numecent®

Cloudpaging Server Administration Guide (Version 9.7)

Legal

Product version: 9.7.0

Guide revision: 3

Last update: 12 January 2024

NOTE: The product version syntax for Cloudpaging Server is: **M.m.p**

(M:major version number, m:minor version number, p:patch version number)

Table of Contents

Introduction

The *Cloudpaging Server Administration Guide* describes how to administer users, applications, license policies, and services in the Cloudpaging Server.

The following are explained in this section:

- Features of Cloudpaging Server
- Related Documentation
- Documentation Conventions
- Getting Support

Features of Cloudpaging Server

- High user-to-server ratio (thousands of users to each server) provides a more cost-efficient approach to on-demand application delivery.
- Fully scalable deployment topology.
- Fully secure, encrypted sessions between server and client with individual, unique keys per session to prevent piracy and spoofing.

Related Documentation

The following manuals are also available for other Cloudpaging products:

- **Cloudpaging Server Supplemental Admin Guide for Enterprise Portal**: Contains the procedures for configuring, maintaining, and using the Enterprise Portal add-on for the Cloudpaging Server.
- **Cloudpaging Studio User Guide:** Contains the procedures for installing and using the Cloudpaging Studio, the application packaging component.
- **Cloudpaging Player User Guide**: Contains procedures for the Cloudpaging Player, the component installed on each user's PC that communicates with the Cloudpaging Server.
- **Cloudpaging Web Portal Integration Supplement Guide**: Contain the procedures and coding examples to integrating Cloudpaging Server with other management tools and consoles.

Document Conventions

The following typefaces are used throughout this guide:

Bold text	Names of windows, panes (portions of a window), dialog boxes, menus, commands, buttons, predefined folders, and keyboard keys. Examples: Click **OK**. Press **Enter**. Navigate to the **My Documents** folder.
Italic text	Indicates references to a section within this document or other documentation. Also used to emphasize new terms. Example: Refer to the *Cloudpaging Web Portal Integration Supplemental Guide* for further information.
> symbol	Sequence in which you should select a menu option. Example: "File > New > Document" means "click the File menu, click New, and then click Document."

1, 2, 3	Numbered items indicate sequential steps in a procedure. Sub steps are indicated with indented bullets. Relevant graphics or screen captures may follow steps.
Screen captures	The screen captures in this document are examples only. They may not exactly match the user interface on your system.

The following icons are used throughout this document:

NOTE

This block style identifies additional information about the preceding text, or the text immediately to follow.

Important

This block style identifies information about actions that might cause problems with an application, your local PC system, or your data, including data loss.

Getting Support

Website

Support information is available online on at: http://support.numecent.com/

Email

You can request technical support by sending an email to support@numecent.com. Our support staff will address your email questions in a timely manner.

Chapter 1: Cloudpaging Server Overview

The Cloudpaging Server is a product that provides a cost-effective method to rapidly deploy and manage applications to a large pool of users. It provides the core functions for Cloudpaging applications to users.

The main features are:

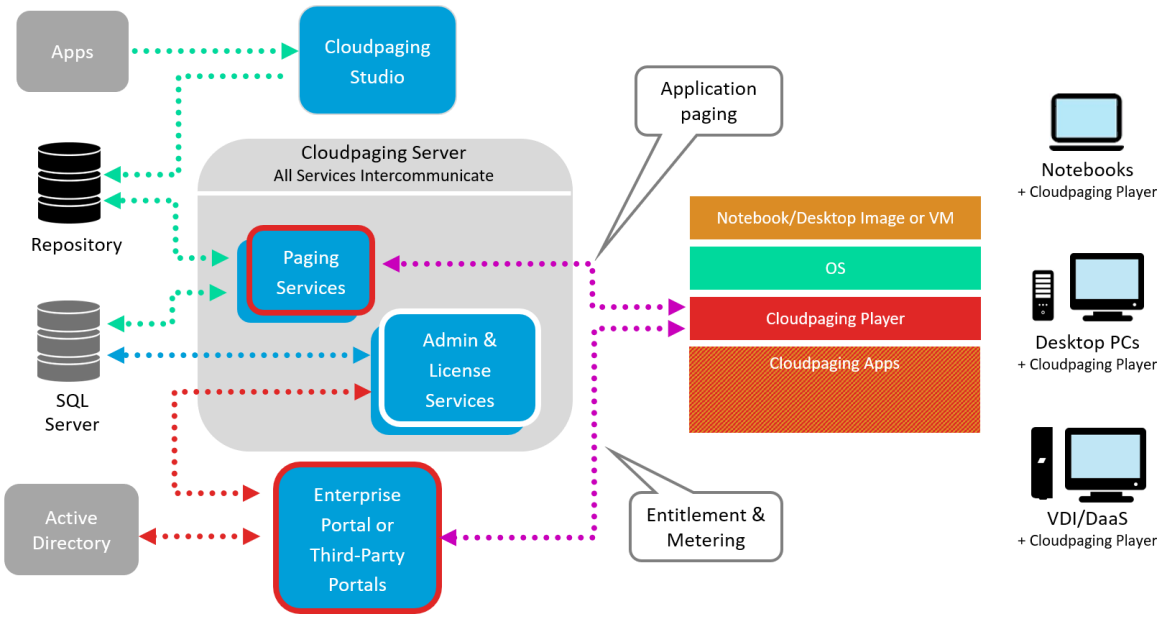
- Patented application Cloudpaging technology
- Anti-piracy protection of application code
- Usage metrics collection
- Licensing policies to enforce all application usage
- Scalability and built-in failover support

The following are explained in this section:

- Cloudpaging Components
- Cloudpaging Server Components
    - Admin Services
    - License Services
    - Paging Services
    - Microsoft SQL Server Database
    - Repository
- Role of the Cloudpaging Server Administrator

Cloudpaging Components

The Cloudpaging system contains three components that deliver a native desktop application experience to end users without a traditional installation. The three components that work together are the Cloudpaging Studio, Server, and Player.



The Cloudpaging Studio is used to package applications into the Cloudpaging format. The Cloudpaging Studio allows packagers to capture applications, dependencies, custom configurations, and custom scripts to create Cloudpaging packages that can be deployed to user endpoints.

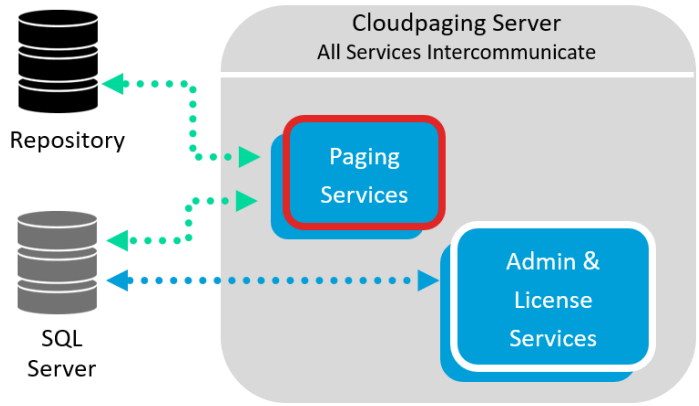
The Cloudpaging Server is where your applications will reside after they've been packaged and where the provisioning and monitoring take place. Once your applications are on the Cloudpaging Server, your IT department can use our Enterprise Portal add-on to provision applications to end users. The Enterprise Portal integrates with Active Directory and allows for entitlement via auto-deploy and a self-service portal. For more information on the Enterprise Portal, please see *Cloudpaging Server Supplemental Admin Guide for Enterprise Portal*.

The Cloudpaging Player is the unique piece of software that "plays" the provisioned applications on the end users' virtual or physical devices.

The Cloudpaging system is designed to be extremely versatile to deliver what you want, when you want, to whomever you want.

Cloudpaging Server Components

Cloudpaging Server provides a Web UI for administration of Cloudpaging Server services. The following components are part of the Cloudpaging Server services and dependencies:



Admin Services

The Admin services provide distinct and configurable web-based interfaces for the administrator and allows for the management of applications via license policies. A web interface is delivered with Cloudpaging Server and is configured to utilize Tomcat to deliver content.

License Services

In the Cloudpaging Server, license policies can be tailored to define parameters that control access and usage of application for a specified period of time. Administrators can create multiple license policies per application. The License services manage license policies and meters application usage by granting, renewing, and expiring user access tokens.

Features include:

- Validating the license policies for applications running on user machines.
- Tracking software license policies that are currently in use.

Paging Services

The Paging services store, manage and page cloudified applications to the Cloudpaging Player on end-users' devices.

Microsoft SQL Server Database

Cloudpaging Server utilizes Microsoft SQL Server database to track user information and server resources. All system and user information is stored in the database. This includes system configuration, topology, application information, application license policies, usage, and user account information.

Repository

The Repository is the component that stores all cloudified applications. This area must be accessible by all Cloudpaging Server components.

Role of the Cloudpaging Server Administrator

The Cloudpaging Server administrator typically performs the following duties:

- Installs and configures Cloudpaging Server software.
- Adds or uploads cloudified applications.
- Manages configuration and delivery of cloudified applications.
- Controls access to the Cloudpaging Server system.
- Monitors the performance of cloudified applications.

Chapter 2: System Requirements and Planning

This chapter provides information you need to plan your Cloudpaging Server system, as well as hardware and software requirements. Use the topology and capacity planning information to determine the number of server machines required. Once you have determined your server capacity requirements, you can procure the appropriate hardware and software.

The following are explained in this section:

- Hardware Requirements
- Software Requirements
- Cloudpaging Server Topology
  - Basic System Topology
  - Distributed System Topology
  - Determining the Number of Servers
  - The Importance of Using Domain Names
- System Security Considerations
  - Service Communication Ports
- Requesting a Product License Key
  - Conditions That Will Affect the Product License Key
  - Importing a Product License Key
- About Credentials and Passwords

Hardware Requirements

Each machine must meet or exceed the system requirements shown below:

Category	Recommended
CPU	2.5 GHz 8–core or higher
RAM	8 GB or higher
Disk Space	200 GB separate partition. SSD or NVMe for cache drive.
Network Interface	5 Gbps or higher

> ***NOTE***
>
> *The number and size of the applications on the server will determine the actual disk space required on your unique system.*

Software Requirements

In addition to the Cloudpaging Server software, the following software applications must be installed.

Software	Requirement
Operating System	Windows Server 2016, Windows Server 2019, or Windows Server 2022
Database Server	Microsoft SQL Server 2017, Microsoft SQL Server 2019, or Microsoft SQL Server 2022
Directory Service (Required for Enterprise Portal add-on)	Active Directory 2012 and later

The Cloudpaging Server installation configures the following prerequisites if necessary:

- Tomcat 9.0
- Adopt OpenJDK 11
- JDBC for MS SQL 9.4

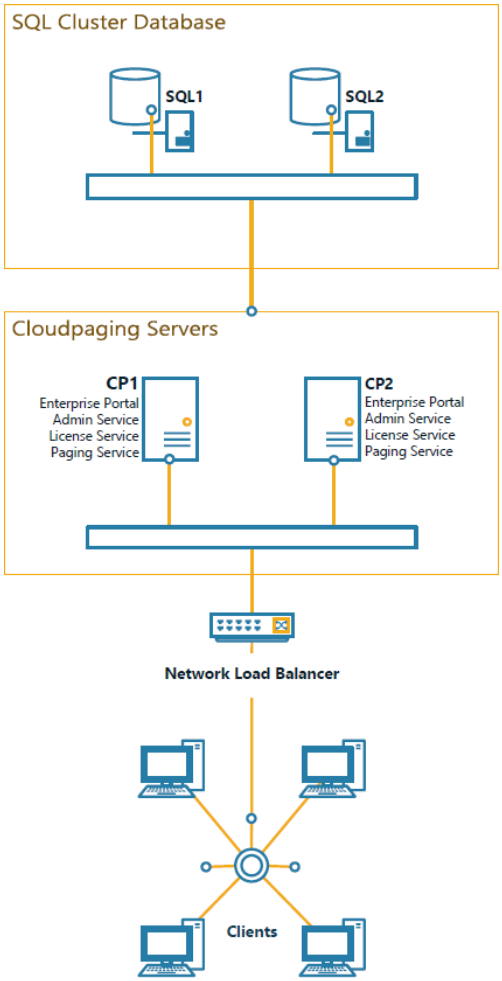
The Cloudpaging Server Web UI works on all modern browsers that support JavaScript, including Microsoft Edge, Firefox, and Chrome.

Cloudpaging Server Topology

Cloudpaging Server is composed of a set of services: the Admin service, Paging service and License service. These Cloudpaging services can be installed on the same machine (e.g. Physical or Virtual machine instance) or separately across machines, depending if the need is simply to test or to provide a scalable production system. Installation of the Cloudpaging Server will prompt you to select the services (Admin, License, Paging) you intend to install.

Basic System Topology

A basic fail-over topology can be used in a pilot phase or with small deployments as it is the simplest configuration to setup and maintain. This topology will provide basic fail-over but is not designed to scale. For this topology we recommend the system configuration to have a SQL database replica or cluster, two servers with the Cloudpaging services, and a load balance switch to route the traffic between the services. This topology can support a maximum of 1000 users, 5000 sessions, and 100 applications. If a greater capacity is needed, then please see the Distributed System Topology.



With this configuration, all the Cloudpaging services will be installed on the same machine. All Cloudpaging services must be connected to the Microsoft SQL database, preferably configured using database clustering,

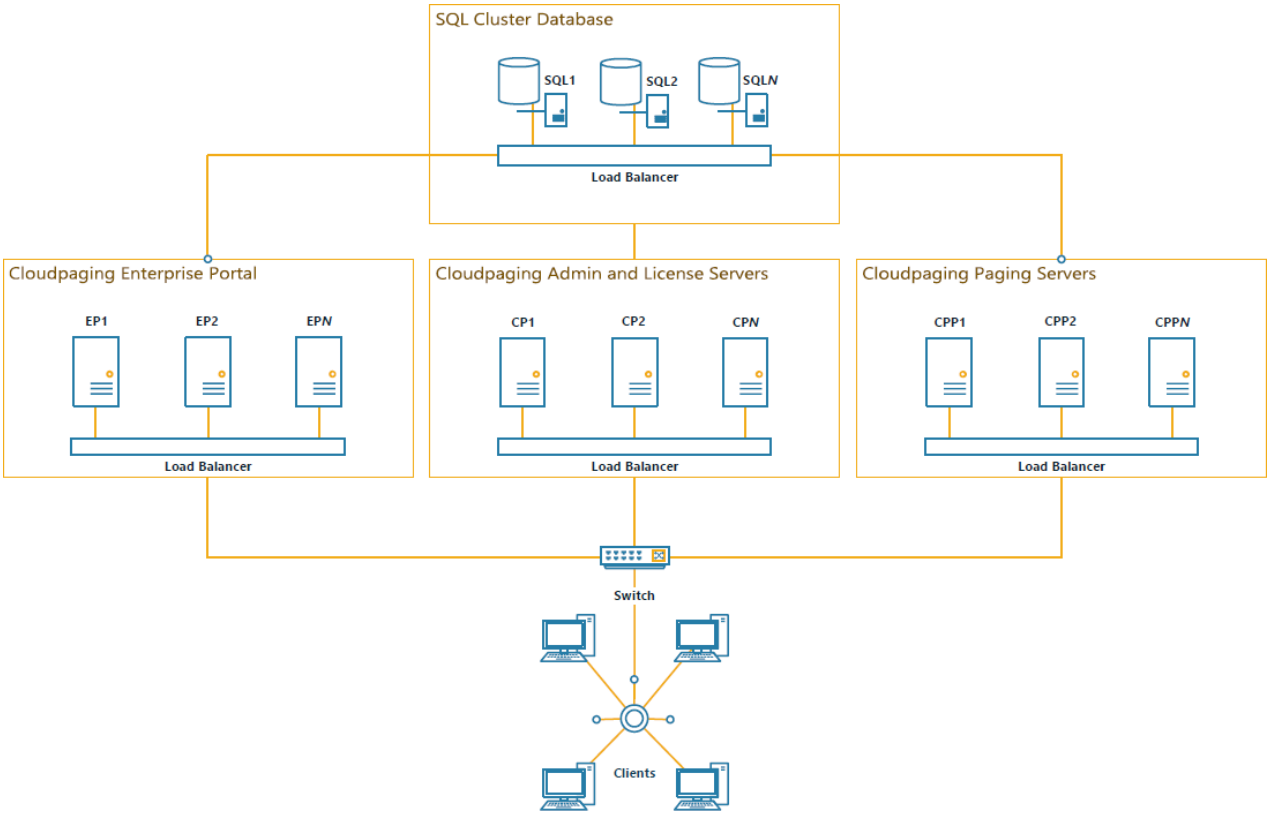
which provides both scaling and fault tolerance. Without a cluster, the database should be replicated to a fail-over system. It is strongly recommended to back up the database nightly.

A load balancer is required to provide fail-over for the Cloudpaging services. This can be a hardware device or software, such as NGINX. When using a load balancer with Admin Service or Enterprise Portal, you must enable persistent binding (or stickiness) on the load balancer. All other services are stateless and do not need persistent binding. Some load balancers require specifying the application session cookie name for stickiness, in which case, specify: JSESSIONID. For more information on configuring load balancers, please see our Support article https://support.numecent.com/support/solutions/articles/1000293062/.

> ***Important***
>
> *The Basic System Topology does not scale if more capacity is required. If the ability to increase capacity is required, then the Distributed System Topology is required.*
>
> *It is not possible to convert a Basic System Topology to a Distributed System Topology.*

Distributed System Topology

For a high availability, fault tolerant distributed system, we recommend the system configuration to have distributed Cloudpaging services, in addition to the basic database cluster and load balancer design. It is important to follow this distributed system recommendation for any large production system as it has been designed to handle a higher volume of traffic and system load. The distributed system will provide both fail-over and the redundancy necessary, and allow for future scaling needs.



When installing Cloudpaging Server on multiple machines, you must run the Cloudpaging Server installer on each machine. During installation, select only the features and Cloudpaging services needed for each machine. The minimum number of Cloudpaging services in this topology is 2 Paging services, 2 Admin/License services (these can be installed on the same machine), and 2 Enterprise Portal services (if being used) for a total of 6 machines.

For example, on a machine designated only as a Paging server, you will only select the Cloudpaging Server feature and the Paging service. On a machine designated for Enterprise Portal, you will only select the Enterprise Portal add-on feature and de-select the Cloudpaging Server feature.

> ***Important***
>
> *For distributed systems, all servers must have the time synchronized for the Cloudpaging Server product licensing to work properly.*

All Cloudpaging services must be connected to the database configured using database clustering or a high-available configuration, which provides both scaling and fault tolerance. The Cloudpaging database (StreamDB) and account (dbjbuser) will be created during the installation by the first machine installing the software. This account requires db_owner permission to the Cloudpaging database, which the installation will configure if not already setup.

A load balancer is required to be in front of all the Cloudpaging services. You must enable persistent binding (or stickiness) on the load balancer when configuring a load balancer with Admin Service or Enterprise Portal. All other services are stateless and do not need persistent binding. Some load balancers require specifying the application session cookie name for stickiness, in which case, specify: JSESSIONID. For more information on configuring load balancers, please see our Support article https://support.numecent.com/support/solutions/articles/1000293062/.

Determining the Number of Servers

Choosing the specific number of services required for the Distributed System Topology is based on capacity planning. We recommend overprovisioning your servers to allow for growth. For more help planning your system capacity needs, visit our Support article at https://support.numecent.com/support/solutions/articles/1000259018.

The Importance of Using Domain Names

It is important to use Domain Name Server (DNS) values for servers' External DNS configurations. The following DNS records should be used in a production system:

- paging service DNS
- license service DNS
- admin service DNS
- enterprise portal DNS (if used)

External DNS values are used when by the Cloudpaging Player to talk to the services. Hence, domain names will need to be resolvable and routable with firewalls and load balancers configured appropriately.

System Security Considerations

It is always a good practice to have a high level of security in place. Cloudpaging is built on web technology standards to allow for common Internet security measures. Consider implementing each of the following security standards when configuring your system:

- **TLS Certificate:** We recommend that you use TLS protocols with your web server. The Cloudpaging Server runs on Tomcat and Tomcat can be configured to use TLS protocols. We recommend using certificates from trusted public certificate authorities as such providers are already trusted by Windows and various browsers.
- **Firewalls:** We recommend that you place firewalls in front of all the Cloudpaging Servers. If you are using TLS, then the only ports you would need to allow open are those to the Paging and License services.
- **System Updates:** We recommend that all Windows updates be applied promptly and kept current to ensure that system security is up to date.

NOTE

Numecent supports the latest patch version for any prerequisites required by a supported product, such as Tomcat 9.0.x. We encourage all customers to maintain their environments with the latest operating systems patches and prerequisite patches. For more information, please visit our Support article https://support.numecent.com/support/solutions/articles/1000278316.

Service Communication Ports

The services communicate through the ports as specified below. A mechanism is in place to resolve port conflicts at installation. When the server installer detects a port conflict, it will try to allocate another port above it (For example: 443+1, 443+2, 443+3, etc.). The following is a list of services and their default communication ports:

Service	Default Port	Traffic Flow	Connection
Admin Service	443	inbound	stateful
License Service	443	inbound	stateless
Paging Service	80	inbound	stateless
Enterprise Portal	443	inbound	stateful

NOTE

For proper service-to-service communication, open the Control Service port (the default 443 or the configured port number) for both inbound and outbound traffic.

Requesting a Product License Key

The Cloudpaging Server system requires a product license key from Numecent. Please contact your Numecent sales representative to discuss licensing options. For more information, visit our Support article https://support.numecent.com/solution/articles/1000259073/

Conditions That Will Affect the Product License Key

You will need a new product license key under any of the following conditions:

- Exceeding the permitted number of Paging or License services.
- Changing an IP address of the database server.
- Adding additional seats.
- Product key expiration.
- External DNS value changes for servers.

Importing a Product License Key

When you import a product license key, it is recommended to import after hours or during a maintenance window. The product key may need to be accepted and can take a few seconds to apply. Please be sure to not use a load-balancer when importing the product license. If the product license key is not valid or the terms are not accepted, then the system may experience an outage until a valid key is imported and accepted.

About Credentials and Passwords

While using Cloudpaging, there are a number of credentials that you should keep in mind. While most system administrators are familiar with these, the Cloudpaging system has specific credentials that require passwords. Below is a list of credentials that require passwords when they are installed.

Component	Description
SQL Server	During the installation of MS SQL Server Enterprise, you will enter a username and password. Be sure to remember the credentials because you will need them for the setup. For MS SQL Server Express installed with Cloudpaging Server as a prerequisite, use the default **sa** account and **Change!it** password during setup.
StreamDB database	The database requires a username and password for executing SQL queries. Each server in Cloudpaging Server requires this password to access the database. This information is stored encrypted in the database connection file at **INSTALLDIR\core\conf\dbConfig.xml**.
Cloudpaging Server	To administer the Cloudpaging Server environment, a default local admin account (username is **admin** with the password set as **password**) is created in the system.
Windows Service	All Cloudpaging servers are required to run as Windows Service with local system privileges. However, if a repository is located on a network file share, then the service needs an account with both local system access and access to the file share. This can be set under the Management Console for Windows Services.
Directory Service	Enterprise Portal uses active directory to manage groups and application associations. Installation of Enterprise Portal is optional. The default local admin account (username is **admin** with the password set as **password**) is used for initial setup and configuration.

NOTE

We strongly recommend that administrators enforce a strong password policy. The Cloudpaging Server requires passwords to be between 8 and 64 characters and contain three of the four below:

- A lower case character
- An upper case character
- A number
- A special character

Chapter 3: Cloudpaging Server Installation

This chapter provides procedures to install, uninstall and upgrade the Cloudpaging Server.

The following are explained in this section:

- Procedure for Installing Cloudpaging Server
  - Step 1 - Configuring a Pre-existing SQL Server
  - Step 2: Installing Prerequisite Components
  - Step 3: Enable SQL Protocols and Setting Ports
  - Step 4: Installing Cloudpaging Server
  - Step 6: Setting Up the Database Server
  - Step 7: Core Services Setup
  - Step 8: Final Installation Wizard Procedures
  - Step 9: Replacing the Certificate and Keystores
  - Step 10: Verifying the Installation
- Completing Configuration and Setup
- Uninstalling Cloudpaging Server
- Modifying Cloudpaging Server
- Upgrading Cloudpaging Server
  - Upgrading the database schema
  - Upgrading server binaries

Procedure for Installing Cloudpaging Server

The installation procedures listed in this chapter should be followed after the required hardware and software have been procured and configured. Ensure that you have performed capacity planning and provisioned your servers to allow for growth. For more information, see Chapter 2: System Requirements and Planning.

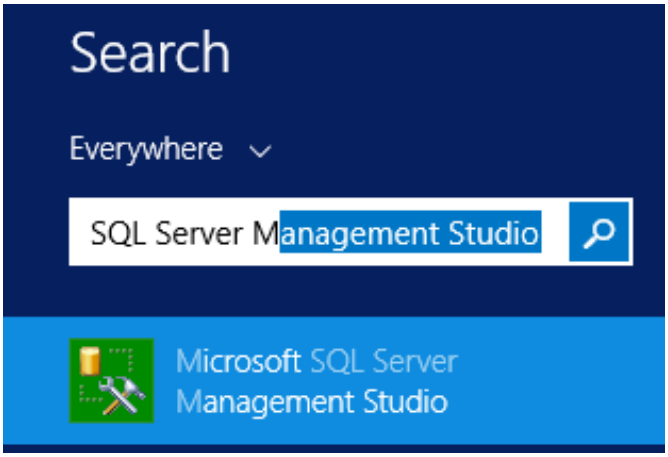
> ***NOTE***
>
> *If you plan to use Windows Authentication and already have a MS SQL Server database installed, please skip Step 1. Please note that in Step 2 you need to run the Cloudpaging Server installer with an account that has **local administrato**r permission and has **sysadmin** role on the MS SQL Server.*

Step 1: Configuring a Pre-existing Microsoft SQL Server

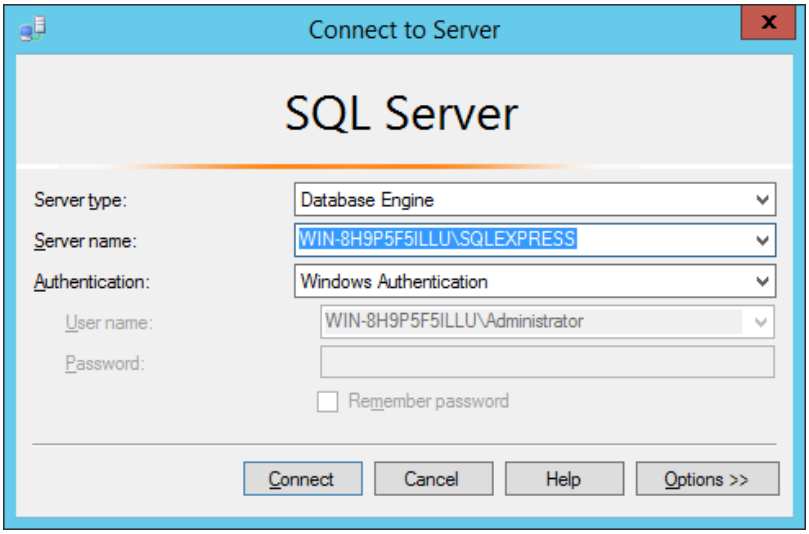
This section only applies to pre-existing database installations. If you do not have a database already installed, please skip to the next step.

To set up SQL Server Authentication mode:

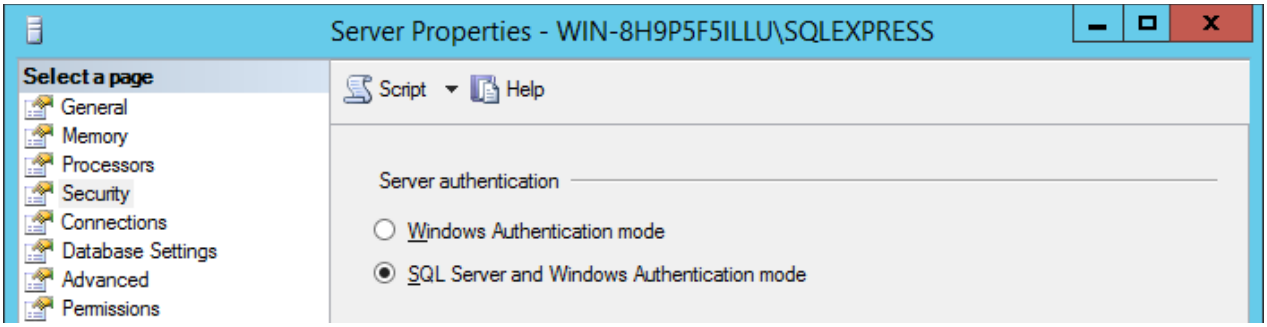
1. Open SQL Server Management Studio



2. Enter information about the SQL database machine and click **Connect**



3. On the left pane, right-click the SQL Server Instance Name or IP address, and select **Properties**.
4. Select the **Security** icon. The **Server Properties** dialog box appears.

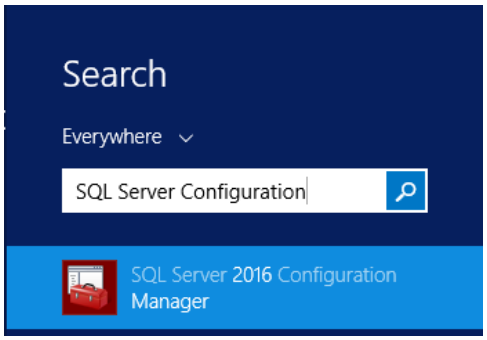


5. Select **SQL Server and Windows Authentication mode**, and click **OK.** SQL Server authentication mode is now set.

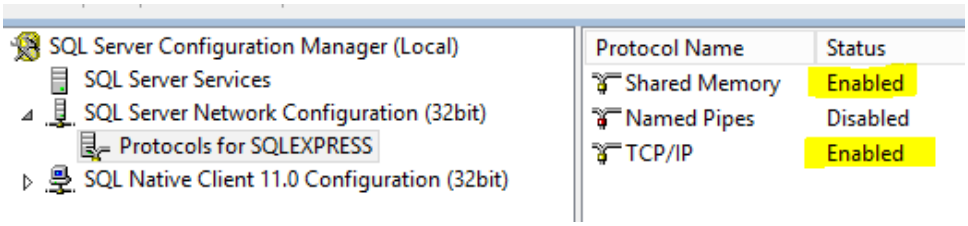
To enable SQL Protocols and Setting Ports:

Please follow these instructions if the database installed is **SQL Server Express Edition**.

