Vocera EVS Integration Deployment Guide

Version 1.1.1

Notice

Last modified: 2023-02-08 09:19

EVS-111Rel-Docs build 14

Contents

Welcome to Vocera EVS!

The Vocera Environmental Services Integration (EVS) allows Environmental Services managers to see what their staff members are working on, receive status updates from staff members, and optimize the efficiency of their teams.

EVS Solution Overview

The Vocera Environmental Services Integration solution sends EVS staff members cleaning requests and automated reminders; staff members respond with voice commands that update the request status in the Epic EVS module.

Hospital EVS managers require real-time visibility into the availability of staff members. However, EVS staff members often provide status updates irregularly, because the process of logging into a work station and then into the EVS system takes additional time.

The Vocera Environmental Services Integration not only saves time by allowing staff members to update the EVS system through voice commands, it *initiates the process* by way of automated reminders that users respond to, removing the barrier of requiring staff members to initiate status updates.

The Vocera Environmental Services Integration:

- Sends cleaning requests to EVS staff members. If the recipient is not on network or not logged in, the system retries every 30 seconds and times out after 10 minutes. If a timeout occurs, the request is removed from the Vocera EVS system.
- Sends EVS staff members automated reminders to update their status for the current cleaning request; staff members respond by saying "In progress" or "Complete".
- Updates the status of cleaning requests in the Epic Environmental Services module, providing necessary visibility for managers.

EVS User Experience

Vocera Environmental Services Integration returns additional value from your existing Vocera investment and leverages the experience the user community already has with Vocera devices.

The basic Vocera Environmental Services Integration user interaction is straightforward:

- The Epic EVS module sends a request to an EVS staff member
- The staff member uses the Vocera device to respond that cleaning is in progress, and again when it is complete
- The Epic EVS module updates staff member status
- If the request is canceled by the Epic system for any reason, the Epic EVS module sends the Epic cancellation message to the staff member. EVS will also cancel any scheduled reminders for the clean request.

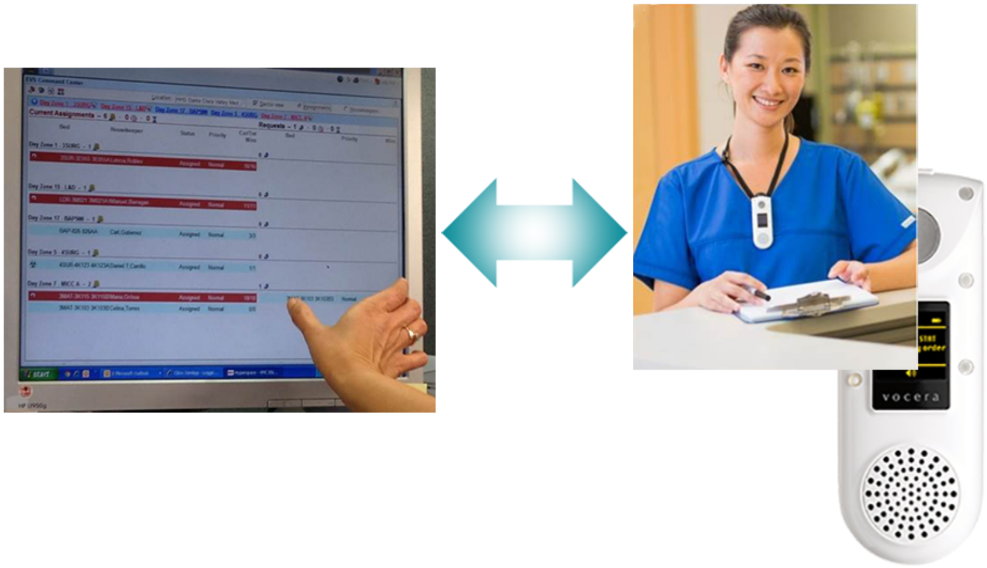
At any time, staff members can further update their status with voice commands.

