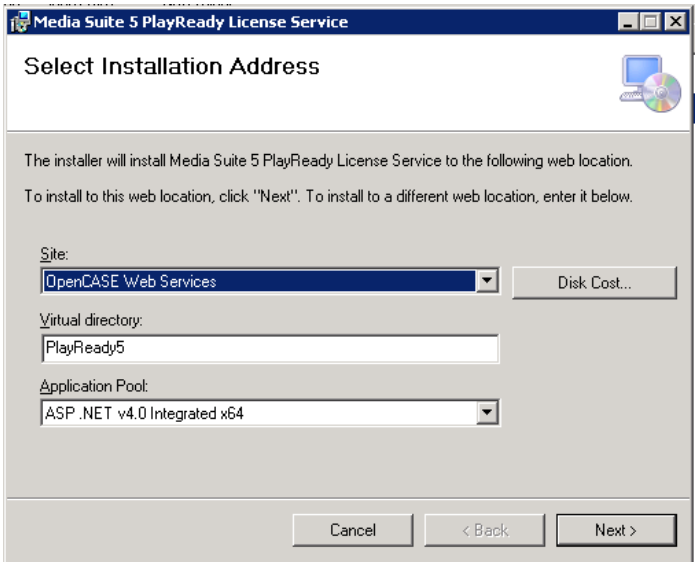
4 On the **Welcome** window, click **Next**. If you have an existing PlayReady License Server instance, you will be given the option to either remove or repair it.

Figure 21 PlayReady License Service Installation Welcome Dialog



Otherwise, the **Select Installation Address** window opens.

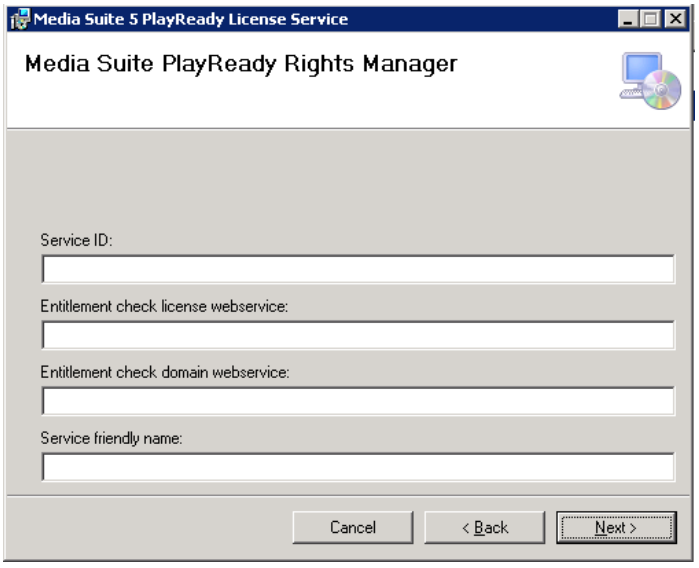
Figure 22 Select Installation Address Window, PlayReady License Service



- For **Site**, select the **Web Services Site** option.
- For **Virtual directory**, accept the **PlayReady5** default or enter a custom name for your directory.
- For **Application Pool**, select the **ASP.NET v4.0 Integrated x64** option.

5 Click **Next**.
The Media Suite PlayReady Rights Manager window opens.

Figure 23 Media Suite PlayReady Rights Manager Window



- For **Service ID**, enter a value or leave it blank. The Service ID is only mandatory if you need domain functionality with your PlayReady licenses.
- For **Entitlement check license webservice**, enter the location for any WSDL that will be used to manage entitlements. A sample URL would be:
  `http://{server}/EntitlementManager/webservices/playreadylicense-service?wsdl`
- For **Entitlement check domain webservice**, enter the location for any WSDL that will be used to manage device domains. A sample URL would be:
  `http://{server}/EntitlementManager/webservices/playreadydomain-service?wsdl`
- For **Service friendly name**, enter a value that will be shown to consumers during the registration of their domains.

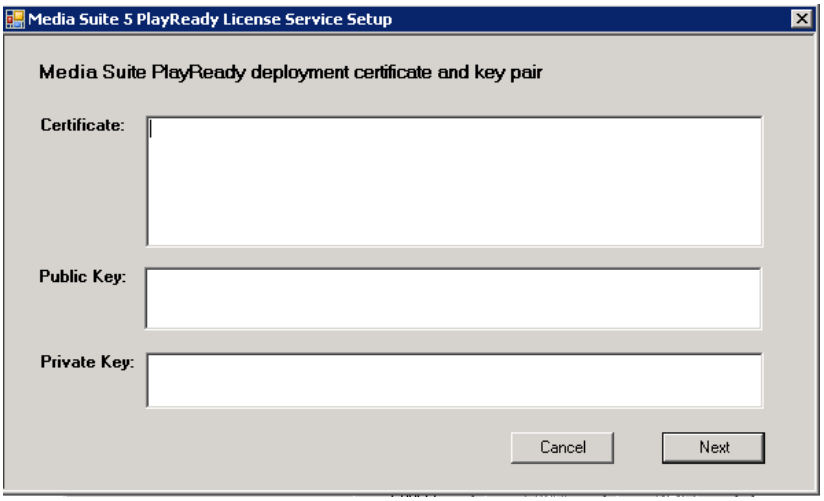
Note For all values in this dialog, if you leave these values blank, no values will be placed in the `web.config` file and entitlement checks and domain functionality will not be configured on your server.

---

6 Click **Next**.

7 On the **Confirm Installation** window, click **Next**.
The **Media Suite PlayReady deployment certificate and key pair** window opens.

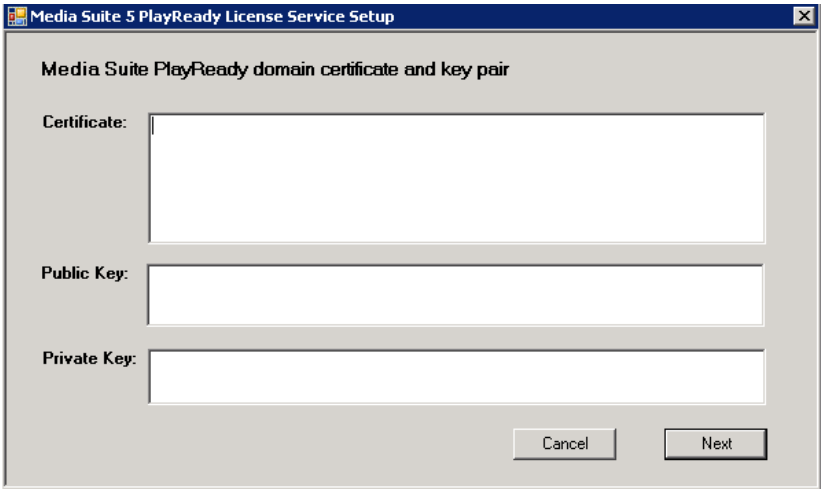
Figure 24 OpenCASE PlayReady Deployment Certificate and Key Pair Window



8 For **Certificate**, **Public Key** and **Private Key**, enter the values provided by Microsoft with your PlayReady licence. They are typically referred to as the Deployment Certificate, Deployment Public Key, and Deployment Private Key. If you leave these fields blank, the PlayReady service will not be configured on your server. Click **Next**.

The **OpenCASE PlayReady domain certificate and key pair window** opens.

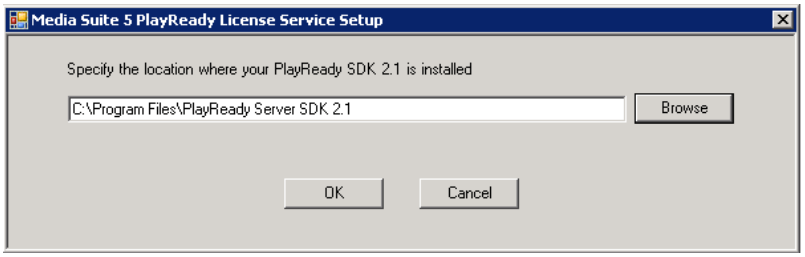
Figure 25 OpenCASE PlayReady Domain Certificate and Key Pair Window



9 For **Certificate**, **Public Key** and **Private Key**, enter the values provided by Microsoft with your PlayReady licence. They are typically referred to as the Domain Certificate, Domain Public Key, and Domain Private Key. Click **Next**.

10 At the prompt **Specify the location where your PlayReady SDK is installed**, browse to the directory PlayReady SDK is installed.

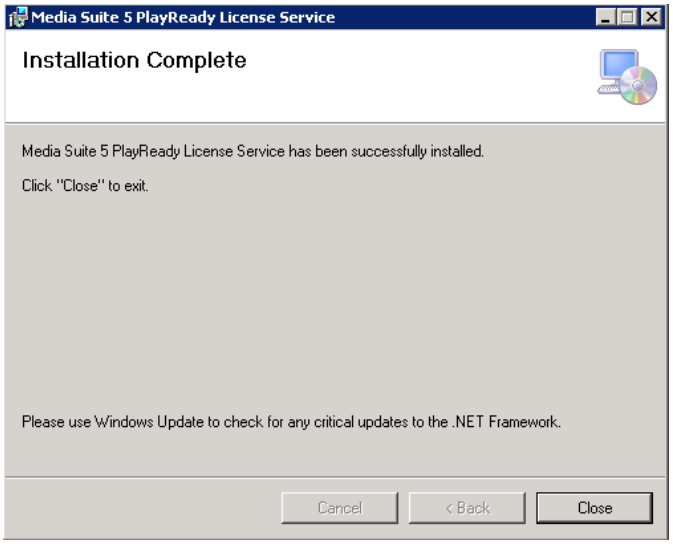
Figure 26 Setting the PlayReady SDK Location



11 Click **OK**.
The Media Suite PlayReady License Service should now install on your system.

12 On the **Installation Complete** window, click **Close**.

Figure 27 Installation Complete Dialog



Updating PlayReady License Service Keys

Depending on the length of your license agreement with Microsoft, you will need to update the license keys on your PlayReady installation before they expire. The following section explains the process for updating license keys on your server.

To update license service keys:

1 Obtain a new set of PlayReady License keys from Microsoft. Give yourself leeway for any administrative delays in the processing of your new keys.

2 Navigate to `c:\VMS_Media\PlayReady5` and run `PlayReadyConfigEditor.exe`. This will open the web.config file.

3 Since the web.config is encrypted, you will see the tag `EncryptedData` within the file. Click **Decrypt**.

4 Overwrite the existing (outdated) key values with those of your new license.

5 Click **Encrypt** to re-encrypt the web.config file.
The new license keys will automatically take effect.

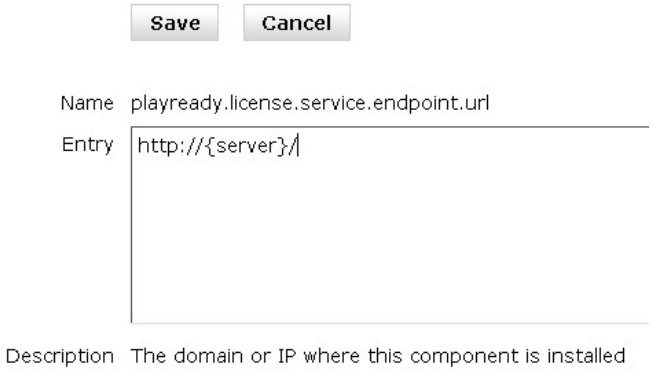
Configuring PlayReady License Settings in Media Suite

After all the Media Suite Web services have been successfully installed on your Windows Server, you will need to configure the Endpoint and WSDL context on the Media Suite system configuration page in the user interface.

To configure the PlayReady endpoint and WSDL within Media Suite:

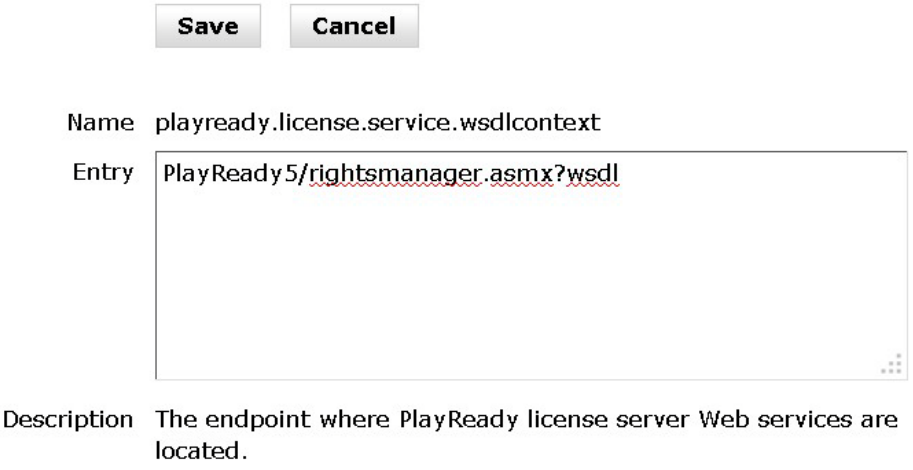
1. Within Media Suite, navigate to **Admin > Setup > Configuration**.
2. Open the `modules > em > playready > playready.license.service.endpoint.url` node.

Figure 28 Setting the PlayReady License Server Endpoint



3. Type the domain or IP where the PlayReady License Server is installed.
4. Click **Save**.
5. Open the `modules > em > playready > playready.license.service.wsdlcontext` node.

Figure 29 Setting the PlayReady License Server WSDL Context



6. Type the WSDL context for the license server Web service. Make sure that the PlayReady number references the correct version.
   For example, `PlayReady5/rightsmanager.asmx?wsdl`
7. Click **Save**.

Chapter 9

Installing RemoteFS

Media Suite 5.x has the capability to install RemoteFS on Linux. The module resides within a self-contained `bin` file, and does not have any external dependencies.

RemoteFS Prerequisite

As with Media Suite, JDK 1.7.0 update 51 must be installed.

Installing RemoteFS

The following process should be used to install RemoteFS:

1. Change the execute permissions on the file.
   `chmod +x vms_remotefs-5.x-buildno.bin`
2. Execute the file.
   `./vms_remotefs-5.x-buildno.bin`
3. Provide the following information requested by the installer.
   - A base directory, which should be entered as an absolute path. Any contents within this base directory will be erased.
   - The RFSUser password. This is the same password that was used to access the RFS Web service.
   - The JMX password.

Starting RemoteFS

To start RemoteFS, use the following command:

`start jboss_init_{rfs_instance_name}.sh`

Verifying RemoteFS Installation

The following section describes the steps that are required to verify your installation.

To verify a RemoteFS Installation:

1. Browse to: `http://{RemoteFS_SERVER}:{Listen_Port}/RemoteFS/vfsCommand?wsdl`.
2. If RemoteFS is properly installed and configured, you will see XML output in the browser.

Other Service Tasks

The following section includes other tasks that may prove useful for your RemoteFS deployment.

Enabling Additional Protocols for RemoteFS

Each time RemoteFS is started, the system reads an XML file to ascertain the protocols for connected filesystems. The following section describes the file and the field values that must be populated to enable additional protocols for an RFS server.