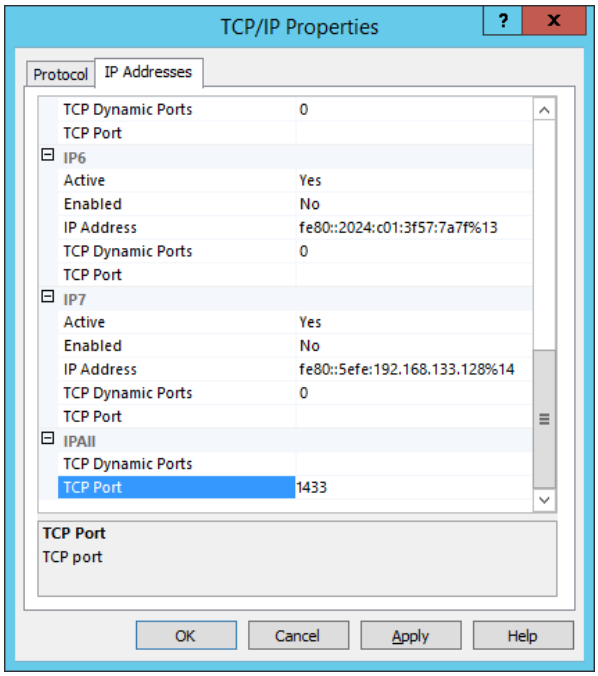
1. From the Server, run **SQL Configuration Manager**. The **SQL Server Configuration Manager** dialog appears.



2. On the **SQL Server Configuration Manager** page, enable the **Shared Memory and TCP/IP protocols**.



3. In the **TCP/IP Properties** dialog, under the **IP Addresses** tab of, set **IP ALL: TCP Port** to a static port (such as 1433), leave **TCP Dynamic Ports** blank, and click **OK**.



4. From the **SQL Service Configuration Manager** dialog, restart the SQL service. Your pre-installed database has been configured.

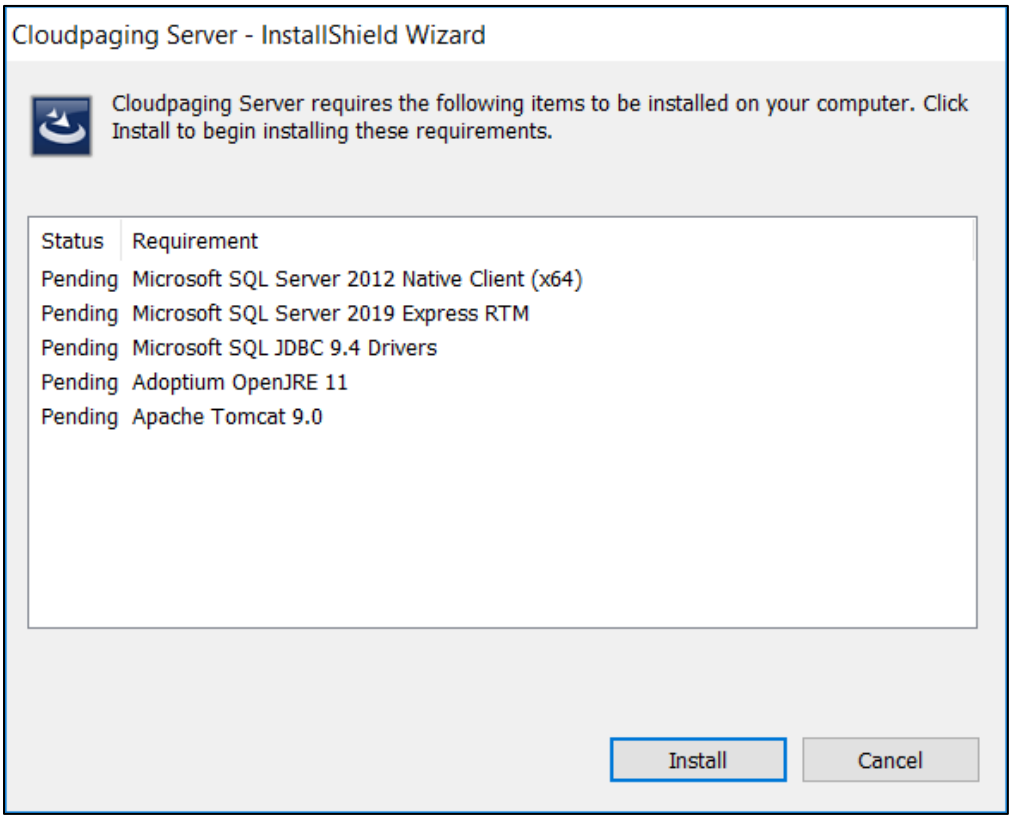
Step 2: Installing Prerequisite Components

To run Cloudpaging Server installation wizard:

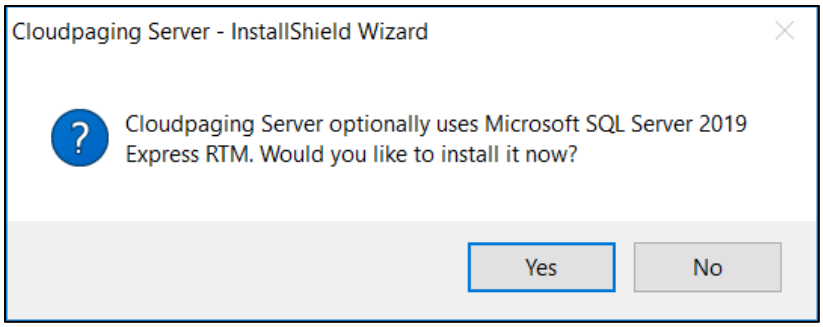
1. Copy the entire Server folder onto the server machine. The contents of the server installation folder include:

   - **ISSetupPrerequisites** folder
   - **Application Jukebox Server.exe** file
   - **Database** folder

2. To launch the Cloudpaging Server installer, double-click **Application Jukebox Server.exe**. You will be prompted to install prerequisite packages required by the Cloudpaging Server.



3. Click **Install** to begin the installation process. You will receive a prompt to install **Microsoft SQL Server 2019 Express RTM**.



4. Click Yes to install the database. Click No if the database is installed on another machine or if an identical or newer version of the database already exists on the local system. Please note that the components will also install prerequisite packages. For more information on prerequisites, see [Software Requirements.]
5. You can select the default options for all prerequisite packages

As the installation of each component is completed, the component's status changes from **Pending** to **Succeeded** in the InstallShield window.

Step 3: Enable SQL Protocols and Settings Port

This step is only required if you installed SQL Server 2019 Express RTM as part of Step 2. If you did not install a database as a part of the previous step, please skip to the next step.

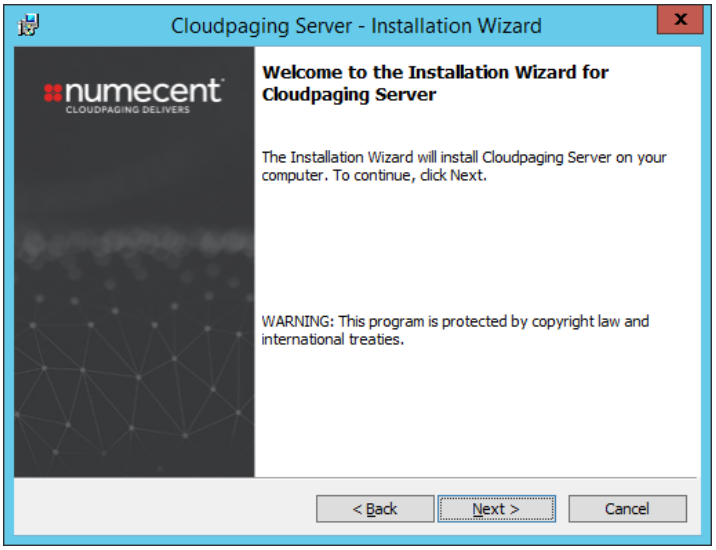
1. Once database installation finishes, please complete the steps in the section [To enable SQL Protocols and Setting Ports](#).

Your database has now been configured.

Step 4: Installing Cloudpaging Server

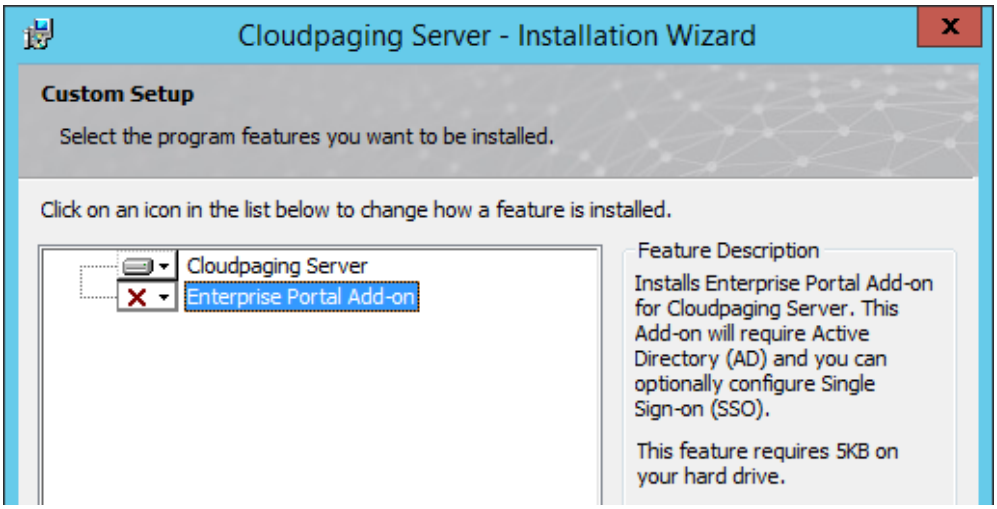
After all requirements have been installed the **Cloudpaging Server Installation Wizard** dialog will appear. This may take a couple of seconds.

1. Click **Next.** The **Welcome** dialog box appears**.**



2. Click **Next.** The **License Agreement** dialog box appears**.**

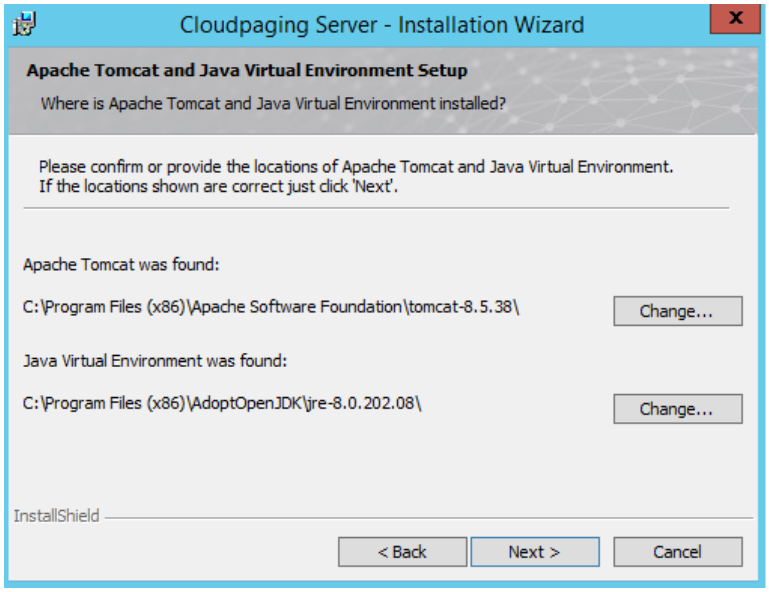
3. Select **I accept the terms in the license agreement**, and click **Next.** The **Custom Setup** dialog appears. The default setting is with the **Cloudpaging Server** feature selected, and with **Enterprise Portal Add-on** deselected.



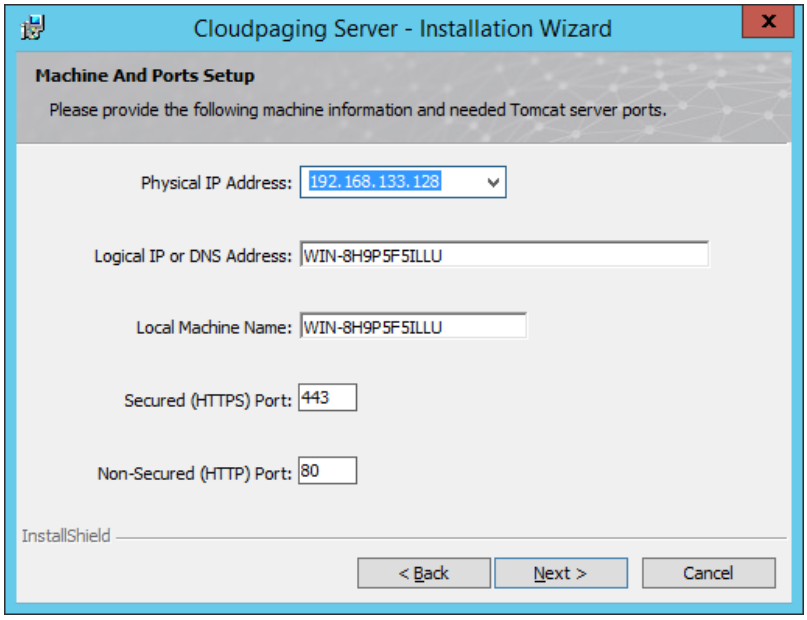
4. Select the desired features then click **Next**. The **Tomcat and JRE Setup** dialog box appears.

Step 5: Setting up Tomcat, JRE, and Cloudpaging Servers

In this screen, you will configure Tomcat and Java, which were installed as part of Step 2.



1. To change the default paths, click **Change** and browse to the appropriate directory.
2. Click **Next. The Machine and Ports setup** dialogue box appears. The following fields are displayed on this dialog box:

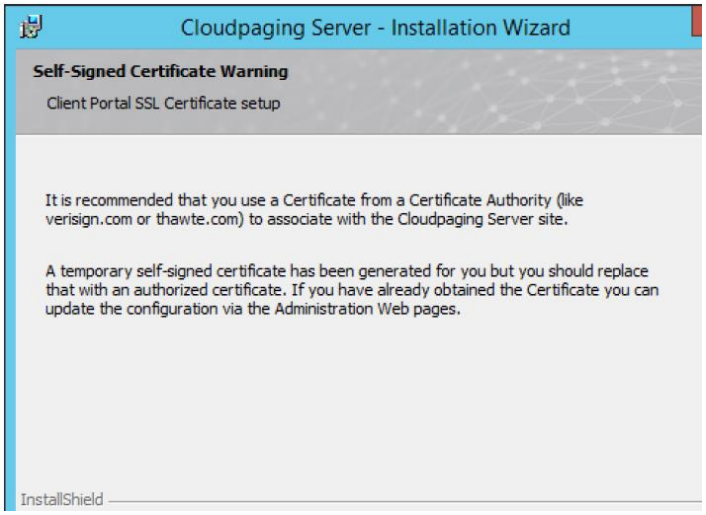


- **Physical IP Address**: The local IP address of the machine.
- **Logical IP or DNS Address**: The address used by Cloudpaging Player to talk to Cloudpaging services on this machine. In most situations, you should use the fully qualified domain name (FQDN) or external DNS name of the machine.
- **Local Machine Name:** The name of the local PC with Cloudpaging Server installed.
- **Secure (HTTPS) port:** The port used for secure HTTP connections. Default is 443.
- **Non-secured (HTTP) port:** The port used for non-secure HTTP connections. Default is 80.

NOTE

The administrator is expected to create all routing rules and DNS entries.

3. Click **Next**. The **Self-Signed Certification Warning** dialogue box appears.

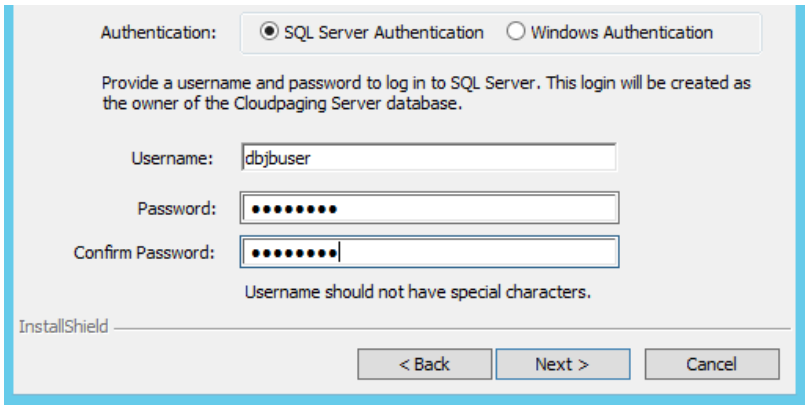


4. Click **Next** to continue. The installer will create a temporary certificate for the Cloudpaging Server. You will be taken to the **Database Owner Credential** screen of the installer dialogue box.

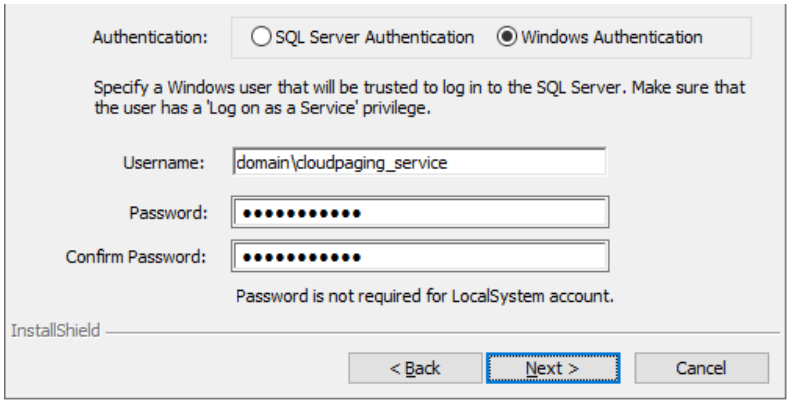
Step 6: Setting up the Database Server

The installer will create a database named **StreamDB**. This screen allows you to choose which database authentication method Cloudpaging Server should be using to connect to the database server. SQL Server Authentication and Windows Authentication methods are supported.

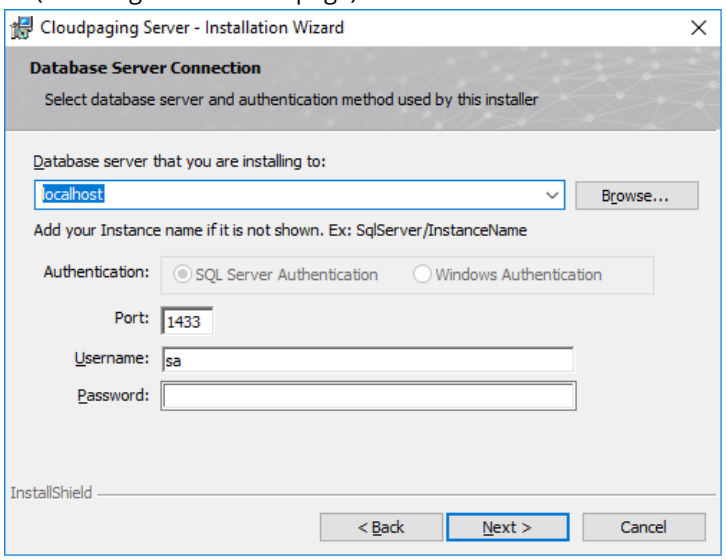
1. Specific the **user account** that the Cloudpaging Server will use to communicate to the database. The user specified requires **db_owner** privilege to access the database.
   - **SQL Server Authentication:** Use this authentication method if you want the Cloudpaging Server to connect to the Database Server via a user account created within the instance of SQL Server. When choosing SQL Server Authentication mode, the user credential is stored and managed by the SQL Server. We suggest you change the default username – do not use "sa" or any existing account name. You will set the password for the new user here.



   - **Windows Authentication**: Use this authentication method if you want the Cloudpaging Server to connect to the Database Server via a Windows account. When selecting Windows Authentication mode, the user credential is managed by the Windows system. We recommend creating a unique Windows service account to use for this communication. The **Username** must be in the format *domain\user*. For example: *ad.domain.com\John.Doe.*



2. Click **Next. The Database Server Connection** dialog box appears. Here, you will configure how the Cloudpaging Server installer connects to the database server and will create the Cloudpaging database during the installation (see image on the next page)



If SQL Server is installed on the remote server, click the Browse button to locate, or enter the remote SQL Server host name directly. The database authentication option is preselected and disabled based on the choice you selected in the previous step.

- **SQL Server Authentication:** Enter the static TCP port (e.g., 1433), username, and password. Unless modified, the default sa account password is Change!it (case-sensitive).
- **Windows Authentication:** Enter the static TCP port (e.g., 1433). The username for the server installer will be displayed in this screen. The username is predetermined at the moment you run the server installer and therefore can't be modified.

For a distributed system topology, you must use the same database credentials for each machine installation.

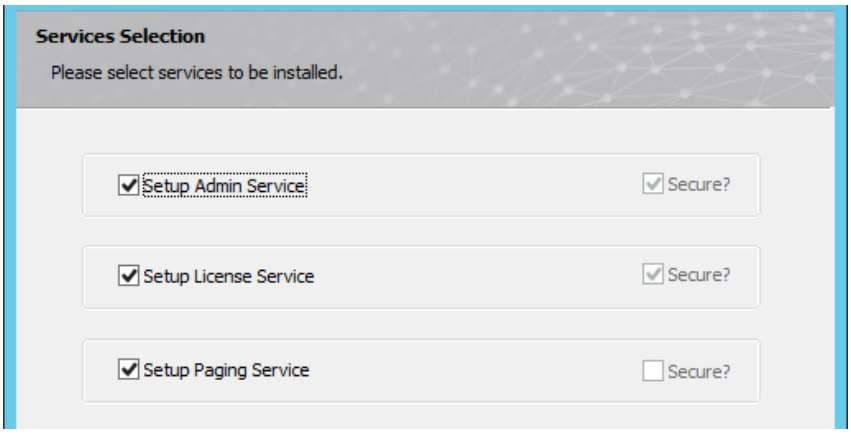
> ***NOTE***
>
> *The username for the Database Server Connection requires a **sysadmin** role to create the database, user account, and set permissions on both.*

3. Click **Next** to continue.

The database server has been configured. In the next step you will configure core services.

Step 7: Core Services Setup

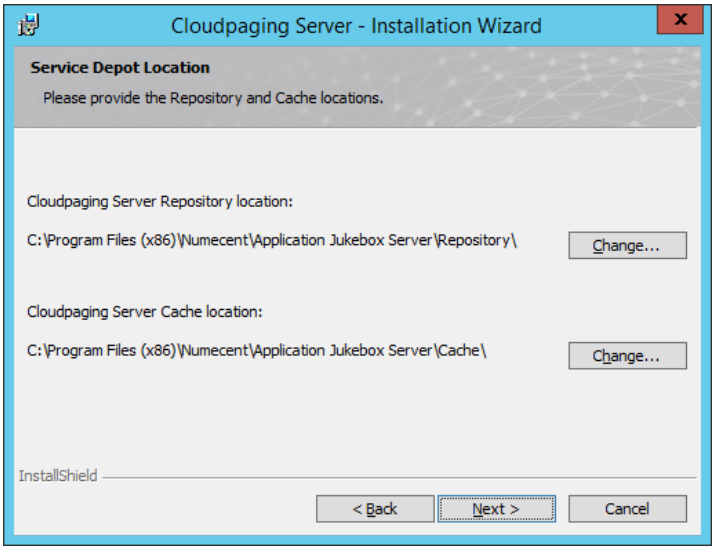
The **Services Selection** dialogue box appears.



1. Select each service you wish to set up. For more information on setting up services based on your topology, see Cloudpaging Server Topology.
   If you choose to install Cloudpaging services on separate machines, you will need to run the Cloudpaging Server installer for each machine.
   - To add an Admin service, select **Setup Admin Service**. The Admin service controls the web interface and is used to configure Cloudpaging Server.
   - To add a License service, select **Setup License Service**. The License service is used by Cloudpaging Player to request an application token.
   - To add a Paging service, select **Setup Paging Service**. The Paging service delivers application pages to Cloudpaging Player.

   Admin and License services will run on the secured port (i.e. 443). Paging service will be on the non-secured port (80).

   Click **Next**. The **Service Depot Location** dialog box appears. The Cloudpaging Server Repository is the folder where appsets are stored after packaging. The Cloudpaging Server Cache is where Cloudpaging services store applications to be paged to clients with Cloudpaging Player.



2. Click **Change** to set the **Repository** and **Server Cache** locations.
   - **Cloudpaging Server Repository location path**: This path must be accessible by all machines. You should use a UNC path or mapped drive path available on all machines. For example: \\[MachineName]\Repository. The machine's WINS name or IP address can be used as long as the machine is accessible. The service account that the Cloudpaging Server runs under should have read permissions to the Repository.
   - **Cloudpaging Server Cache location path**: The cache should be on fast local storage to optimize delivery to end-users. The cache folder path must be the same for all Cloudpaging Servers in your topology. For example, if you choose e:\cache, this path must exist on all machines running Cloudpaging services.
3. Click **Next**

The installer will create the Repository and Cache directories.

Step 8: Final Installation Wizard Procedures

1. Click **Install**. The **Installing Cloudpaging Server** dialog box appears. When the installation process is complete, the **Installation Completed** dialog appears.
2. Click **Finish**. The installation has completed.

The Cloudpaging Server module runs through a Windows service called **Cloudpaging Server Service**. This Windows service runs all Cloudpaging services, including Admin service, License service, and Paging service.

> ***NOTE***
>
> *When you are prompted to restart the **Cloudpaging Server Service**, this is referring to the Windows service..*

Step 9: Replacing the Certificate and Keystores

Both the Admin service and Enterprise Portal add-on install with a temporary SSL Certificate. These should be replaced with valid signed certificates that you obtained from a certificate vendor such as Verisign or Thawte.

For more information on creating certificates and keystores, please visit http://docs.oracle.com/javase/7/docs/technotes/tools/windows/keytool.html

The Keystore provided during installation is located in the Cloudpaging Server installation directory:

- C:\Program Files (x86)\Numecent\Application Jukebox Server\

If a new keystore file is created, the reference to it in the Tomcat server.xml file needs to be updated. By default, the server.xml file is located in:

- C:\Program Files (x86)\Numecent\Application Jukebox Server\core\conf

By default, the keystore password is **changeit**. If your new keystore has a different password it will need to be specified in the server.xml file. You can add the attribute "keystorePass=" to the connector element.

```
<Connector acceptCount="100" clientAuth="false"
disableUploadTimeout="true" enableLookups="false"
keystorePass="CustomPassword" keystoreFile="C:\Program Files
(x86)\Numecent\Application Jukebox Server\Secured.keystore"
maxHttpHeaderSize="8192" maxSpareThreads="75" maxThreads="150"
minSpareThreads="25" URIEncoding="UTF-8" port="443"
scheme="https" secure="true" sslProtocol="TLS" />
```

For additional instructions on updating the keystore, visit our Support article https://support.numecent.com/support/solutions/articles/1000214387 .

Step 10: Verifying the Installation

Database Setup

- Check that StreamDB was created.
- Check that the database user (default: dbjbuser) was added to the database.
- Check that the added database user can login to the database.

Windows Services

- Verify that the Cloudpaging Server Service has been created and started.
- Verify that temporary certificate and keystore were replaced by signed certificate and keystore.

Repository access

- Confirm that the Cloudpaging Server Service (Windows service) Logon as setting has access to the Repository fileshare.

Once Cloudpaging Server is installed, you should access the Admin service web UI and save the link for future access to the site. The Admin service web UI can be accessed from a web browser by visiting https://localhost/jukeboxserver/.

> ***NOTE***
>
> *The Cloudpaging Server might take a few minutes to initialize the deployment of the portals. Wait until the portal shortcuts are available on the desktop before trying to connect to the web portals.*

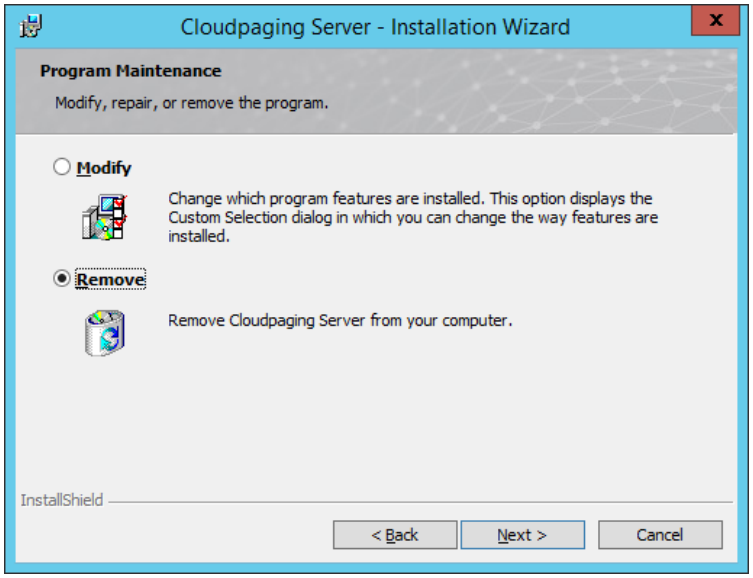
Completing Configuration Setup

After verifying that all the Cloudpaging Server components are properly installed, it is necessary to complete the initial configuration and setup procedures as described in Chapter 4.

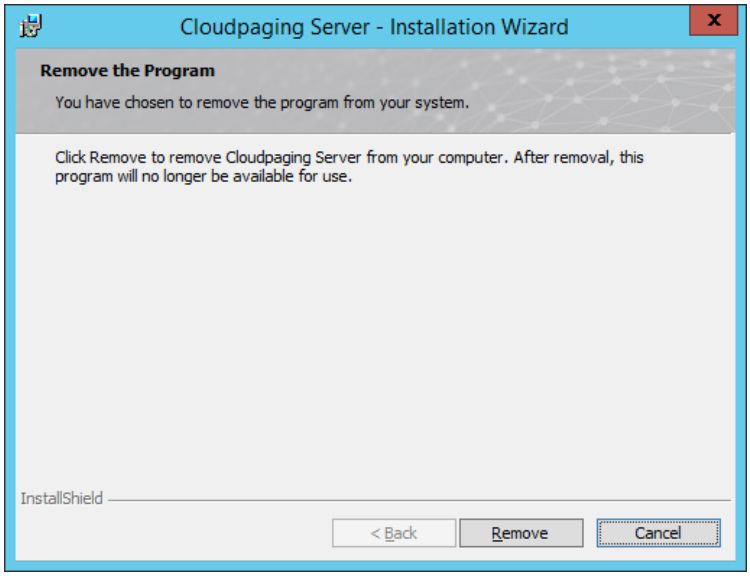
Uninstalling Cloudpaging Server

Uninstalling the Cloudpaging Server will remove the Windows Service and the files under the Cloudpaging Server install directory, by default, C:\Program Files (x86)\Numecent\Application Jukebox Server\.