- The staff member uses the Vocera device to issue the "Start Break" command:

  If the staff member is eligible for a break, the Epic EVS module updates staff member status and the Vocera device tells the user, "Starting Break". If the staff member is not eligible for a break, the Vocera device plays an appropriate response to the user, stating that the request is denied.
- The staff member uses the Vocera device to issue the "End Break" command:

  If a break is in progress, the Epic EVS module updates staff member status and the Vocera device tells the user, "Ending Break". If a break is not in progress, the Vocera device plays an appropriate response to the user.
- The staff member uses the Vocera device to issue the "End Shift" command:

  If a shift is in progress, the Epic EVS module updates staff member status and the Vocera device tells the user, "Ending Shift". If a shift is not in progress, the Vocera device plays an appropriate response to the user, stating that a shift has not been started.



EVS Server Requirements

Install the Vocera Environmental Services Integration solution on a dedicated server that does not host any other Vocera products. You can set up the Vocera Environmental Services Integration solution in either a physical or a virtual environment.

EVS Physical Server Requirements

The server that supports the Vocera Environmental Services Integration solution has the following minimum requirements in a physical environment.

Table 1: Minimum Physical Server Requirements for Vocera Environmental Services Integration Server

Component	Minimum Server Requirement
Processor	Intel Xeon Dual Core
RAM	2 GB
Available Hard Disk Space	25 GB
Operating System	Windows 2008 R2 Standard Windows 2012 R2 Standard

EVS Virtual Server Requirements

The server that supports the Vocera Environmental Services Integration solution has the following minimum requirements in a virtual environment.

You can install the Vocera Environmental Services Integration server in any of the following virtual environments

- VMware Virtual Infrastructure 3.5 (ESX or ESXi)
- VMware vSphere 4 (ESX or ESXi)
- VMware vSphere 5 (ESX or ESXi)

You must also have a VMware vCenter Standard license or greater.

Table 2: Minimum Virtual Resource Reservations for Vocera Environmental Services Integration Server

Component	**Minimum Server Requirement**
Processors per VM	2
RAM per VM	2 GB
Host Resource Reservation	500 MHz CPU 512 MB RAM
Network Interface	VMware vSphere 4: VMXNET 3 VMware ESX 3.5: Enhanced VMXNET (VMXNET 2)
Minimum Disk Space	25 GB
Operating System	Windows 2008 R2 Standard Windows 2012 R2 Standard

Installing the Environmental Services Integration

The EVS installation program provides a wizard that guides you through the process of installing the software. EVS also provides an uninstallation program that intentionally leaves behind files necessary for an upgrade.

Installing EVS Software

The topics in this section show you how to install the EVS software; they also describe the directories and environment variables that the installation program creates.

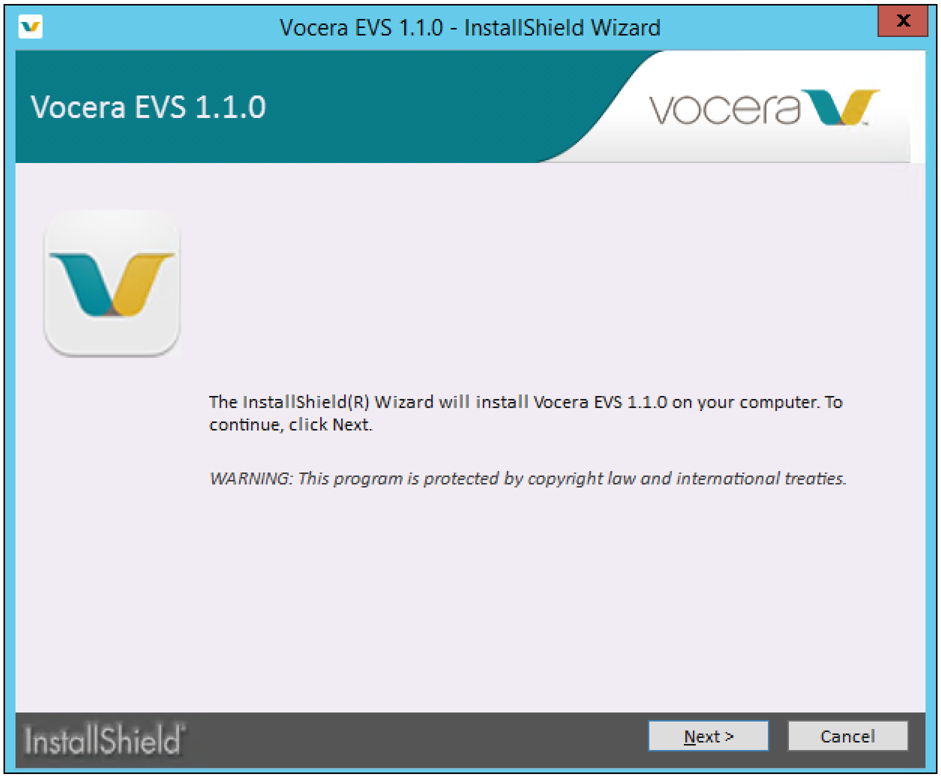
How to Install the EVS Software

Use the steps in this topic to install the Vocera Environmental Services Integration software.