To add additional protocols to RemoteFS:

1 Edit `(SERVER}\JBoss\server\RemoteFS\deploy\remote-vfs-systems.xml`

The original file appears as such:

```
<?xml version="1.0" encoding="utf-8" ?>
<systems>
  <system type="ftp" junctionName="FTPSAMPLE1" />
</systems>
```

2 To configure additional protocols, you need to create a block of new fields with related values for each protocol as per the following example. Irrelevant fields values may be left blank, but all fields in the set should be included.

```
<systems>
  <system type="value" junctionName="value">
     <host>value</host>
     <port>value</port>
     <userName>value</userName>
     <password>value</password>
     <domain>value</domain>
  </system>
</systems>
```

Note If you are copying the original XML from step 1, make sure that you remove the "/" that appears after the "junctionName" value so that the XML is properly parsed.

3 After any changes have been made, you will need to restart your RemoteFS service for them to take effect.

Explanations for the protocol field values are shown in the following table:

Table 20 Protocol Fields to Configure for RemoteFS

Protocol	Fields
file	**system type** - the relevant protocol type e.g. "file" **junctionName** - the repository node name as defined in the Repository Manager page on the Media Suite user interface. **host** - the filesystem path that you are mounting. **port** - the port number (if required by the protocol) **userName** - leave this field blank. It is not applicable. **password** - leave this field blank. It is not applicable. **domain** - leave this field blank. It is not applicable.

Table 20 Protocol Fields to Configure for RemoteFS

Protocol	Fields
ftp	**system type** - the relevant protocol type e.g. “ftp” **junctionName** - the repository node name as defined in the Repository Manager page on the Media Suite user interface. **host** - the filesystem path that you are mounting. **port** - the port number (default is 21) **userName** - the filesystem username **password** - the filesystem password **domain** - the domain (if required by your deployment)

Protocol	Fields
sftp	**system type** - the relevant protocol type e.g. “sftp” **junctionName** - the repository node name as defined in the Repository Manager page on the Media Suite user interface. **host** - the filesystem path that you are mounting. **port** - the port number (default is 22) **userName** - the filesystem username **password** - the filesystem password **domain** - the domain (if required by your deployment)
smb	**system type** - the relevant protocol type e.g. “smb” **junctionName** - the repository node name as defined in the Repository Manager page on the Media Suite user interface. **host** - the filesystem path that you are mounting. **port** - the port number (if required by the protocol) **userName** - the filesystem username **password** - the filesystem password **domain** - the domain (if required by your deployment)

Configuring RemoteFS in Media Suite

After modifying the `remote-vfs-systems.xml` file to account for your required protocols, you will need to configure the corresponding repository node information within the Media Suite interface.

To configure RemoteFS within Media Suite:

1. Within the Media Suite interface, navigate to **Workflow > Repository Manager**.
2. Click on the underlined Repository Node name or create a new repository node, if necessary.
3. Type in the Repository Node name, the RemoteFS service credentials, choose the PFS protocol from the dropdown, and type the server name and port number from which the public will download the media.
4. Save your changes.

Changing the RemoteFS User Credentials

If you need to verify or update the RemoteFS service username or password, they are stored within the following file:

```
c:\JBoss\server\RemoteFS\conf\props\RemoteFSApplication-users.properties
```

Uninstalling RemoteFS

Remove the RemoteFS directory.

Chapter 10

Configuring MOS

Introduction

Videoscape Media Suite (VMS) is a modularized system that can facilitate all backend aspects of automated workflow processing, packaging, and the delivery of monetizable digital assets. VMS is highly customizable, and supports a broad range of licensing, transcoding, encryption, and distribution models without the need to develop or redeploy new software.

As part of that customizability, Cisco can create Custom ESB (Enterprise Service Bus) services to interact with other systems such as Cisco's MOS (Media Origination System). MOS is a system that takes input from media encoding systems and produces output that can be distributed by CDNs and consumed by various client devices at different bit rates.

Purpose

The purpose of this document is to guide administrators in configuring Media Suite and MOS so that those systems can share assets and metadata via the MOS VOD ESB service. Testing criteria and scenarios are also described.

MOS VOD ESB Service

MOS Setup

Once the MOS environment is running (independent of VMS), settings need to be configured so that VMS can communicate with MOS. When configuring MOS, you will need to have access to the Cisco Media Origination System Release 2.3 User Guide. Consult the "Setting Up the MOS Infrastructure" section, paying particular attention to the following items:

1. NAS Store (Output)
2. NAS Media Storage (Input)
3. The Publish Template
4. Creating and Enabling Media Services
5. The Capture Endpoint
6. The Asset Workflow Template

Setup Considerations

During and after configuration, it is important to note some details that might affect the workflow of an asset:

1. The NAS Store contains the processed assets that MOS will reference. MOS examines those files to create a playlist for any required devices. This location has to be visible and accessible to VMS and any other systems in the deployment. Ensure that the VMS repository ESB read/write commands reference the directory that corresponds to this NAS Store location.
2. The NAS Media Storage is used by MOS as its output folder. VMS requires read access to this location.
3. Virtual IPs are not needed for Capture and Playback Endpoint setup within MOS.
4. Ensure that all services are active and that all asset workflow templates are enabled.

Protocols

The NAS Store and NAS Media Storage repositories can only be accessed by the NFS protocol. Armada (Cisco Transcode Manager) can only access file systems that use the SMB or CIFS protocols.

VMS Setup

To implement a full workflow encoding setup, the following items must be configured in VMS:

1 Workflow Template

2 Action Template for MOSVOD Service

3 Action Template for the Armada Service (or any other transcoder that will be used)

4 Action Profile for DELETE MOSVOD ESB

5 Action Profile for CREATE MOSVOD ESB

6 Action Profile for MOS Input (Repository Node ESB)

7 Action Profile for Images (Repository Node ESB)

8 Bind Profile for creating bundles that may contain manifests, videos, metadata, and images.

9 Policy (for Product Entitlement)

10 VMS Repository Node Hot Folder (for asset encoding)

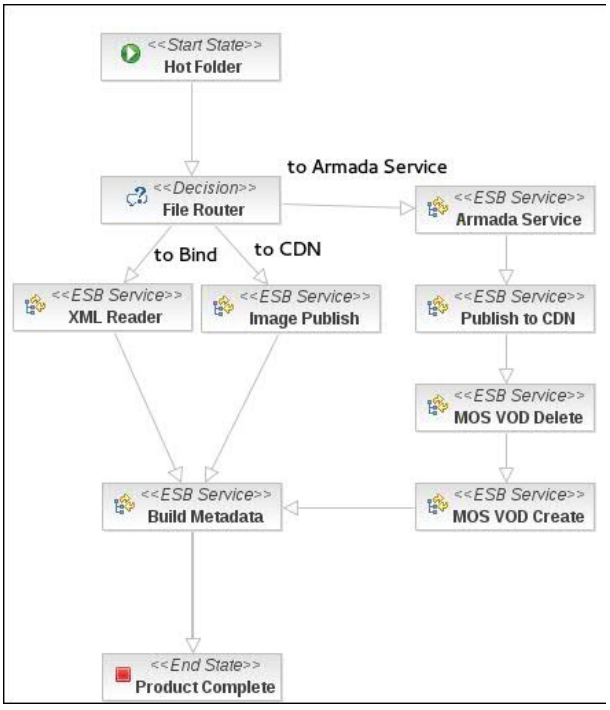
11 VMS Repository Node for MOS source files (This is the same location as the NAS Media Store.)

12 Workflow Definition

VMS Workflow

The image below represents a sample Media Suite workflow that utilizes the MOS VOD ESB service for processing XML metadata and video assets.

Figure 30 Sample Workflow for VMS, MOS, and Armada Usage



MOSVOD Action Template

Prior to setting up a workflow that enables communication between VMS and MOS, you must create an Action Template. For details on creating Action Templates, refer to the Media Suite User Guide. The following section, however, shows specific fields that are involved in this instance.

Figure 31 Choosing the MOS VOD ESB Service

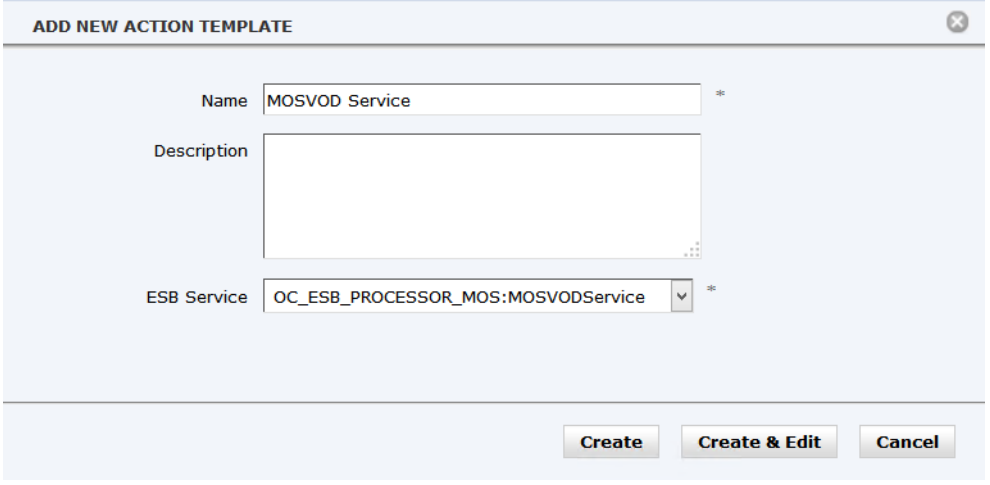
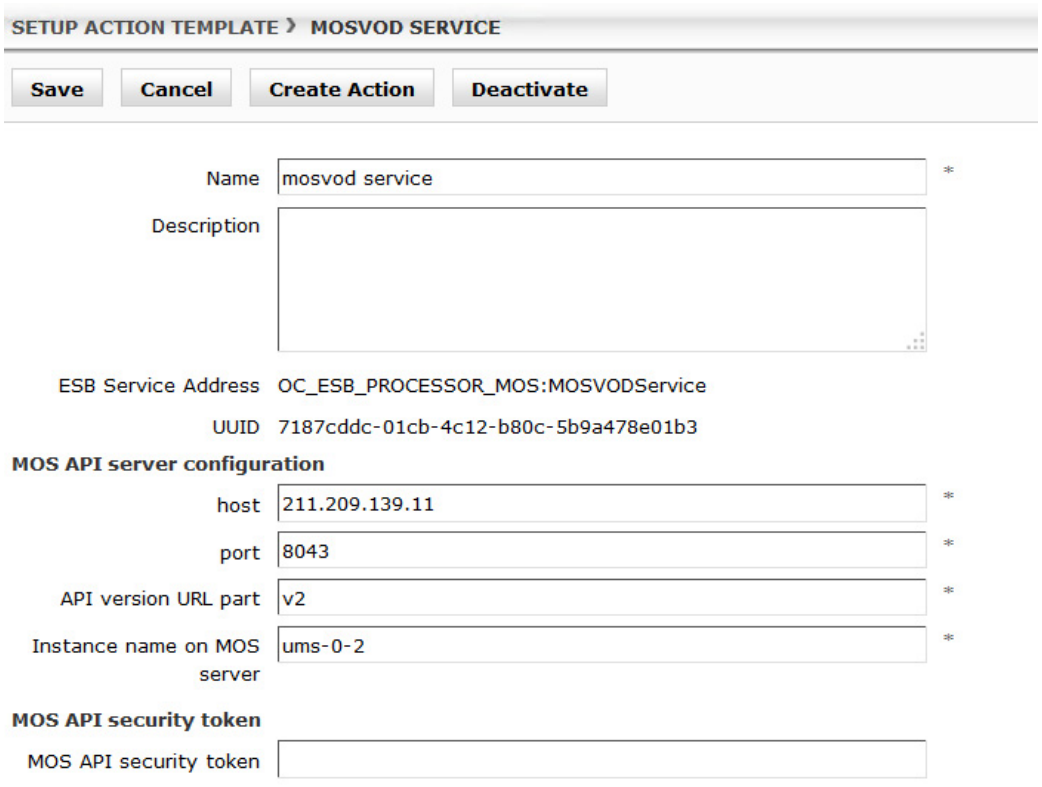


Figure 32 Configuring the MOS VOD Service Action Template



Host

The HOST is the domain name or IP address of MOS. By default, HTTPS is used for communications between VMS and MOS.

Port

This value is the port number of the MOS Service Manager. The Service Manager binds to port 8043 for external HTTPS access and to port 8001 for local HTTP access via the loopback address (127.0.0.1). The MOS GUI binds to port 8443 for HTTPS access.

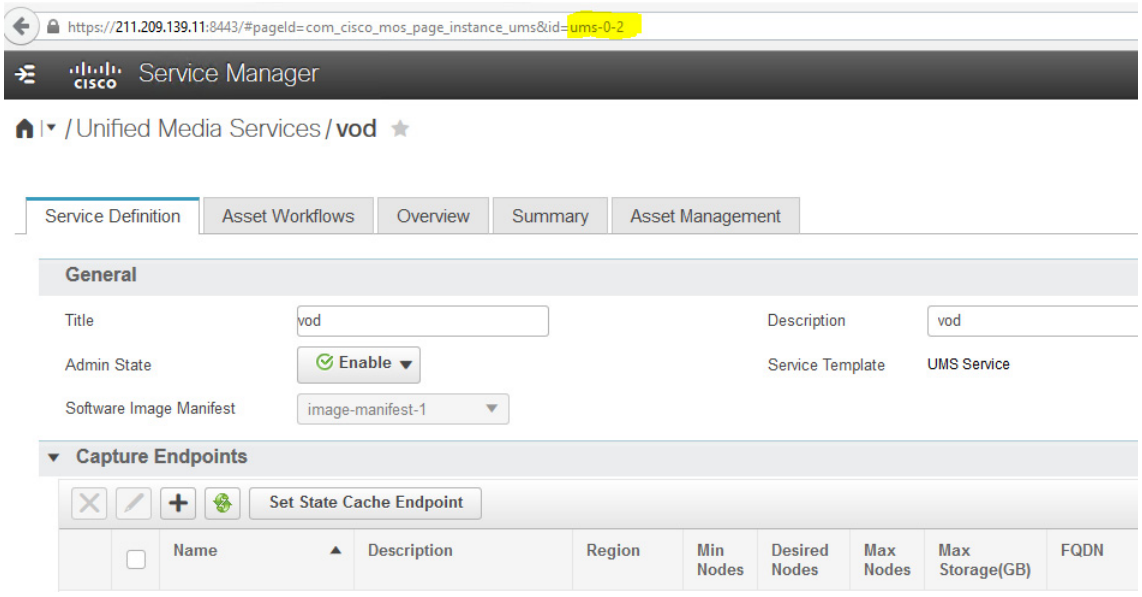
API Version

The API version is a string found in the REST URL requests sent to MOS. For MOS release 2.3, that string is "v2". The string may change depending upon the version of MOS that is being used. Consult the MOS User Guide for API version updates as required.

Raw Service Instance Name

The Raw Service Instance Name can be found in the MOS user interface. When the service instance of choice is selected, the name can be seen in the browser URL. The diagram below shows the highlighted name as "`ums-0-2`".