1. **Backup** contents of the installation directory, if desired.
2. **Double click** on the Application Jukebox Server.exe or Uninstall from the **Add or Remove Program** in the control panel.
3. Select **Remove** and click **Next.**



4. The installer will show the following confirmation dialog for removal.

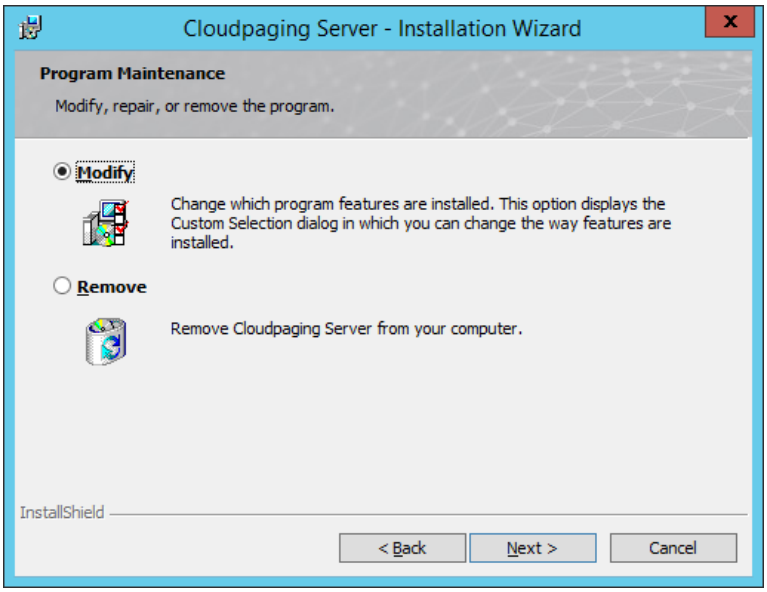


5. Manually delete the cache and repository folders
6. Manually remove the StreamDB database and the dbjbuser created during installation.

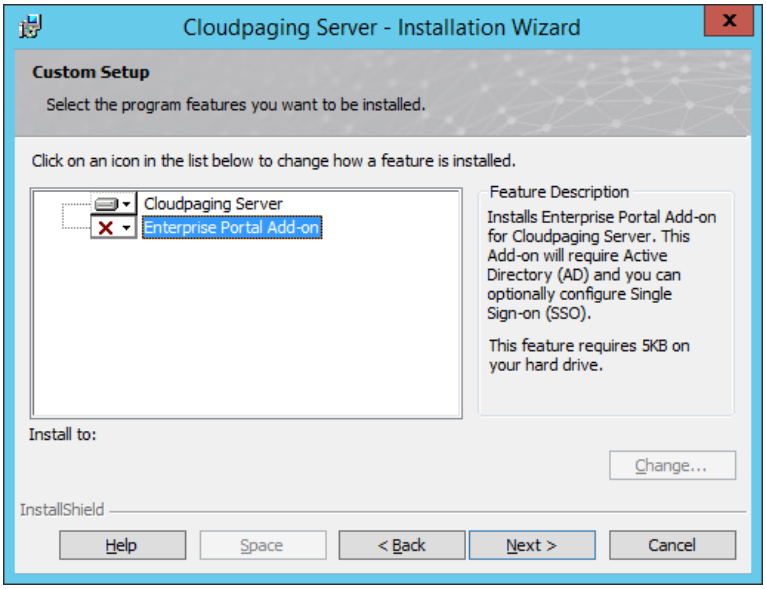
Uninstallation of the Cloudpaging Server is now complete.

Modifying Cloudpaging Server

The Cloudpaging Server installer will go into Modification mode automatically if Application Jukebox Server.exe is launched again after the initial installation.



In Modification mode, features not yet installed can be selected for installation and existing features can be selected for uninstallation.



This dialog is similar to the one in the initial setup. However, the installation directory cannot be changed. Depending on the features selected for modification, different dialogs will be displayed. The installer will not prompt again for Tomcat, JRE, Machine, Ports, and installation directory information.

Upgrading Cloudpaging Server

Use the following procedure for each Cloudpaging Server machine. Multiple machines should be upgraded by taking the first offline and performing the steps below, then restarting the first machine before taking all other machines offline to upgrade. This reduces system downtime and establishes a rollback procedure if something goes wrong with the process.

Upgrading the database schema

Database schema maybe different between the versions of the Cloudpaging Server. If the database schema has been modified, there will be a migration script for each affected feature, (e.g., web_x.x_to_x.x.sql and drm_x.x_to_x.x.sql for Core and the Enterprise Portal Add-On respectively). Running the migration script is required to update schema for new or modified functions to work properly. The script will handle data migration as well when required.

> ***Important***
>
> *This block style identifies information about actions that might cause problems with an application, your local PC system, or your data, including data loss.*

All Cloudpaging Server machines should be taken offline before applying the database migration script. To apply the database changes, execute the database migration script for this version of the server (if there is one).

Upgrading server binaries

When you launch a newer version of **Application Jukebox Server.exe** installer than the currently installed version, the installer will automatically go into Upgrade mode.

Clicking on **Upgrade** in the next dialog will begin the upgrade. As an Apache Tomcat webapps application, the Cloudpaging Server is a collection of webapps, or WAR (Webapps Archive) files, i.e. jukeboxserver.war, jukeboxdrm.war.

This upgrade works first by stopping the Cloudpaging Server Windows service. It then removes the WAR files and their unpacked folders (jukeboxserver and jukeboxdrm) under the C:\Program Files (x86)\Numecent\Application Jukebox Server\core\webapps directory. Once the existing application contents are cleaned, the upgrade will copy new WAR file for each installed features (Core and Enterprise).

This means all installed features will be upgraded at the same time. Finally, the Upgrade will restart Cloudpaging Server service.

> ***Important***
>
> *The Cloudpaging Server requires write access to the folder containing the Cloudpaging Server installer, setup files, and prerequisites. For example, when running installer from a network share with limited access, the Cloudpaging Server installer fails when trying to install Tomcat.*

For more information, please refer to the Cloudpaging Server Upgrade Notes of the version to which you are upgrading.

Chapter 4: Cloudpaging Server Configuration

The configuration procedures in this chapter should be followed immediately after completing the installation and component verification procedures (see Chapter 3, Cloudpaging Server Installation). These setup procedures apply to Cloudpaging Server and should be set up before configuring Enterprise Portal. The following are explained in this section:

- Initial Cloudpaging Server configuration procedures
  - Step 1: Log In to Admin Service Website
  - Step 2: Install the Cloudpaging Server Product license
  - Step 3: Verify Communication Between Services
  - Step 4: Change Admin Password
  - Step 5: Verify Repository Path
- Database Configuration
  - Windows Authentication
  - Active Directory Configuration
  - Log In to Admin Service Website

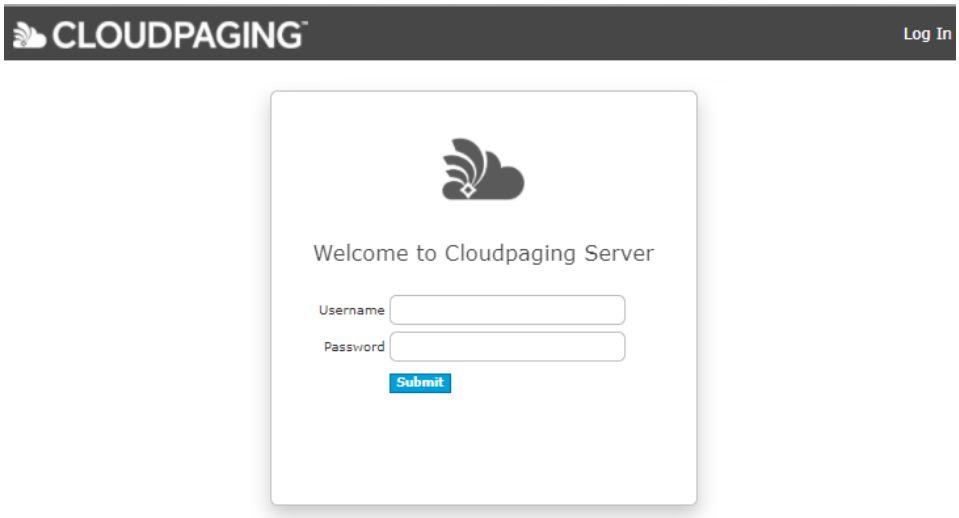
Initial Cloudpaging Server Configuration Procedures

All Cloudpaging services, including Admin service, License service, and Paging service, run within the Apache Tomcat web server and are listed in the Manage Services page.

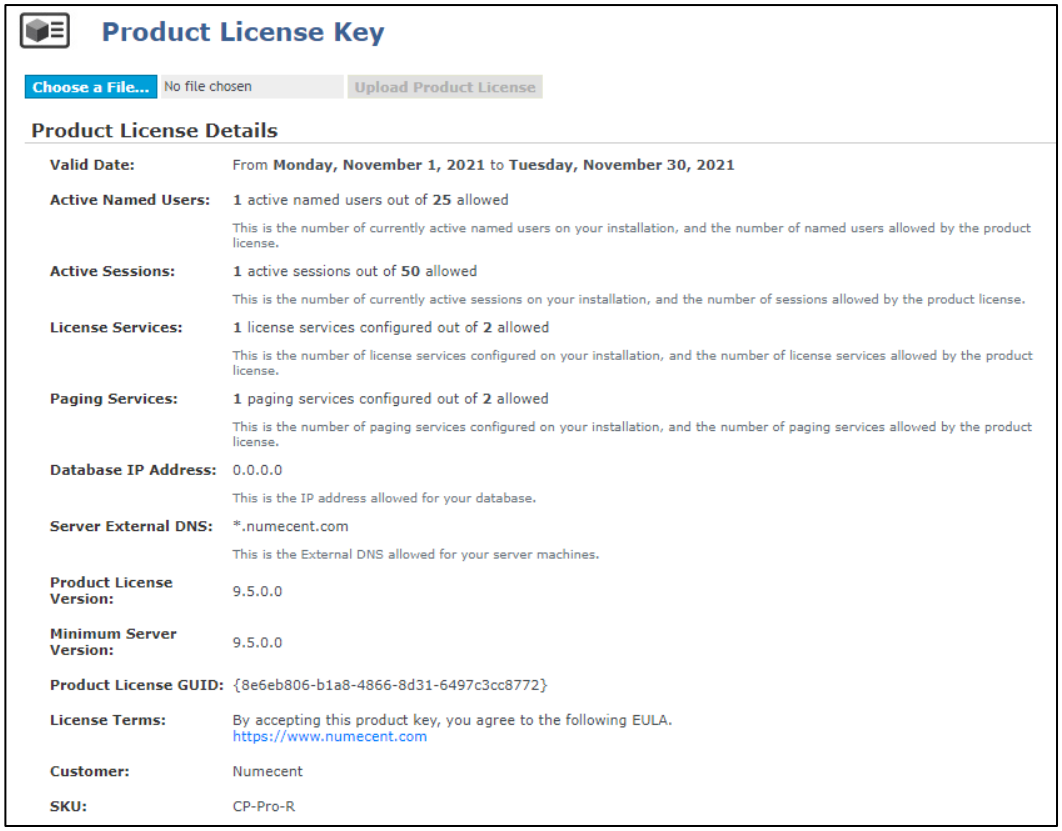
Step 1: Log into Admin Service Website

Once Cloudpaging Server is installed, it is necessary to access the Admin Service website both for initial configuration and for making any future configuration changes.

1. Navigate to the Admin service web page at https://localhost/jukeboxserver/. The Admin login page appears



2. Enter the default Username "**admin**" and password "**password**"
3. Click **Submit** to display the **Product License Key**



4. **Accept** the License Terms

A Product License Key enables various Cloudpaging features and determines parameters like number of users, servers, and duration the server is valid for. This page on the Cloudpaging Server will display product license details as specified in the Product License Key that was issued. For more information on the Product License Key, please visit our Support article https://support.numecent.com/solution/articles/1000259396.

Step 2: Install the Cloudpaging Server Product license

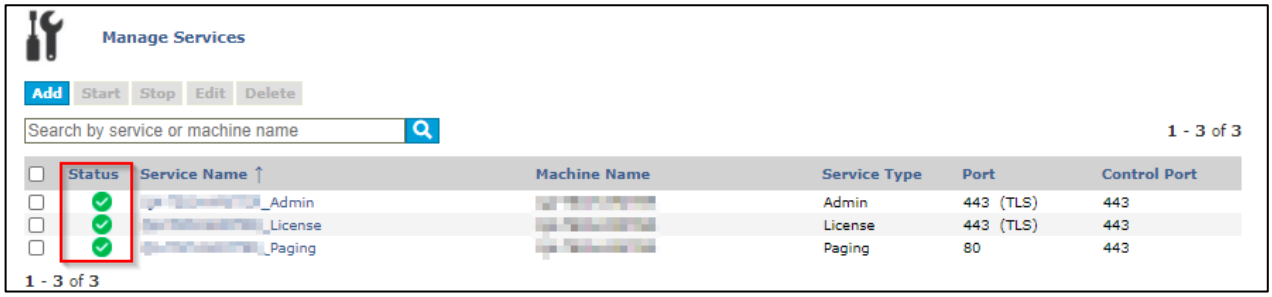
1. On the **Product License Key** page, click **Choose a file.** A dialog box will open to allow you to select the license file from your local system. To request a Product License Key, see Requesting a Product License.

2. Identify the location of the Product License Key that was sent to you and click **Open**.
3. Click **Upload Product License** to display the Product License Details section.

4. Restart the Cloudpaging Server service (a Windows service) on each machine running a Cloudpaging service.

Step 3: Verify Communication Between Services

1. From the **Systems** menu, select **Services** to display the **Manage Services** page.
2. Verify that there is a **Green check mark** next to each service listed on the page



Step 4: Change Admin Password

It is recommended that you change the password for the default Admin account.

1. Click on the name (Admin) displayed in the top right hand corner and select Change Password to display the Change Password page.

2. Enter the current password (the default password is “password”) in the **Current Password** field.
3. Enter a new password in the **New Password** field.
4. Re-enter the new password in the **Confirm Password** field.
5. Click **Update**

The password for the Admin account has been changed. We recommend that you create individual accounts for each administrator responsible for managing the Cloudpaging Server. See Creating a User for more information on creating administrative accounts.

Step 5: Verify Repository Path

The Cloudpaging Server uses a folder called Repository to add and publish applications. This folder must be accessible to all machines containing Cloudpaging services.

By default, the path for the Repository folder is **C:\Program Files (x86)\Numecent\Application Jukebox\Repository**

To test the repository path:

1. Upload an appset packaged with the Cloudpaging Studio to the repository using the Admin web page. See Adding an Application for more information.
2. Verify that the appset has been published to all Paging servers on the **Edit Application** page.

The initial Cloudpaging Server setup configuration procedures have been completed. Please read Chapter 4 to continue adding applications, license policies, and learn more about administering the Cloudpaging Server.

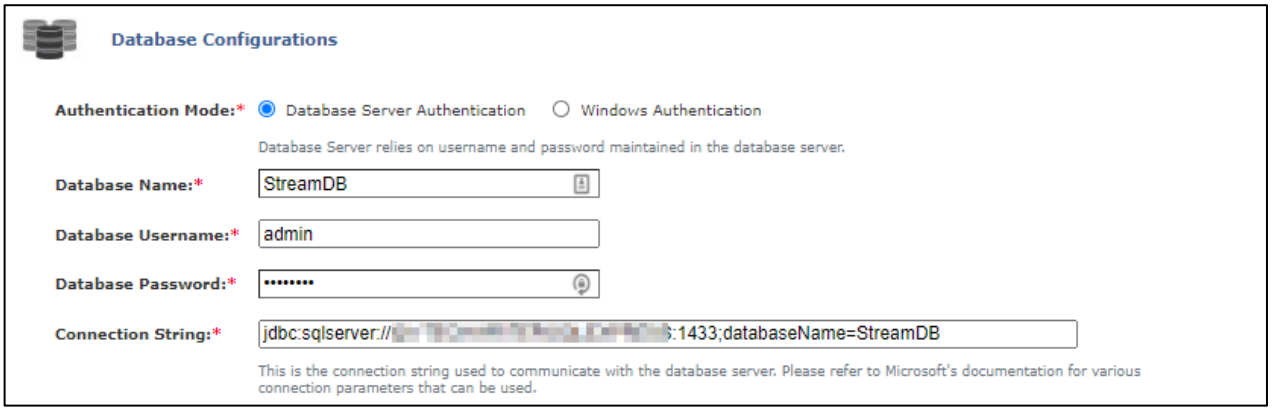
Database Configuration

During installation, the database configuration is setup and stored encrypted in a file called **dbConfig.xml.** If this database information is incorrect, corrupted, or missing, the server administrator will be asked to configure the database. The configurations can be changed at any time by the administrator using the **Database Configurations** page under the **Systems** menu.

> ***NOTE***
>
> *If the Cloudpaging Server is unable to reach the database, you will be presented with an error message and will not be able to login to manage the server. However, you can access the Database Configurations page by accessing https://localhost/jukeboxserver/showDatabase.do locally from the machine where the Cloudpaging Server is installed.*

To change the database configurations:

1. From the **Systems** menu, select **Database Configurations** to display the **Database Configurations** page. The following options are displayed on this page:



- **Authentication Mode:** Configures how the Cloudpaging Server authenticates with the database. Database Server Authentication uses credentials stored in the database. Windows Authentication uses local or domain Windows account to authenticate.
- **Database Name:** Refers to the name of the Cloudpaging Server database that was configured during installation. The default is StreamDB.
- **Database Username:** Refers to the user account configured to be the Cloudpaging Server database owner. By default, dbjbuser is used when authentication mode is set to Database Server Authentication and Local System is used when authentication mode is set to Windows Authentication.
- **Database Password:** Corresponds to the password for the Database Username account when authentication mode is set to Database Server Authentication.
- **Connection String:** Connects the Cloudpaging Server to the database using the specified URL. The default value is the format used for Microsoft SQL Server. Please see the chart of examples for the servers we support.

Database	Connection String
MS SQL Server	jdbc:sqlserver://10.10.1.2:1433
MS SQL Server Express	jdbc:sqlserver://127.0.0.1\SQLEXPRESS:1433

2. Click **Update.** You will receive a confirmation message that the database configurations have been successfully updated.

Windows Authentication

During installation, you can have the database connection configured to use integrated Windows Authentication. If you choose to change the database authentication method post installation, then follow these steps to configure integrated Windows Authentication. This feature allows the Cloudpaging Server to communicate with Microsoft SQL Server using Windows credentials. This approach simplifies deployment of the Cloudpaging Server by eliminating the need to create or maintain separate user and login accounts in SQL Server.

When enabling the Windows Authentication, you can use either local or domain Windows accounts to authenticate the Cloudpaging Server with SQL Server. This depends on whether the SQL Server is installed on the same Windows server as the Cloudpaging Server or a different Windows server.

This section describes how to enable Windows Authentication using a local or remote SQL server. Before proceeding, make sure you have:

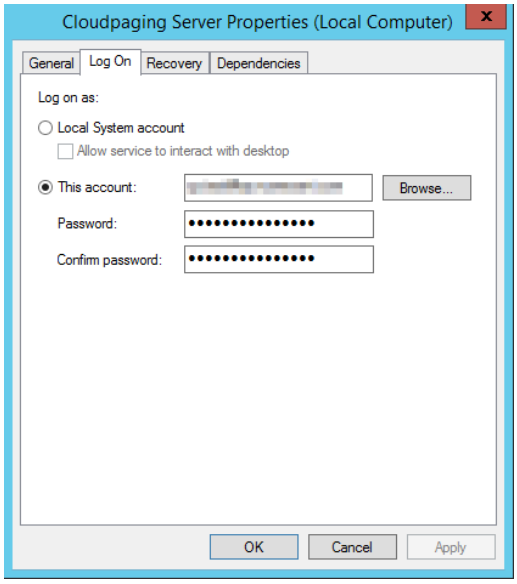
- A local or domain Windows user account that has access to the SQL Server instance with a sysadmin role, or db_owner role to the database that the Cloudpaging Server will use (default is StreamDB).
- The ability to access SQL Server using the sysadmin login.

To enable Windows Authentication with SQL Server:

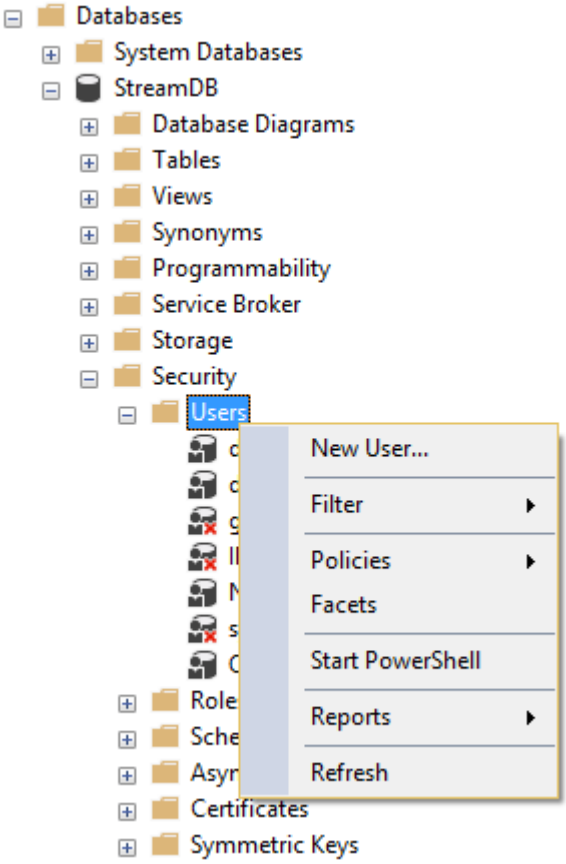
If your Microsoft SQL Server and Cloudpaging Server are installed on a different Windows Server machine, reconfigure Cloudpaging Server Windows service's Log on as account in Service Control Manager.

> ***NOTE***
>
> *For Windows Authentication features to work with a remote SQL Server, all participating Windows Server machines must be joined to the same domain according to the following procedure.*

1. Stop Cloudpaging Server service.
2. In the Windows Service Control Manager, find Cloudpaging Server service.
3. Open the **Properties** dialog of Cloudpaging Server service, and go to the Log On tab.
4. Under the **Log On** tab, switch the **Log on as** radio button from **Local System account** to **This account**.
5. Specify a domain Windows domain account and enter the password. This account should have local administrative privileges.

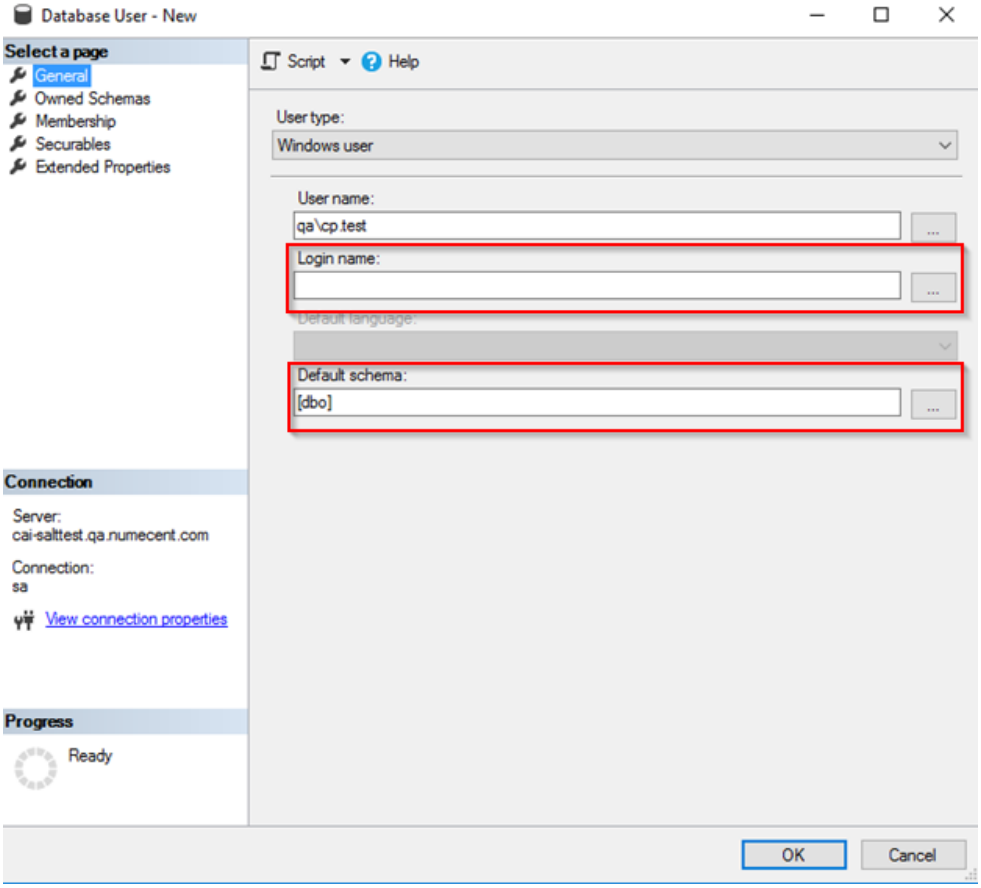


6. Connect to SQL Server using Management Studio and create a new User account under StreamDB in Object Explorer.

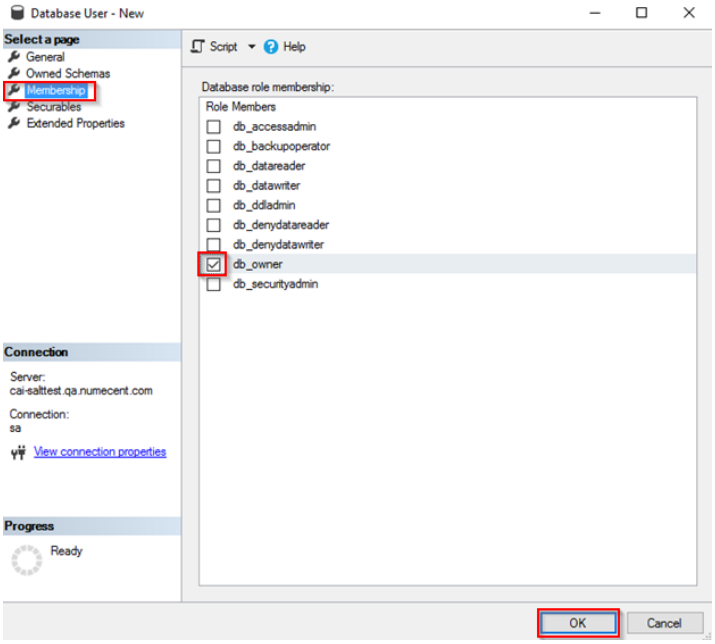


7. In the **Database User - New** dialog box, select **Windows user as User Type**, and then select the same Windows account you used in the Log on as configuration step.

8. Leave **Login name** empty because this user does not need instance-level login account for security purposes. Optionally, you can select **[dbo]** as the default schema for the new user.



9. While in the **Database User - New** dialog, switch to **Membership** and check the **db_owner** check box to assign membership.
10. Click **OK.**



You have now configured SQL Server to use Windows Authentication with Cloudpaging Server.

11. Start the Cloudpaging Server service.
12. Go to the Admin service web page and from the **Systems** menu, select **Database Configurations** to display the Database Configurations page.
13. Switch **Authentication Mode** from Database Server Authentication to **Windows Authentication**. The Windows account you specified in Log on as appears automatically in the **Database Username** field. This is the account used to run the Cloudpaging Server.

14. Review other fields on the page, and then click the **Configure Database** button. The following message appears: **Database is configured successfully**. P**lease restart Cloudpaging Server service** for new changes to take effect.
15. Restart Cloudpaging Server service.

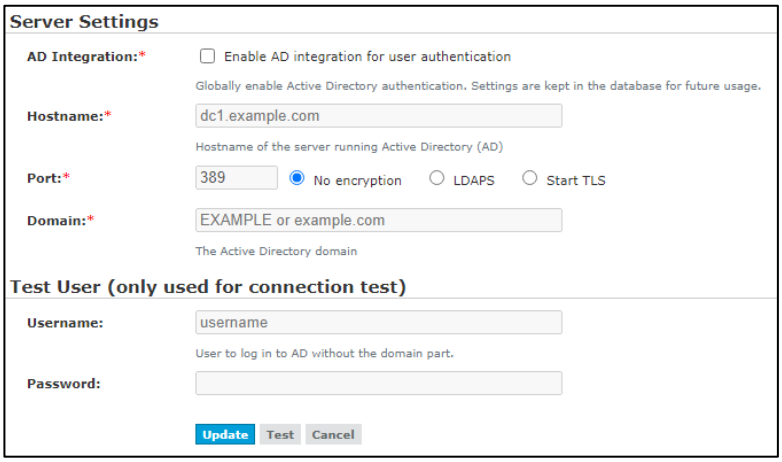
Your Cloudpaging Server now uses Windows Authentication mode.

> ***NOTE***
>
> *You cannot change the values in the database user name and password fields on the Cloudpaging Server portal. To change the user used to authenticate with the SQL Server in Windows Authentication mode, repeat the steps above and assign the database ownership permissions appropriately.*

Active Directory Configuration

The **Active Directory Configuration** page allows you connect to the appropriate Active Directory server on your network or in the cloud.

From the Systems menu, select **Active Directory** to display the **Active Directory Configurations** page.



Server Settings

- **AD Integration:** Control to enable or disable the integration with active Directory.
- **Hostname:** The host name of your Active Directory server. Example: server.ad.domain.com, ad.domain.com.
- **Port:** The port on which your directory server is listening. Examples: 389, 636 (for TLS).
- **Encryption Options:** Select the type of encryption: No encryption, LDAPS, Start TLS.
- **Domain:** The Active Directory domain. For example, example.com.