1. Download the Vocera Environmental Services Integration application from Thru.
2. Log into the Vocera Environmental Services Integration computer with administrator privileges.
3. Copy the VoceraEVS-Installer-1.1.0.*BuildNumber*.exe file to the server and double-click it to start the installation.

   BuildNumber is an integer that refers to the current product release.

   The Welcome dialog box appears.
4. Click **Next** in the Welcome dialog box.

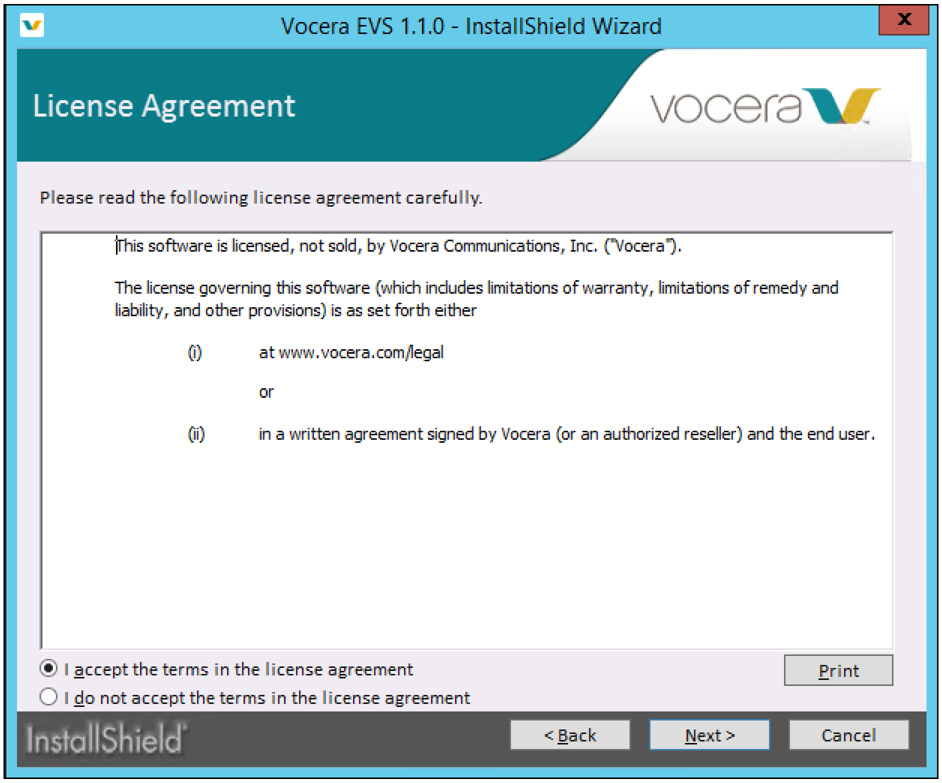


The InstallShield(R) Wizard will install Vocera EVS 1.1.0 on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

The License Agreement dialog box appears.

5. In the License Agreement dialog box, select **I accept the terms in the license agreement** and click **Next**.



Please read the following license agreement carefully.

This software is licensed, not sold, by Vocera Communications, Inc. ("Vocera").

The license governing this software (which includes limitations of warranty, limitations of remedy and liability, and other provisions) is as set forth either