Figure 33 Finding the raw service instance name in MOS



MOSVOD Action Profile

The following section describes fields that must be populated to create the MOS VOD Action Profile:

Command Type

There are two command types: DELETE and CREATE.

The DELETE command type deletes incoming file sets via the workflow. The MOS Content ID found in the VMS physical asset (i.e. component) is used to find the corresponding asset on MOS, which is then removed from the MOS environment.

The CREATE command type allows nodes to create a VOD playback playlist with the provided file set as input. This input follows the accepted MOS formats (H.264 files contained within MPEG TS or the Microsoft Smooth Streaming). For content updates, the MOS Content ID is always generated prior to executing the CREATE ESB service.

- MOS updates are performed in the following manner:
- VMS directs MOS to delete the MOS Content and its ID (DELETE workflow note)
- VMS updates the physical asset(s) (CREATE workflow node)
- VMS generates a new MOS Content ID for the updated assets
- VMS sends the updated content and MOS Content ID to MOS
- MOS creates the playlist using the MOS Content ID
- MOS confirms the playlist creation to VMS

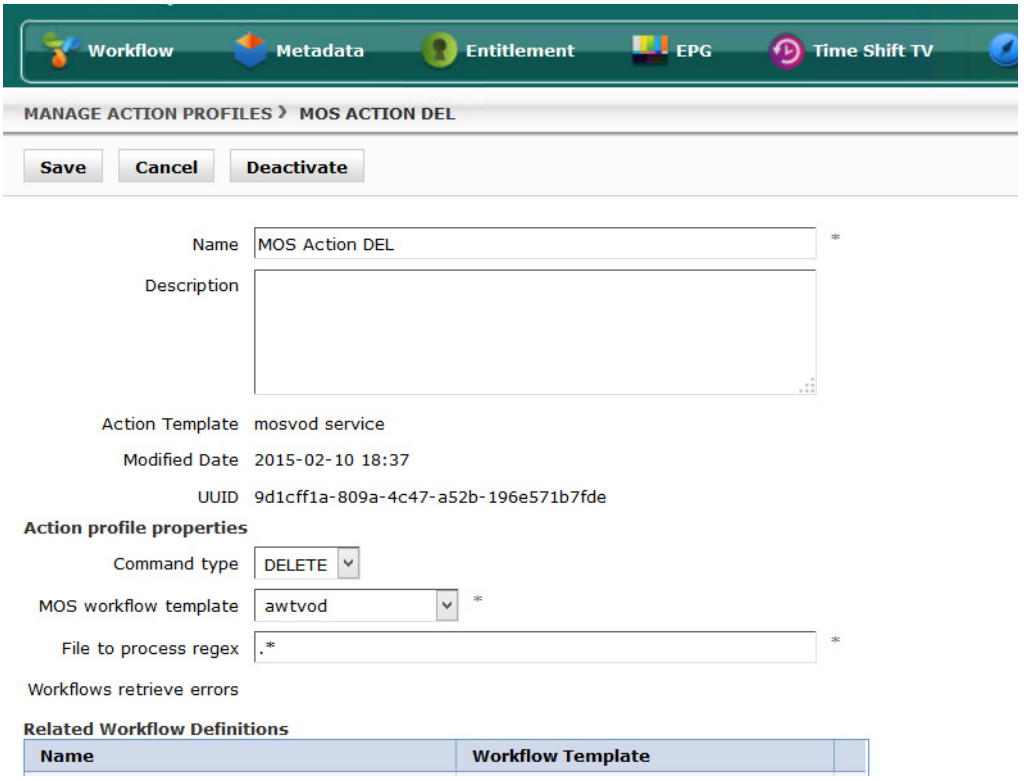
MOS Workflow Template

This dropdown field displays the available asset workflow templates that are available on the MOS environment. If no templates are listed, check the MOS environment or the connectivity between VMS and MOS to ensure that the template information is being shared between the two systems. The selected MOS asset workflow template will dictate the output of MOS and the VOD playback playlists.

File to Process Regex

This field contains a regex string that determines which files from a given file set will be used in the DELETE or CREATE request.

Figure 34 Sample MOS VOD Action Profile



Creation of Assets

The successful creation of a VMS asset will be accompanied by the creation of a bundle contains a manifest with a public URL showing a playlist file (typically in a .m3u8 or .ism format). The physical assets within that bundle will contain a MOS Content ID.

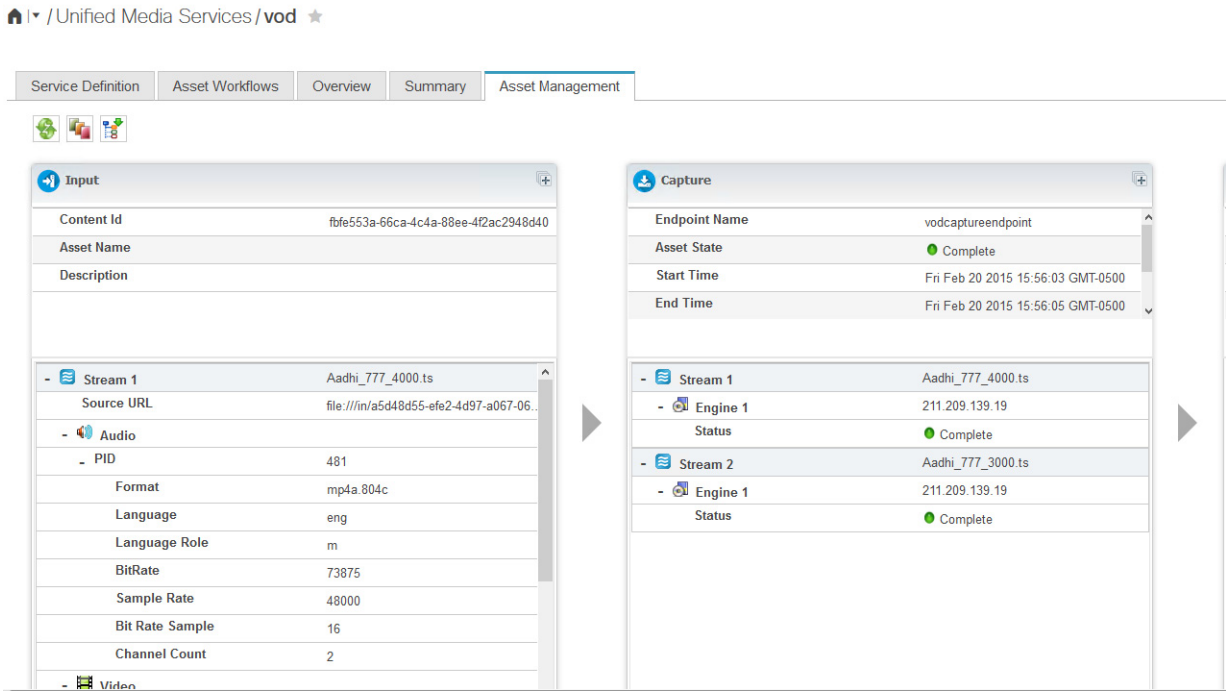
MOS Content ID

The MOS Content ID is a unique read-only foreign key that is shared by all physical assets within a bundle. VMS uses this ID to reference MOS Content for the creation and deletion of assets on the MOS system. MOS Content IDs may still exist even if an error has occurred during a workflow. In

that instance, related assets may have been created on the MOS system. In order for the workflow to successfully recover from the error, the assets would need to be deleted on the MOS system before they could be recreated.

Notice on the screen below that the capture status is shown as "Complete" for all streams. Also note that the Publish URL is populated. That Publish URL field contains the MOS-generated playlist.

Figure 35 Successful Asset Processing by MOS



MOS Logging

The MOS system records the results of all video content that it processes. If errors occur during ingestion, they will be found on the Asset Management UI page in MOS. Other error details can be accessed on the VMS workflow status page or in the server logs.

Configuring JAVA SSL Certificates

If you have not placed Java SSL Security Certificates into the VMS JVM, then an error will be thrown when you try to send requests to MOS via SSL. SSL is a requirement for setting up a MOS deployment.

Figure 36 Sample MOS Security Error

```
2015-02-10 14:03:48,082 ERROR
[com.extend.opencase.esb.services.mos.impl.MOSVODService] (pool-345-thread-1) cannot
obtain workflow names from MOS service API exception
message:[javax.net.ssl.SSLHandshakeException:
sun.security.validator.ValidatorException: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid
certification path to requested target]
```

The CA (Certified Authority) certificate and Server certificates need to be copied from MOS to all nodes containing "server_esb" in the VMS environment. This will ensure that SSL handshakes between VMS and MOS are successful. You can find the "ca.crt" and "server.crt" at the follow location on the MOS server:

`/etc/opt/cisco/mos/public/`

To configure Java SSL Certificates on VMS:

1. Run the following command on the VMS server. Ensure that you have the proper permissions for executing the commands.

   ```
   keytool -import -noprompt -trustcacerts -alias MOSVODCA -file /
   {certificate location}/ca.crt -keystore /usr/lib/jvm/{active JVM version}/jre/lib/
   security/cacerts -storepass {your-password}

   keytool -import -noprompt -trustcacerts -alias MOSVODSVR -file /{certificate
   location}/server.crt -keystore /usr/lib/jvm//{active JVM version}/jre/lib/
   security/cacerts -storepass {your-password}
   ```

   Note Replace **{active JVM version}** with the JVM that is being used by VMS. For example "`java-1.7.0-openjdk-1.7.0.51.x86_64`". Also, the default keystore password is "changeit". We recommend that you use a new secure password.

2. Add the keystore with the newly added certificates as a run parameter for VMS. This is done by editing the "run.conf" file located in:
   `/opt/cisco/vms/var/jboss/bin/`

3. Within `run.conf`, search for "`JAVA_OPTS`" and add the following line to that parameter:
   `-Djavax.net.ssl.trustStore=/usr/lib/jvm/{active JVM version}/jre/lib/security/cacerts"`

That parameter line will appear at the end of the `JAVA_OPTS` section as shown here:

```
if [ "x$JAVA_OPTS" = "x" ]; then
   JAVA_OPTS="-server -Xms${JVM_XMS_RAM_ACTUAL}m -Xmx${JVM_XMX_RAM_ACTUAL}m -
XX:MaxPermSize=1024m -XX:ReservedCodeCacheSize=128m -
Dcom.iplanet.am.cookie.encode=true -Dcom.sun.identity.configuration.directory=/opt/
cisco/vms/var/sso -Djboss.messaging.ServerPeerID=620 -
Djboss.messaging.groupname=OlGIvyzv_5_6-92626 -Djgroups.bind_addr=vms-app-
02.vdc3.vcte.com -XX:+HeapDumpOnOutOfMemoryError  -XX:+CMSPermGenSweepingEnabled -
XX:+CMSClassUnloadingEnabled -XX:+UseCompressedOops -XX:+UseConcMarkSweepGC -
XX:+UseCodeCacheFlushing -Dvms.node.id=620 -Dvms.oci.version=5.6-92626 -
Djava.awt.headless=true -Dffs.home=/opt/cisco/vms/state/ffs/slave_home -
Djboss.partition.name=OlGIvyzv_5_6-92626 -Djboss.partition.udpGroup=228.1.2.3 -
Djgroups.udp.mcast_addr=228.1.2.3  -Djboss.default.jgroups.stack=tcp -
Djboss.platform.mbeanserver -Djava.rmi.server=vms-app-02.vdc3.vcte.com -
Djavax.management.builder.initial=org.jboss.system.server.jmx.MBeanServerBuilderImpl
-Dorg.jboss.resolver.warning=true -Dsun.rmi.dgc.client.gcInterval=3600000 -
Dsun.rmi.dgc.server.gcInterval=3600000 -Djavax.net.ssl.trustStore=/usr/lib/jvm/
{active JVM version}/jre/lib/security/cacerts"
fi
```

4. Restart the VMS server. SSL requests from VMS to MOS will no longer throw any errors.

Chapter 11

Configuring Merchandiser

Understanding Merchandiser

Merchandiser (formerly VCM) is an external Cisco component that integrates with Media Suite to manage catalog content, create site navigation structures, and to aid in the monetization of content. Configuring Media Suite and Merchandiser enables the synchronization of metadata between both applications and the reporting of bundle warning messages when required.

Note The names Merchandiser and VCM will be used interchangeably in the following chapter. While Merchandiser is the official Cisco product name, VCM appears within various places in the user interface, configuration files, and code.

Configuring Media Suite for Merchandiser

The following section details the process of configuring Media Suite to work with Merchandiser.

To configure Media Suite to use Merchandiser:

1. Navigate to **Admin > Setup > Configuration**.
2. Within the system configuration tree, open the **modules > cm > vcm** node.
3. Make any necessary changes to the following child nodes of the VCM node. Ensure that the values are identical within both Media Suite and, later, when configuring Merchandiser:

Table 21 Default Values in VMS System Configuration Nodes

VMS Node	Default Value	Value for VCM Integration
URL	Empty	Type the appropriate callback URL. This is the VCM endpoint to which Media Suite submits bundle updates. For example: http://{server-web.cisco.com}:5430/gateway/submit
app.name	DEFAULT	assetSource.default.httppost.appName This is the identifier configured within VCM that uniquely identifies response messages intended for the current Media Suite instance.

Table 22 Default Values in VMS System Configuration Nodes

VMS Node	Default Value	Value for VCM Integration
asset.source.id	Empty	Same as `assetSource.default.name` in the vcm-cmsstatus configuration. This is the identifier configured within VCM that uniquely identifies messages sent by the current Media Suite instance.
prg.name	StatusMessage	assetSource.default.httppost.prgName The name of the VCM service that Media Suite expects responses from.
http.socket.timeout	60000	This is the read timeout (in milliseconds) for VCM HTTP connections.
http.conection.manager.timeout	60000	This is the connection timeout (in milliseconds) for VCM HTTP connections
logicalvideo.xslt.template (within the XSLT node)	vcmBundleXslt	This plugin transforms VMS logical video bundles into VCM bundles before they are posted to VCM. All XSLT templates must be activated prior to use. Validation XSLTs will generate VCM bundle warnings when required.
logicalvideo.deleted.xslt .template (within the XSLT node)	vcmDeletedBundlesXslt	All XSLT templates must be activated prior to use.
videocollection.xslt.template (within the XSLT node)	vcmVideoCollectionXslt	This plugin transforms VMS video collection bundles into VCM bundles before they are posted to VCM. All XSLT templates must be activated prior to use. Validation XSLTs will generate VCM bundle warnings when required.
videocollection.deleted.xslt .template (within the XSLT node)	vcmDeletedVideoCollectionXslt	All XSLT templates must be activated prior to use.
logicalvideo.validation.xslt.template	vcmBundleValidation Xslt	An XSLT transformer that validates VMS bundle XML. Validation is performed for NDS ADI XML compliance. Once validation is complete, all bundle validation warnings can be found on the standard bundle warning page within Media Suite.
videocollection.validation.xslt.template	None	An XSLT transformer that validates VMS video collection bundle XML. Validation is performed for NDS ADI XML compliance. Once validation is complete, all bundle validation warnings can be found on the standard bundle warning page within Media Suite.