Test User

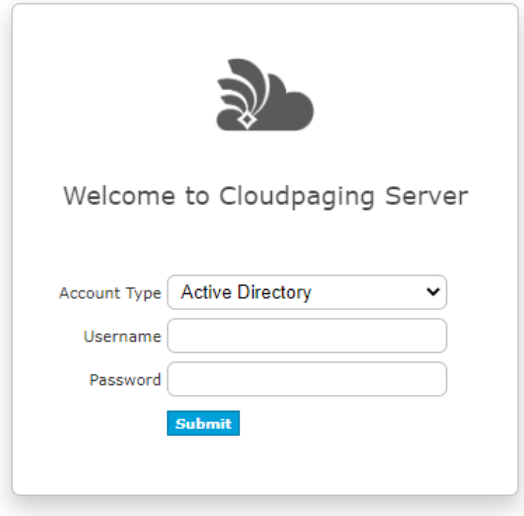
The following information is only used to test the Active Directory supplied above and ensure the integration work well. This information is not stored by Cloudpaging Server.

- **Username:** The name of the user that will be used temporarily to test the connection to Active Directory.
- **Password:** The password of the user specified above.

Log into Admin Service Website

Once you have configured Cloudpaging Server with Active Directory, then you will be able to login using your Active Directory domain credentials.

1. Navigate to the Admin service web page at https://localhost/jukeboxserver/. The Admin login page appears.



2. Select the **Account type** as “Active Directory”
3. Enter your Active Directory domain username and password.
4. Click **Submit**

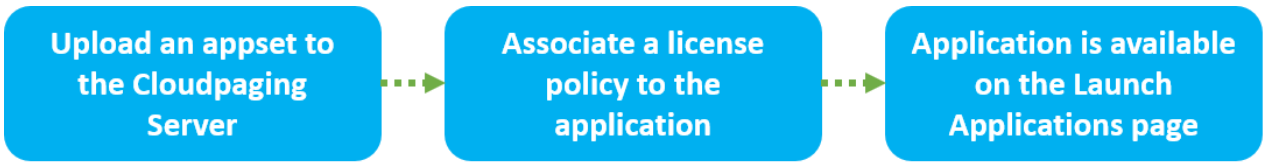
Chapter 5: Managing Application Deployments

This chapter provides the procedures for managing applications and license policies from the Admin service web UI. The following are explained in this section:

- Applications
  - Adding an Application
  - Editing an Application
  - Creating a License Policy for an Application
  - Viewing License Policies for an Application
  - Deleting an Application
- License Policies
  - Creating a License Policy
  - Editing a License Policy
  - Upgrading a License Policy
  - Cloning a License Policy
  - Viewing Sessions of a License Policy
  - Deleting a License Policy
- Upgrades
  - Creating an Upgrade
  - Editing an Upgrade
  - Deleting an Upgrade
- Sessions
- Testing Applications

Applications

Cloudifying applications is the process of converting a software application to a format that can be paged to the Cloudpaging Player. Administrators use the Cloudpaging Studio to package required components and configuration information of an application into a compressed, encrypted file called an appset. The appset is then uploaded to the Cloudpaging Server, assigned license policies and is ready to be deployed to end users. Once an appset has been uploaded to the Cloudpaging Server it is referred to as an application.



NOTE: Once an appset has been uploaded to the Cloudpaging Server it is referred to as **an application**.

Adding an Application

To enable users to launch applications cloudified using the Cloudpaging Studio, it must be added to the Cloudpaging Server. There are two methods to add an appset.

- **Upload from local system**: You can upload an appset (.stp file) from any location accessible from your machine.
- **Import from server repository**: You can add an appset that has already been copied over to the Cloudpaging Server's repository.

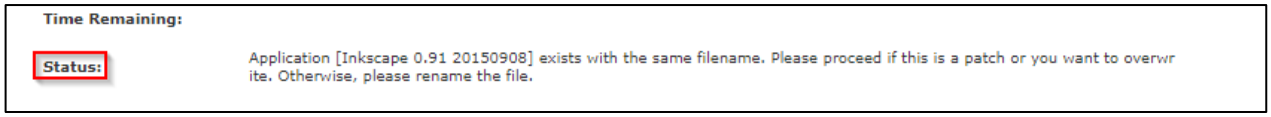
You can upload or import a new appset to the Cloudpaging Server or add a patch version to an appset already in the repository.

To upload from local system:

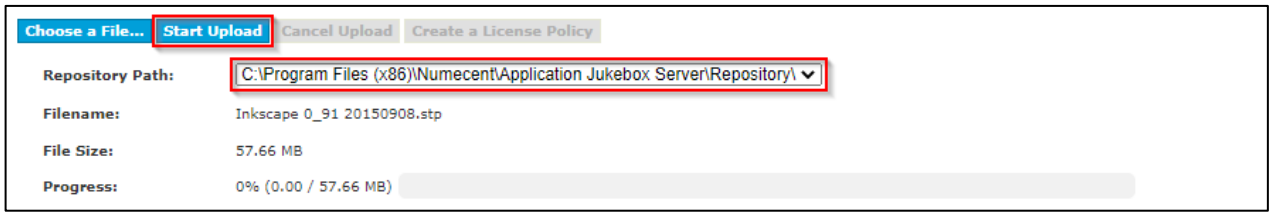
1. From the **Deployments** menu, select **Applications** to display the **Manage Applications** page.
2. Click the **Add Application** drop-down and choose **Upload from local system** to display the **Upload Application** page.

3. Click on **'Choose a File...'**. A dialog box will open to allow you to select the .stp file from your local system.

4. Choose the .stp file and click on Open. The name of the file will appear in the Filename field. If this application already exists in the repository, there will be a message indicating that the upload process will patch the existing appset.

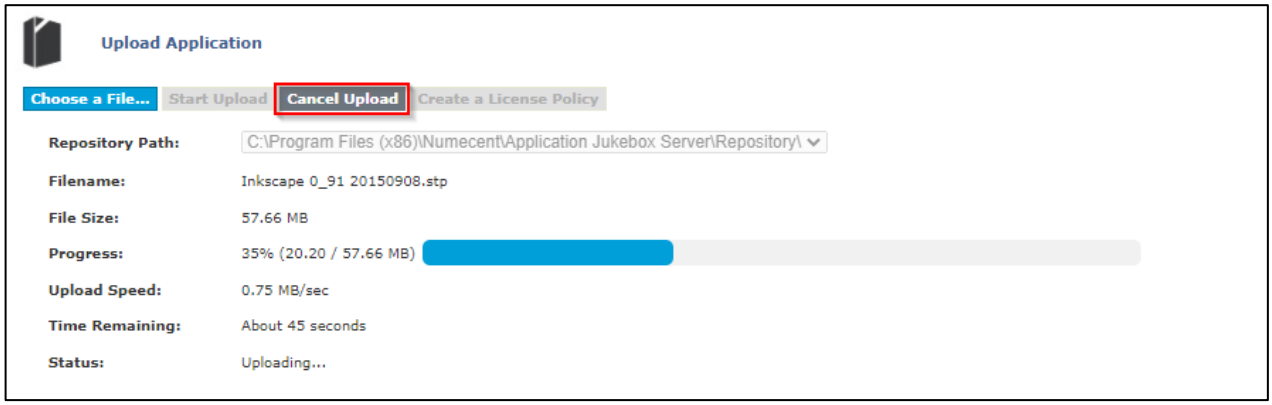


5. Choose the Repository Path to which the appset should be uploaded.
6. Click on Start Upload.

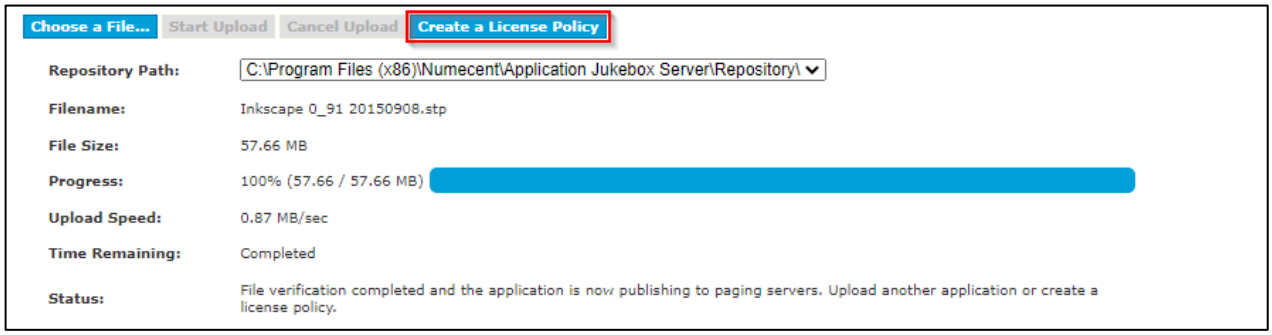


The appset upload process will commence.

The **Progress, Upload Speed, Time Remaining,** and **Status** options provide additional information on the status of the upload. You can also **Cancel** an upload in progress.



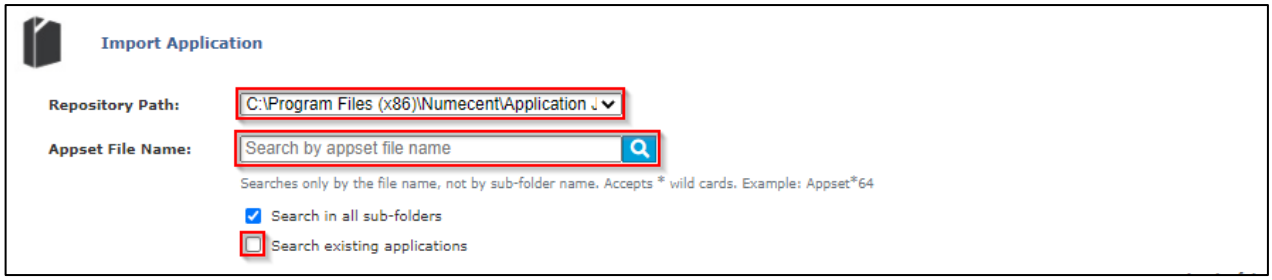
Once the upload has completed, the **Time remaining** field will change to **Completed**. You may select another application to upload or click on the **Create a license policy** option to create a license policy for the application you just uploaded.



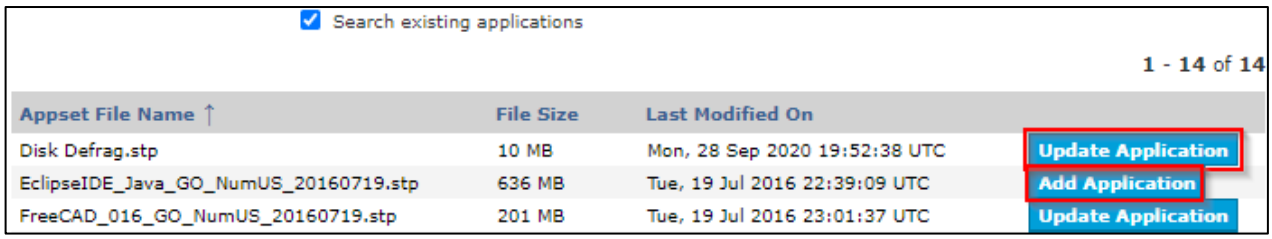
To import from server repository:

1. From the **Deployments** menu, select **Applications** to display the **Manage Applications** page.
2. Click the **Add Application** drop-down and choose **Import from server repository** to display the **Import Application** page. A list of applications is displayed.

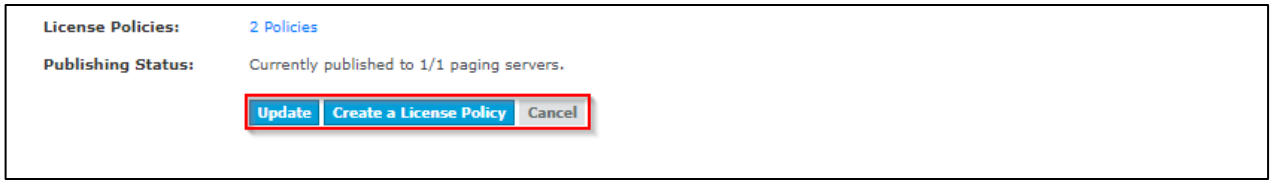
3. Choose the **Repository Path** where the application is located. You can also search for an appset by entering its file name in the **Appset File Name** field. If you wish to import a patch version for an existing application, select the **Search existing applications** check box.



4. Click on the **Add Application** button next to the appset you wish to add to the Cloudpaging Server. For adding a patch version, click on the **Update Application** button. The **Edit Application** page is displayed with a confirmation message.



5. You may edit the application's **Title, Description**, and **Publisher Name** and click on **Update** or click on the **Create a license policy** option to create a license policy for the application you just imported.



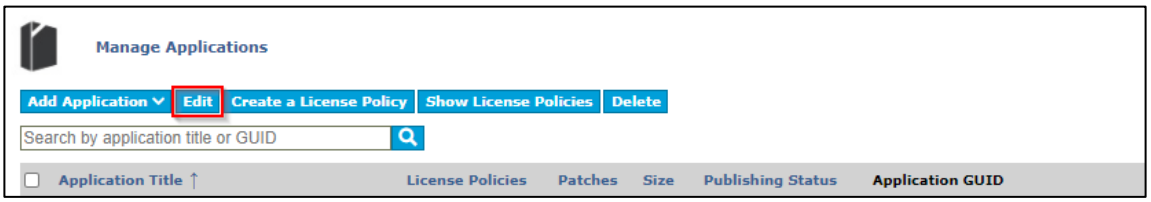
Once an application is uploaded or imported to the Cloudpaging Server, it will be automatically published to all available Paging Servers. If there are any errors during the publishing process, an error message will be displayed along with the name of the Paging Server(s) where the application could not be published.

Editing an Application

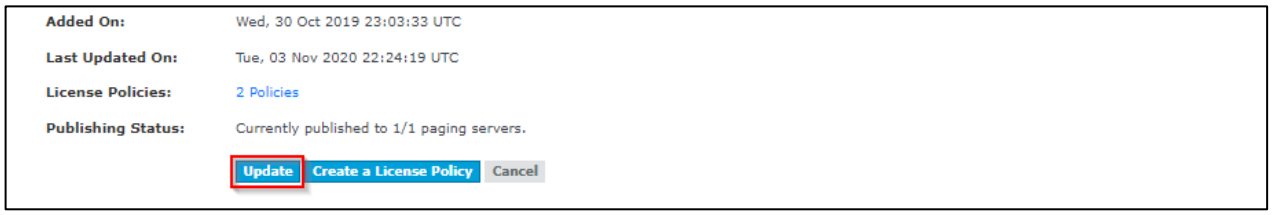
You may edit the application's title once it has been uploaded to the Cloudpaging Server. For example, you could change the application title from **Office 2016** to **Office 2016 Enterprise Edition**. In addition to the title, an application's Description and Publisher name can also be modified on the Cloudpaging Server.

To edit the default information of an application:

1. From the **Deployments** menu, select **Applications** to display the **Manage Applications** page.
2. Select the application you wish to edit and click the **Edit** button to display the **Edit Application** page.



3. Details of the application that were imported from the .stp file are displayed on the page. Only **Application Title**, **Description**, and **Publisher Name** can be modified. Make the desired changes and click **Update**.



The Command Line, Working Folder, and Windows Compatibility fields cannot be edited in the application but can be overwritten when creating a license policy. See Creating a License Policy for more information.

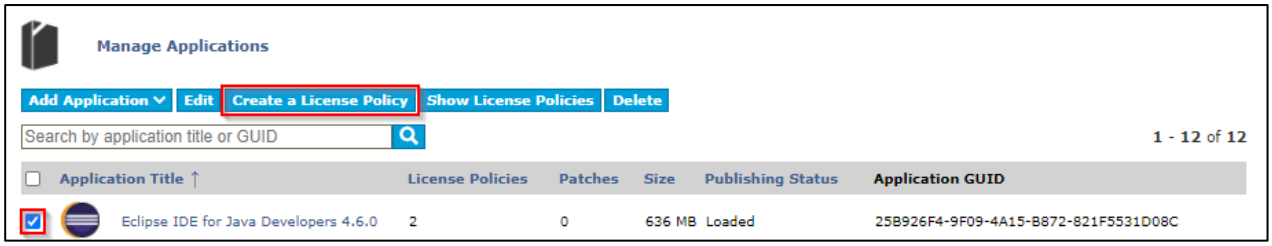
> ***NOTE***
>
> *The application's description is displayed in the Launch Applications page. Once an application's description has been updated, you will need to edit the existing license policies associated to the application and click on Update to update the description in the Launch Applications page. Past usage information in Usage reports will not be affected by changes made to the application.*

Creating a License Policy for an Application

You may create a new license policy for an application from the Manage Applications page.

To create a new license policy:

1. From the **Deployments** menu, select **Applications** to display the **Manage Applications** page.
2. Select the application for which you wish to create a license policy and select the **Create a license policy** option.

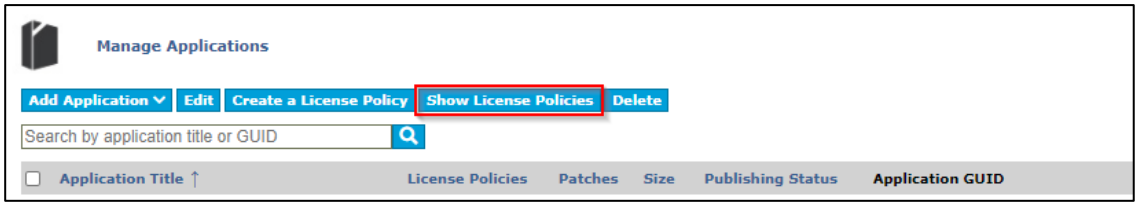


3. Enter or update the license policy options. For more information on license policy options, see Creating a License Policy.
4. Click on Create.
   The license policy has been created.

Viewing License Policies for an Application

You can view all the license policies for an application from the **Manage Applications** page

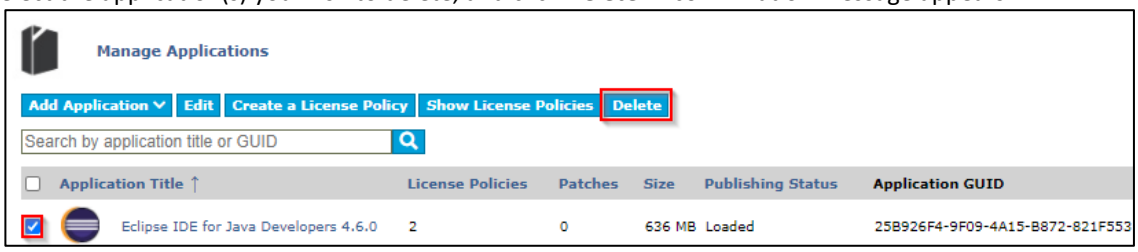
1. From the **Deployments** menu, select **Applications** to display the **Manage Applications** page.
2. Select the application for which you wish to view all license policies and click **Show License Policies**.



The **Manage License Policies** screen will appear with all license policies for that application. For information on additional options on the **Manage License Policies** page**,** see License Policies.

Deleting an Application

1. From the **Deployments** tab, select **Applications** to display the **Manage Applications** page.
2. Select the application(s) you wish to delete, and click **Delete**. A confirmation message appears.

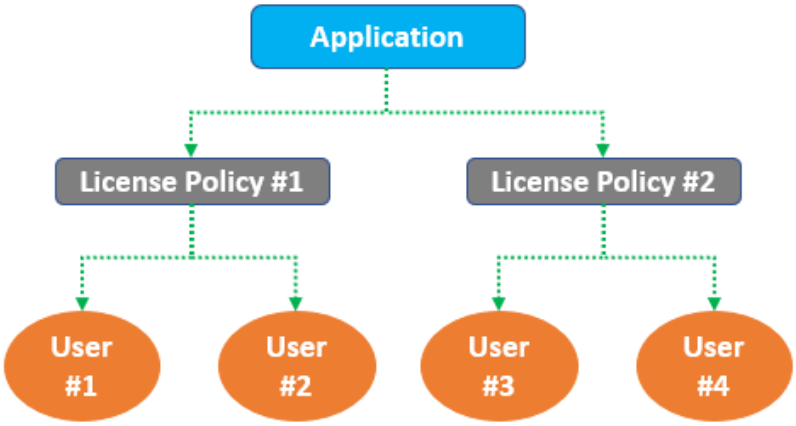


3. Click **OK** to continue with the deletion process or click **Cancel** to abort the deletion. The application is no longer listed on the **Manage Applications** page.

> ***Important***
>
> *Deleting an application will disrupt sessions currently in use (see Sessions to find out how to identify if an application is in use).*

License Policies

An application's *license policy* dictates the allowed usage of an application. After adding an application, one or more license policies can be created. Every application added to the Cloudpaging Server must be associated with at least one license policy in order to make it available for deployment to the user's machine.



Creating a License Policy

The license policy specifies the conditions under which the application can be used defined by settings, such as policy type, total number of seats, offline duration (fixed licenses only), etc.

To create a license policy:

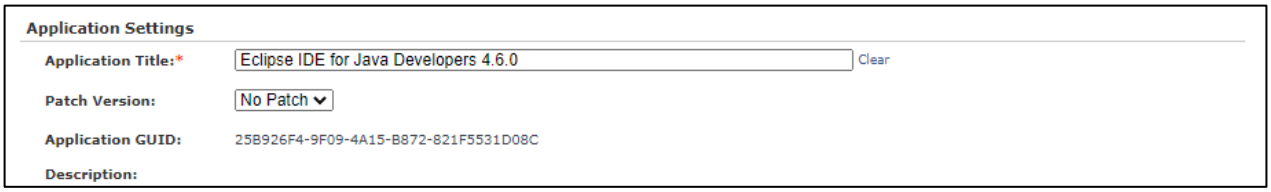
1. From the **Deployments** menu, select **License Policies** to display the **Manage License Policies** page.
2. Click **Create** to display the **Create a License Policy** page.

3. Fill in the required fields, or choose the desired settings for this license policy. See License Policy Options for more information on these options.
4. Click **Create**.
   The license policy has been created.

License Policy Options

There are four sections on this page.

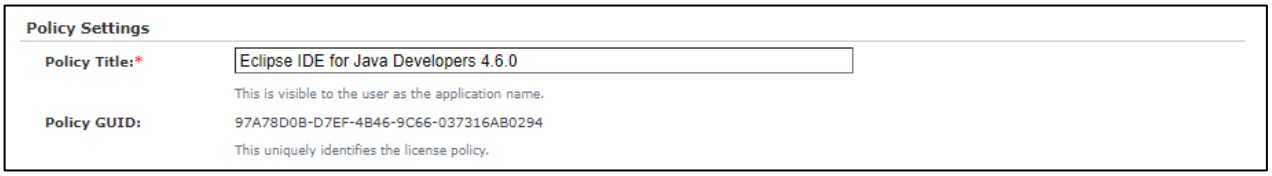
Application Settings



- **Application Title:** Select the application name from the drop-down list which will be associated to this license policy.

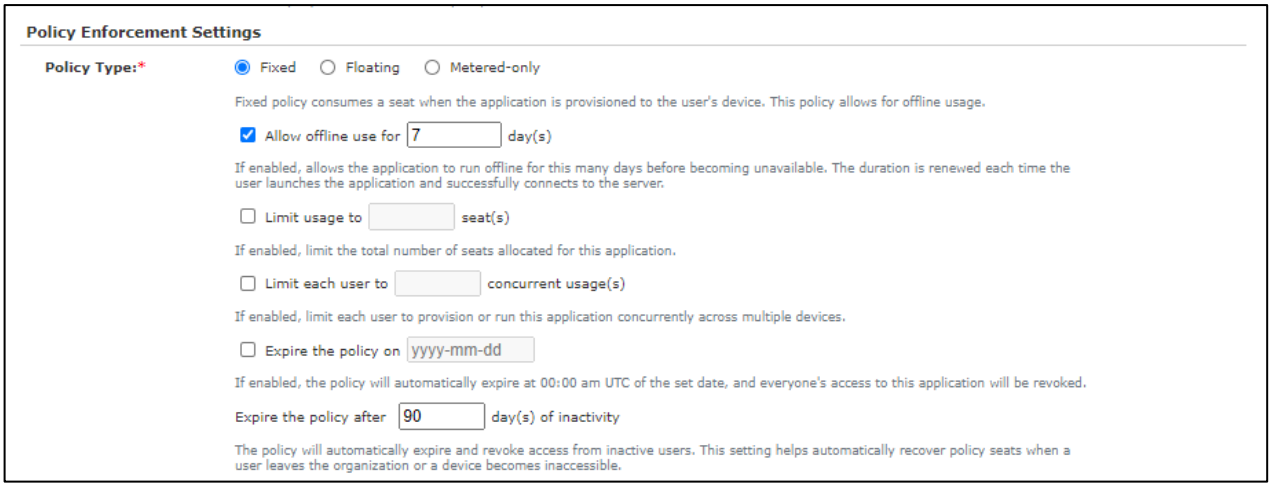
- **Patch Version**: Select the application patch from the drop-down list to associate it with the license policy. **No Patch** will be displayed if the application does not have an available patch version.
- **Application GUID**: This is the application's auto-generated unique identifier and cannot be modified.
- **Description**: This is the description of the application as entered on the **Manage Applications** page. This description will be displayed on the **Launch Applications** page.

Policy Settings



- **Policy Title:** The name entered here will be visible to users as the name of the application. The default value is same as the Application Title field but can be changed.
- **Policy GUID:** The is the license policy's auto-generated unique identifier and cannot be modified.

Policy Enforcement Settings

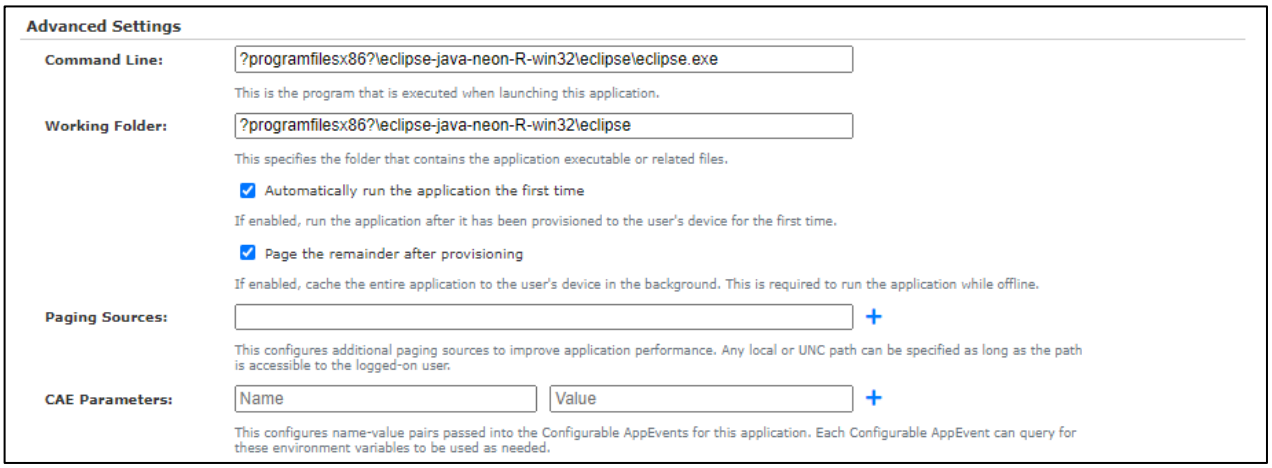


- **Policy Type:** There are three types of license policies:
  - **Fixed (Default):** An application with a fixed license policy will consume a seat as soon as the application is added to the Cloudpaging Player, regardless of whether the application is running or not. Fixed license policies can be used offline.
  - **Floating:** An application with a floating license policy will consume a seat while the application is in a Running state in the Cloudpaging Player. Floating license policies cannot be used offline.
  - **Metered-only:** Similar to a Fixed license policy, a seat is used when the application is provisioned to the user's device. However, this license policy does not expire or limit offline usage.
- **Allow offline use:** (For fixed license policies only) If enabled, this option allows the application to run offline for the specified number of days before becoming unavailable. The duration is renewed each time the user launches the application and successfully connects to the Cloudpaging Server. The default duration is **7 days**.
- **Limit usage of seats:** If enabled, this option will limit the total number of seats allocated for this application.

- **Limit per user concurrent usages:** If enabled, this option restricts the number of computers on which each user can simultaneously run the application. For example, to limit a user from running the application on more than five devices simultaneously, set the value to five concurrent usages.
- **Expire policy on:** If enabled, you can specify a date (in 'yyyy-mm-dd' format) when the policy will automatically expire. Once an application is expired, it will be removed from the Cloudpaging Player for all users who have paged this application.
- **Expire after day(s) of inactivity:** This option allows you to revoke a user's access to an application if they have not used it for the specified number of days. If the session has not been in the 'Running' state for the specified duration, the license policy will expire for that user and the application will be removed from the Cloudpaging Player. The default duration is 365 days.

Advanced Settings

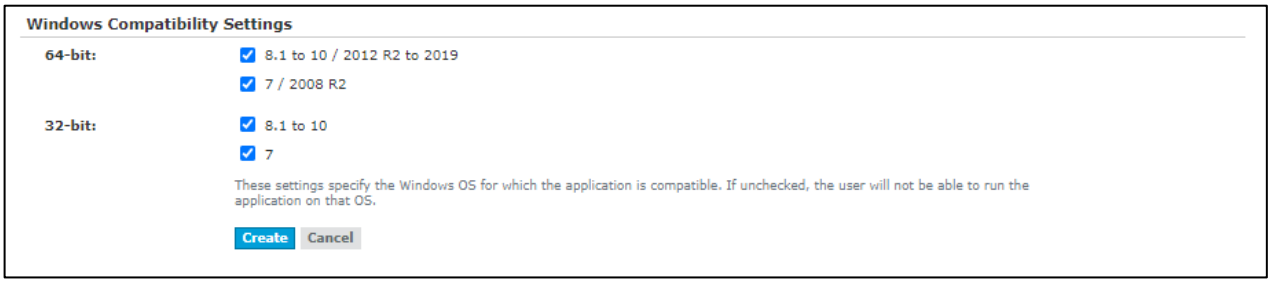
Some options set in the Cloudpaging Studio for an appset can be overwritten by the license policy in the Cloudpaging Server.