(i) at www.vocera.com/legal

or

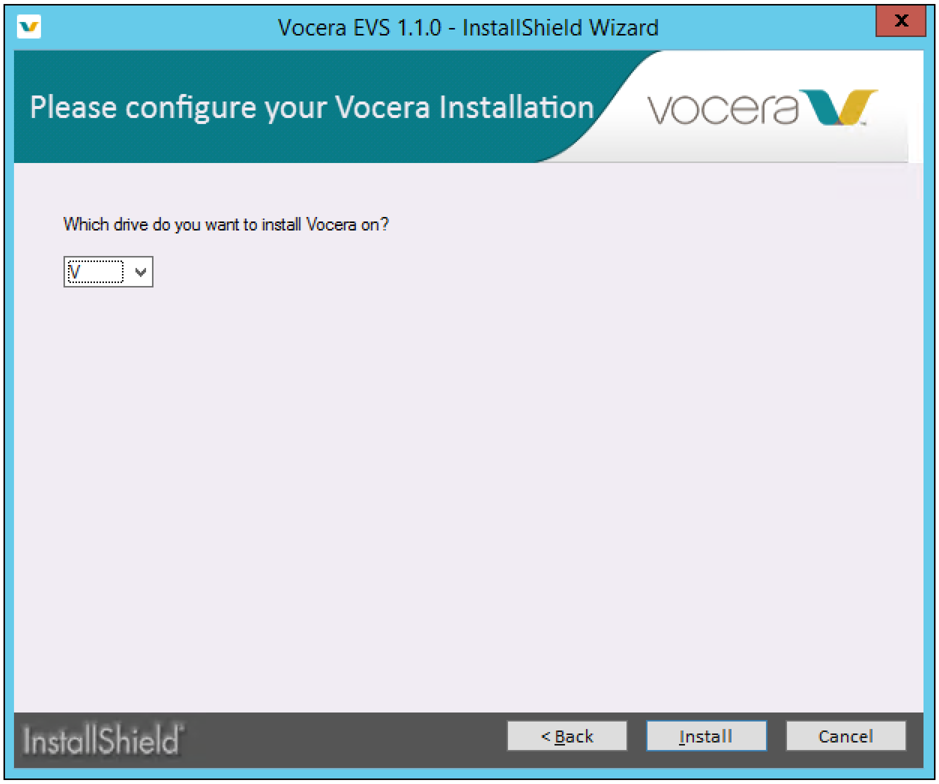
(ii) in a written agreement signed by Vocera (or an authorized reseller) and the end user.

● I accept the terms in the license agreement

○ I do not accept the terms in the license agreement

The Configuration dialog box appears.

6. Specify the installation drive (the default is C) in the Configuration dialog box and click **Install**.

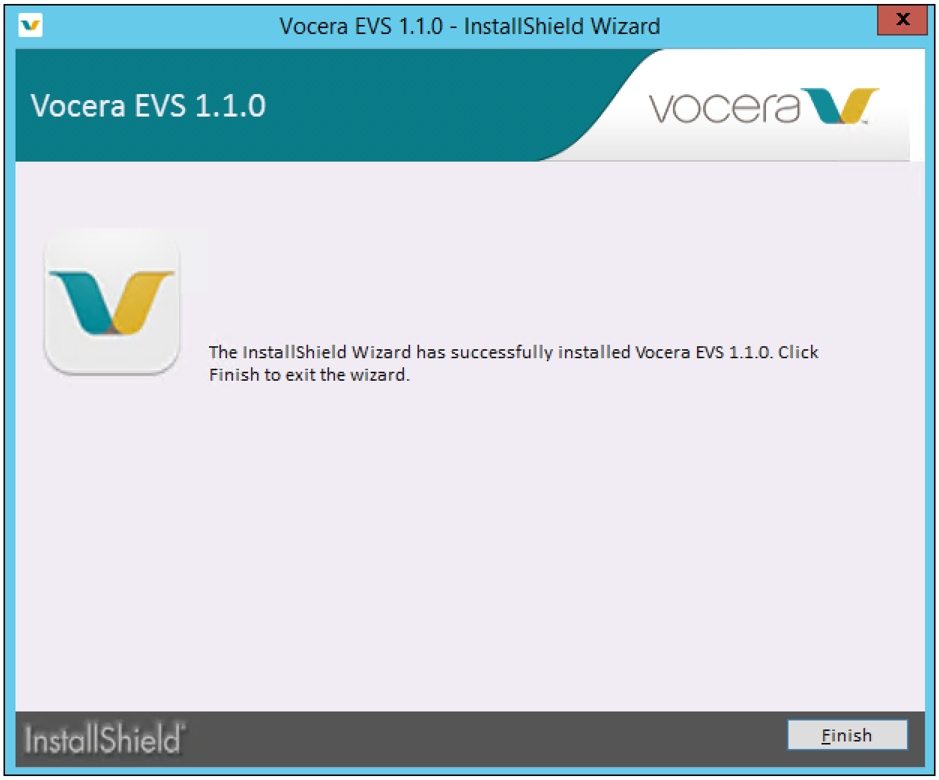

Vocera EVS 1.1.0 - InstallShield Wizard
Please configure your Vocera Installation
vocera
InstallShield
< Back
Install
Cancel

Which drive do you want to install Vocera on?