4 Set the following value in system configuration:

`modules > cm > bundle > record.deleted.bundle.xml = true`

Configuring Merchandiser

Configuration settings within Merchandiser are set within its user interface (on the Asset Source Management page) and as well as within the `config.properties` configuration file located on the Merchandiser servers. For details, see the *Merchandiser Installation, Configuration and Invocation Guide* (named *AIM-USR-386 CMS VCM Server ICI*).

Ensure that you have configured the following within the VCM cms-status page:

1 Update the following value in the `config.properties` file.

```
assetSource.default.httppost.postbackURL=
http://{server}:80/opencase/ContentManager/resource/rest/VCM/callback
```

This postback URL is used by Merchandiser to notify Media Suite about the status of the bundle that VMS posted earlier on.

After you have finished configuring Merchandiser, you will need to restart all Merchandiser services by using the following command: `nds_cmsstatus restart`

Activating Bundle XSLT Files

By default, all bundle XSLTs within Media Suite are inactive and relevant XSLTs must be activated in order to use Merchandiser.

To active the XSLT templates:

1 Navigate to **Metadata > Setup > Bundle XSLT.**

2 Click the **VCM Logical Video Bundle XSLT Template**.

3 Confirm that the XSLT ID matches the value in **modules>cm>vcm>xslt>logicalvideo.xslt.template** within the system configuration.

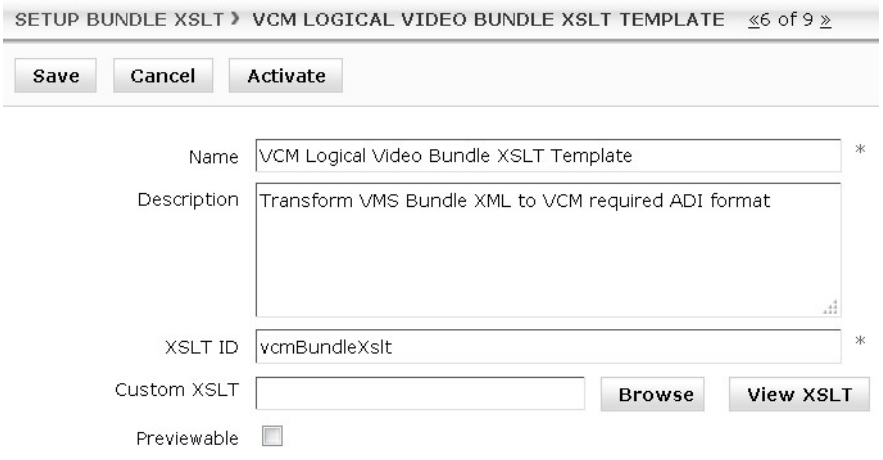
4 Click **Activate**.

5 Navigate to **Metadata > Setup > Bundle XSLT.**

6 Select the **VCM Logical Video Delete Bundle XSLT Template**.

7 Confirm that the `XSLT ID` matches the value in **modules>cm>vcm>xslt>logicalvideo.deleted.xslt.template** within the tree node.

8 Click **Activate.**



9 Repeat the above steps for any other VCM XSLT Templates that are required by your deployment.

Activating Merchandiser Plugins

By default, all Merchandiser plugins within Media Suite are inactive.

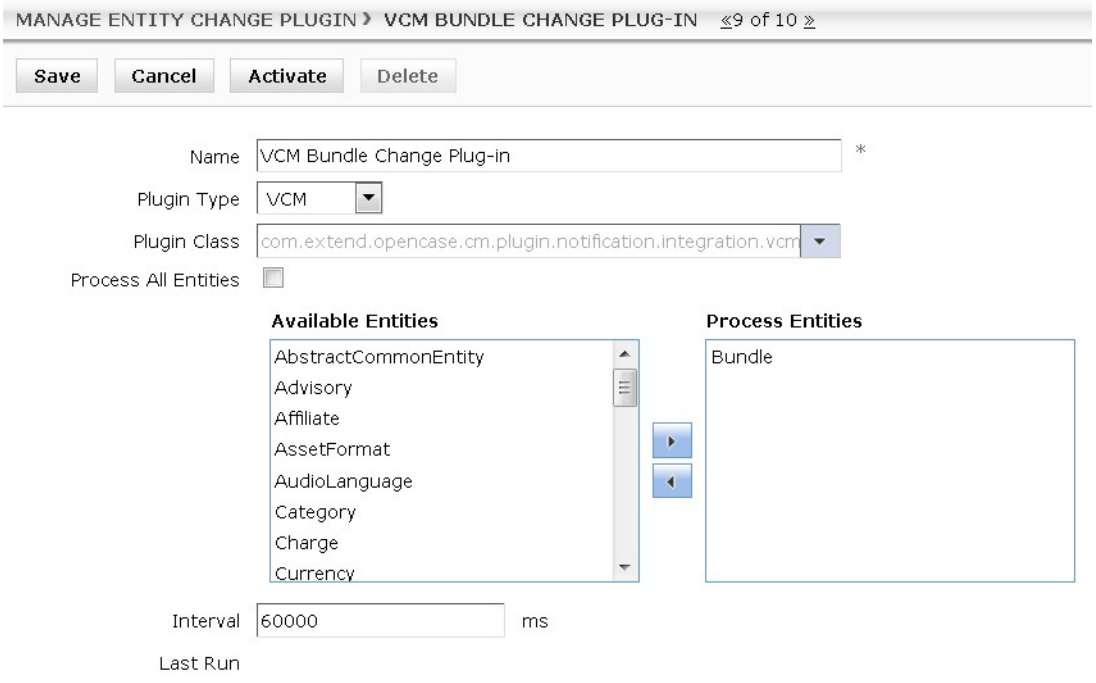
To activate Merchandiser plugins:

1 Navigate to **Metadata > Setup > Plugins > Entity Change Plugins**

2 Click **VCM Bundle Change Plug-in.**

3 Ensure that the **Bundle** entity exists in the **Process Entities** pane.

4 Click **Activate.**



Appendix A

LINUX RPM PROCEDURES

The following appendix explains various Linux RPM procedures and syntax that are important for installing and managing RPM packages.

Installing RPM Packages

This section describes the process of installing RPM package files, which must be performed on each deployment node.

To install RPM packages:

1. Log in as a root or a user with sudo priviledges.
2. Type the command:
   ```
   rpm -Uvf /path/to/vms_package_name.rpm
   ```
3. A message will be displayed indicating that the installation has started. Once it is complete, the cursor will come back. If any prerequisites are not met, you will get an error message indicating any details and the package installation will abort.

Listing Running Packages

To list packages that are running on a particular node:

1. Type the command:
   ```
   rpm -qa | grep vms
   ```
2. A list will be displayed that indicates any installed packages with vms in the name. A sample follows:
   ```
   [node_Hostname~]$ rpm -qa |grep vms

   vms_base-5.x.x-BuildNo.x86_64
   vms_search_config-5.x.x-BuildNo.x86_64
   vms_epg_tstv-5.x.x-BuildNo.x86_64
   vms_search_master-5.x.x-BuildNo.x86_64
   vms_server_esb-5.x.x-BuildNo.x86_64
   vms_keymanagement-5.x.x-BuildNo.x86_64
   vms_reporting-5.x.x-BuildNo.x86_64
   vms_cms-5.x.x-BuildNo.x86_64
   vms_workflow-5.x.x-BuildNo.x86_64
   vms_modcluster-5.x.x-BuildNo.x86_64
   vms_admin-5.x.x-BuildNo.x86_64
   vms_entitlement-5.x.x-BuildNo.x86_64
   vms_ocesb_epg-5.x.x-BuildNo.x86_64
   ```

Removing RPM Packages

To remove an installed package:

1. Type the command:
   `rpm -e package_name` (without the version number)
   For example, `rpm -e vms_base`

Package Dependencies

It is important that all dependencies that are required for a package be met. To facilitate your examination of any required or supplied dependencies, Linux has some built-in command line options available. This section will explain those options.

Note To avoid software incompatibilies, do not mix package versions from different releases. VMS will enforce package version dependencies.

Querying Packages for Dependencies

on an rpm file, regardless of whether it is installed.

To query a given package for RPM dependencies:

1. Type the command:
   ```
   rpm -qp --requires vms_ocesb_epg-5.0.0-BuildNo.x86_64.rpm
   ```
2. A list will be displayed that indicates any dependencies for the specified RPM package. Sample output follows:
   ```
   vmsvp_jbossas-5.0-BuildNo
   vmsvp_jbossesb-5.0-BuildNo
   vms_workflow = 5.0-BuildNo
   ```

The following lines show system level output that is outside of VMS's control and that may be ignored:

```
/bin/sh
/bin/sh
/bin/sh
/bin/sh
rpmlib(PayloadFilesHavePrefix) <= 4.0-1
rpmlib(CompressedFileNames) <= 3.0.4-1
```

Querying Packages for Provided Components

on an rpm file, regardless of whether it is installed.

To query a package for components that it provides:

1. Type the command:
   ```
   rpm -qp --provides vms_server_esb-5.x.x-BuildNo.x86_64.rpm
   ```

2 A list will be displayed that indicates any components that are provided by this package.

Sample output follows:

```
vms_server_esb-5.x.x-76139.x86_64.rpm
vmsvp_jbossas-5.x.x-BuildNo
vmsvp_jbossesb-5.x.x-BuildNo
vms_server_esb = 5.x.x-BuildNo
vms_server_esb(x86-64) = 5.x.x-BuildNo
```

Appendix B

ADMINISTERING MEDIA SUITE INSTALLS

The following appendix explains various tasks that you might need to occasionally perform on your Media Suite deployment. Keep in mind that this is different from administering Media Suite from its user interface on a operational basis.

Managing Install Agents

The Installation Agents are programs that are intended to run continuously in the background on all nodes, except for the Installer Manager and any RemoteFS nodes. The following section explains how to manage the agents.

Starting Agents

To start an install agent, run the following command as the `ciscovms` user:

```
execute sh /opt/cisco/vms/var/agent/agent.sh start -fc -nc -au
```

The arguments are:

- `start`
  Starts the agent and registers the node with the database.
- `-fc`
  Perform offline local changes if JBoss is stopped.
- `-nc`
  Perform online local changes when JBoss is running.
- `-au`
  Autostart JBoss with the agent. Typically, JBoss will take a few minutes to start. If you intend on stopping/starting JBoss manually after the agent has made changes, stop the agent and re-launch it without the `-au` argument.

Note The `-nc` argument depends on `-fc`, and the `-au` argument depends on `-nc`.

Starting Agents for the First Time

Agents generate a random 3-digit node ID when they are first launched. There is a (small) possibility that a node with the same ID already exists in the system, and that the new node will fail to register.

If an agent fails to register:

1 Stop the Agent on the new node.

2 Uninstall the packages on the node.

3 Remove `/opt/cisco/vms`,

4 Reinstall the packages.

5 Relaunch the Agent.

Stopping Agents

To stop an install agent, run the following command as the `ciscovms` user:

```
sh /opt/cisco/vms/var/agent/agent.sh stop
```

Querying Agent Status

To query an agent on its running status, execute the following command as `ciscovms` user:

```
sh /opt/cisco/vms/var/agent/agent.sh status
```

Managing Nodes

The following section provides various procedures for managing nodes within a deployment.

Adding Nodes to a VMS Cluster

To add a node to a deployment:

1 Install any required RPM packages on the node.

2 Copy the `database.oci` file into `/opt/cisco/vms/state` after you have installed the `vms_base` package. The `database.oci` file will automatically be used once the agent is started.

3 Launch the Agent for the node via the Linux command line:

```
execute sh /opt/cisco/vms/var/agent/agent.sh start -fc -nc -au
```

4 Navigate to **Status > Nodes** to confirm that the node has joined the cluster.

Deleting Nodes from a VMS Cluster

When deleting a node from a Media Suite cluster, it is important to maintain service continuity, Make sure that you have redirected load balancer traffic prior to deleting the node. Also, ensure that you have sufficient redundancy to continue operations for which a deleted node was responsible. Failure to maintain service continuity will lead to errors.

The process of deleting a node from a cluster involves two separate processes:
1) Shutting down the node.
2) Deleting the node.

Note The shutdown command cannot be reverted. Once the button has been pressed (and JBoss/Agent are stopped), the node must be removed from the system. The node can be added again to the system at a later time.

To delete a node:

1 Within Installer Manager, navigate to **Status > Nodes**.

2 Click an underlined Node name to see its details.

3 Click **Shutdown**.

4 Wait approximately 90 seconds.

5 Click **Delete Node**. (This button only appears once the node is inactive.) The Delete Node button removes this node from the Media Suite database so that it is forgotten by the system.

Note If an SSO primary node is deleted (and if multiple SSO nodes are running), a new primary node will automatically be selected once the old one is deleted. Otherwise, if no other SSO nodes are running, the SSO Primary will be unassigned and the system will become inoperable until a new SSO node is brought up.

Making an SSO Node Primary

This section details the procedure for changing the SSO Primary node used by Media Suite. This process would be performed if an existing SSO primary node fails or needs to be taken down for any other reason.

To make a node the SSO primary node:

1 Within Installer Manager, navigate to **Configure > Module-Specific Settings**.

2 Within the OpenAM Single-sign-on section, select an available SSO node from the Primary Node dropdown.

Note Ensure that any potential SSO primary node has been configured and administered by the Agent prior to selection. This state can be verified by logging into the SSO Management Console at `{hostname}:8080/opensso` as the `amadmin` user.

Disabling Swagger REST API Documentation

Two forms of REST APIs are available for a Media Suite instance: online (available via a URL) and offline (within the build). By default, these APIs are available to developers, but an option exists to disable access if necessary.

The URL for accessing the APIs is as follows:

`{server}:8080/opencase/ContentManager/api-docs/`

Other modules with APIs include:

Common API -

Content Manager -

Content Processor -

Workflow Service -

Switching to a Search Standby Node

Whenever a Search Master instance fails (for whatever reason) it is necessary to switch Media Suite over to use an existing standby node.

To switch from a failed Search Master to a Search Standby node:

1 Shut down the existing Search Master node either via the Installer Manager interface or the command line.

2 Redirect load balancer traffic to the Search Standby node.

3 Navigate to **Installer Manager > Status > Nodes**.

4 Click the node with the Standby Master to go to the node details page.

5 Click **Switch this standby master into an active master**.
The standby will become the active Search Master node but will not have a standby after this process has been performed.

To revert to the Search Master (with a Standby node):

1 Correct any problems with the Search Master.

2 Restart the Search Master node.

3 Redirect load balancer traffic to the Search Master.

4 Reinstall and relaunch the Standby node. You will now have a setup with a Search Master and Standby.

Propagating Security Settings

If your organization has a requirement to rotate passwords at a specified interval, the following section provides details on that procedure.

To propagate passwords:

1 Navigate to **Configure > Passwords and Security**.

2 Click **Update** for any existing password that you need to change.

3 Type and confirm the new password. The password must be 8 characters or longer.

4 Click **Save**.

5 If your organization has a requirement to rotate passwords at a specified interval, click **Propagate**. That button will push out any updated passwords to the database and to SSO nodes.

Considerations when propagating passwords:

1 Momentary outages in Web service calls should be expected when changing passwords on a running instance.

2 Ensure that external and third-party systems calling Media Suite (using `wsuser` credentials) are updated to reflect any new credentials.