- **Command Line:** This field is used to set the target of the application. This target will be launched by the Cloudpaging Player when the user clicks the **Launch** button, when **Automatically run the application the first time** is configured, or when the start command is issued via our Cloudpaging Player APIs.
- **Working folder:** This field is used to set the Current directory for the process set in the Command Line. However, this only applies when the application is launched from the Cloudpaging Player. If the application is launched outside of the Cloudpaging Player (for example, through the desktop shortcut), the Working folder setting will be ignored.
- **Automatically run the application the first time:** This optional setting will launch and run the application automatically after it has been provisioned to the user's device for the first time. This setting is enabled by default.
- **Page the remainder after provisioning:** This optional setting will page the entire application to the Cloudpaging Player in the background after the application is provisioned. This is required to run applications when the device is offline.
- **Paging Sources:** You can configure additional paging sources to improve the application's performance. Local or UNC paths can be specified but the paths must be accessible to the user. There is no limit on the number of additional paging sources, however, we recommend no more than one or two additional paging sources.
- **CAE Parameters:** This configures name-value pairs passed into the Configurable AppEvents for this application. Each Configurable AppEvent can query for these environment variables to be used as needed.

Windows Compatibility

These settings specify the supported Operating Systems for an application. A user is only allowed to run the application on the Operating Systems selected in the license policy.



Editing a License Policy

All license policy settings except Application Settings and Policy Type can be modified by editing the policy. Changes to a license policy do not take effect on applications already provisioned to an end-user's device until the application is relaunched.

To edit a license policy:

1. From the **Deployments** menu, select **License Policies** to display the **Manage License Policies** page.
2. Select the license policy you wish to edit, and click **Edit** to display the **Edit License Policy** page.

3. Make your desired changes, and click **Update**.

The license policy you selected has been updated.

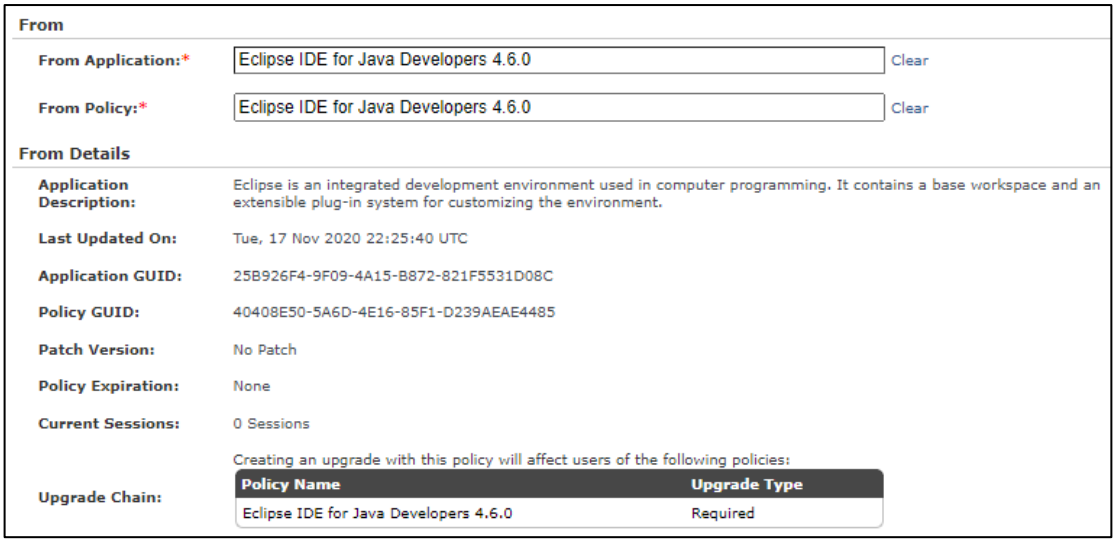
Upgrading a License Policy

License policies can be assigned an upgrade path from the **Manage License Policy** page.

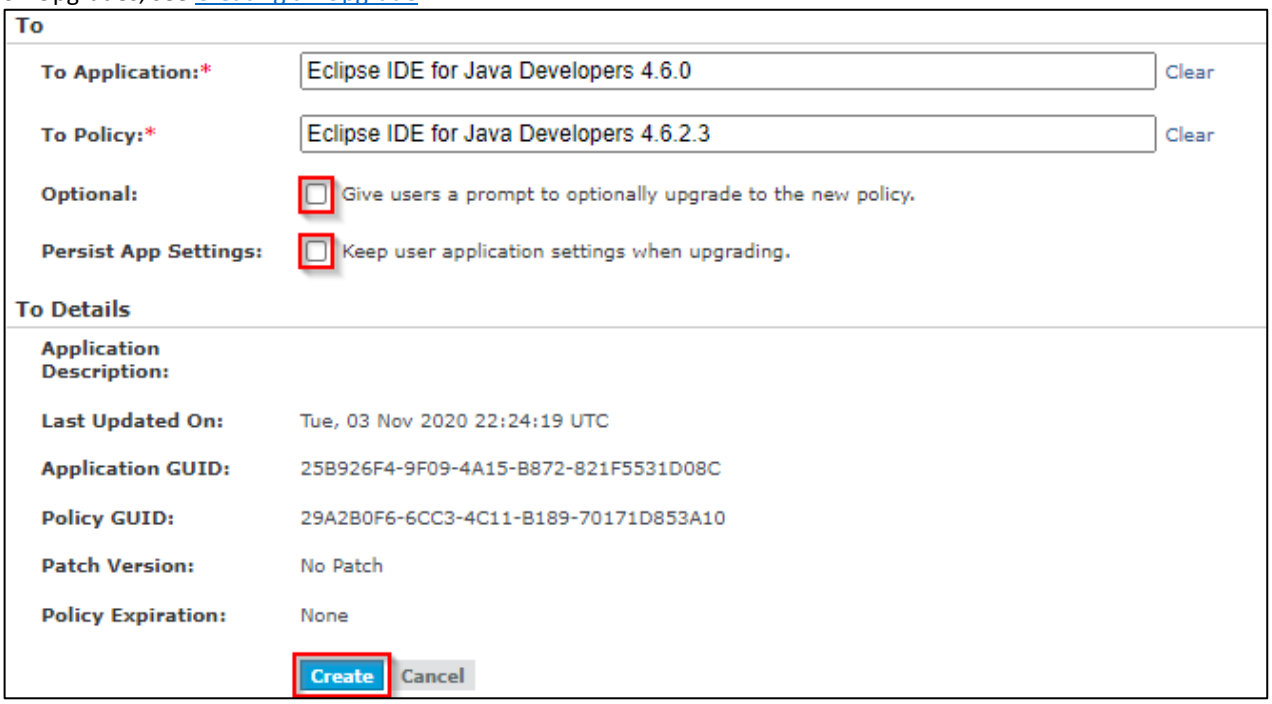
To upgrade a license policy:

1. From the **Deployments** menu, select **License Policies** to display the **Manage License Policies** page.
2. Select the license policy you wish to upgrade, and click **Upgrade** to display the **Add Upgrade** page.

3. The **From Application** and **From Policy** are automatically selected based on the license policy selected.



4. Update the **To Application** and **To Policy** and click on **Create** to create the upgrade. For more information on Upgrades, see [Creating an Upgrade](#).



The upgrade will be created.

Cloning a License Policy

Cloning a license policy allows administrators to quickly create similar license policies for an application or apply the same license policy settings to a new application.

To clone a license policy:

1. From the **Deployments** menu, select **License Policies** to display the **Manage License Policies** page.
2. Select the license policy you wish to clone and click **Clone** to display the **Create a License Policy** page.

3. The cloned license policy will have the text **CLONE** prepended to the original license policy's title.
4. Make the desired changes to the license policy and click **Create**.

A new license policy has been created.

Viewing Sessions of a License Policy

Administrators can view all the sessions for a license policy directly from the **Manage License Policy** page.

1. From the **Deployments** menu, select **License Policies** to display the **Manage License Policies** page.
2. Select the license policy you wish to view all sessions for and click **Show Sessions**.

The **Manage Sessions** page will appear with all current sessions for that license policy. For options on the **Manage Sessions** page**,** see Sessions.

Deleting a License Policy

To delete the license policy:

1. From the **Deployments** menu, select **License Policies** to display the **Manage License Policies** page.
2. Select the license policy you wish to delete and click **Delete**. A confirmation message appears.

3. Click **OK** to continue the deletion or click **Cancel** to abort the deletion. The license policy is no longer listed on the **Manage License Policies** page.

> ***Important***
>
> *Deleting a license policy will disrupt application sessions currently in use (see Sessions to find out how to identify if a license policy is in use).*

Upgrades

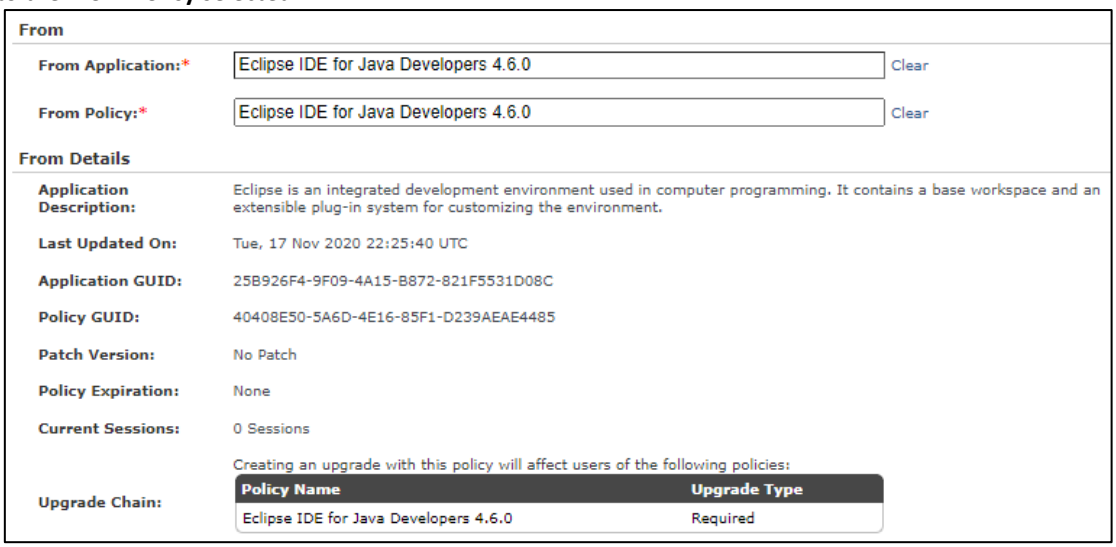
The Cloudpaging Server makes it simple to distribute upgrades or even replace an application with a different application. The administrator has to specify the to and from license policies to create an upgrade. The Cloudpaging Player must be online to be notified of a pending upgrade. The user will be prompted to upgrade their application when the Cloudpaging Player starts or when they launch the application. Administrators can configure the upgrades to be applied immediately or allow users to defer upgrades until they are ready to apply them.

Creating an Upgrade

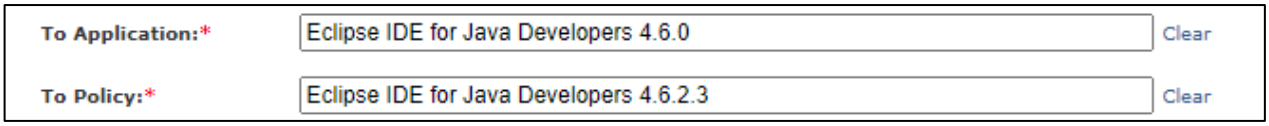
To add an upgrade:

1. From the **Deployments** menu, select **Upgrades** to display the **Manage Upgrades** page.
2. Click **Create** to display the **Create Upgrade** page.

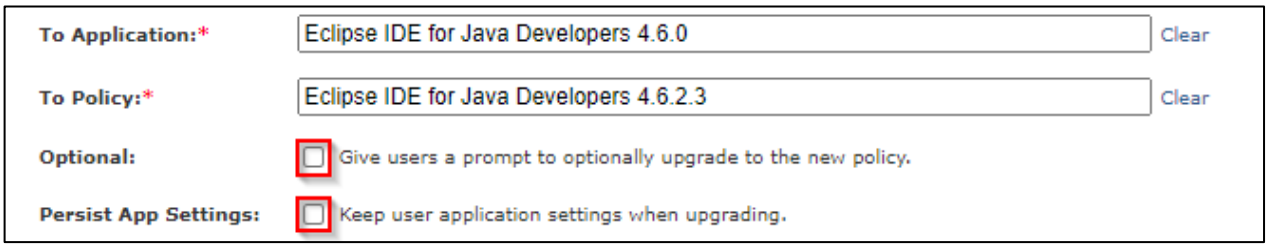
3. Choose the **From Application** and the **From Policy** from the drop-down lists. Details of the application and license policy will be displayed under **From Details** along with information on existing upgrades associated to the **From Policy** selected.



4. Select the **To Application** and **To Policy** from the lists. Details of the application and license policy will be displayed under **To Details**.



5. Optionally, select if the users should be displayed a prompt to defer the upgrade by selecting the **Optional** check box.
6. When upgrading from an application's license policy to another license policy of the same application, administrators can choose to optionally retain the application's settings when upgrading. Click on the check box in the **Persist App Settings** field to retain the settings.



7. Click **Create**.

The **Manage Upgrades** page displays the upgrade path by listing the from and to license policies.

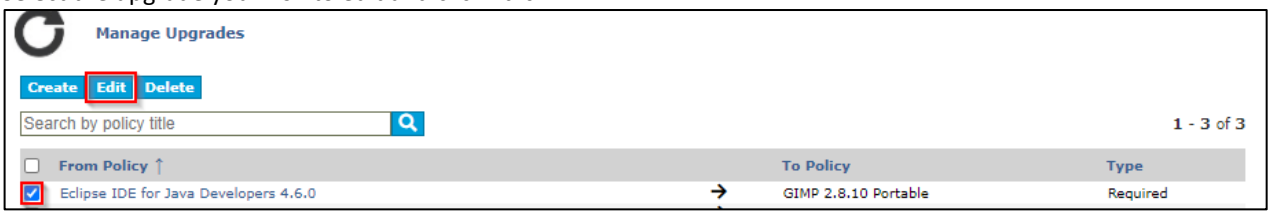
> ***NOTE***
>
> *Circular references are not allowed. For example, you cannot set an upgrade from Application 1 to Application 2 and set another upgrade from Application 2 to Application 1.*

Editing an Upgrade

Changes to an application's upgrade do not affect users that have already upgraded. Only users that have not yet upgraded will receive new upgrade.

To edit an application upgrade:

1. From the **Deployments** menu, select **Upgrades** to display the **Manage Upgrades** options.
2. Select the upgrade you wish to edit and click **Edit**.



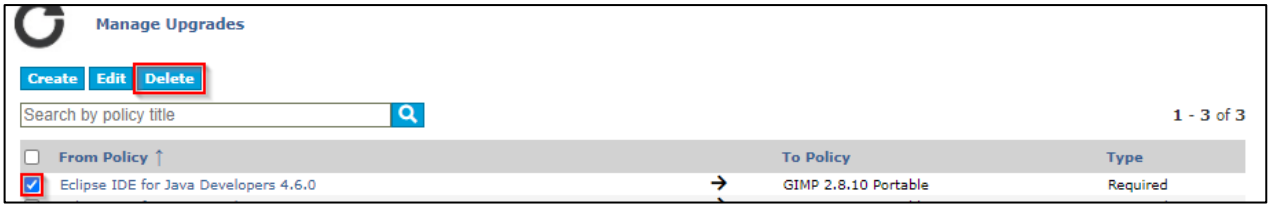
3. Make your desired changes and click **Update**.

The changes you made to the upgrade have been saved.

Deleting an Upgrade

To delete an application upgrade:

1. From the **Deployments** menu, select **Upgrades** to display the **Manage Upgrades** page.
2. Select the upgrade path you wish to delete and click **Delete**.



The upgrade is no longer listed on the **Manage Upgrades** page.

Sessions

When an application is added to the Cloudpaging Player, a session is created on the Cloudpaging Server for that application's license policy properties. The session contains the license policy type (fixed, floating, or metered-only), the IP address of the Player when the session was first created, and the username. The status of a session can change from Registered to Activated, Virtualized, Ready, and Running depending on the state of the application within the Cloudpaging Player. This session remains on the Cloudpaging Server until either the application is removed from the Cloudpaging Player or the session is expired due to policy terms (e.g. **expiration time** or **invalidation time**) or forced expiration by an administrator. The session is removed from the Cloudpaging Server's database once the application is removed or it has been expired. The status of the session goes from Ready to Devirtualized, Deactivated, and Unregistered before being removed.

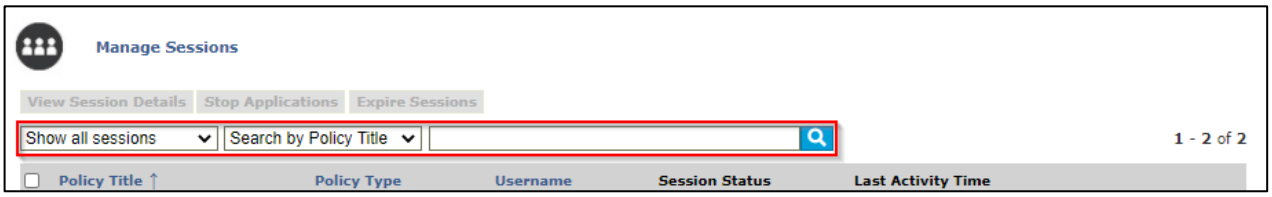
Managing Sessions

The **Manage Sessions** page allows you to view and control sessions. Once an application has been provisioned, a user can continue using the application until the session is expired. The Manage Sessions page can be used to view, stop applications, or expire sessions. Administrators can free up license policy seats by using **Stop Applications** or **Expire Sessions**.

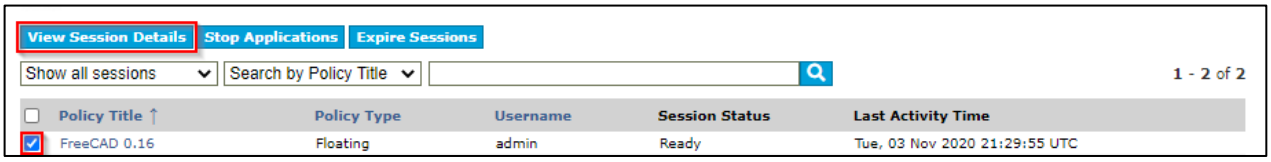
- **Stop Applications** - A session in the **Running** state can be stopped. Once an administrator clicks on **Stop Applications**, the applications will be stopped by the Cloudpaging Player within 5 minutes. The session status will now be displayed as **Stopping**.
- **Expire Sessions** - A session in any state can be expired by clicking on **Expire Sessions**. If the session is in the **Running** state, the application is first stopped and then expired. Once a session has expired, the application can no longer be used.

To view a session:

1. From the **Deployments** menu, select **Sessions** to display the **Manage Sessions** page.
2. You can filter the page by searching for a **Policy Title** or **Username** and selecting the session state. Click on **Search.**



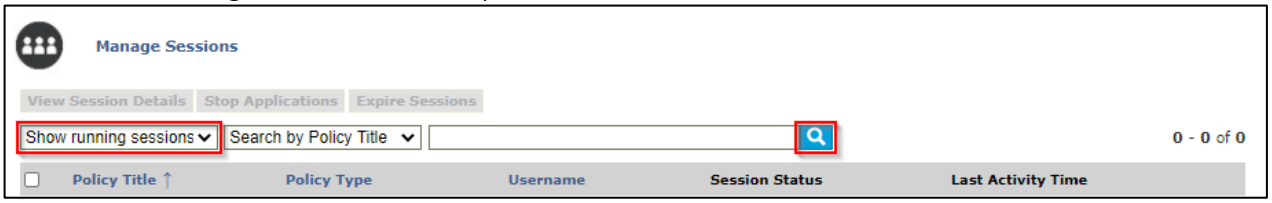
3. Select the session that you wish to view details of and click the **View Session Details** button to display the **View Session** page.



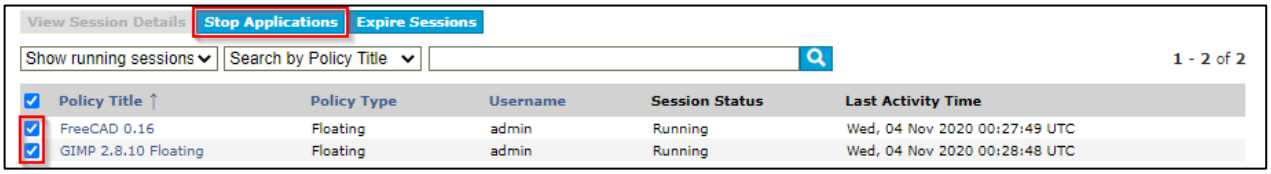
4. Information about the session, such as **Session ID**, **Policy Title**, **Username**, **Physical IP** address, and client statistics is shown.
5. You can click on the **Stop Application** or **Expire Session** buttons, if desired.

To stop an application that is running:

1. From the **Deployments** menu, select **Sessions** to display the **Manage Sessions** page.
2. Choose **Show running sessions** from the drop-down and click **Search**.



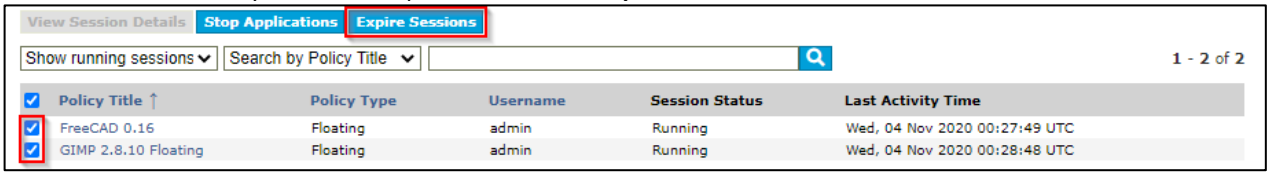
3. Select one or more sessions which you wish to stop and click the **Stop Applications** button.



Within 5 minutes, the application will be forced to close. However, a user can still relaunch a stopped application. To prevent the user from launching that application again, you must expire the session.

To expire a session:

1. From the **Applications** tab, select **Sessions** to display the **Manage Sessions** options.
2. Choose an application state from the drop-down, if desired and click **Search**. If the session is in a **Running** state, then it will be stopped before the session is expired.
3. Select the session that you wish to expire and click the **Expire Sessions** button.



This application can no longer be launched from Cloudpaging Player.

> ***NOTE***
>
> *If you also have the Enterprise Portal configured in your environment, you will need to unsubscribe the license policy for the user in addition to expiring the session to fully revoke access to the application. Please see the Manage License Policies section in the Cloudpaging Server Supplemental Admin Guide for Enterprise Portal for more information.*

How are sessions and seats in a license policy related?

Administrators can limit the usage for a license policy to a specific **number of seats**. You can also limit each user to a specific **number of concurrent usages**. These seats are tracked based on all active sessions containing that license policy GUID. For fixed policies, every session is considered active and is therefore counted as an application seat. For floating policies, only sessions in the **Running** state are considered active and are counted as application seats. Cloudpaging Server will guarantee that the active session seat count for a license policy across all users is less than or equal to the **number of seats** specified in that policy. If the license policy has a limit specified on the **number of concurrent usages**, then the Cloudpaging Server will ensure that the number of active sessions for a given user does not exceed the limit.

Example: If a license policy's **number of seats** count is set to 5, then up to five different users (as identified by their usernames) would be able to run the application on a single machine each. If, additionally, the **number of concurrent usages** count is set to 2 in that same license policy then a single user would be able to run the application on two separate machines. If users attempt to provision an application which would result in exceeding the limits set, they will be presented with an error message and will not be allowed to provision the application.

> ***NOTE***
>
> *The Cloudpaging Server uses the session table to determine the number of users and/or devices which are currently connected to the server.*

What happens when a session is expired?

When a session is expired it is deleted from the database. An expired session implies that the application can no longer be launched from the Cloudpaging Player. A session can be expired in any of the following circumstances:

- The License policy's **expiration date** has been reached.
- The session status has not changed within the duration specified in the **Day(s) of inactivity** configuration in the license policy. For more information, see Policy Enforcement Settings.
- The administrator manually expires the session on the **Manage Sessions** page.

When the Cloudpaging Player is online and an application is in a **Running** state, the session status is updated at every heartbeat interval. There is a grace period that defines how long the Cloudpaging Server will wait after a heartbeat. If this grace period passes, then the Cloudpaging Server will stop the session and the Cloudpaging Player will force the user to stop the application.

How is a session expired when the application is offline?

For a fixed license policy with offline duration, the heartbeat check to the Cloudpaging Server will continually refresh the offline duration. For example, if the offline duration is set to 7 days, the Cloudpaging Player will count down the time until the heartbeat is successful. On a successful heartbeat, the offline duration is renewed back to 7 days. If the session is expired due to policy terms or by the administrator, the application will be removed from the Cloudpaging Player. When the Cloudpaging Player is offline or if the heartbeat fails the offline duration continues to count down. If the time becomes 0 the session is stopped on the Cloudpaging Server.

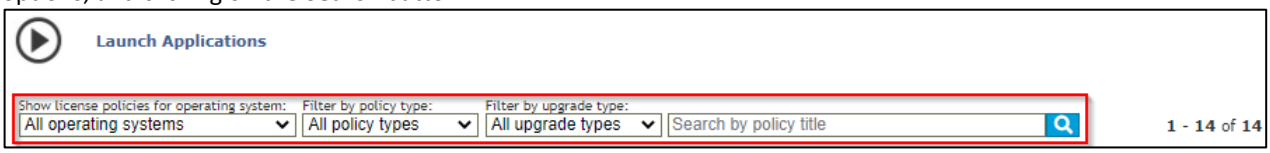
Testing Applications

Administrators can test application provisioning before deploying them to users.

To test application provisioning:

1. From the **Deployments** menu, select **Launch Applications** to display the **Launch Applications** page. Applications are displayed in a list along with their license policy information like Title, Description, Type, Windows Compatibility Settings and Upgrade Path (if applicable).

2. You can filter the results using the search by **Operating Systems, Policy Type**, **Upgrades** or **Policy Title** options, and clicking on the **Search** button.



3. Click on the **Play** button on the right of the application you wish to provision.

The application will be provisioned to the Cloudpaging Player on your machine.

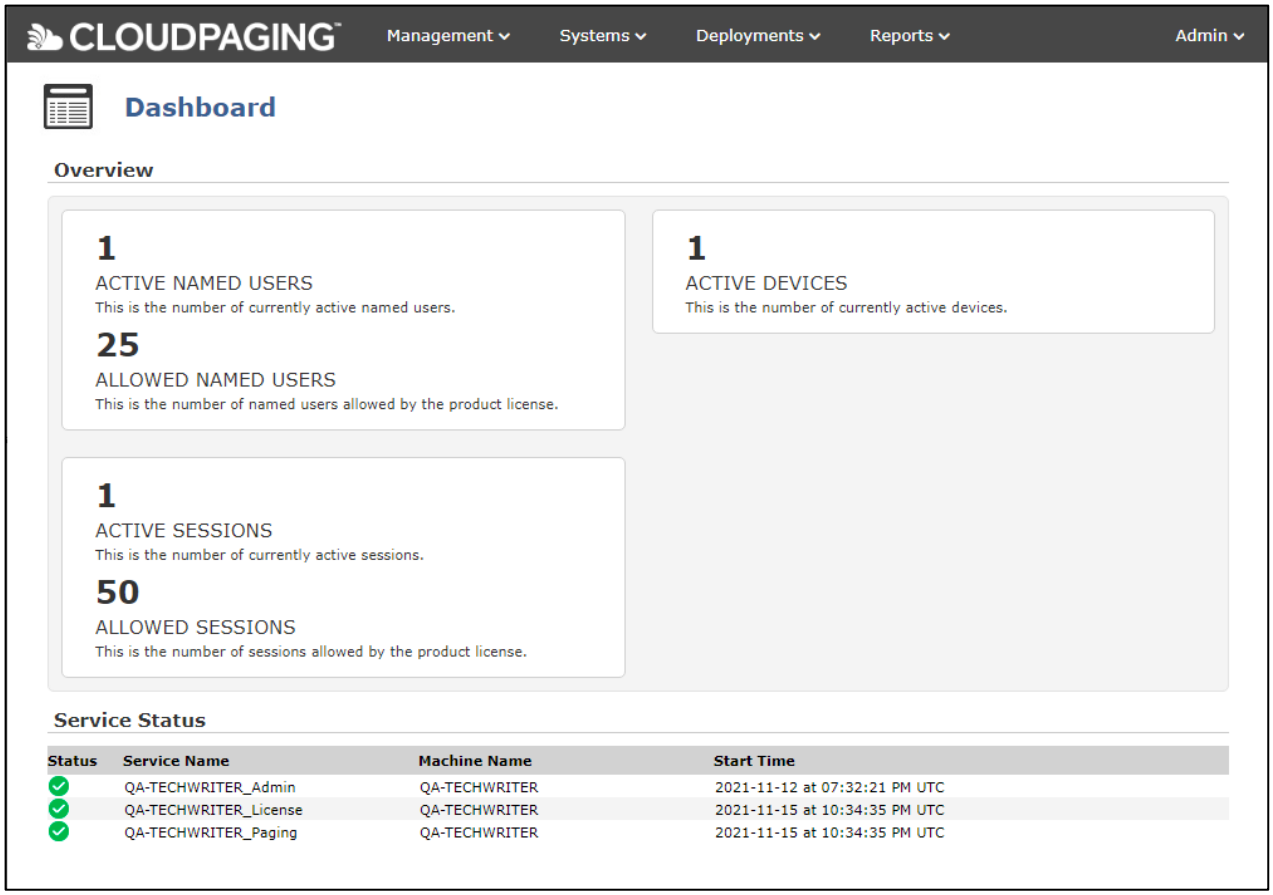
Chapter 6: System Management

The following are explained in this section:

- Dashboard
- Services
  - Adding a Service
  - Editing a Service
  - Starting and Stopping a Service
  - Deleting a Service
- Machines
  - Adding a Machine
  - Editing a Machine
  - Deleting a Machine
- Server Configurations
- User Accounts
  - Creating a User
  - Editing Users
  - Deleting Users
- System Logging
  - Managing Logging
- System Monitoring
  - Server Health Checks
- Backup Recovery
  - Database Tuning

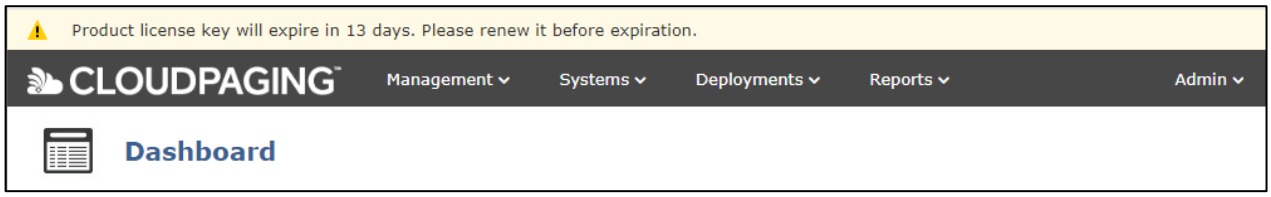
Dashboard

The Dashboard provides quick and easy system status information, including available Cloudpaging services. By default, the Dashboard page appears each time you log in to the Admin service.