V

The Installer runs and a progress bar indicates the status.

7. When the installation is complete, click Finish.



The InstallShield Wizard has successfully installed Vocera EVS 1.1.0. Click Finish to exit the wizard.

The installer exits, and EVS application shortcuts (EVS Release Notes and Send Vocera Logs) are visible on the desktop.

The installation is now complete.

Installation Folders

The EVS installation program creates a `vocera` directory at the root of the target drive and creates sub-directories that contain the core product, third-party software, logs, and other ancillary files.

Table 3: EVS Install Folders

Folder	Description
\InstallLogs-Vocera	Installation logs
\vocera\agent	Vocera Agent (messaging broker)
\vocera\evs	EVS application
\vocera\jre17	Java Runtime Environment 1.7.0
\vocera\logs	EVS logs
\vocera\myodbc52	MySQL ODBC Connector
\vocera\mysql51	MySQL database
\vocera\mysql-workbench	MySQL Workbench
\vocera\support	Vocera Support tools (including AutoSendLogs)

Environment Variables

The EVS installation program creates environment variables that the product uses to locate necessary files at runtime.

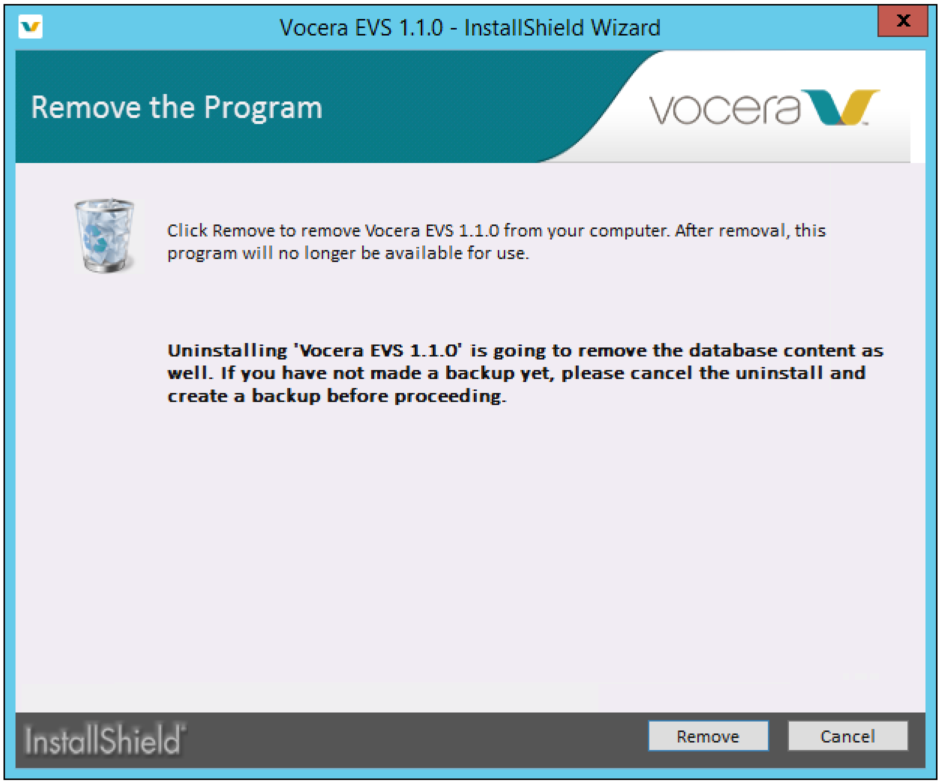
Table 4: Environment variables

Environment variable	Default value
JAVA_HOME	%VOCERA_DRIVE%\vocera\jre17
MYSQL_HOME	%VOCERA_DRIVE%\vocera\mysql51
VOCERA_DRIVE	C: (By default)

How to Uninstall EVS

The EVS uninstallation program removes the product but intentionally leaves behind log files and files necessary to upgrade the software.