Note JMX passwords are set per node and cannot be propagated.

Appendix C

CONFIGURING APACHE FOR VMS

Media Suite 5.0+ releases have been pre-integrated with mod_cluster 1.2.0 to provide a simple way of setting up multi-node instances for non-production environments. The following section describes how to install and configure Apache mod_cluster for use with Media Suite.

Installing mod_cluster

The specifics of configuring Apache is entirely up to the customer. From a security perspective, we do not provide or endorse any particular security approach. Configuring that functionality is the sole responsibility of the customer.

Warning Do not install a standby master node using mod_cluster. Doing so will result in unpredictable search functionality.

To install mod_cluster:

1. Download mod_cluster from: http://downloads.jboss.org/mod_cluster/1.2.0.Final/mod_cluster-1.2.0.Final-linux2-x64-ssl.tar.gz
2. Extract the archive to it's native full path /`opt/jboss/httpd`... (the similarity of the jboss name with the Media Suite application server is an unfortunate coincidence).
3. Edit /opt/jboss/httpd/httpd/conf/httpd.conf with the following modifications:
   - change `Listen 80` to `Listen <your.lb.fqdn.example.com>:80`
   - comment out `LoadModule advertise_module` (i.e. `# LoadModule advertise_module`)
   - uncomment `#ServerName www.example.com:80`
   - set `ServerName` to `<your.lb.fqdn.example.com>:80`
   - At the bottom of the file, replace the `IfModule manager_module` section with the following, and update the FQDN appropriately:

```
<IfModule manager_module>
  Listen <you.lb.fqdn.example.com>:6666
  ManagerBalancerName mycluster
  <VirtualHost <your.lb.fqdn.example.com>:6666>
      <Location />
        Order deny,allow
        # Deny from all
        Allow from all
      </Location>

      KeepAliveTimeout 300
      MaxKeepAliveRequests 0
```

```
        # ServerAdvertise on [http://@IP@:6666]
        # AdvertiseFrequency 5
        # AdvertiseSecurityKey secret
        # AdvertiseGroup      (ADVIP):23364
        EnableMCPMReceive
        AllowDisplay On

        <Location /mod_cluster_manager>
          SetHandler mod_cluster-manager
          Order deny,allow
          # Deny from all
          Allow from all
        </Location>
      </VirtualHost>
    </IfModule>
```

Completing the Apache Configuration

The following procedures should be followed when starting Apache:

- Ensure that no other process is bound to ports 80 and 443 on the same interface
- As root start Apache with:
  `/opt/jboss/httpd/sbin/apachectl start`
- To verify that mod_cluster is running, access:
  `http://<your.lb.fqdn.example.com>:6666/mod_cluster_manager`

  If mod_cluster is running, you will see a page titled **mod_cluster/1.2.0.Final**

- Select the vms_modcluster package in the Installer Manager at **Configure > Select Packages**.

- On the **Configure > Module-Settings** in the JBoss Mod-Cluster section. enter `<your.lb.fqdn.example.com>:6666` in the mod_cluster host:port field. `<your.lb.fqdn.example.com>` is the load-balancer's address)

- Deploy the `vms_modcluster` RPM on each VMS node.

Appendix D

INSTALLER AUTOMATION API

Media Suite includes an installer automation service that is presented as a SOAP API. The following appendix provides details related to that API.

Installer API WSDL

The WSDL for the automation API can be accessed at the following address:
http://<host>:<port>/InstallerManager/webservices/installerautomation-service?wsdl

Automation API Calls

Executing the calls listed below is broadly equivalent to installing through the Installer Manager user interface. The following section presents available API calls by logical groupings:

Package Selection

`getPackagesAvailableForSelection`

- Gets a list of package names and versions available for selection. All nodes must be inactive for this call to proceed.
- Throws `InstallerAutomationWebServiceFault`
  - if any of the nodes are active
  - if the lookup operation encounters an error

`selectPackagesByName`

- Selects packages with names in the specified set. If this is a 5.x -> 5.y upgrade, the specified package set must include all packages already selected in the 5.x install. The selected packages can be a superset of existing packages if new packages are being added to the system. All nodes must be inactive for this call to proceed.
- Throws `InstallerAutomationWebServiceFault`
  - if there are active nodes
  - if any of the specified packages are not available for selection
  - in case of dependency failures
  - if the selection process encounters an error

Endpoints

`listEndpoints`

- Lists all endpoints for all modules in the system.
- Throws `InstallerAutomationWebServiceFault` in case of error.

`getSSODomain`

- Gets the SSO cookie domain.
- Throws `InstallerAutomationWebServiceFault` in case of error.

`updateEndpoints`

- Updates endpoint configuration, setting protocols, hosts, and the SSO cookie domain. Only protocol and hostname values are user-settable; ports, context paths, and traffic types are fixed. All web user interface endpoints must reside under the specified SSO cookies domain. The specified endpoints collection must include all endpoints in the system. Updating a subset at a time is not allowed.
- Throws `InstallerAutomationWebServiceFault`

  - if all endpoints are not included in the specified collection
  - if an invalid endpoint is specified (with no pk, for example)
  - in case of validation failures. Rules are:
  >> SSO cookie domain must start with a dot
  >> All Web user interface endpoints must reside under the specified SSO cookies domain
  >> Hostnames must pass basic validation

Note Results of strict hostname validation that show warning messages in the UI are ignored.

Database Inventory

`listDatabases`

- Lists all databases in the system.
- Throws `InstallerAutomationWebServiceFault` in case of error.

`createDatabase (removed in Release 5.5)`

Datasource Mapping

`listDatasources (removed in Release 5.5)`

`mapDatasourceToDatabase (removed in Release 5.5)`

Passwords and Security

`setAdminPassword`

- Sets the VMS admin password (8 characters minimum). This password is applied to SSO at configuration/administration time. Changing it here without propagating to SSO (via the InstallerManager user interface or the SSO management console) will have no effect on a running system.
- Throws `InstallerAutomationWebServiceFault`

  - if the password does not meet the minimum length requirement
  - if the save operation encounters an error

`setJMXPassword`

- Sets the JBoss JMX console and connector password (8 characters minimum). This password is applied to JBoss at Agent runtime. Changing this password via this call without later reinstalling the RPMs will have no effect on the running system, but will break the InstallerManager

features that use the JMX connector. Those features are JGroups membership check and instantaneous log level change.

- Throws `InstallerAutomationWebServiceFault`

  - if the password does not meet the minimum length requirement
  - if the save operation encounters an error

`setSSLClientTruststoreAndPassword`

- Uploads a client truststore and sets its password. A client truststore is required if any VMS endpoints are configured to use HTTPS. The truststore and its password are applied to JBoss at Agent runtime. Changing them via this call without later reinstalling the RPMs will have no effect on the running system. The client truststore file must contain all necessary certificates to enable SSL communication over applicable endpoints.
- Throws `InstallerAutomationWebServiceFault`

  - if none of the endpoints use HTTPS
  - if the save operation encounters an error

`setSSODirectoryManagerPassword`

- Sets the SSO directory manager password (8 characters minimum). This password is applied to SSO at configuration/administration time.

Warning Changing this password via this call without propagating to SSO (via the InstallerManager user interface or the SSO management console) will cause VMS to become inoperable.

- Throws `InstallerAutomationWebServiceFault`

  - if the password does not meet the minimum 8 character length requirement
  - if the save operation encounters an error

`setWsuserPassword`

- Sets the VMS `wsuser` password (8 characters minimum). This password is applied to SSO at configuration/administration time, and to various VMS Web services at database update time.

Warning Changing this password via this call without propagating to SSO (via the Installer Manager UI or the SSO management console) and to VMS Web services (via the Installer Manager UI) will cause VMS to become inoperable.

- Throws `InstallerAutomationWebServiceFault`

  - if the password does not meet minimum length requirement
  - if the save operation encounters an error

Module-Specific Settings

`getDefaultAdminLanguage`

- Gets the default admin language.
- Throws `InstallerAutomationWebServiceFault` in case of error.

`getDefaultSystemLanguage`

- Gets the default system language.
- Throws `InstallerAutomationWebServiceFault` in case of error.

`getJBossLogLevel`

- Gets the JBoss log level for VMS and related packages.
- Throws `InstallerAutomationWebServiceFault` in case of error.

`setJBossLogLevel`

- Sets the JBoss log level for VMS and related packages. This value is applied to JBoss at Agent runtime. Changing it here without reinstalling the RPMs has no effect on the running system. Changing log levels at runtime (for troubleshooting purposes) can be done via the Installer Manager user interface. Supported values are `DEBUG, INFO, WARN, or ERROR`.
- Throws `InstallerAutomationWebServiceFault`
  - if the specified value is invalid
  - if the save operation encounters an error

`getSelectedLanguages`

- Gets the language set selected for VMS.
- Throws `InstallerAutomationWebServiceFault` in case of error.

`listSupportedLanguages`

- Lists all languages supported by VMS. Entries in this list are of the format '`aa_BB-cc`', where '`aa`' is the language code, '`BB`' is the country code, and '`cc`' is the language group code. For example, '`ko_KR-as`' represents Korean/Korea/Asia.
- Throws `InstallerAutomationWebServiceFault` in case of error.

`selectLanguages`

- Selects the language set used by VMS, the default administrative language, and the system language. All language parameters passed to this call must belong to the supported language set, returned by `listSupportedLanguages()`.

Warning If this is an upgrade and there is data in the system that references locales outside of the selected set, that data will be deleted during the database update. A warning is generated when starting the database update and the user is given the option to expand the locale selection to prevent data loss.

Note Select only those locales needed for your service. Selecting extra locales will degrade system performance.

- Throws `InstallerAutomationWebServiceFault`
  - if any of the specified locales are outside the supported set
  - if the save operation encounters an error

`getModClusterProxy`

- Gets the mod_cluster proxy host:port.
- Throws `InstallerAutomationWebServiceFault` in case of error.

`setModClusterProxy`

- Sets mod_cluster proxy host:port.
- Throws `InstallerAutomationWebServiceFault` in case of error.

Database Update

`isDatabaseUpdateReady`

- Returns whether the global database updater is ready to update.
- Throws `InstallerAutomationWebServiceFault` if the lookup operation fails.

`isDatabaseUpdateInProgress`

- Returns whether the global database updater state is in progress.
- Throws `InstallerAutomationWebServiceFault` if the lookup operation fails.

`isDatabaseUpdateComplete`

- Returns whether the global database updater state is complete.
- Throws `InstallerAutomationWebServiceFault` if the lookup operation fails.

`isDatabaseUpdateInError`

- Returns whether the database update is in an error state.
- Throws `InstallerAutomationWebServiceFault` if the lookup operation fails.

`performDatabaseUpdate`

- Starts the database update process. All nodes must be inactive and the global database updater state must be ready. This call performs validation on items required for database updater operations and then starts the update process asynchronously. Callers need to poll the state using the provided '`isDatabaseUpdate...()`' methods.
- Throws `InstallerAutomationWebServiceFault`

  - if any nodes are active

  - if the global database update state is not ready

  - if any of the individual database updaters are not in the ready or complete states

  - if EntitlementManager is included and a random entry from `OEM_LOGIN` has a password that cannot be decrypted

  - if this is a 4.x -> 5.x upgrade and the 4.x version is below the minimum required for a direct upgrade (4.1.2.2 at the time of this writing)

  - if any of the system passwords have not been set (SSO directory manager, admin, wsuser, JMX admin, or client truststore, if used)

  - if any of the endpoints use HTTPS and no client truststore has been provided

  - if this is an upgrade and any of the included modules have data referencing locales outside of the selected locale set

Node Status

`listJBossNodes`

- Lists all registered nodes.
- Throws InstallerAutomationWebServiceFault in case of error.

`removeJBossNode (VMS 5.1.1 and higher)`

- Removes the specified node from the system. In order to be removed, the node must be inactive, meaning the Agent must be out of communication for more than the standard node TTL (90 seconds by default). If the node being removed is the SSO primary, the system will select an alternate SSO primary from any of the other active running SSO nodes.

Warning If no alternate SSO primary can be selected, the system will switch the SSO state to "Packages Installed", and VMS will become inoperable until an SSO primary can be brought online.

- Throws `InstallerAutomationWebServiceFault`
  - if the node cannot be found by ID.
  - if the node is still active.
  - if the removal operation encountered an error.

`shutdownJBossNode (VMS 5.5 and higher)`

- Performs a graceful shutdown of JBoss and the Agent on the specified node. Shutdown mode blocks the execution of scheduled tasks and will cause all SOAP calls to throw 500-series errors. We strongly recommend that you remove the node from the load-balancer pool immediately prior to making this call. This call should be used to gracefully stop and remove a Media Suite node, especially at runtime.

Warning This shutdown command cannot be reverted. Once this call has been made (and JBoss/Agent are stopped), the node must be removed from the system. The node can be added again to the system at a later time.

- Throws `InstallerAutomationWebServiceFault` in case of error.

`partitionNodesForUpgrade (VMS 5.5 and higher)`

- Partition nodes for an A-side/B-side upgrade. To enable the upgrade mode, all nodes in the deployment must be partitioned into 2 groups: upgrade FIRST and upgrade LAST. The nodes in the FIRST partition are upgraded first while the nodes in the LAST partition remain running. Once the FIRST partition upgrade is complete, the nodes in the LAST partition are then upgraded. There must be sufficient redundancy to enable the upgrade mode, meaning that each module must have nodes in both partitions.
- Throws `InstallerAutomationWebServiceFault`:
  - If the partitioning is invalid, meaning insufficient nodes for redundancy or improper split of modules across partitions (there must be at least one node from each module in each partition).
  - In case of another error.

`isUpgradeModeEnabled (VMS 5.5 and higher)`

- Returns true if the upgrade mode is enabled.
- Throws `InstallerAutomationWebServiceFault` in case of error.

`enableUpgradeMode (VMS 5.5 and higher)`

- Enables the upgrade mode. In order to enable, all nodes in the system must be partitioned with sufficient redundancy. Once the flag is set, all running nodes (of the same version) will stop reloading the installer configuration. This effectively caches the configuration in memory until shutdown.
- Throws `InstallerAutomationWebServiceFault` in case of error.

`disableUpgradeMode (VMS 5.5 and higher)`

- Disables the upgrade mode. If running nodes (of the same version) have already loaded the upgrade mode flag, their installer configuration will be cached until restart, so this change will not have an effect on them.
- Throws `InstallerAutomationWebServiceFault` in case of error.

Package Status

`listInstalledPackages`

- Lists all packages selected by the user whether they are installed on node(s) or not.
- Throws `InstallerAutomationWebServiceFault` in case of error.

`listInstalledModules (5.1.1 and higher)`

- Lists all modules contained in all packages selected by the user, whether they are installed on node(s) or not.
- Throws `InstallerAutomationWebServiceFault` in case of error.

Custom File Deployment

`listCustomFiles (5.1.1 and higher)`

- Lists all custom files registered in the system.
- Throws `InstallerAutomationWebServiceFault` in case of error.