To view alerts:

Alerts, such as when the product key will expire, will be shown at the top of the page above the navigation bar.



Services

Cloudpaging services (Admin, License, and Paging) can be added or modified once they are registered on each machine in the system. For descriptions of the three Cloudpaging services, refer [to Cloudpaging Server Components](#).

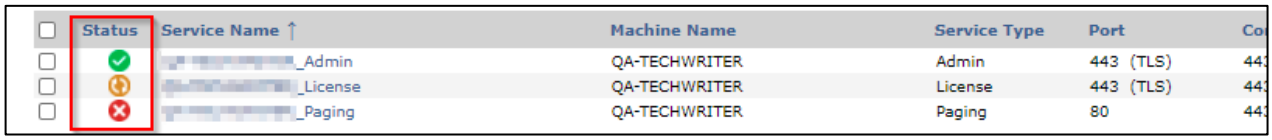
Checking the Status of a Service

The **Manage Services** page displays a list of Cloudpaging services and status associated with each machine.

To check the status of a service:

From the **Systems** menu, select **Services** to display the **Manage Services** page. The status of a service is indicated by the **Status** icons as follows:

- **Green (check mark)**: the service is running normally.
- **Yellow (spinning animation)**: the service is loading.
- **Red (X mark)**: the service is not running. Possible reasons for this are that the service had a problem starting or that service was stopped by an Administrator.



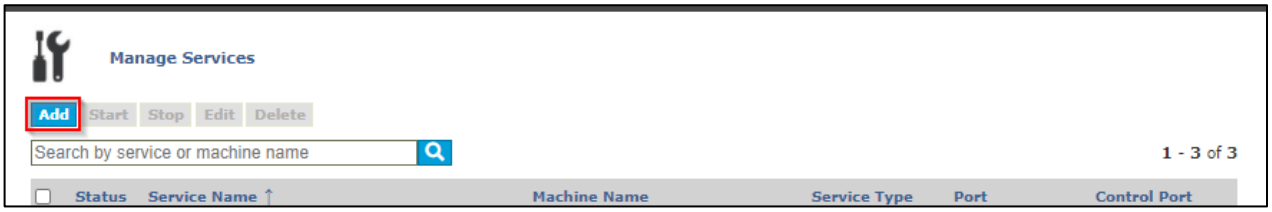
Adding a Service

You can add a Cloudpaging Server service to an existing machine to increase capacity of the Cloudpaging Server system.

> ***NOTE***
>
> *Each machine can only run one instance of each Cloudpaging Service. For example, a machine can run an Admin, Paging, and License service, but not two Paging services.*

To add a service to a machine

1. From the **Systems** menu, select **Services** to display the **Manage Services** page.
2. Click **Add** to display the **Add Service** page



3. Specify the following information
   a. **Machine:** Select the machine on which the service I hosted from the drop-down menu.
   b. **Service Type**: Select the service type.
   c. **Name:** Enter a name to uniquely identify the service. This name is for internal system reference only.
   d. **Port:** Specify the port through which the Cloudpaging Player will communicate with the Cloudpaging Server. The default port number for the Admin and License services is 443 and the

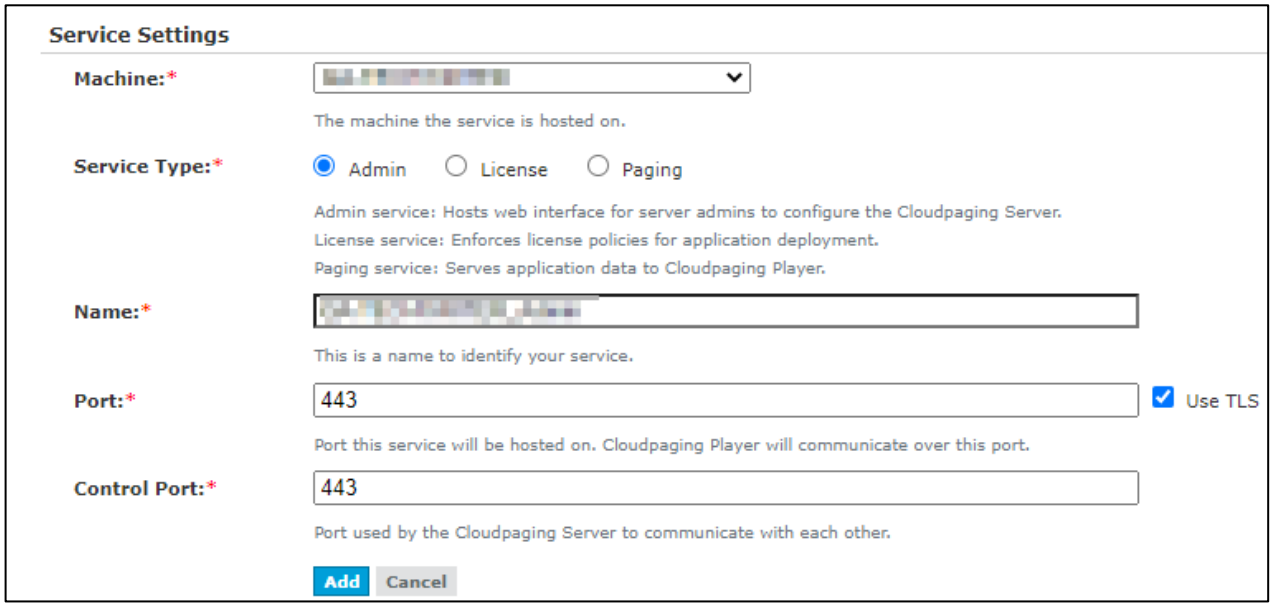
default for the Paging service is 80. You can change the port numbers, if desired. However, this will required additional updates to the Apache Tomcat configuration (see Additional Apache Tomcat configurations for more information).

e. **Use TLS:** You can optionally change whether the services will communicate over a secure or a non-secure port. The Admin and License services use TLS by default. The paging service does not use TLS. Please note changing the default setting will required additional updates to the Apache Tomcat configuration (see Additional Apache Tomcat configurations for more information).

f. **Control Port:** The control port number is used by multiple Cloudpaging Servers to communicate with each other.

4. Click **Add.**

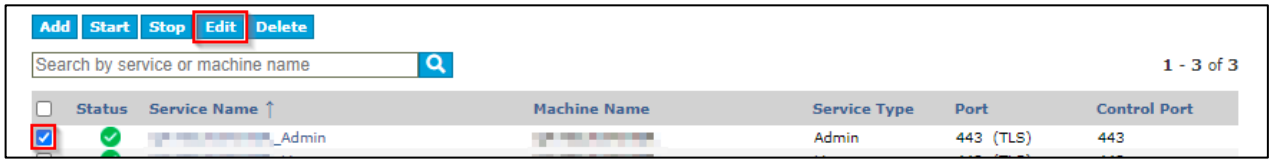
The new service will be displayed on the **Manage Services** page.



Editing a Service

To edit a Cloudpaging Service

1. From the **Systems** menu, select **Services** to display the **Manage Services** page.
2. Select the check box next to the desired service and click **Edit** to display the **Edit Service** page.



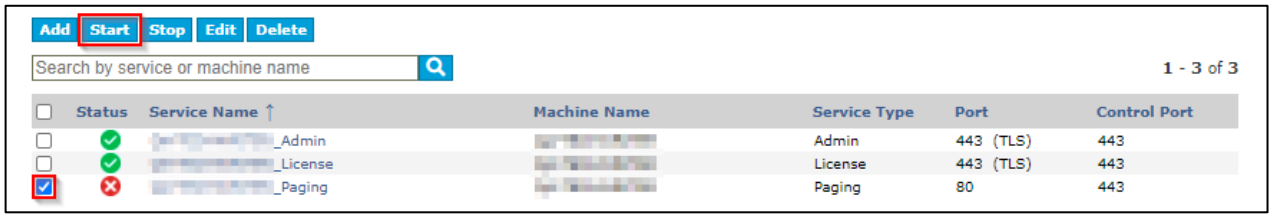
3. Make the desired changes and click **Update**. A confirmation dialog appears.
4. Click **OK** to continue or click **Cancel** to abort the update.

> ***NOTE***
>
> *Modifications to the Port number, Use TLS and Control port settings may require updates to the Apache Tomcat configurations. See Additional [Apache Tomcat configurations](#) for more information.*

Starting and Stopping a Service

To start a service:

1. From the **Systems** menu, select **Services** to display the **Manage Services** page.
2. Select the check box next to the desired service and click **Start**.



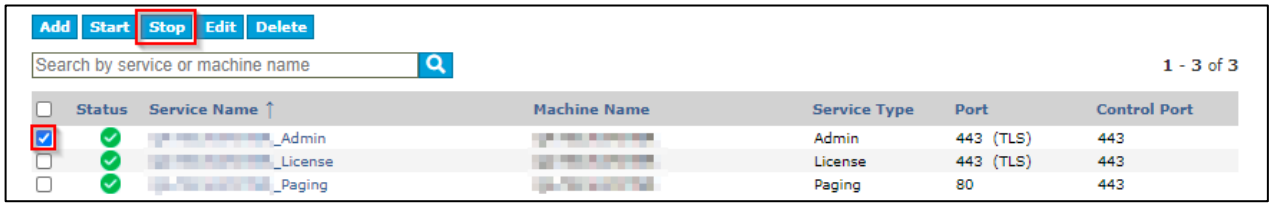
3. From the **Management** menu, select **Dashboard** to display the **Dashboard** page.

   The red x mark status indicator next to the stopped service in the Services table changes to a yellow spinning icon, indicating that the service is loading. Once the service has loaded, it will display a green check mark.

> ***NOTE***
>
> *You may need to refresh your browser to see the updates in the status indicator.*

To stop a service:

1. From the **Systems** menu, select **Services** to display the **Manage Services** page.
2. Select the check box next to the desired service and click **Stop**.



3. From the **Management** menu, select **Dashboard** to display the **Dashboard** page.

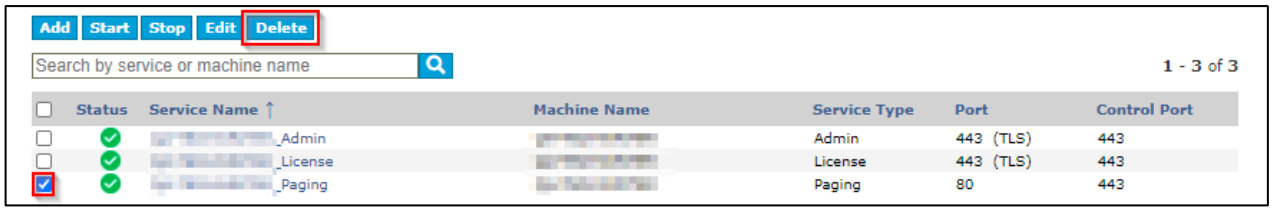
   The green check mark status indicator next to the service in the Services table changes to a red x mark, indicating that the service has stopped.

> ***NOTE***
>
> *You may need to refresh your browser to see the updates in the status indicator.*

Deleting a Service

To delete a service:

1. From the **Systems** menu, select **Services** to display the **Manage Services** page.
2. Select the service you wish to delete and click **Delete**. A confirmation message appears.



3. Click **OK** to continue the deletion, or click **Cancel** to abort the deletion.
   The deleted service no longer appears on the **Manage Services** page.

Additional Apache Tomcat configurations

The Cloudpaging Server uses an instance of Apache Tomcat to setup the ports to access Cloudpaging services. The port configurations are defined in the server.xml file in the installation path(s). Some changes require modifying the server.xml file. This file can be found at the path INSTALLDIR\core\conf\server.xml. Here are some scenarios that require updates to the server.xml file:

- **Changing the port number:** The Admin and License services use secure port 443 by default while the Paging service uses non-secure port 80. You can change the port numbers from their default value to a custom port number. However, changing the port number requires modifying the server.xml file. Update the port number in the Cloudpaging Server and then please see our Support article https://support.numecent.com/a/solutions/articles/1000235279 for more information on how to update the server.xml file.
- **Changing TLS settings:** The 443 port used by the Admin and License services are secure by default. The Paging service uses the non-secured port 80 by default because the pages are encrypted by Cloudpaging Studio and performance is improved when provisioning applications. There is roughly a 15% performance improvement to using HTTP with pre-encrypted application data to using HTTPS. If you wish to change service ports to be secure or non-secure, the server.xml file will have to be updated. Change the TLS settings in the Cloudpaging Server and then please see our Support article https://support.numecent.com/a/solutions/articles/1000295079 for more information on how to update the server.xml file.
- **Using the root of your server IP or DNS:** You can configure the Apache Tomcat web server to use the root context path of your server IP and/or DNS. For example, you can access the Cloudpaging Server by visiting https://<your-ip-or-dns>/ instead of https://<your-ip-or-dns>/jukeboxserver. This change requires modifying the server.xml file. Please see our Support article https://support.numecent.com/a/solutions/articles/1000295222 for more information on how to update the server.xml file.

Machines

A machine is a physical PC or a virtual machine on which Cloudpaging Server services are running. All Cloudpaging Server machines must be registered with the Cloudpaging Database. This is done automatically during installation.

Adding a Machine

If the current capacity of the Cloudpaging Server system is not sufficient, or if a machine needs to be replaced, then you can add machines to the existing configuration. After you add the machine, the system recognizes that the new machine exists at a specified IP address and has a specific name.

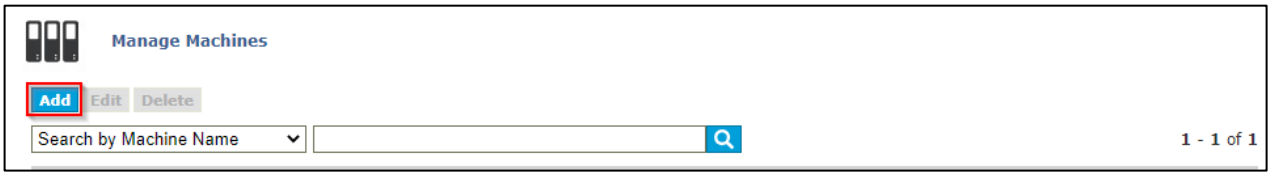
> ***NOTE***
>
> *Before you add a machine, make sure all necessary server files have been installed (see [Chapter 3, Cloudpaging Server Installation](#)).*

Once a machine has been added, Cloudpaging services can be configured to run on that machine.

To add a machine:

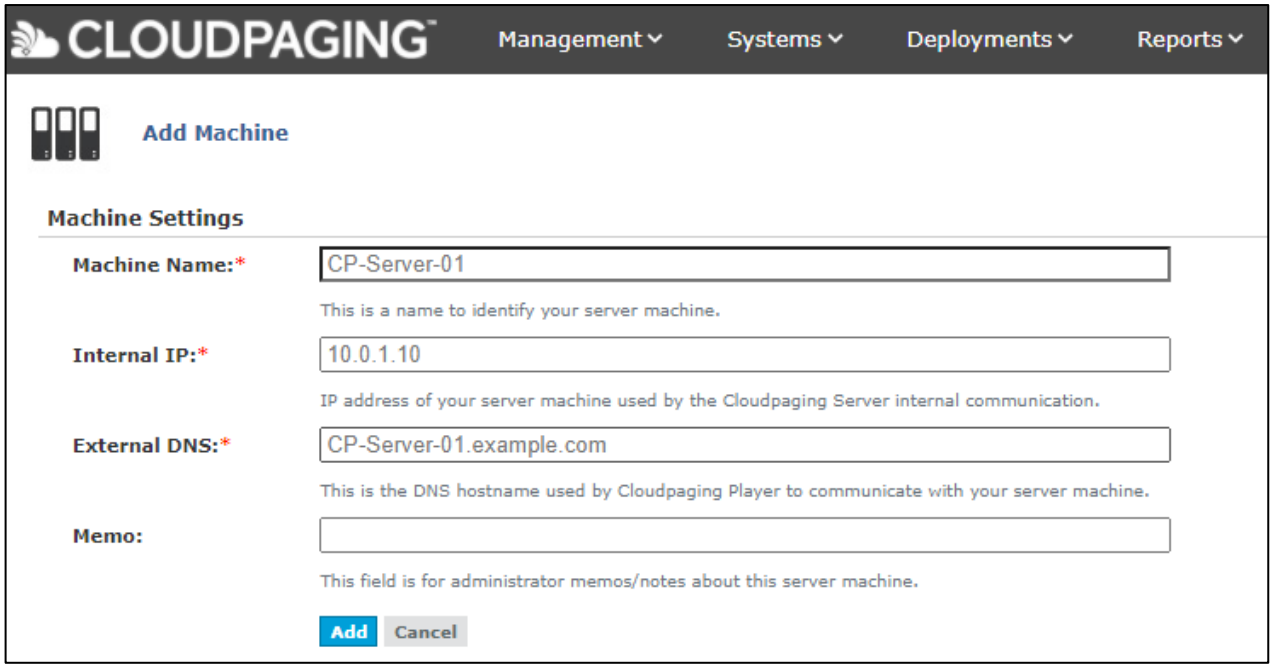
To add a machine:

1. From the **Systems** menu, select **Machines** to display the **Manage Services** page.
2. Click **Add** to display the **Add Machine** page.



The following options are displayed on this page:

a. **Machine Name:** This machine name is to uniquely identify your machine and is an internal system reference only.
b. **Internal IP**: This is the physical IP address of the machine where Cloudpaging services will be running and communicating with each other.
c. **External DNS:** The DNS hostname that the Cloudpaging Players use to contact the machine. If you add a Load Balancing switch in your environment, the External DNS needs to point to the address of the Load Balancing switch.
d. **Memo:** This field is for any additional notes that you may want to add about the server machine.



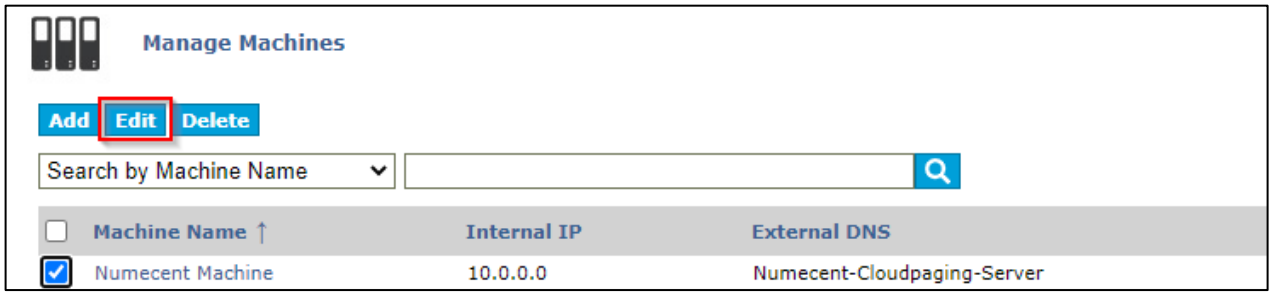
3. Click **Add.**
   The machine has been added and will appear on the **Manage Machines** page**.**

Editing a machine:

The Cloudpaging Server allows you to edit the Internal IP and External DNS of a machine. If IP addresses are changed locally on a machine, then the Cloudpaging Server will not be able to communicate with that machine. You must update the machine's configuration in the Cloudpaging Server.

To edit a machine:

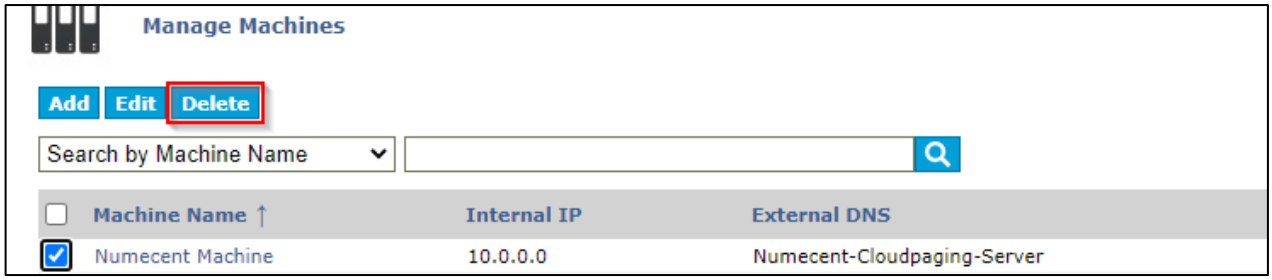
1. From the **Systems** menu, select **Machines** to display the **Manage Services** page.
2. Select the machine you wish to edit and click **Edit.**



3. Make the desired changes and click **Update.** For descriptions of these fields, see Adding a Machine. The changes will appear on the **Manage Machines** page.

Deleting a machine:

1. From the **Systems** menu, select **Machines** to display the **Manage Services** page.
2. Select the machine you wish to delete and click **Delete.** A confirmation appears.



3. Click **OK** to continue with the deletion process or click **Cancel** to abort the deletion. The deleted machine no longer appears on the **Manage Machines** page.

> ***NOTE***
>
> *If you need to delete a specific Admin, Paging, or License service, see Deleting a Service).*

Server Configurations

Server Configurations are settings that apply to the entire Cloudpaging Server system. Most of these settings are configured during system installation or are assigned default values by the system. After system installation is completed, you can view, modify, or add new settings from the Server Configurations page.

To view or edit the server configurations:

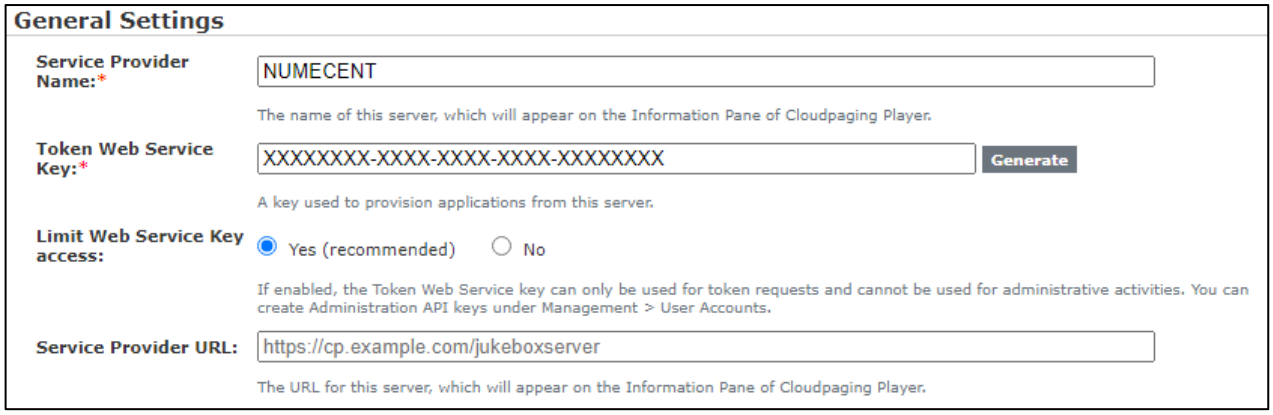
1. From the **Systems** menu, select **Server Configurations** to display the **Server Configurations** page.
2. Make the desired changes and click **Update**. (See Server Configurations Options for more information on these settings).

Server Configurations Options

There are seven sections on this page.

General Settings

- **Service Provider Name:** The name provided in this field will appear on the Information Pane of the Cloudpaging Player. This can help your users identify your system from others.
- **Token Web Service Key:** The Token Web Service Key is a required parameter when using Cloudpaging Server for token requests to provision applications from third party systems. See *Cloudpaging Web Portal Integration Supplement Guide* for more information.



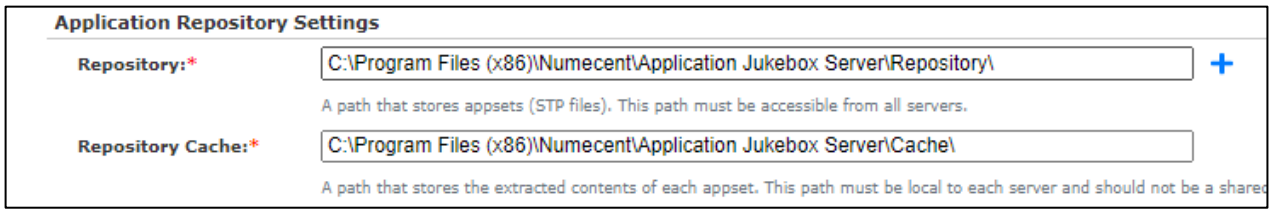
> ***Important***
>
> *Any updates to the **Token Web Service Key option** requires restarting the **Cloudpaging Server Service** on all machines running Cloudpaging services.*

> ***NOTE***
>
> *If Enterprise Portal is configured in your environment, you should update the configuration to use an API Key in the Enterprise Portal's **System Configuration**. Please see the Cloudpaging Server Supplemental Admin Guide for Enterprise Portal for more information.*

- **Limit Web Service Key access:** This control will disable the Token Web Service Key from being able to access the Cloudpaging Server APIs. This providers a higher level of security.
- **Service Provider URL:** This URL will appear on the Information Pane of the Cloudpaging Player. It also identifies the web page that users will need to visit to provision applications to their machine.

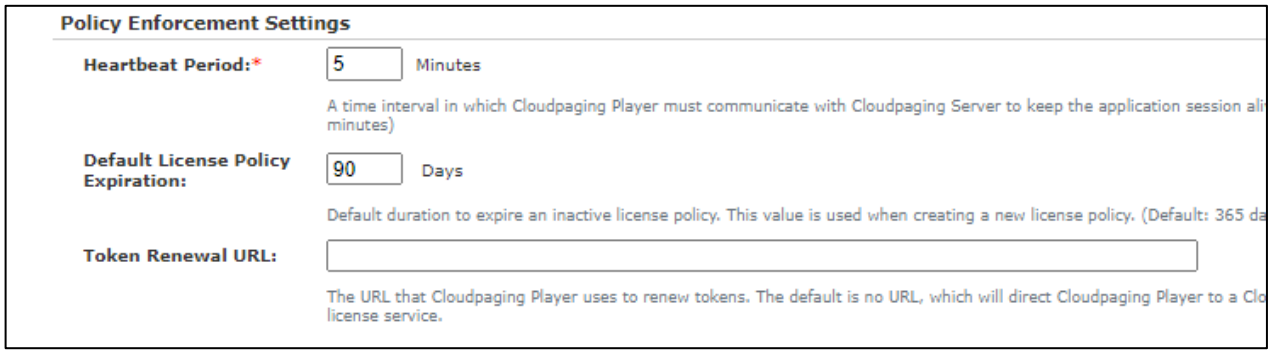
Application Repository Settings

- **Repository:** The path to the folder that contains all appsets (STP files) uploaded to the Cloudpaging Server. You can add multiple repository paths by clicking on the **Add (+)** button. Any updates to this option requires restarting the **Cloudpaging Server Service** on all machines running Cloudpaging services. The repository path must be accessible to all Cloudpaging Servers in a distributed topology.
- **Repository Cache:** Stores the extracted contents of appsets published to the Paging Service. Any updates to this option requires restarting the **Cloudpaging Server Service** on all machines running Cloudpaging services. This path should be a local system path on the machine.



Policy Enforcement Settings

- **Heartbeat Period:** This is the time in minutes within which the Cloudpaging Player must communicate with the Cloudpaging Server to inform that a session is still active. The default period is 5 minutes. If the Cloudpaging Player fails to communicate within this period, the Cloudpaging Server will stop running the application.
- **Default License Policy Expiration:** This value indicates how many days a session can remain unused (not in a Running state) before it will be expired by the Cloudpaging Server. The default duration is 365 days.
- **Token Renewal URL:** The URL that Cloudpaging Player uses to renew tokens. By default, the Cloudpaging Player will be directed to a Cloudpaging Server license service. Token Renewal URL should be in format https://server-dns-name/jukeboxserver/license/token/renew.tok.

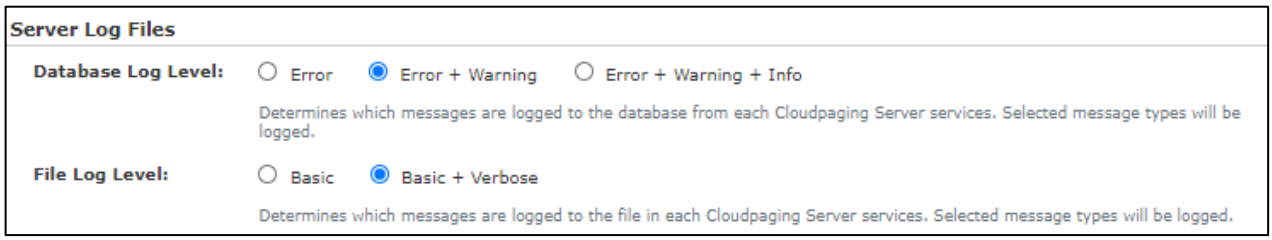


Player Settings

- **Player Upgrade URL:** This URL allows the end-users to upgrade their Cloudpaging Player version without requiring help from an administrator.
- **Minimum Version Required:** This option is used to specify a minimum version number of the Cloudpaging Player that can be used when provisioning applications from the Cloudpaging Server. The version should be specified in the format of x.x.x or x.x.x.xxxxx, where x is an number between 1 and 99. For example, 9.1.9 or 9.2.0.28774. If the end-users' Cloudpaging Player version is less than the version specified, they will be prompted to upgrade their Cloudpaging Player by using the Player Upgrade URL.
- **Collect Machine Details:** Enabling this setting will record end-user's machine information, such as IP address, OS, CPU, and bandwidth. This information can be viewed on the View Session page and the Usage reports.