1. Choose **Start > Control Panel > Programs > Uninstall a Program**.
2. Right-click **Vocera EVS 1.1.0** and select **Uninstall**.
3. When the **Vocera EVS 1.1.0 InstallShield Wizard** appears, click **Remove** to uninstall the application.

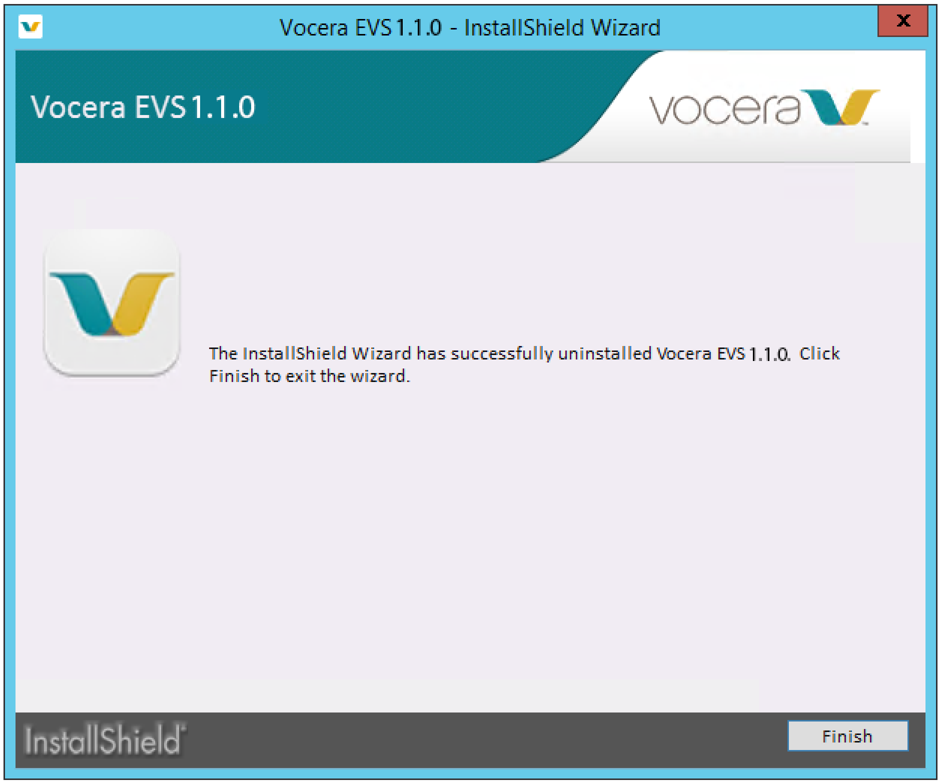


Click Remove to remove Vocera EVS 1.1.0 from your computer. After removal, this program will no longer be available for use.

Uninstalling 'Vocera EVS 1.1.0' is going to remove the database content as well. If you have not made a backup yet, please cancel the uninstall and create a backup before proceeding.

The uninstall proceeds and a progress bar indicates the status.

4. When the uninstall is complete, click **Finish**.



The InstallShield Wizard has successfully uninstalled Vocera EVS 1.1.0. Click Finish to exit the wizard.

The software is removed.

Configuring the Environmental Services Integration

After you install the Vocera Environmental Services Integration software, you need to configure integration parameters to allow it to interact with the Epic Environmental Services module.

Note: When providing field values in an XML file, make sure that special characters are encoded using the proper escape sequences. See *Appendix: Special Characters in XML* on page 47 for more details.

Configuring the System Level Integration

The Vocera Environmental Services Integration uses an XML file to define parameters and defaults for the system-level integration. You can optionally override these defaults with the configuration file provided for each vendor-specific integration (for example, the Epic EVS integration).

The system-level configuration file, `evs.xml`, is located in the `\vocera\evs\resources` directory on the Vocera Environmental Services Integration server.

The content of `evs.xml` is structured XML with begin tags (for example `<retry>`) and end tags (for example `</retry>`). You enter values between the tags to configure the integration parameters.

The `evs.xml` file contains the following main sections:

Section	Description
voice	Provides system-level defaults for message send and delete retries, and also extends the vendor-specific email integration with additional voice integration capabilities.
plugin	Configures the web services retry attempts for cleaning request updates that Vocera sends to the Environmental Services vendor.