`createCustomFile (5.1.1 and higher)`

- Creates a custom file. The custom file object must be populated to include a name, file name, valid target module, and optionally a target path. Files can be activated or deactivated as needed by setting the active flag in the Installer Manager user interface.
- Throws `InstallerAutomationWebServiceFault`

  - if the specified custom file already has a primary key

  - if a name or file name are missing

  - if a target module is missing, invalid, or refers to the Agent or the Installer itself

  - if a custom file with the same name already exists

  - if the custom file is configured to be placed inside the target module's deployment home, but the target module does not have a deployment home

  - if another custom file that resolves to the same target path (file or directory) already exists in the system

  - if the save operation encounters an error

`updateCustomFile (5.1.1 and higher)`

- Updates custom file content. Agents running on nodes hosting this custom file's target module will redeploy the custom file.
- Throws `InstallerAutomationWebServiceFault`

  - if the primary key is invalid

  - if the save operation encounters an error

Downloading database.oci

There are two ways to obtain the `database.oci` file. You may:

1. Download the `database.oci` file from `http://<host>:<port>/InstallerManager/database.oci` and place it in `/opt/cisco/vms/state` after installing the required packages.

Note The URL used to access the database.oci file is authenticated. If you try to access it while logged into the Installer Manager, the download will occur, otherwise, you will receive an `HTTP 401` error.

---

2. Have the Agent download the file (assuming network access is granted) by exporting the environment variable `VMS_DATABASE_OCI_URL` prior to launching the Agent. The Agent will immediately ask for the Installer Manager administrator password. This is the recommended approach for automated deployments. The required password can be piped into the Agent launch script.

Figure 37 Sample Shell Script for Agent Launch and Automating database.oci Download

```
export VMS_DATABASE_OCI_URL=http://<host>:<port>/InstallerManager/database.oci
echo {password_for_installer_manager} > tempfile
sh /opt/cisco/vms/var/agent/agent.sh -fc -nc -au < tempfile
rm tempfile
```

Appendix E

SYSTEM CONFIGURATION

Configuration settings for Media Suite are set automatically during installation, but there may be an occasional need to manually change settings afterward. The System Configuration page contains configuration information for each Media Suite module, and provides a user interface for manually changing related values. This appendix describes the contents of the System Configuration page, and can serve as a guide when updating configuration settings.

Note Always take care when modifying configuration settings. Incorrect entries can affect the proper operation of application modules or of Media Suite itself.

The system configuration page is organized in a tree node structure. At the highest level, three nodes are available: general, modules, and services. Each of these nodes has subordinate nodes, or subnodes, representing configuration settings to which values are assigned. The values assigned to each subnode are either the default value (if only one) or a list of all possible values for that setting.

The following tables describe each node and its subnodes in some detail. Subnodes are identified in the tables by the node name followed by a > symbol, with the subnode name appearing just underneath. The Descriptions for each subnode provide details on the configuration setting and its possible values.

General Nodes

General node settings define locale and time zone information and establish connection limits.

Table 23 general Node Hierarchy

Node Level	Possible Values	Description
config.ttl		The duration (in milliseconds) that configuration values will be cached before they expire and new values are fetched.
httpclient > max.host.http.connections	1 to n	Sets the maximum number of connections the Apache HTTP client pool can have when connecting to a single host. Currently, this is only used for the proxy service for serving WMRM license requests in EntitlementManager. Since this is only used in the EM module, this number can be the same value as the connection pool size.
httpclient > max.total.http.connections	1 to n	Sets the maximum number of open connections that the Apache HTTP client pool can have. Currently, this setting is only used by the VMS proxy service for serving WMRM license requests in EntitlementManager.

Table 23 general Node Hierarchy

Node Level	Possible Values	Description
last.modified.date		Displays the date when any configuration was last updated.
server.timezone	As with the `client supported.timezones` option, these nodes specify the time zone that corresponds to the relevant region/city.	As with the client `supported.timezones` option, these nodes list a general region and a city within that region.
supported.timezones (For client devices.)	Possible values vary by deployment. Specify the time zone that corresponds to the region/city at right. For example: time.zone.central.standard time.zone.western.european time.zone.gulf.standard	Displays a list of general regions and a city within that region. For example: America/Chicago Europe/Lisbon Asia/Dubai
webservice.client > javax.xml.ws.client.connectionTimeout		Connection timeout for webservices in milliseconds.
webservice.client > javax.xml.ws.client.receiveTimeout		Receive timeout for webservices in milliseconds.

Module Nodes

Module node settings define parameters that are specific to all modules installed within Media Suite and their Web services.

Notes on All Web Services

The combination of a module's public server URL and its wsdlcontext create the full URL for the WSDL location of the Web service. This value is internally referenced by Media Suite when calling Web services, and never needs to be modified. Additional parameters, such as `namespace`, `servicename`, `defaultuser`, and `defaultpassword` are also required in order to configure Web services. Those values are internally referenced by Media Suite and should also never be modified.

Notes on Search Manager returnFields

When specifying an override, you must list all the fields that you want returned for each document. There are two reasons for setting these return values:

1. To improve performance by limiting the fields that are returned to consumers.
2. To make custom fields visible to consumers.