Server Log Files

- **Database Log Level:** Determines level of messages logged to the database for the Cloudpaging services. The default log level is Error + Warning + Info. Any updates to this option requires restarting the Cloudpaging Server Service on all machines running Cloudpaging services.
- **File Log Level:** Determines level of messages logged to the file for the Cloudpaging services. The default log level is Basic. Any updates to this option requires restarting the Cloudpaging Server Service on all machines running Cloudpaging services.

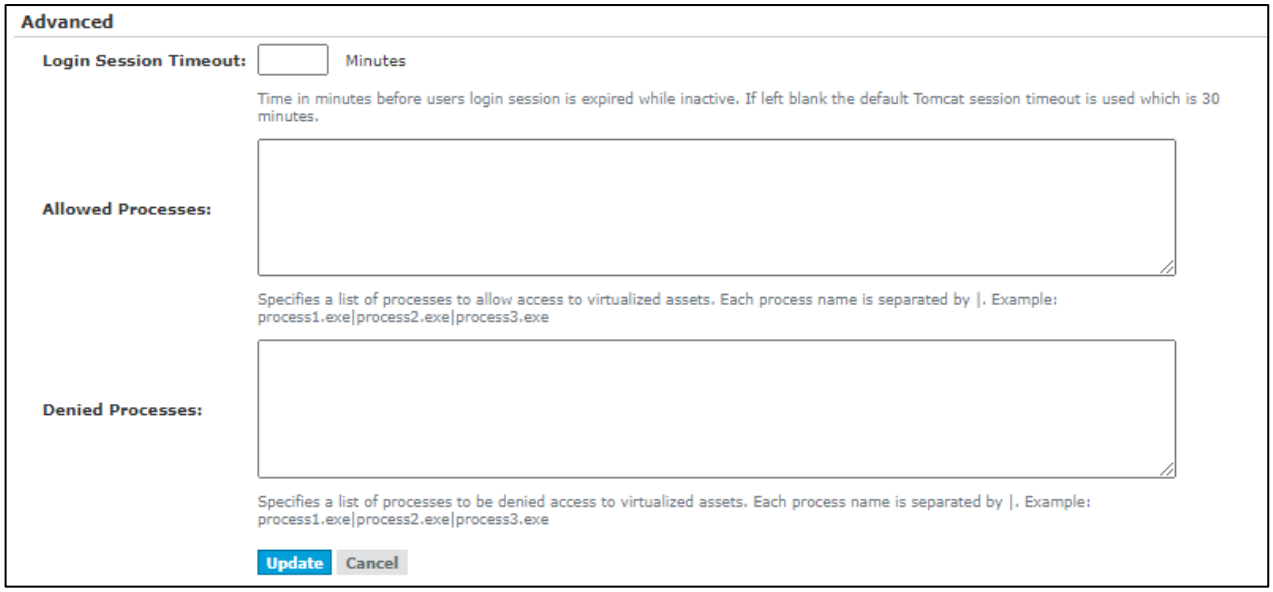


Web Caching Settings

- **Allow Web Caching:** This option allows web caching through 3rd party web caching service.
- **Cache Period:** The period of time that pages should be stored in the web cache before they can be overwritten.

Advanced Settings

- **Login Session Timeout:** Time in seconds before end-users' login session is expired while inactive. If left blank, the default Apache Tomcat session timeout is used (30 minutes).
- **Allowed Processes:** Specify a list of processes that are allowed to access the virtualized assets. Multiple process names can be entered separated by a vertical bar or pipe symbol (|).
- **Denied Processes:** Specify a list of processes that are not allowed to access the virtualized assets. For example, you may want to deny anti-virus processes from accessing the assets. Multiple process names can be entered separated by a vertical bar or pipe symbol (|).



User Accounts

The Cloudpaging Server supports role-based access control. This feature allows administrative users to manage various Cloudpaging Server functions based on user roles and restricts access to unauthorized users. Each role is predefined and grouped according to logical functions in the Cloudpaging Server.

The table below describes each role and the privileges associated with the role. The privileges defined for each role are hierarchical; this means higher level roles inherit permissions of lower level roles. For example, the Session Manager role inherits all privileges of the Reporter role in addition to having its own privileges.

Logging Level	Description
Reporter (default level)	View reports for usage, license policies, and applications
Session Manager	View, stop, and expire Cloudpaging sessions. This role inherits all Reporter privileges.
Applications Manager	Manage Cloudpaging applications, license policies, and upgrades. This role inherits all Session Manager privileges.
System Manager	Manage machines, services, configurations, and product license keys. This role inherits all Application Manager privileges.
Administrator (Highest level)	Manager user accounts. This role inherits all System Manager privileges.

Creating a User

To enable role-based access control, start by creating a new administrative user. This user can be an internal user or an existing user in Active Directory. When you create a new administrative user, you must assign them a username and password, along with a role that determines the Cloudpaging Server functions they can access. Internal users can change the password set for them after their first login.

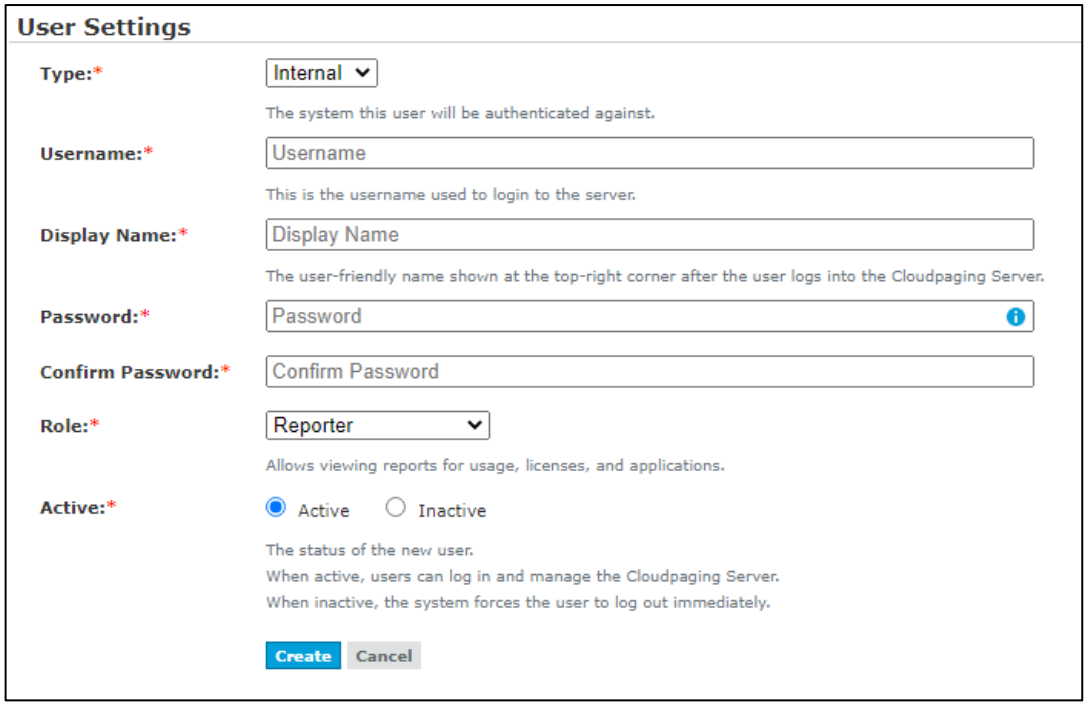
Access to the Cloudpaging Server API's can also be controlled under the Manage User page by creating an "API Key" account. An "API Key" is a uniquely generated password that allows API level access to your Cloudpaging Server configuration, settings, and applications. This is used to keep DevOps pipelines and automation secure.

To create a new user:

1. From the **Management** menu, select **User Accounts** to display the **Manage Users** page.
2. Click **Create** to display the **Create User** page.

The following options are displayed on the **Create User page:**

a. **Type:** Select either **Internal** or **Active Directory**, if configured.
b. **Username:** A unique identifier in the user management subsystem. The username must start with an alphanumeric character and is not case sensitive. It can include the following special characters: ! # @ ( ) . - _ ~ and spaces are not allowed. For Active Directory, this name must match the username (SAMAccountName or User Principal Name) in Active Directory without the domain.
c. **Display Name:** The user-friendly name shown at the top-right corner after the user has logged in to the Cloudpaging Server.
d. **Password** (this option is not available for Active Directory users)**:** A case-sensitive password that is 8 to 64 characters long. The password must contain at least three of the following four items:
    i. One number
    ii. One uppercase letter
    iii. One lowercase letter
    iv. One special character
e. **Confirm Password:** Retype the same case-sensitive password.
f. **Role:** The role to which the new user will be assigned. The default role is Reporter. See User Accounts for more information on roles.
g. **Active:** The status of the new user. By default, new users have an active status. If you change the status to Inactive, the system forces the user to log out immediately and the user will no longer have access the Cloudpaging Server.

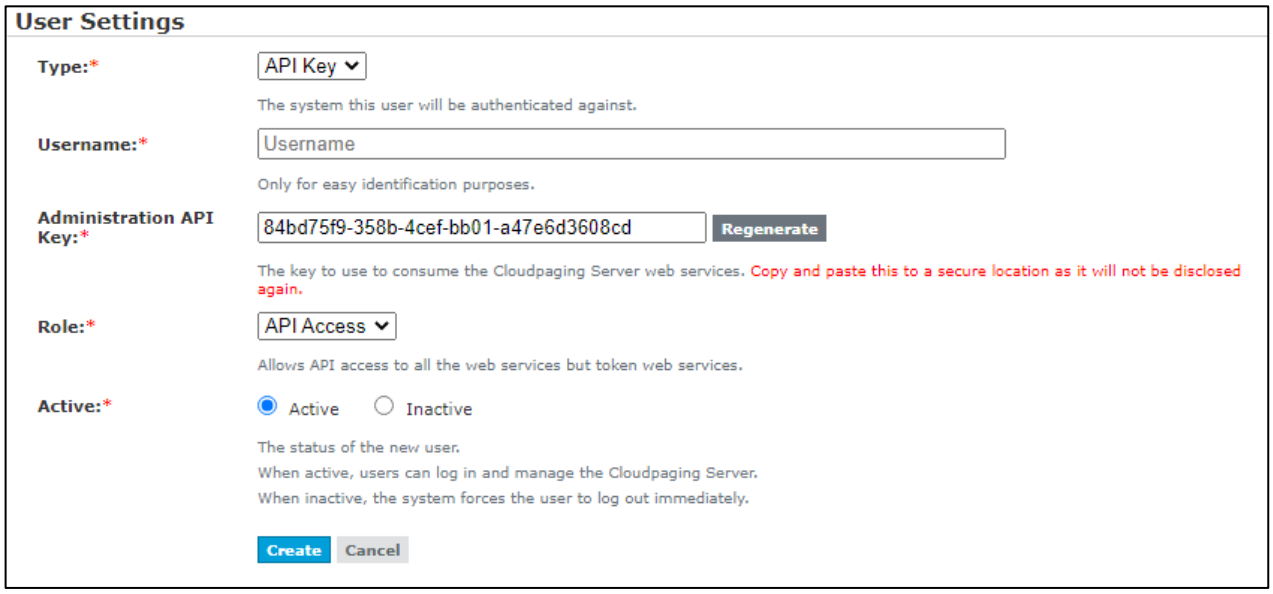


3. Click **Create**. A new administrative user has been created.

After the new user logs into the system, the available tabs correspond to the privileges associated with the role assigned to that user.

To create an API key:

1. From the **Management** menu, select **User Accounts** to display the **Manage Users** page.
2. Click **Create** to display the **Create User** page.



The following options are displayed on the **Create User page:**

a. **Type:** Select **API Key**
b. **Username:** A unique name to identify the intended use of the API access.

c. **Administration API Key: X**
d. **Role:** Will be set to API Access.
e. **Active:** The status of the new key. By default, the new key will have an active status. If you change the status to **Inactive** the system will no longer allow that key to access the Cloudpaging Server.

3. Click **Create**. A new administrative API key has been created.

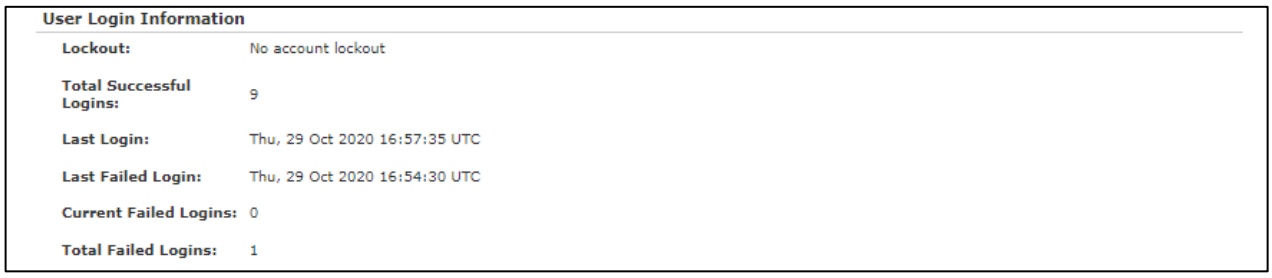
Editing Users

Once an administrative user has been created, you can change their display name, password, role, or active status. However, the username cannot be modified.

> ***NOTE***
>
> *If you edit an administrative user, the system logs out the user automatically. The user must re-login for the edits to take effect.*

To edit users:

1. From the **Management** menu, select **User Accounts** to display the **Manage Users** page.
2. Select the user you wish to edit and click **Edit** to display the **Edit User** page. In addition to providing fields for editing the user, this page shows login statistics. You may find information helpful when determining whether the user has logged into the system successfully.



3. Make the desired changes and click **Update**. The user you edited is logged out automatically and must re-login for the edits to take effect.

For security purposes, if the user enters wrong passwords consecutively, the Cloudpaging Server will lock the user out for 5 minutes after a set number of attempts. A user with Administrator privileges can, however, click the **Unlock** button in the Lockout field on the **Edit User** page to unlock the user.

Deleting Users

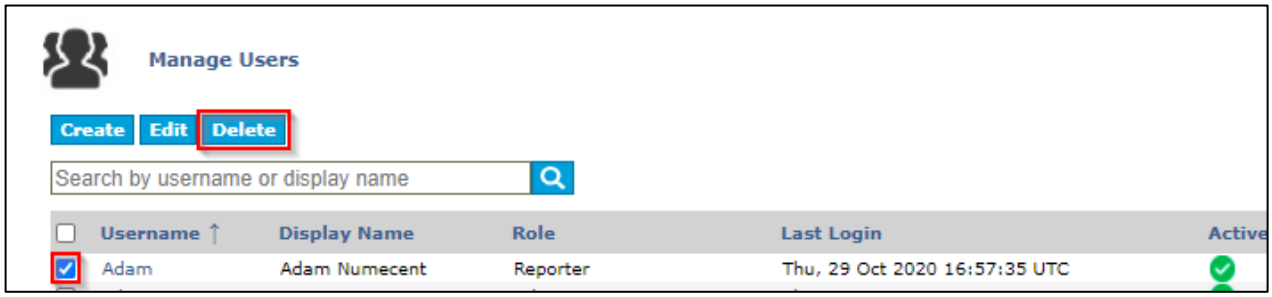
The Cloudpaging Server allows users with Administrator privileges to delete users from the system. However, if you only want to revoke a user's access to the system, you can change the user's status from Active to Inactive on the Edit User page instead of deleting the user. The system forces the inactive user to log out immediately. See Editing Users for more information.

Additional notes on deleting users:

- Users cannot delete themselves.
- After a user account has been deleted, the user is forced to log out from the system. No additional changes can be made by the deleted user.
- To prevent multiple users from being deleted accidentally, only one user account can be deleted at a time.
- Active Directory users are only deleted from the Cloudpaging Server and do not affect Active Directory.

To delete users:

1. From the **Management** menu, select **User Accounts** to display the **Manage Users** page.
2. Select the user you wish to delete and click **Delete.** A confirmation message will appear.



3. Click **OK** to continue the deletion, or click **Cancel** to abort the deletion. The deleted user no longer appears on the Manage Users page and the system forces the user to log out from the system.

System Logging

Each Cloudpaging Server machine will log information to the database and to a file based on the **Server Log Settings** under **Server Configurations**.

Server logs provide extensive data about hardware resource usage and the Cloudpaging Server services usage. You can use the logs to analyze web server problems when a browser or the Cloudpaging Player cannot communicate with the Cloudpaging Server. Logging is available for all services, and a summary of the logging output is shown on the Dashboard.

Managing Logging

The Cloudpaging Server uses Log4j to perform all file logging and the default is to output to the Tomcat console and to a 100MB roll-over log file. Log files can be found on each machine under the installation folder, such as **C:\Program Files\Numecent\Application Jukebox Server\core\logs**, in a file named **core_service.log**. The default log level is **INFO**, however different levels could be set for each of the different services. The table at the top of next page lists the supported logging levels.

Logging Level	Description
ALL	All levels including custom levels.
TRACE	Designates finer grained informational events than DEBUG.
DEBUG	Designates fine-grained informational events that are most useful to debug an application.
INFO	Designates informational messages that highlight the progress of the application at coarse-grained level.
WARN	Designates potentially harmful situations.
ERROR	Designates error events that might still allow the application to continue running.
FATAL	Designates very severe error events that will presumably lead the application to abort.
OFF	Highest possible rank and is intended to turn off logging.

To change the log level:

1. Edit the **log4j.xml** file located under **C:\Program Files\Numecent\Application JukeboxServer\core\webapps\jukeboxserver\WEB-INF\classes**.
2. Find the following XML element with the name attribute **name="com.st.k2"**.

```
<logger name="com.st.k2" additivity="false" >
<level value="INFO" /> <-- Change here
<appender-ref ref="fileAppender"/>
<!-- appender-ref ref="emailAppender"/-->
</logger>
```

3. Change the level element's value from INFO to the desired log level.
4. Find the following text in the configuration file:

```
<appender name="fileAppender"
class="org.apache.log4j.RollingFileAppender">
<param name="Threshold" value="INFO" /> <-- Change here
<param name="MaxFileSize" value="100MB"/>
<param name="File" value="${catalina.base}/logs/
drm_service.log"/>
```

5. Change the text **value="INFO"** to **value="DEBUG"**.
6. Save the file.
7. Restart the Cloudpaging Server service.

To setup logging specific to a service, use the following services name(s): com.st.k2.service.stream, com.st.k2.service.token, or com.st.k2.service.admin.

To set up email alerts for fatal errors:

1. Edit the **log4j.xml** file located under **C:\Program Files\Numecent\Application JukeboxServer\core\webapps\jukeboxserver\WEB-INF\classes**.

```
<appender-ref ref="emailAppender"/>
```

2. Search the file for the following XML elements, and remove the comments around the elements.

3. Search the file for the following XML elements, and modify the element's value attribute with the desired values.

```
<param name="SMTPHost" value="localhost.localdomain" /> <!--
Replace with SMPT server URL -->
<param name="From" value="from@domain.com" /> <!-- Replace
with from email address -->
<param name="To" value="to@domain.com" /> <!-- Replace with
recipient's email address -->
```

4. Search the file for the triggering Policy element, and remove the comments around the element.

```
<triggeringPolicy
class="org.apache.log4j.rolling.FilterBasedTriggeringPolicy">
<filter class="org.apache.log4j.filter.LevelRangeFilter">
<param name="levelMin" value="INFO" />
<param name="levelMin" value="FATAL" />
</filter>
</triggeringPolicy>
```

5. Save the file.
6. Restart the Cloudpaging Server service. An email will be generated for all FATAL errors, which appear like the following:

Log session start time Tue Oct 05 18:00:30 PDT 2010

Time	Thread	Level	Category	Message
11450	Thread-1	ERROR	com.st.k2.license.b	dbjb

```
java.net.UnknownHostException: dbjb
    at java.net.InetAddress.getAllByName0(Unknown Source)
    at java.net.InetAddress.getAllByName0(Unknown Source)
    at java.net.InetAddress.getAllByName(Unknown Source)
    at java.net.InetAddress.getByName(Unknown Source)
    at com.st.k2.license.b.isValidIP(b.java:419)
    at com.st.k2.service.base.CoreService.prepareForService(CoreService.java:157)
    at com.st.k2.service.base.CoreService.onInit(CoreService.java:88)
    at com.st.k2.service.stream.StreamService.onInit(StreamService.java:54)
    at com.st.k2.service.base.BaseService.doInit(BaseService.java:75)
    at com.st.k2.service.base.BaseService.init(BaseService.java:64)
    at com.st.k2.service.base.BaseService.init(BaseService.java:54)
    at org.apache.struts.action.ActionServlet.initModulePlugIns(ActionServlet.java:869)
    at org.apache.struts.action.ActionServlet.init(ActionServlet.java:336)
    at javax.servlet.GenericServlet.init(GenericServlet.java:212)
    at org.apache.catalina.core.StandardWrapper.loadServlet(StandardWrapper.java:1139)
    at org.apache.catalina.core.StandardWrapper.load(StandardWrapper.java:966)
    at org.apache.catalina.core.StandardContext.loadOnStartup(StandardContext.java:3956)
    at org.apache.catalina.core.StandardContext.start(StandardContext.java:4230)
    at org.apache.catalina.core.ContainerBase.start(ContainerBase.java:1014)
    at org.apache.catalina.core.StandardHost.start(StandardHost.java:736)
    at org.apache.catalina.core.ContainerBase.start(ContainerBase.java:1014)
    at org.apache.catalina.core.StandardEngine.start(StandardEngine.java:443)
    at org.apache.catalina.core.StandardService.start(StandardService.java:448)
    at org.apache.catalina.core.StandardServer.start(StandardServer.java:700)
    at org.apache.catalina.startup.Catalina.start(Catalina.java:552)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
    at java.lang.reflect.Method.invoke(Unknown Source)
    at org.apache.catalina.startup.Bootstrap.start(Bootstrap.java:295)
    at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:433)
```

See apache-common-log for more details about logging configuration: http://commons.apache.org/logging/guide.html.

System Monitoring

We strongly recommend having system hardware monitoring in place. Third-party system monitoring software can be configured to watch and report on system status including:

- Ping availability and response times
- % CPU Utilization
- Memory availability
- Bandwidth consumption

Server Health Checks

To assist with third-party health checks, the Cloudpaging Server supports a PING type mechanism for all services. PING services can be accessed as through the Core and Enterprise Portal web pages:

- Admin Service: /jukeboxserver/ping.do
- License Service: /jukeboxserver/license/token/renew.tok?msid=ping
- Paging Service: /jukeboxserver/stream/client.do?msid=ping

Response format will be an HTTP Response. (Ex. HTTP 200 No Error, 403 forbidden, 500 Internal Error, 503 unavailable, etc.). If the server is down, it will return with a time-out.

To check server health:

A sample script server_health_check.vbs is provided in the install location. This script can be used to check the health of each service using the following command:

```
server_health_check.vbs (admin|license|paging) ipaddress port
```

The script will return a 0 if the service is running and a 1 if it is not.

Backup Recovery

It is strongly recommended that all system data be backed up for recovery purposes. The Cloudpaging Server stores all system, topology, application, and usage data within the database. All information is contained in the **StreamDB** database. By backing up the database, you will be able to recover all system data. Use standard backup procedures common to database servers.

Appsets, stored in the **Repository**, should also be backed up to a redundant system in case the system needs to be recovered using standard backup procedures for data files. The Cloudpaging Server will verify that the local cache versions are valid based on a CRC stored in the database and if not valid, the files will be restored from the repository.

In addition, for a fault-tolerant system, it is also recommended to replicate the database for fail-over.

Database Tuning

The Cloudpaging Server database StreamDB comes with predefined indices for all important tables, such as StApplication, StLicense, and StSession. Basic performance is therefore already optimized with the installed version. However, for a Cloudpaging Server system that has been in service for a long period of time, additional tuning is recommended to ensure optimal performance.

The additional tuning strategy will be unique for each organization. This is due to the fact that each system will eventually contain different clusters of data that are unique to that organization. For instance, one organization may have more applications than servers; and in that case, optimization to the StApplication table is more important compared to the StServer table.

Microsoft SQL Enterprise Edition provides two built-in Profiler and Tuning-Advisor utilities for that purpose:

- Profiler How-To: http://msdn.microsoft.com/en-us/library/ff650699.aspx
- Tuning How-To: http://msdn.microsoft.com/en-us/library/ms166575.aspx

Database Administrator (DBA) should read these articles and perform recommended tasks to gain performance improvements periodically. In general, tuning procedure includes:

- Running Profiler using **Tuning** template for a period of time (e.g., one hour of normal use).
- Opening the saved Profiler output in a new Session in Tuning-Advisor.
- Running **Start Analysis** in Tuning-Advisor.
- Inspecting the Partition and Index Recommendations for valid use-case.
- Applying the recommendations and collecting the new performance baseline.

> ***NOTE***
>
> *Additional disk space on the database server might be required to store additional index and statistics. It is up to the DBA to find a balance between space and speed.*

Pruning Database Records

The size of the database tables will increase over time, especially the StUsage, StAuditSession, and StServerLog tables. If any table has grown too large, administrators can archive or delete old records. These tables can be safely truncated and/or backed up without affecting server functionalities as they are for auditing purposes only. For more information on pruning database records, visit our Support article https://support.numecent.com/support/solutions/articles/1000199272.

Refining tuning policy, collecting performance baselines, and reducing database size by pruning records are good practices for a long-lasting and healthy user experience.

Chapter 7: Reports

The Cloudpaging Server has the capability of generating reports that can be used for different purposes such as auditing, historical usage, etc. It is also possible to generate custom reports using popular reporting tools such as Microsoft Access and Crystal Reports with the Cloudpaging Server APIs.

The following are explained in this section:

- Reports
- Reports API
  - License Usage
  - Application Usage
  - Paging Usage Reports
- Custom Reports
  - Cloudpaging Database Structure
  - Importing Data from the Cloudpaging Database
  - Sample Queries

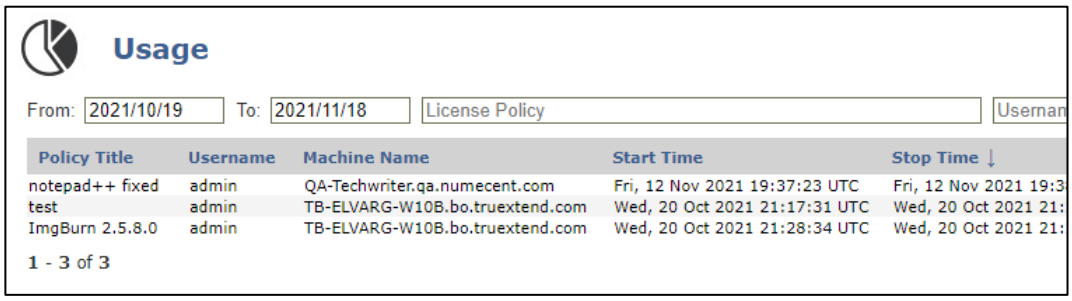
Reports

Cloudpaging Server provides a report to view information about application usage. This report can be used to understand when application have been running, when, where, and the duration.

To generate a report:

1. From the Reports menu, select the Usage report.
2. If you need to limit the scope of the report, set the various filter parameters appropriately. Otherwise, leave entries blank and options set to default values.
    a. **From**: This is the starting date of the report data to be displayed. Must be in yyyy/mm/dd format for year, month, and day.
    b. **To**: This is the ending date for the report data to be displayed. Must be in yyyy/mm/dd format for year, month, and day.
    c. **License Policy:** The name of the license policy for the application used to filter the report results. Partial matching and wildcard (*) searches are allowed.
    d. **Username**: The name of the user that will be used to filter the report results. Partial matching and wildcard (*) searches are allowed.
    e. **Machine Name:** The machine name that will be used to filter the report results. Partial matching and wildcard (*) searches are allowed.

A report similar to the one shown in the figure below will be displayed.



- Click on a column heading to sort the list according to that heading
- Click on the current sort column to toggle between ascending and descending sort order

> ***NOTE***
>
> *Deleting a user or application will not affect the historical information reported in the Standard Usage report.*

Reports API

There may be times when you want to query for past usage information about a specific application or user. A set of HTTP API's are available to service these types of requests and the response is formatted in CSV format.

Standard Usage

Format

https://cloudpaging.corp.com/jukeboxserver/usageReport.do?<parameter>=<value>&<parameter>=<value>

Parameter	Description
accessKey	**Required**. The Cloudpaging server's **web service key**.
appGuid	Filters the report by application, using the application's GUID.
licenseguid	Filters the report by a license policy. If this is supplied, the service ignores the appGuid.
userName	Filter the report by user name.
startTime	Filter by session start times after the value. Parameter must be UTC time in this format: **yyyy-MM-dd hh:mm:ss.S**
endTime	Filter by session end times before the value. Parameter must be UTC time in this format: **yyyy-MM-dd hh:mm:ss.S**
physicalIP	Filter by user's physical IP address.
dataType	Current or Past (Default). If current, report includes only currently active sections.
duration	Used with the next two parameters to filter the report by duration of session. This is an integer value.
durationUnits	**Seconds, Minutes, Hours,** or **Days**. Time unit for the duration parameter.
durationComparison	Possible values are **Equal To, Greater Than, Greater or Equal To, Less Than, Less Than,** or **Equal To**. This is used with the other duration parameters to filter the report by duration.
pageSize	The maximum number of records returned with the report. Maximum value = 1000.
pageNumber	The service uses this number to calculate the row offset.
clientReport	**True** or **False**. The default is **False**. If **True**, the report will include client information, including OS type, player version, and so on.

Example

https://cloudpaging.corp.com/jukeboxserver/usageReport.do?AccessKey=7a474BC5-6090-493E-A8D9-28ABA099B7C5&appGuid=19FECDA3-49D4-4F9E-A6B6-5F9D4D603DF9

License Usage

Format

https://cloudpaging.corp.com/jukeboxserver/licenseReport.do?<parameter>=<value>&<parameter>=<value>

Parameter	Description
accessKey	**Required**. The Cloudpaging server's **web service key**.
appGuid	Filters the report by application, using the application's GUID.
licenseguid	Filters the report by a license policy. If this is supplied, the service ignores the appGuid.
startTime	Filter by session start times after the value. Parameter must be UTC time in this format: **yyyy-MM-dd hh:mm:ss.S**
pageSize	The maximum number of records returned with the report. Maximum value = 1000.
pageNumber	The service uses this number to calculate the row offset.