This value should be set to a comma delimited list of field names that are contained within the EPG, VoD, or EPG and VoD indices. Refer to the Search Manager API for a list of possible fields.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
cm > bind >	frequency.process.previous.request	The millisecond interval between when the binding service checks for completeness of the bundles that it has bound.
	remove.expired.bind.request	The number of days before an unfulfilled bind request is removed.
	unprocessed.batch.size	The number of bind requests that the bind job will process in a single pass.
cm > bundle >	bundle.delete.batch.size	The number of bundles with expired licensing windows that will be fetched for deletion as a group. These deletions will repeat for the duration of the maxruntime interval.
	delete.inactive.bundle.validation. warnings	Specifies the number of days that validation warnings will remain on the system. Those warnings can be seen on the bundle details page by clicking **Warnings**.
	physicalasset.delete.batch.size	Sets the batch size for deletion of expired assets.
	record.deleted.bundle.xml	A boolean value (true/false) that specifies whether to archive the structure and metadata of deleted bundles.
	remove.deleted.bundles	Specifies the number of days that archived deleted bundle data will remain on the system.
cm > caching >	ttl.bundle	Bundle REST cache time-to-live (in seconds). Applied only at server startup.
	ttl.feed	Feed REST cache time-to-live (in seconds). Applied only at server startup.
	ttl.physicalasset.releaseurl	Physical asset by release URL object cache time-to-live (in seconds). Applied only at server startup.
	ttl.product	Product REST cache time-to-live (in seconds). Applied only at server startup.
cm > custom.attributes >	max.custom.attributes.allowed	Specifies the maximum number of custom attributes that are allowed instance wide. **Warning** Setting a number that is higher than the default is not supported and can negatively impact system performance.
	max.custom.attributes.per.component .type	Specifies the maximum number of custom attributes that are allowed per component type. **Warning** Setting a number that is higher than the default is not supported and can negatively impact system performance.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
cm > index > sm >	bundles.batchSize	Specifies how many BundleGenerated records fetch in single request from DB during VOD indexing. Value must be an Integer > 1.
	calc.unavailable.days.ahead	Controls the calculation of the number of unavailable days between multiple product offers for the same bundle. The period defines a safety net for situations where indexing is not possible for those days. Value must be an Integer > 1. **Warning** Setting this value too high will result in a large VoD index, which will affect indexing and replication times. Since the job runs daily, a default value of 7 days or less is recommended.
	feedType.name	Specifies which of the available feed types you would like to use to generate a VoD schema. This Feed Type will be used by the publish job that runs in the background to prepare the index for the Search Manager.
	recalc.unavailable.hours.buffer	Configures number of days that is used as a buffer to recalculate unavailable fields in a MAINTAIN OFFER WINDOW STEP. Value must be an Integer > 1.
	unavailable.field.granularity	Configures granularity of unavailable fields. Value must be either 'hourly' or 'daily'.
cm > notifications >	delete.expired.assets.job.maxruntime	Specifies the maximum number of milliseconds that the expired asset deletion job can run.
	delete.expired.bundles.job.maxruntime	Specifies the maximum number of milliseconds that recurring instances of the "delete expired bundles" quartz job can run for.
	entity.change.queue.job.cpu.load	Specifies the CPU load factor that is used to determine the number of threads that the entity change queue “executor” thread pool will attempt to run concurrently. Values must be in the range of 0.00 to 1.00. A value of 0.25 attempts to allocate 25% of available processors. The entity change queue thread pool always uses at least 1 processor. **Note** For changes to take effect, all Content Manager nodes must be restarted.
	entity.change.queue.job.maxruntime	Specifies the maximum number of minutes that an entity queue job will run.
	entity.change.queue.job.maxThread PoolSubmissions	Specifies the maximum number of entity change queue items that can be processed concurrently.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
	entity.change.queue.lock.time.minutes	Specifies the maximum number of minutes that a queue row will be locked while it is being processed.
	notification.batch.size	The number of notifications that will be processed in a single notification job cycle.
	notification.job.maxruntime	Specifies the maximum number of minutes that a notification job will run.
	update.associated.components.batch.size	Specifies the maximum number of components associated with core entities (such as common entities, common attributes, origin mappings, and URL signings) that can be updated at one time. These batch updates will continue until the '`entity.change.queue.lock.time.minutes`' is reached. Therefore, to avoid problems, batch process time should never exceed that lock time.
cm > origin.mapping >	cm.origin.mapping.url.regex	The regular expression that sets the pattern for URL pattern for origin base URL and public URL.
cm > product	record.deleted.product.xml	Indicates whether product XML should be archived when deleting a product. Acceptable values: true/ false.
	remove.deleted.products	The number of days to keep deleted products before they are permanently deleted. Default value is 365 days. Acceptable value: any integer greater than 0.
cm > pt > wizard > images	allowed.types	A comma-delimited list that specifies the image formats that are allowed for the Producer bundle wizard.
	max.file.size	Specifies the maximum size (in bytes) for images that will be imported into the Producer bundle wizard.
cm > releaseUrl >	includePhysicalAsset	A boolean field that enables the visibility of physical asset videos within bundle XML. Set to True to enable and False to disable. **Note** Changes to this setting will result in feed regeneration. Depending upon the size of the catalog, changes may take a fairly long time to take effect.
cm > search >	compass.queue.batch.size	Specifies the number of records that will be processed at one time during a compass queue job.
	maxcomponents.hits	For bundle searches, the maximum number of components that can be found by the search term before an error is thrown. **Warning** Setting this number too high can result in performance issues on the user interface.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
cm > vcm >	app.name	Identifier configured in VCM to uniquely identify response messages intended for the current Media Suite instance.
	asset.source.id	Identifier configured in VCM to uniquely identify messages sent by the current Media Suite instance.
	deleted.xslt.template	Specifies an XSLT transformer to transform Media Suite deleted bundle XML to NDS ADI compliant XML.
	http.connection.manager.timeout	Connection timeout (in milliseconds) for VCM HTTP connections
	http.socket.timeout	Read timeout (in milliseconds) for VCM HTTP connections.
	prg.name	Name of the VCM service that Media Suite is expecting responses from.
	url	VCM endpoint URL to which Media Suite submits bundle updates.
	xslt.template	Specifies an XSLT transformer to transform the Media Suite bundle XML to NDS ADI compliant XML.
cm > webservices >	componentservice.wsdlcontext	The Web service endpoint where the creation and management of components and bundles are performed along with related tasks. For more details, see “Notes on All Web Services” on page 106.
	componentservice.namespace componentservice.servicename componentservice.defaultuser componentservice.defaultpassword	Endpoint connection parameters that are required by the internal componentservice. For more details, see “Notes on All Web Services” on page 106.
	entitlementservice.wsdlcontext	The Web service endpoint that handles entitlement tasks, such as those related to the licensing and provisioning of assets. For more details, see “Notes on All Web Services” on page 106.
	entitlementservice.namespace entitlementservice.servicename entitlementservice.defaultuser entitlementservice.defaultpassword	Endpoint connection parameters that are required by the internal entitlementservice. For more details, see “Notes on All Web Services” on page 106.
	kmsservice.user kmsservice.password	Key Management Server Client Identity credentials for TSTV schedule generation.
	searchservice.wsdlcontext	The Web service endpoint where search customizations are performed for Content Manager. For more details, see “Notes on All Web Services” on page 106.
	searchservice.namespace searchservice.servicename searchservice.defaultuser searchservice.defaultpassword	Endpoint connection parameters that are required by the internal searchservice. For more details, see “Notes on All Web Services” on page 106.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
	urlsigningservice.wsdlcontext	The Web service endpoint that handles tasks related to URL Signing. For more details, see “Notes on All Web Services” on page 106.
	urlsigningservice.namespace urlsigningservice.servicename urlsigningservice.defaultuser urlsigningservice.defaultpassword	Endpoint connection parameters that are required by the internal urlsigningservice. For more details, see “Notes on All Web Services” on page 106.
cp > armada > armada.kms.settings >	certificate	Optional certificate used for Key Management Server Client Identity certificate authentication for the HLS/AES key service.
	username password	Key Management Server Client Identity credentials for the Armada key service.
	rsa.private.key	The private key used to decrypt key material from the Key Management Server.
cp > repository >	content.file.retention	The number of days to retain data for `OCP_CONTENT_FILE` and `OCP_CONTENT_FILE_METADATA`.
	ftp.mode.passive.enable	Sets whether FTP connections will use a passive or active mode for connecting to FTP locations. If the value is set to “true”, passive mode will be used; if the value is set to “false”, active mode will be used.
	ftp.timeout	The number of milliseconds before an FTP connection will timeout.
cp > repository > hls.collator >	manifest.wf.instance.interval	An interval (in milliseconds) where the HLS collator would retry finding all items contained in the manifest.
	manifest.wf.instance.retry.count	The number of retries that are made when trying to find items referenced by an HLS manifest. Beyond this number, the collation attempt will be considered a failure.
cp > repository >	hotfolder.exclusion.list	A regular expression that specifies files to exclude from hot folder scans. By default, files with a `.db` extension will be excluded since those file are generally created by Windows and shouldn't be picked up for processing.
	hotfolder.scan.batch.enabled	A value of “true” indicates that batch processing will be turned on. If the value is set to “false”, batch processing will not be turned on and any new files detected by the hot folder scan will be queued for future processing in an ad hoc manner.
	hotfolder.scan.batch.size	The number of files the hot folder scan job will pick up and queue for processing.
	path.processor.plugin.classes	A comma-separated list of path processor plug-in classes referring to by their Seam names.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
	retry.event.batch.enabled	The number of RetryEvent entries that the RetryEventManger will fetch from the `OCP_RETRY_EVENT` table every minute.
	retry.event.batch.size	Enables or disables the retry.event.batch.size setting. If set to false, then all entries in the `OCP_RETRY_EVENT` table will be fetched.
cp > repository > retry.on.io.failure >	recoverable.exceptions	A comma-delimited list of exceptions that may be considered recoverable for repositories in this deployment. When one of these exceptions is encountered, a retry will be attempted (see `retry.count`).
	retry.count	The number of retries that will be attempted before considering a repository connection a failure. Acceptible value: any integer greater than 0.
cp > repository >	sftp.timeout	The number of milliseconds before an SFTP connection will timeout.
	supported.filesystem.types	A comma delimited list of all file types that are supported by Media Suite repositories.
cp > webservices >	contentfileservice.wsdlcontext	The Web service endpoint where content file creation and management are performed along with related tasks. For more details, see “Notes on All Web Services” on page 106.
	contentfileservice.namespace contentfileservice.servicename contentfileservice.defaultuser contentfileservice.defaultpassword	Endpoint connection parameters that are required by the internal contentfileservice. For more details, see “Notes on All Web Services” on page 106.
	eventservice.wsdlcontext	The Web service endpoint where event creation and management are performed along with related tasks. For more details, see “Notes on All Web Services” on page 106.
	eventservice.namespace eventservice.servicename eventservice.defaultuser eventservice.defaultpassword	Endpoint connection parameters that are required by the internal eventservice. For more details, see “Notes on All Web Services” on page 106.
	hotfolderservice.wsdlcontext	The Web service endpoint where hot folder creation and management are performed along with related tasks. For more details, see “Notes on All Web Services” on page 106.
	hotfolderservice.namespace hotfolderservice.servicename hotfolderservice.defaultuser hotfolderservice.defaultpassword	Endpoint connection parameters that are required by the internal hotfolderservice. For more details, see “Notes on All Web Services” on page 106.
	processorservice.wsdlcontext	The Web service endpoint that supports ESB functionality for all services. For more details, see “Notes on All Web Services” on page 106.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
	processorservice.namespace processorservice.servicename processorservice.defaultuser processorservice.defaultpassword	Endpoint connection parameters that are required by the internal processorservice. For more details, see “Notes on All Web Services” on page 106.
	repositoryservice.wsdlcontext	The Web service endpoint where repository creation and management are performed along with related tasks. For more details, see “Notes on All Web Services” on page 106.
	repositoryservice.namespace repositoryservice.servicename repositoryservice.defaultuser repositoryservice.defaultpassword	Endpoint connection parameters that are required by the internal repositoryservice. For more details, see “Notes on All Web Services” on page 106.
em > account.validation.plugin	classname	The fully qualified class name of the default account validation plug-in that will be used.
em > accountmanager. defaultplugin		Parent node for settings for the default account manager plug-in class.
	classname	The fully qualified class name of the default account manager plug-in that will be used.
	config	Any configuration data required by the plug-in class.
em >	base.release.url	An optional field that can store the base origin location for this deployment’s PhysicalAssets.
em > caching >	affiliate.ttl	The time (in seconds) the affiliate cache will live before it expires and is refreshed on any future access.
	component.ttl	The time (in seconds) the component cache will live before it expires and refreshes on the next load.
	drmtype.ttl	The time (in seconds) the DRM type cache will live before it expires and refreshes on the next load.
	licenserequestinfo.ttl	The time (in seconds) the license request info cache will live before it expires and refreshes on the next load.
	licensingwindow.ttl	The time (in seconds) the licensing window cache will live before it expires and refreshes on the next load.
	product.ttl	The time (in seconds) the product cache will live before it expires and refreshes on the next load.
	subproductsbybundleuuid.ttl	The time (in seconds) that the subscription products cache will live before it expires and refreshes on the next load. In this instance, subscription products are found by bundle UUID.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
	subproductsbyreleaseurl.ttl	The time (in seconds) that the subscription products cache will live before it expires and refreshes on the next load. In this instance, subscription products are found by releaseURL.
	urlsigning.ttl	The time (in seconds) the URL signing cache will live before it expires and refreshes on the next load.
em > device.validation.plugin	classname	The fully qualified class name of the default device validation plug-in that will be used.
em > drm.key.provider.plugin >		Parent node for settings for the DRM key provider plug-in class.
	classname	The fully qualified class name of the default plug-in that will be used.
	config	Any configuration data required by the plug-in class.
em > drmtxmanager.plugin >		Parent node for settings for the DRM transactions plug-in class.
	classname	The fully qualified class name of the default plug-in that will be used.
	config	Any configuration data required by the plug-in class.
em > entitlementmanager.plugin >		Parent node for settings for the entitlement manager plug-in class. This class is used to implement custom logic for pre and post creation, modification, and the deletion of entitlement data.
	classname	The fully qualified class name of the default entitlementmanager plug-in that will be used.
	config	Any configuration data required by the plug-in class.
em > entitlementmanager.plugin .product.lookup >		Parent node for settings for the entitlement manager plug-in class that looks up products.
	classname	The fully qualified class name of the default entitlementmanager product lookup plug-in that will be used.
	config	Any configuration data required by the plug-in class.
em > hlsaes > hlsaes.kms.settings >	certificate	Optional certificate used for Key Management Server Client Identity certificate authentication for the HLS/AES key service.
	username password	Key Management Server Client Identity credentials for the HLS/AES key service.
	rsa.private.key	The private key used to decrypt key material from the Key Management Server.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
em > license.terms.plugin >		Parent node for settings for the license terms plug-in class.
	classname	The fully qualified class name of the default plug-in that will be used.
	config	Any configuration data required by the plug-in class.
em > login.validation.plugin	classname	The fully qualified class name of the default login validation plug-in that will be used.
em > notifications >	batch.size	The number of notifications that will be processed in a single notification job cycle.
	max.retry.count	Specifies the maximum number of retries for an entitlement notification before the attempt errors out.
	notification.job.maxruntime	Specifies the maximum number of minutes that a notification job will run before it is considered defunct and is made available for execution by another process.
em > oauth >	timeout	The number of milliseconds that an inactive OAuth session will remain alive for.
em > playready > playready.kms.settings	certificate	Optional certificate used for Key Management Server Client Identity certificate authentication for the PlayReady key service.
	username password	Key Management Server Client Identity credentials for the PlayReady key service.
	rsa.private.key	The private key used to decrypt key material from the Key Management Server.
em > playready >	playready.license.service.endpoint.url	The domain or IP where the PlayReady license server is installed.
	playready.license.service.wsdlcontext	The endpoint where PlayReady license server Web services are located. For more details, see “Notes on All Web Services” on page 106.
em > playready.license.router. plugin >		Parent node for settings for the PlayReady license router plug-in class.
	classname	The fully qualified class name of the default manager plug-in that will be used.
	config	Any configuration data required by the plug-in class.
em > webservices >		
	accountmanagerservice.wsdlcontext	The Web service endpoint where accountmanagerservice tasks are performed for user accounts. For more details, see “Notes on All Web Services” on page 106.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
	accountmanagerservice.namespace accountmanagerservice.servicename accountmanagerservice.defaultuser accountmanagerservice.defaultpassword	Endpoint connection parameters that are required by the internal accountmanagerservice. For more details, see “Notes on All Web Services” on page 106.
	devicemanagerservice.wsdlcontext	The Web service endpoint where devicemanagerservice tasks are performed. For more details, see “Notes on All Web Services” on page 106.
	devicemanagerservice.namespace devicemanagerservice.servicename devicemanagerservice.defaultuser devicemanagerservice.defaultpassword	Endpoint connection parameters that are required by the internal devicemanagerservice. For more details, see “Notes on All Web Services” on page 106.
	notificationmanagerservice.wsdlcontext	Endpoint connection parameters that are required by the internal notificationmanagerservice. For more details, see “Notes on All Web Services” on page 106.
	notificationmanagerservice.namespace notificationmanagerservice.servicename notificationmanagerservice.defaultuser notificationmanagerservice.default password	Endpoint connection parameters that are required by the internal notificationmanagerservice. For more details, see “Notes on All Web Services” on page 106.
	rightslockerservice.wsdlcontext	The Web service endpoint where rightslockerservice tasks are performed. For more details, see “Notes on All Web Services” on page 106.
	rightslockerservice.namespace rightslockerservice.servicename rightslockerservice.defaultuser rightslockerservice.defaultpassword	Endpoint connection parameters that are required by the internal rightslockerservice. For more details, see “Notes on All Web Services” on page 106.
	urlsigningservice.wsdlcontext	The Web service endpoint where urlsigningservice tasks are performed. For more details, see “Notes on All Web Services” on page 106.
	urlsigningservice.namespace urlsigningservice.servicename urlsigningservice.defaultuser urlsigningservice.defaultpassword	Endpoint connection parameters that are required by the internal urlsigningservice. For more details, see “Notes on All Web Services” on page 106.
em > wmrm >	wmrm.license.service.endpoint.url	The domain or IP where the WMRM license service server is installed.
	wmrm.license.service.wsdlcontext	The Web service endpoint where WMRM tasks are performed. For more details, see “Notes on All Web Services” on page 106.
em > wmrm.license.router.plugin>		Parent node for settings for the WMRM license router plug-in class.
	classname	The fully qualified class name of the default plug-in that will be used.
	config	Any configuration data required by the plug-in class.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
kms > jasypt >	password.provider	Implementation of `com.cisco.vms.kms.plugin.jasypt.api.SEKAcquisitionPlugin` that is responsible for returning the SEK (storage encryption key) that will be used for Jasypt encryption.
kms > webservices >	keyset.import.size	The maximum number of Keysets that can be passed to the KMS import function in a single call.
lm > cm.generate.bundles >	batchSize	Batch size for generating bundles from stations operation. Defaults to 100.
	maxThreads	Maximum number of threads for ContentManager interaction. Defaults to 20.
lm > index		EPG indexing properties.
lm > index > caching >	fullreindex.db.batchsize	Database batch size to prepare caches for the 'Future Only', 'Everything' and 'Reverse EPG Future Only' reindexing. Defaults to 1000.
	targetedReindex.db.batchsize	Database batch size to prepare caches for the 'Changes only' reindexing. Defaults to 500.
lm > index >	connection.timeout.milliseconds	Solr server connection timeout. Defaults to 60000.
	fallback.locales	Fallback locales are used to populate missing critical metadata in the language being indexed. This prevents blank spots in the EPGgrid. The default locale is tried first, and if metadata cannot be found, fallback locales are tried in the order they are defined until metadata is found. The list of fallback metadata locales is separated with commas. The default value is empty.
	maxThreads	Maximum number of threads to index data to Solr master. Defaults to 20.
lm > index > notification >	url	A mechanism to notify external systems that new and updated EPG data is available to consumers.
lm > index > programs	db.batchsize	Database batch size for Programs indexing. Defaults to 4000.
lm > index >	read.timeout.milliseconds	Solr server read timeout. Defaults to 1800000.
lm > index > regions >	db.batchsize	Database batch size for Regions indexing. Defaults to 10000.
lm > index > schedules >	db.batchsize	Database batch size for Schedule indexing. Defaults to 40000.
	repg.queue.size	The queue size (in records) for Reverse EPG indexing.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
	repg.solr.maxThreads	Solr maximum number of threads for Reverese EPG indexing.
lm > index > stations	db.batchsize	Database batch size for Stations indexing. Defaults to 500.
	solr.batchsize	Batch size for filling station index cache from Solr master. Defaults to 100.
lm > ingest >	allow.past.schedule.ingest	Allows ingestion of schedules with start dates in the past. Defaults to false.
	batchSize	EPG ingest batch size. Defaults to 500.
	cm.bundle.rest.path	Relative path to ContentManager REST web service to get bundle XML. Defaults to '/bundle'.
	cm.bundle.xslt.code	Used to inject bundle and product information into channel lineups for each station and channel. The value of the XSLT property is sent to the ContentManager REST web service to get the bundle XML. Defaults to 'epgstation'.
	cm.product.rest.path	Relative path to ContentManager REST web service to get product XML. Defaults to '/product/{uuid}'.
	cm.role.default	Default role set to program contributors with unknown role. Defaults to 'castMember'.
lm > ingest > cm.role.mapping >		This group of setting provides a mapping between EPG based roles and roles defined for on demand assets managed in Content Manager.
	castMember	Role name to identify Cast Member. Default value is empty.
	creator	Role name to identify Program Creator. Defaults to 'Creator'.
	director	Role name to identify Program Director. Defaults to 'Director'.
	producer	Role name to identify Program Producer. Defaults to 'Producer'.
	writer	Role name to identify Program Writer. Defaults to 'Writer'.
lm > ingest >	contributor.roles	This setting limits the type of contributor roles and the maximum number that can be ingested per role for each program or schedule. Defaults are: Actor:10,Guest star:10,Writer:-1,Director:-1,Producer:-1,Executive producer:-1,Host:-1,Screenwriter:-1, Voice:10,Original music:10,Contestant:10,Guest:10,Musical guest:10,Music:10

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
	create.active.regions	Determines whether lineups are marked as active when created by feed ingestion. True means that lineups are marked as active. False means that lineups will be inactive until manually activated by an administrator.
	generate.missing.programs	Allowing creating missing program records from schedule records. Defaults to true.
	image.url.prefix	Public images URL prefix. Defaults to 'http://some.cdn.com/some/base/path/'. The URL prefix is required if EPG data contains relative image URLs. This value will be used as a prefix for relative URLs and should contain the CDN URL with a base path where EPG images can be found.
	max.history.days	Maximum number of days that historical EPG data will be retained. A value of -1 specifies that EPG data never expires. **Warning** Keeping all EPG data will negatively affect indexing time and disk usage
	max.ingeststep.idle.minutes	Used to determine if a workflow process step is hung. If a new workflow attempts to invoke an EPG process belonging to the current workflow, and the current workflow is unresponsive (for longer than the set time), the current workflow will be considered defunct and will be terminated. This will allow the new workflow to run. **Warning** Setting this value too low can result in the improper termination of valid workflows.
	max.workflow.idle.minutes	Maximum time between process step invocation for a currently running workflow. This setting is used to ensure that a defunct workflow does not lock up ingestion. If another workflow attempts to invoke EPG processes and the current workflow has been idle for more then the set time, the current workflow will be considered defunct and will be terminated. Afterward, the new workflow will be allowed to run.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
	maxThreads	Maximum number of threads for EPG ingest operations. Defaults to 20. This value should be tuned according to the number of CPUs available on the system. **Warning** Depending on the deployment, setting this value too high can result in an unresponsive user interface.
	override.schedule.with.program	Mode of copying matching data fields from programs to schedules. Possible values: ALL, MISSING, NONE. ALL copies all matching fields, MISSING copies only fields that are not present in the schedule, NONE does not copy any fields. Defaults to ALL.
lm > vcs >		LinearManager (EPG) specific settings for Videoscape Control Suite.
	programImages.batchsize	Specifies how many Program Images should be send to Videoscape Control Suite in a single request.
	programs.batchsize	Specifies how many Programs should be send to Videoscape Control Suite in a single request.
	programsSeries.batchsize	Specifies how many Program Series should be send to Videoscape Control Suite in single request.
	schedules.batchsize	Specifies how many Schedules should be send to Videoscape Control Suite in a single request.
lm > webservices	ingestservice.wsdlcontext	The Web service endpoint where ingestservice tasks are performed. For more details, see “Notes on All Web Services” on page 106.
	ingestservice.namespace ingestservice.servicename ingestservice.defaultuser ingestservice.defaultpassword	Endpoint connection parameters that are required by the internal ingestservice. For more details, see “Notes on All Web Services” on page 106.
	kmsservice.user kmsservice.password	Key Management Server Client Identity credentials for TSTV schedule generation.
	programservice.wsdlcontext	The Web service endpoint where programservice tasks are performed. For more details, see “Notes on All Web Services” on page 106.
	programservice.namespace programservice.servicename programservice.defaultuser programservice.defaultpassword	Endpoint connection parameters that are required by the internal programservice. For more details, see “Notes on All Web Services” on page 106.
	stationsourceservice.wsdlcontext	The Web service endpoint where stationsourceservice tasks are performed. For more details, see “Notes on All Web Services” on page 106.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
	stationsourceservice.namespace stationsourceservice.servicename stationsourceservice.defaultuser stationsourceservice.defaultpassword	Endpoint connection parameters that are required by the internal stationsourceservice. For more details, see “Notes on All Web Services” on page 106.
occ > caching	cache.event.cpu.load	Specifies the CPU load factor that is used to determine the number of threads that the cache event thread pool will attempt to run concurrently. Values must be in the range of 0.00 to 1.00. A value of 0.25 attempts to allocate 25% of available processors. The cache event thread pool always uses at least 1 processor. **Note** For changes to take effect, all OCC nodes must be restarted.
occ > external.services > crs >	url	The URL to connect to the Content Resolution Service.
occ > external.services > dps >	namespace	The namespace for the XML schema for the Device Profile Service.
	username password	The username/password required to connect to the Device Profile Service.
	url	The URL to connect to the Device Profile Service.
occ > external.services > smrs >	url	The URL to connect to the Search Manager Resolution Service.
occ > external.services > vcs > contentAPI	programImages.path	Relative path to post Image ContentInstanceMetadata to contentAPI WS. Full WS path = `{vcs/host}` + `{contentAPI/programImages/.path}`. Value must start with a '/' symbol.
	programs.path	Relative path to post ContentMetadata to contentAPI WS. Full WS path = `{vcs/host}` + `{contentAPI/programs/path}`. Value must start with a '/' symbol.
	programSeries.path	Relative path to post ContentGroupMetadata to contentAPI WS. Full WS path = `{vcs/host}` + `{contentAPI/programSeries/path}`. Value must start with a '/' symbol.
	schedules.path	Relative path to post Linear ContentInstanceMetadata to contentAPI WS. Full WS path = `{vcs/host}` + `{contentAPI/schedules/path}`. Value must start with a '/' symbol.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
occ > external.services > vcs >	enabled	Determines whether Videoscape Control Suite support is enabled for all modules. The default value is false.
	host	URL of the Video Control Service host. Acceptable value: protocol://hostname:port
occ > external.services > vcs > mappings >	credits	Defines mappings for Content credits. For example: Actor=urn:tva:metadata:cs:TVARoleCS:2005:4, Producer=urn:tva:metadata:cs:TVARoleCS:2005:7
	genre	Defines mappings for Content genres.
	mimeType	Defines mappings for Image mime-types.
	parentalGuidance	Defines mappings for Content parental guidance warnings.
	soundType	Defines mappings for sound types. EPG currently supports only 'Surround', 'Stereo' and 'Mono' sound types. For example: Surround=urn:nnds:mercury:metadata:soundType/DigitalSurround, Stereo=urn:nnds:mercury:metadata:soundType/SimpleStereo, Mono=urn:nnds:mercury:metadata:soundType/Monaural
occ > external.services > vcs >	password	The password for the Videoscape Control Suite API.
	username	The username for the Videoscape Control Suite API.
occ > external.services > vosm >	namespace	The namespace for the XML schema for the VOS Manager.
	username password	The username/password required to connect to the VOS Manager.
	url	The URL to connect to the VOS Manager.
occ > webservices >	configservice.wsdlcontext	The Web service endpoint where system configuration tasks are performed. For more details, see “Notes on All Web Services” on page 106.
	configservice.namespace configservice.servicename configservice.defaultuser configservice.defaultpassword	Endpoint connection parameters that are required by the internal configservice. For more details, see “Notes on All Web Services” on page 106.
	permissionservice.wsdlcontext	The Web service endpoint where permissionservice tasks are performed. For more details, see “Notes on All Web Services” on page 106.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
	permissionservice.namespace permissionservice.servicename permissionservice.defaultuser permissionservice.defaultpassword	Endpoint connection parameters that are required by the internal permissionservice. For more details, see “Notes on All Web Services” on page 106.
ocesbcore > webservices >	asynccontinuationservices.wsdlcontext	The Web service endpoint where asynccontinuationservices tasks are performed. For more details, see “Notes on All Web Services” on page 106.
	asynccontinuationservices.namespace asynccontinuationservices.servicename	The namespace for the XML schema for the Device Profile Service.
	eventservice.wsdlcontext	The Web service endpoint where eventservice tasks are performed. For more details, see “Notes on All Web Services” on page 106.
	eventservice.namespace eventservice.servicename	The namespace for the XML schema for the Event Service.
	exceptionhandlingservice.wsdlcontext	The Web service endpoint for the exceptionhandlingservice, which manages exceptions. For more details, see “Notes on All Web Services” on page 106.
	exceptionhandlingservice.namespace exceptionhandlingservice.servicename	The namespace for the XML schema for the exception handling Service.
rm > virtualizer >	virtualizer.repository	Specifies the location where temporary files will be stored for report generation.
	virtualizer.type	Specifies the format (such as GZIP) in which temporary files are stored for report generation. Refer to the Jasper Reports user guide for details on available formats.
scm >	locales.in.use.epg	Sets the locales that are in use by EPG. This is a read-only field that should not be changed.
scm > sm > apis > channel	returnFields	Overrides the default fields that are returned for channel via the Search Manager API. For details, see “Notes on Search Manager returnFields” on page 106.
scm > sm > apis > channels >	returnFields	Overrides the default fields that are returned for channels via the Search Manager API. For details, see “Notes on Search Manager returnFields” on page 106.
scm > sm > apis >	connectionTimeoutInMilliseconds	The maximum number of milliseconds that a connection can timeout without being disconnected from the Search Manager API.
scm > sm > apis > content > epg >	returnFields	Overrides the default fields that are returned for content via the Search Manager API. For details, see “Notes on Search Manager returnFields” on page 106.
	searchFields	Specifies EPG fields that are searched for content via the Search Manager API. These values are submitted in the “q” parameter.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
scm > sm > apis > content > epgAndVod >	returnFields	Overrides the default fields that are returned for EPG and VoD via the Search Manager API. For details, see “Notes on Search Manager returnFields” on page 106.
	searchFields	Specifies EPG and VoD fields that are searched for via the Search Manager API. These values are submitted in the “q” parameter.
scm > sm > apis > content >	phrase.search.proximity	Specifies the number of words apart that each word in a search phrase can appear in the sentence to produce a hit. For example if the customer is searching for "bouncing ball", a document with a title, "Bouncing big blue ball rolled down the hill" would be a match if the value of phrase search proximity was set to a value greater then 2.
	search.tie	Controls how hits in multiple fields of the document are combined to produce a relevancy rating that also takes into account field weights. As this is a complex subject, refer to SOLR online documentation for further details.
scm > sm > apis > content > vod >	returnFields	Overrides the default fields that are returned for VoD via the Search Manager API. For details, see “Notes on Search Manager returnFields” on page 106.
	searchFields	Specifies VoD fields that are searched for content via the Search Manager API. These values are submitted in the “q” parameter.
scm > sm > apis > contentdetails > epg >	returnFields	Overrides the default fields that are returned for the Content Details Search Manager API when requesting EPG entities only. For details, see “Notes on Search Manager returnFields” on page 106.
scm > sm > apis > contentdetails > vod >	returnFields	Overrides the default fields that are returned for the Content Details Search Manager API when requesting VoD entities only. For details, see “Notes on Search Manager returnFields” on page 106.
scm > sm > apis >	defaultMax	Specifies the maximum default number of documents that can be returned by the Search Manager API.
	maxConnections	The maximum number of connections that are allowed from the Search Manager business layer to the Solr slave data store.
scm > sm > apis > programschedule >	returnFields	Overrides the default fields that are returned for the Program Schedule Search Manager API when requesting EPG entities only. For details, see “Notes on Search Manager returnFields” on page 106.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
scm > sm > apis >	readTimeoutInMilliseconds	Sets a maximum timeout value for the Search Manager API when reading data. Any reads that exceed this value will terminate the connection.
scm > sm > apis > recommendcontent > epg >	returnFields	Specifies EPG fields that are returned for recommended content from the Search Manager API.
scm > sm > apis > recommendcontent > vod >	returnFields	Overrides the default fields that are returned for recommendedcontent via the Search Manager API. For details, see “Notes on Search Manager returnFields” on page 106.
scm > sm > apis > recommendschedules >	returnFields	Overrides the default fields that are returned for recommendschedules via the Search Manager API. For details, see “Notes on Search Manager returnFields” on page 106.
scm > sm > apis > regions >	returnFields	Overrides the default fields that are returned for regions via the Search Manager API. For details, see “Notes on Search Manager returnFields” on page 106.
scm > sm > apis > related > epg >	returnFields	Overrides the default fields that are returned for the Related Search Manager API when requesting EPG entities only. For details, see “Notes on Search Manager returnFields” on page 106.
scm > sm > apis > related > vod >	returnFields	Overrides the default fields that are returned for the Related Search Manager API when requesting VoD entities only. For details, see “Notes on Search Manager returnFields” on page 106.
scm > sm > apis > schedule > program >	returnFields	Overrides the default fields that are returned for the Schedule Search Manager API for the program portion of each scheduling event. For details, see “Notes on Search Manager returnFields” on page 106.
scm > sm > apis > schedule > schedule >	returnFields	Overrides the default fields that are returned for the Schedule Search Manager API for the schedule entity portion of the scheduling event. For details, see “Notes on Search Manager returnFields” on page 106.
scm > sm > apis > schedule > station >	returnFields	Overrides the default fields that are returned for the Schedule Search Manager API for the station portion of each scheduling event. For details, see “Notes on Search Manager returnFields” on page 106.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
scm > sm > apis > schedules >	returnFields	Overrides the default fields that are returned for the Schedules Search Manager API when requesting schedules entities only. For details, see “Notes on Search Manager returnFields” on page 106.
scm > sm > apis > typeahead >	max	Specifies the maximum number of entries that will be returned when using typeahead functionality.
	phrase.search.proximity	Specifies the number of words apart that each word in a search phrase can appear in the sentence to produce a hit. For example if the customer is searching for "bouncing ball", a document with a title, "Bouncing big blue ball rolled down the hill" would be a match if the value of phrase search proximity was set to a value greater then 2.
scm > sm > cache >	flush.minimum.interval	The minimal amount of seconds that you must wait between flush events. This mechanism prevents frequent cache flushes, which would have negative performance implications.
	flush.wait.cycles.program	The minimum number of cycles to wait for multi-lingual data updates for flushing the Search Manager content cache. If you need to flush immediately, set this value to 0. This value can be used if you only have metadata in a single language. By default, each cycle is 30 seconds long.
	flush.wait.cycles.schedules	The minimum number of cycles to wait for multi-lingual data updates to arrive before flushing the Search Manager schedules cache. If you need to flush immediately, set this value to 0, which can be performed if you have metadata only in a single language. By default, each cycle is 30 seconds long.
	flush.wait.cycles.vod	The minimum number of cycles to wait for multi-lingual data updates for flushing the Search Manager content cache. If you need to flush immediately, set this value to 0. This value can be used if you only have metadata in a single language. By default, each cycle is 30 seconds long.
scm >	solr.master.defaultuser solr.master.defaultpassword	Sets the default user and password for the Solr master.
scm > webservices >	cacheflushwebservice.wsdlcontext	The Web service endpoint where cacheflushwebservice tasks are performed. For more details, see “Notes on All Web Services” on page 106.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
	cacheflushwebservice.namespace cacheflushwebservice.servicename cacheflushwebservice.defaultuser cacheflushwebservice.defaultpassword	Endpoint connection parameters that are required by the internal cacheflushwebservice. For more details, see “Notes on All Web Services” on page 106.
	searchconfigwebservice.wsdlcontext	The Web service endpoint where searchconfigwebservice tasks are performed. For more details, see “Notes on All Web Services” on page 106.
	searchconfigwebservice.namespace searchconfigwebservice.servicename searchconfigwebservice.defaultuser searchconfigwebservice.defaultpassword	Endpoint connection parameters that are required by the internal searchconfigwebservice. For more details, see “Notes on All Web Services” on page 106.
tstv >		Section describes Time Shift TV related properties
tstv > bundle.generation >		Section describes properties related to Content Manager Logical Video TSTV bundles generation
	lock.timeout	Time in seconds. Indicates how long Linear Manager node can hold a lock for particular TSTV bundle generation entry. Once timeout is expired, the TSTV bundle generation entry will be available for re-generation. Defaults to 1800.
	retry.count	The maximum number of retries after which failed TSTV bundle generation entry will be ignored forever. Defaults to 5.
tstv > capture.schedules >		Section describes properties related to TSTV Capture Schedules generation.
	maxThreads	Number of maximum allowed threads for Capture Schedules generation. Defaults to 10.
tstv >	cm.update.repg.expiry.buffer	Determines the number of seconds for the Scheduler Service to register physical assest in Content Manager. Physical assets are registered after the event broadcasting is finished. Defaults to 1800.
	cm.update.repg.expiry.enabled	Determines if Reverse EPG expiry date should be updated (true) or not (false) in ContentManager. Defaults to true.
	max.days	Allowed number of days for which Capture schedules will be generated and Reverse EPG data will be indexed. Defaults to 3. **Warning** Setting this value too high can have a negative impact on indexing time.
	recorded.content.prefix.catchup	URL prefix for Catchup release URL field in index (corresponding field name is 'rurl') which is appended during indexing. Defaults to `/TSTVC/`.

Table 24 modules Node Hierarchy

Node Levels	Value(s)	Description
	recorded.content.prefix.epg	URL prefix for Reverse EPG release URL field in index (corresponding field name is 'rurl') which is appended during indexing. Defaults to `/TSTVR/`.
tstv > tstv.scheduler >		Section describes properties related to Scheduler service.
	host	URL of Scheduler service. Acceptable value: protocol://hostname:port.
	namespace	The namespace for the XML schema for the Scheduler service.
	username password	The username/password required to connect to Scheduler service. Optional.
	schedules.notify.path	Path that will be added to host information. Resulted URL is used to notify Scheduler service about Capture Schedule changes. Defaults to /scheduler/sa/schedules/update.
	station.notify.path	Path that will be added to host information. Resulted URL is used to notify Scheduler service about station source changes. Defaults to `/scheduler/sa/station/update`.
ws > webservices >	processdefinitionservice.wsdlcontext	The Web service endpoint for the processdefinitionservice, which manages Workflow Definitions. For more details, see “Notes on All Web Services” on page 106.
	processdefinitionservice.namespace processdefinitionservice.servicename processdefinitionservice.defaultuser processdefinitionservice.defaultpassword	Endpoint connection parameters that are required by the internal processordefinitionservice. For more details, see “Notes on All Web Services” on page 106.

Services Nodes

Services node settings define settings for services that are global to Media Suite, such as SSO (Single Sign On).

Table 25 services Node Hierarchy

Node Level	Value	Description
sso >	keepalive.interval	The interval at which Media Suite checks with SSO to verify if the login cookie is still valid.