Example

https://cloudpaging.corp.com/jukeboxserver/licenseReport.do?accessKey=7a474BC5-6090-493E-A8D9-28ABA099B7C5

Application Usage

Format

https://cloudpaging.corp.com/jukeboxserver/applicationReport.do?<parameter>=<value>&<parameter>=<value>

Parameter	Description
accessKey	**Required**. The Cloudpaging server's **web service key**.
appGuid	Filters the report by application, using the application's GUID.
licenseguid	Filters the report by a license policy. If this is supplied, the service ignores the appGuid.
startTime	Filter by session start times after the value. Parameter must be UTC time in this format: **yyyy-MM-dd hh:mm:ss.S**
endTime	Filter by session end times before the value. Parameter must be UTC time in this format: **yyyy-MM-dd hh:mm:ss.S**
pageSize	The maximum number of records returned with the report. Maximum value = 1000.
pageNumber	The service uses this number to calculate the row offset.

Example

https://cloudpaging.corp.com/jukeboxserver/applicationReport.do?accessKey=7a474BC5-6090-493EA8D9-28ABA099B7C5&startTime=2013-12-01 00:00:00.0&endTime=2014-01-30 23:59:59.0

Paging Usage Reports

The default number of records returned from a usage report query is 100 records. The number of records returned from a usage report query is 1000, which can be set using the 'pageSize' parameter. The results of one usage report request is known as a 'page'. By parsing special HTTP response headers the user can tell making another request for the next 'page' or previous 'page' will return additional usage records. This concept is often referred to as 'paging'.

The usage report HTTP response header contains two custom headers that contain information related to paging usage reports. The names of the two HTTP response headers are 'numecent-page-next' and 'numecent-page-prev'. The 'numecent-page-next' header's value will indicate if making another request for the next page will return more results. If the value of the 'numecent-page-next' header is not equal to the string 'null' appending the value of the header to the original HTTP usage report request will return the next page of usage results. If the value of the 'numecent-page-next' header is equal to the string 'null' making an HTTP request for next page will return no additional results. The 'numecent-page-prev' header will indicate if making another usage report request for the previous page will return any results. If the value of the 'numecent-page-prev' is not equal to the string 'null' appending the value of the header to the original HTTP usage report request will return the previous page of usage report records. If the value of the 'numecent-page-prev' header is equal to the string value 'null' making a request for the previous page will not return any results.

Example of Usage report paging

The initial usage report request is just a default request with no additional filter set in the parameters. The response of this request will return 100 usage report records.

HTTP Request 1:

https://cloudpaging.corp.com/jukeboxserver/usageReport.do?accessKey=129cf1a0-f7a7-11e1-a21f-0800200c9a66

HTTP Response 1:

```
Cache-Control:no-cache, no-store, must-revalidate
Content-disposition:attachment; filename=usage.csv
Content-Type:text/html;charset=utf-8
Date:Tue, 11 Mar 2014 20:47:11 GMT
Expires:Thu, 01 Jan 1970 00:00:00 GMT
numecent-page-next:&pageNumber=2&pageSize=100
numecent-page-prev:null
Pragma:no-cache
```

Looking at the 'numecent-page-next' and 'numecent-page-prev' headers in the HTTP response we can determine that making an additional request for the next page will return more usage records, however making a request for the previous page will not return any usage report records. The next example shows the request and response to get the next usage report page.

HTTP Request 2:

```
https://cloudpaging.corp.com/jukeboxserver/
usageReport.do?accessKey=129cf1a0-f7a7-11e1-a21f-
0800200c9a66&pageNumber=2&pageSize=100
```

HTTP Response 2:

```
Cache-Control:no-cache, no-store, must-revalidate
Content-disposition:attachment; filename=usage.csv
Content-Type:text/html;charset=utf-8
Date:Tue, 11 Mar 2014 20:48:40 GMT
Expires:Thu, 01 Jan 1970 00:00:00 GMT
numecent-page-next:null
numecent-page-prev:&pageNumber=1&pageSize=100
Pragma:no-cache
```

Looking at the 'numecent-page-next' and 'numecent-page-prev' headers in the HTTP response we can determine that making an additional request for the next page will not return any usage report records. However making a request for the previous page will return additional usage records, which should be the same set return in HTTP Request 1.

The next example shows the request and response to get the previous report page.

HTTP Request 3:

```
https://cloudpaging.corp.com/jukeboxserver/
usageReport.do?accessKey=129cf1a0-f7a7-11e1-a21f-
0800200c9a66&pageNumber=1&pageSize=100
```

HTTP Response 3:

```
Cache-Control:no-cache, no-store, must-revalidate
Content-disposition:attachment; filename=usage.csv
Content-Type:text/html;charset=utf-8
Date:Tue, 11 Mar 2014 20:49:55 GMT
Expires:Thu, 01 Jan 1970 00:00:00 GMT
numecent-page-next:&pageNumber=2&pageSize=100
numecent-page-prev:null
Pragma:no-cache
```

Custom Reports

If you have specific reporting needs that go beyond what the standard reports offer, you can generate tailored reports outside of the Cloudpaging Server using popular client reporting tools such as Microsoft Access and Crystal Reports that can directly access the Cloudpaging Server database.

This section demonstrates how to accomplish just that. Specifically, it will:

- Describe relevant parts of the Cloudpaging Server database structure.
- Explain how to connect to the database using Microsoft Access 2007.
- Provide some sample queries to generate reports from a client reporting software.

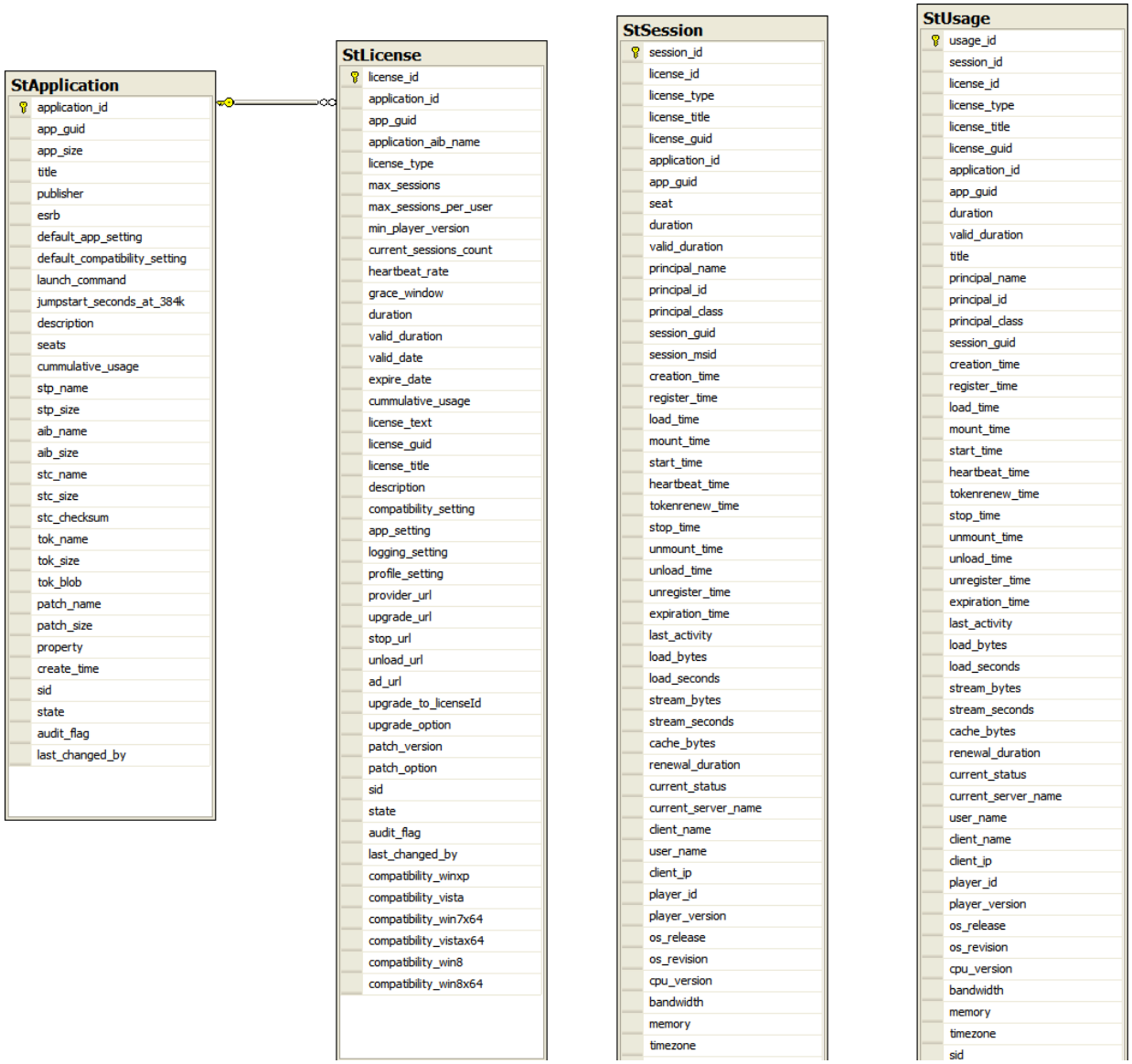
This will provide you with the tools you need to create your own custom reports.

Cloudpaging Database Structure

The Cloudpaging Server database is divided into four groups: Server Management, User Management, Application Management, and Usage Management. The schemas for the Usage Management and Audit Database are presented in the sections.

Usage Management Schema

Usage Management tables store past historical data from the StSession table. It can be used to generate reports on applications usage, such as when and for how long an application was ran by a specific user. This metered data can then be used to better understand application daily usage patterns, application license consumption, and applications used by user.



Audit Database Schema

Audit tables store the accumulative data from the StApplication, StLicense, and StSession tables. They are used mainly for querying and generating usage reports.

StAuditApplication

audit_id
audit_on
audit_column_mask
application_id
app_guid
app_size
title
publisher
esrb
default_app_setting
default_compatibility_setting
launch_command
jumpstart_seconds_at_384k
description
seats
cummulative_usage
stp_name
stp_size
aib_name
aib_size
stc_name
stc_size
stc_checksum
tok_name
tok_size
tok_blob
patch_name
patch_size
create_time
sid
state
audit_flag
last_changed_by
property

StAuditLicense

audit_id
audit_on
audit_column_mask
license_id
application_id
app_guid
application_aib_name
license_type
max_sessions
max_sessions_per_user
min_player_version
current_sessions_count
heartbeat_rate
grace_window
duration
valid_duration
valid_date
expire_date
cummulative_usage
license_text
license_guid
license_title
description
compatibility_setting
app_setting
logging_setting
profile_setting
provider_url
upgrade_url
stop_url
unload_url
ad_url
upgrade_to_licenseId
upgrade_option
patch_version
patch_option
sid
state
audit_flag
last_changed_by
compatibility_winxp
compatibility_vista
compatibility_win7x64
compatibility_vistax64
compatibility_win8
compatibility_win8x64

StAuditSession

audit_id
audit_on
audit_column_mask
session_id
license_id
license_type
license_title
license_guid
application_id
app_guid
seat
duration
valid_duration
principal_name
principal_id
principal_class
session_guid
session_msid
creation_time
register_time
load_time
mount_time
start_time
heartbeat_time
tokenrenew_time
stop_time
unmount_time
unload_time
unregister_time
expiration_time
last_activity
load_bytes
load_seconds
stream_bytes
stream_seconds
cache_bytes
renewal_duration
current_status
current_server_name
client_name
user_name
client_ip
player_id
player_version
os_release
os_revision
cpu_version
bandwidth
memory
timezone
sid
state
audit_flag
last_changed_by
session_key_ex
process_name
external_ip
min_player_version
provider_name

Some database fields use specific numerical ID's for states or statuses. Below is a listing by table to show what the various ID's mean:

License Policy Types

Session	Current Status	Description
1	**fixed**	Fixed license policy is where the seat is checked out on activation and checked back in when the application is removed.
2	**floating**	Floating license policy is where the seat is checked out as the application is being launched and checked back in when stopped.
3	**metered-only**	Like Fixed license policy whose seat is checked out when an application is activated, Metered-only allows unlimited offline duration and invalidation (inactivity) times. This license policy type will never expire and is allowed to go offline as long as it needs it. With this license policy type, the Player performs less frequent heartbeats than other types of license policy.

License Policy Upgrades

Session	Current Status	Description
1	**forced major**	Forced upgrade from one application to another application.
2	**optional major**	Optional upgrade from one application to another application.
3	**forced minor**	Forced upgrade to a patch of the application.
4	**optional minor**	Optional upgrade to a patch of the application.

Session Status

Session	Current Status	Description
0	**invalid**	(reserved flag – not currently used)
1	**pending**	Temporary session created for Cloudpaging Player to initialize paging
2	**registered**	Application added to Cloudpaging Player application list
3	**loaded**	Cloudpaging Player downloaded all application components (such as, AIB, EULA, Custom AppEvents, icons)
4	**mounted**	Application has been successfully virtualized (such as registry, file associations, fonts)
5	**running**	Application is currently running
6	**heartbeat**	Cloudpaging Player sends confirmation that application is running as well as usage data
7	**evicted**	Admin revokes session so that the application is no longer usable

8	**stopped**	Application has stopped running
9	**unmounted**	Application is devirtualized
10	**unloaded**	Cloudpaging Player has removed downloaded data (such as AIB, TOK, and EULA)
11	**unregistered**	Application has been removed from the Cloudpaging Player application list
12	**expired**	Application license policy has expired
13	**upgraded**	Application has been upgraded
14	**error**	Application error

Importing Data from the Cloudpaging Database

Microsoft Access will be used to create a project comprised of data imported from the Cloudpaging Database. Once connected to the server, two examples will be given that illustrate database queries that produce reports. To import data from the Cloudpaging Database to an Access database, do the following:

1. Start Microsoft Access.
2. Select New.
3. Enter a name for your Access project file, then click Create.
4. Follow the instructions for importing SQL data to an Access database from the CloudpagingDatabase found here: https://support.numecent.com/solution/articles/1000259075-linking-cloudpaging-with-an-accessdatabase-for-additional-reporting
   Microsoft Access will now connect to the Cloudpaging Database.

Sample Queries

This section presents two samples of queries to report on the following Cloudpaging Server information:

- Application usage for a specific user
- License usage for a specific time period

Sample Query 1: Application Usage for a specific user

This query will return all paging applications usage for a specific user.

Input: User name (e.g. *'Jim'*).

Output: Listing of all paging applications used by that user.

Sample Query #1

```
SELECT
title, user_name, start_time, stream_seconds, stream_bytes,
load_bytes
FROM
StUsage
WHERE
user_name like 'admin'
ORDER BY
title asc
```


	title	user_name	start_time	stream_seconds	stream_bytes	load_bytes
1	Gvim71	admin	2008-03-13 00:06:36.000	119	999638	0
2	Gvim71	admin	2008-03-13 00:20:47.000	70	999638	0
3	Gvim71	admin	2008-03-20 18:06:25.000	7	118742	0
4	Gvim71	admin	2008-03-20 18:06:36.000	29	120674	0

Sample Query 2: Application Usage for a specific time period

This query will return the utilization data within a specified period of time.

Input: All of the following:

- Start time (e.g., 'Dec/1/2012/)
- End time (e.g., 'Dec/13/2012/)

Output: Listing of the total usage for each license between the start and end dates.

Sample Query #2

```
SELECT
license_title, license_type, COUNT(usage_id) as total_sessions,
sum(stream_seconds) as total_stream_seconds
FROM
StUsage
WHERE
start_time >= '12/1/2012' and start_time < '12/13/2012'
GROUP BY
license_title, license_type
ORDER BY
license_title asc
```

	license_title	license_type	total_sessions	total_stream_seconds
1	Firefox usage all OS OK	2	7	1078
2	Gvim71	2	55	1608
3	Gvim71 fixed	1	1	3
4	Gvim71 Usage	2	4	11
5	TextPad	2	1	3
6	TextPad fixed	1	1	34
7	TextPad usage	2	7	41
8	TextPad usage 1	2	1	5

Glossary

The definitions in this glossary are in the context of Cloudpaging Server and may not necessarily correspond to more general definitions.

Term	Definition
access token	A file that is used to monitor the license agreement of a cloudified application, which is passed back and forth between Cloudpaging Player and Cloudpaging Server. When Cloudpaging Player receives an access token, permission is granted to run the application. If a user is running multiple or expired sessions, Cloudpaging Player may be denied an access token.
activation	The first of six stages that launch an application from Cloudpaging Player.
admin	See **system administrator**
Admin service	The Cloudpaging Server component that provides configurable web-based interfaces for the administrator, as well as web interfaces for users to launch cloudified applications.
Application Installation Blueprint (AIB) file	An Application Installation Blueprint (AIB) file contains all the files' metadata and registry information for the appset. The extension for this file is .aib.
AppEvent	A Cloudpaging Studio process or action that uses a *trigger* to run an executable program or script. In turn, the script contains instructions to perform one or more specified runtime actions that are required by the application being cloudified, such as launching another application. AppEvent scripts and executable programs can be written in any scripting or programming language. AppEvents are also called *Configurable AppEvents* (*CAEs)*.
application	Any Windows software program or game, such as Microsoft Word or Adobe Acrobat. See also **cloudified application**.
application license policy	A policy that controls the conditions under which users can access and use an application. These conditions are defined by settings such as license type, total number seats, offline duration (fixed licenses only), etc.
appset	See **cloudified application**

artifacts	For Cloudpaging Server, artifacts are unwanted changes or additions, such as DLLs, that occur most often when “All processes” is selected during the application installation capture portion of the packaging process.
authentication	The process by which the system validates a user's logon information. The user's name and password are compared against an authorized list, and if the system identifies a match, access is granted to the extent specified in the permission list for the user.
cache	A temporary storage place (often a specified portion of computer disk space or RAM memory) for frequently needed data that can be retrieved quickly. See also Client cache.
Configurable AppEvents (CAE)	There is one CAE file per AppEvent trigger which generates a response such as running a VB script that installs other needed applications. See also AppEvent.
Common Gateway Interface (CGI) script	A script that uses the CGI (Common Gateway Interface) protocol for interfacing external application software with an information server, commonly a web server. In the case of a web server, the CGI script responds to requests from client web browsers by returning output. Each time a request is received, the server analyzes what the request asks for, and returns the appropriate output.
clean packaging PC	For Cloudpaging Studio, a PC that has the Windows operating system installed, as well as the most recent drivers (and possibly imaging software), but is free of any other unnecessary applications.
client	For Cloudpaging Server, a PC with Cloudpaging Player installed. In general, a client is a computer that accesses shared network resources provided by another computer, called a server. Also, a client can be an application or process that requests a service from a process or component. In a client-server environment, the workstation is usually the client computer.
cloudified application	An application that has been converted to a cloudified format using Cloudpaging Studio. A user can access this program from any Windows computer with an Internet broadband or company intranet connection using Cloudpaging Server, without installing the program on a client PC. There are three different names for a “cloudified application” depending on the point of

	view (POV) of the person using it: • From a Appset Designer's POV - *"appset"* • From a Cloudpaging Server admin's POV - *"application"* • From a user's POV - *"cloudified application"*
cloudifying	The process of converting a software application into a form that can be cloudpaged to users' PCs. See also **cloudified application**.
Cloudpaging	The proprietary protocols and technologies developed by Numecent which allow cloudified applications to be delivered from Cloudpaging Server to a client device on demand.
Cloudpaging database	The Cloudpaging Server component that tracks user information and server resources for a given enterprise. All system and user information is stored in this database, including system configuration information, topology, logs, application information, application licenses, usage, and user account information. See also **Database Server**.
Cloudpaging service	This is a general term that refers to an Cloudpaging Server service: *Paging service, License service,* or *Admin service*.
Configurable AppEvent	See **AppEvent**
database server	A machine that stores profile, usage, and system information of Cloudpaging Player users. The database server is a component of the Cloudpaging Server system. See also **Cloudpaging database**.
disposition layers	The layers used to configure individual system resources, such as a file, folder, registry key, or registry value. The layer determines whether the system resource can be seen by the local system, and whether it is permanent or can be removed. **Layer 1 (Installed-permanent)** Copies assets (files, folder, registry keys, and registry values) permanently onto the local system, and can be seen by the entire local system. **Layer 2 (Installed-temporary):** Installs assets during the activation process, and uninstalls assets during the deactivation process. The original asset is backed up before the new asset is installed, and when the new asset is uninstalled, the original asset is restored. **Layer 3 (Virtual-integrated):** Assets that can be seen both by the virtualized application and the local system, but are not physically installed on the local system.

	Layer 4 (Virtual Isolated): The default setting. Assets that can only be seen by the virtualized application, and are not physically installed on the local system. See also **virtualization**.
Data Source Name (DSN) file	This file is used in the process of linking a Cloudpaging database to a Microsoft Access database.
End User License Agreement (EULA)	This is typically an HTML file that consists of a legal agreement the user acknowledges regarding the warranties and conditions of use for a software program.
fixed license	An application license that permits access to the application when the user is working offline (not connected to the enterprise's network). NOTE: A fixed license is in use when an application is in the Ready or Running state. See also **floating license**.
floating license	An application license that requires the user's PC to be online (connected to the enterprise's network) in order to access the application. With this license the application cannot be accessed offline. NOTE: A floating license is only in use when an application is Running, but not when it is in the Ready state. See also **fixed license**.
Fully Qualified Domain Name (FQDN)	An unambiguous domain name that specifies the exact location in the Domain Name System's tree hierarchy through to a top-level domain, and finally to the root name server. Some applications, such as web browsers, will try to qualify the domain name part of a Uniform Resource Locator (URL) if the DNS resolver cannot find the domain. An FQDN differs from a regular domain name by its absoluteness; a default domain name will not be added.
Globally Unique Identifier (GUID)	16-byte code that identifies an interface to an object across all computers and networks. Such an identifier is unique because it contains a time stamp and a code based on the network address hard-wired on the host computer's LAN interface card. These identifiers are generated by a utility program.
fileshare	A folder and its contents that is shared with other users on a network.
firewall	A firewall is a software and/or hardware barrier that protects private and company information from

	external threats. For companies, all communication is routed through a proxy server outside of the organization's network, and the proxy server decides whether it is safe to let a particular message or file pass. The proxy server hides the true network addresses.
ICO file (application icon file)	This is the graphic file for the application icon provided by the software vendor (for example, the Word icon for each Word file).
keytab file	Used by Cloudpaging Server/Enterprise Portal, this file contains pairs of Kerberos principals and DES-encrypted keys (derived from the Kerberos protocol). The Keytab file can be used to log into Kerberos without being prompted for a password. The most common use of keytab files, however, is to allow scripts to authenticate to Kerberos without human interaction or store a password in a plaintext file.
license key	A key used by software products to protect against piracy. Typically a license key includes a digital signature that locks to a specific computer to ensure the software cannot be executed on other computers.
License Service	A Cloudpaging Server component that manages *application licenses* and meters application usage. The primary functions are to grant, renew, and delete access tokens, record application usage, and help with server *load balancing*.
load balancing	The process of distributing processing and communications activity evenly across a computer network so that no single device is overwhelmed. Load balancing is especially important for networks for which it is difficult to predict the number of requests that will be issued to a server. If one server starts to get swamped, requests are forwarded to another server with more available capacity. Load balancing refers to a server cluster sharing information requests equally across all of its active nodes. This can be done either statically, by tying clients directly to different back-end servers, or dynamically by having each client tied to a different back-end server controlled by software or a hardware device. See also **failover**.
local paging	The process used by Cloudpaging Player to cloudpage an appset from a file located on the Target

	PC. Local paging allows you to test several appset iterations quickly, without publishing the appset to Cloudpaging Server.
log file	A file that stores messages generated by an application, service, or operating system. These messages are used to track the operations performed.
machine	See **server**
network segment	A portion of a network separated by a Layer 3 switch.
node	For Network topology, equipment situated at a point of branching of physical connections, or terminating a physical connection.
Appset Designer	The person responsible for using Cloudpaging Studio to convert Windows-based applications into a cloudified format.
page	A fixed-size portion of an application that is fetched from the Cloudpaging Server system to Cloudpaging Player on users' PCs.
paging	The process of transferring a cloudified application from Cloudpaging Server to a user's PC.
persistent binding	A setting that allows network switching hardware to constantly route traffic between two computers so that a *session* is maintained. With load-balancing switches, or Layer 3 switches, persistent binding settings facilitate this constant route instead of using the most efficient route.
Player Cache	The segment of disk storage on the Cloudpaging Player PC that temporarily stores pages that are paged from an enterprise's Cloudpaging Server. See also **cache**.
Prefetching	The process of fetching application pages prior to their being explicitly requested by Cloudpaging Player, in anticipation of being needed by the application. **Stage-1 Prefetch:** This group of application pages are those that are needed to start the application, as well as to perform typical actions for the first few minutes of using the program, such as opening files, saving files, etc. **Stage-2 Prefetch:** This group of pre-cloudpaged data consists of application pages that are needed to

	accomplish other important tasks the user is likely to perform.
product license key	The license key that is required in order to start the Cloudpaging Server system and register users.
project file	See **Project Workspace**.
Project Workspace	The Cloudpaging Studio file (.stw) that stores all packaging configuration settings for an application that is being cloudified. This file is also known as the *project file*.
protocol	Set of rules and conventions for sending information over a network. These rules govern the content, format, timing, sequencing, and error control of messages exchanged among network devices.
realm	A database of user names and passwords that verifies valid users of a web service or application and displays a list of roles associated with each valid user.
Repository	The Cloudpaging Server folder where all cloudified applications are stored. This folder must be accessible by all other Cloudpaging Server components.
scaled topology	The installation of Cloudpaging Server components on multiple machines. For example, a multi-machine topology might include: *Paging*, *License*, and *Admin services* each installed on three separate machines, duplicates of these three machines for *load balancing* and *failover*; a machine for the **Cloudpaging** *database*; and (for Enterprise Portal) a machine for the Web Portal.
seats	The total number of users that can concurrently run an application that is authorized by a particular application license.
security	Techniques for ensuring that data stored in a computer, or transferred from one computer to another, cannot be read or compromised. Most security measures involve data encryption and passwords. Data encryption is the translation of data into a form that is unintelligible without a deciphering mechanism. This typically requires a password, which is a secret word or phrase that a user enters to gain access to a particular program or system.
symmetric key	See **session key.**
server	A computer (or “machine”) onto which Cloudpaging Server components or other system components are installed.

service	For Cloudpaging Server, the sub-component of of the system that delivers application *pages* to *clients* and provides other Cloudpaging Server functions. See also **Cloudpaging service** and **Windows service**.
service interruption	An interruption in the delivery of pages from Cloudpaging Server. Cloudpaging Player handles and recovers from service interruptions that may occur during normal use. For example, Internet traffic can affect how data travels to and from networks.
session	An open line of communication between a user and another computer, or between two computers. A session can remain open even if no communication is on-going. This allows for the communication to continue at a later time. Once a session has been evicted, then the communication will no longer be allowed and a new session must be established.
session key	An encrypted key that is randomly generated to ensure the security of a communications session between a user and another computer or between two computers. Session keys are sometimes called symmetric keys, because the same key is used for both encryption and decryption. A session key can contain information about the user that is communicating.
signed certificates	Web sites use SSL certificates to secure communication with a web browser. A certificate is digitally signed by the creator and this signature is used by application software to validate the legitimacy of the certificate by others. Most web sites use a signed certificate issues from a certificate authority (CA), such as Thawte or Verisign.
STC file	See **secure sockets layer**.
STP file	This file contains all the data pages of the application being cloudified. This file is the largest portion of the STP file.
Paging service	The Cloudpaging Server component that stores, manages, and pages cloudified applications to Cloudpaging Player on user's PCs. The Paging service adheres to restrictions established by the *License service*. See also **License Service**.
strong password	A password that satisfies these basic guidelines to protect against password-guessing system attacks: • Include numbers, punctuation, and upper and lowercase letters in the password.

	• Use a password having at least eight characters. • Avoid a password based on repetition, dictionary words, letter or number sequences, user names, or biographical information such as names or dates.
system administrator	The person who manages the enterprise's Cloudpaging Server. In general, the person responsible for setting up and managing domain controllers or local computers and their user and group accounts, assigning passwords and permissions, and helping users with networking issues.
target application	An application that has been selected to be cloudified using Cloudpaging Studio.
templates	Aliases that map to actual Windows folder paths. For example, WindowsDir maps to c:\windows.
token file (TOK)	This file contains the type of license and other security data for the cloudified application. It also contains general information about the application, including location of servers and other paging sources, game rating, and operation system compatibility. The token file extension is **.tok**.
token renewal frequency	Frequency at which Cloudpaging Player must renew its access tokens in order to continue using Cloudpaging Server applications. Currently, the frequency is set at 10 minutes.
Tomcat	Apache Tomcat is a web container, or application server developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management, but can also be configured by editing configuration files that are normally XML-formatted. Tomcat includes its own internal HTTP server.
trigger	A process or condition that causes an action to start, such as running a VB script.
virtual machine (VM)	Software that mimics the performance of a hardware device, such as a program that allows applications written for an Intel processor to be run on a Motorola processor.

	A VM is a computer that does not exist as a physical device, but is simulated by another computer.
Virtualization	The process that occurs after activation and prefetch of an appset, which makes the application appear to the user's Windows system as if it were physically installed.
Web Portal	(For Enterprise Portal systems) A website that is integrated with Cloudpaging Player and Cloudpaging Server. Enterprise Portal is the web interface that functions as the front-end that an admin uses to configure and control all Cloudpaging services.
Web server	(For Enterprise Portal systems) The machine that contains the Web Portal.
Windows service	An application that starts when Windows is booted and runs in the background as long as Windows is running. Windows services can have special privileges that regular applications do not. For instance, on a limited user account a service could access certain parts of the Windows Registry that a regular application could not access. Most Windows services do not have a user interface and are loaded when Windows starts.
Windows Internet Name Service (WINS)	Microsoft's implementation of NetBIOS Name Service (NBNS), a name server and service for NetBIOS computer names. Effectively WINS is to NetBIOS names what DNS is to domain names - a central mapping of host names to network addresses. Like DNS, it is broken into two parts: (a) a Server Service that manages the encoded Jet Database, server to server replication, service requests (query & registration, renewals, de-registration), and conflicts; and (b) a TCP/IP Client component which manages the client's registration, renewal of names and takes care of queries.