Configuring the System-Level Voice Integration

The system-level voice integration allows EVS to send Vocera device users auto-generated reminders about updating the status of cleaning requests; it also sets up system-level defaults for the voice integration. You can optionally override these defaults in the vendor-specific sections of the integration.

Specify any parameters for the system-level voice integration in the `<voice>` section of the `evs.xml` file. You can customize the following main features of the voice integration:

- Completion query
  When a device user indicates that a cleaning request is in progress, Vocera waits a configurable amount of time and then asks if the user wants to mark it as complete (the "Completion query"). If the user says "Yes", Vocera updates the Epic system with the new status; if the user says "No" or does not respond, Vocera waits a designated amount of time and then re-sends the query to the user.
  The configuration parameters for the completion query allow you to customize the behavior of this feature.
- In-progress query

When a user receives a cleaning request, Vocera immediately asks if the user wants to mark it in progress (the "In-progress query"). If the user says "Yes", Vocera updates the Epic system with the status; if the user says "No" or does not respond, Vocera waits a designated amount of time and then re-sends the query to the user.

The configuration parameters for the in-progress query allow you to customize the behavior of this feature.

- Retry settings for message deletion attempts

  Vocera attempts to delete a message from a user's device after every successful state transition. Device users initiate a state transition by acknowledging messages that they receive. For example, when a user responds with a "Yes" to the prompt "Do you want to mark this request in progress", Vocera attempts to delete this message from the user's device. If the deletion fails, the values in the `<delete><retry>` section determine how often Vocera attempts to retry the deletion.

  The `<voice><delete>` section of `evs.xml` specifies system-level defaults for the delete retry feature. You can optionally override these settings in the `pluginproperties.xml` file for each vendor.

- Retry settings for message send attempts

  Vocera sends cleaning requests and status queries to a user's device as part of the Environmental Services integration. If the first delivery attempt for one of these messages fails due to a potentially recoverable error (for example, the user is not logged in), the values in the `<voice><send>` section determine how often Vocera attempts to re-send the message.

  The `<voice><send>` section of `evs.xml` specifies system-level defaults for the message retry feature. You can optionally override these settings in the `pluginproperties.xml` file for each vendor.

- Message escalation settings

  In the EVS integration, an escalation is a message that is sent to an administrator's Vocera device when an unrecoverable error occurs. The following events result in an escalation during the voice integration:

  - Sending a message to an invalid user ID
  - A timeout that occurs as a result of a message retry failure
  - A timeout that occurs as a result of a message delete failure

  The `<voice><escalation>` section of `evs.xml` specifies system-level defaults for message escalations. You can optionally override these settings in the `pluginproperties.xml` file for each vendor.

- Message settings

  In the EVS integration, a message is a cleaning request or reminder that is sent to the Vocera device of an Environmental Services employee. You can customize the tone that the device plays when this message arrives, as well as the priority of the message.

  The `<voice><message>` section of `evs.xml` specifies system-level defaults for the message customizations. You can optionally override these settings in the `pluginproperties.xml` file for each vendor.

How to Configure the System-Level Voice Integration

Configure the system-level voice integration by editing the `<voice>` section of the `evs.xml` file, a structured XML document.

In most situations, the default values Vocera provides in the `<voice>` section of `evs.xml` are sufficient. However, if you want Vocera to send an escalation notification when a message send or deletion fails, you must specify a value for `<voceraid>` in the message escalation parameters. When `<voceraid>` is not specified, Vocera discards the escalation message.

In all other cases, the default values Vocera provides for the parameters in `evs.xml` are sufficient, unless you want to customize the voice integration.

To configure the system-level voice integration:

1. On the Vocera Environmental Services Integration server, open the `\vocera\evs\resources\evs.xml` file in a text editor.

2. Customize the behavior of the completion query by setting values for the tags in the `<configuration><voice><completion-query>` section of the `evs.xml` file.

   Specify a value within each pair of begin and end tags as follows:

Tag	Value

`<configuration>`
`  <voice>`
`    <completion-query>`
`      <nack-reminder-delay/>`
	Use the `<nack-reminder-delay/>` tag to specify the amount of time (in seconds) to wait before sending a follow-up completion query when a user fails to acknowledge the previous query with either a "Yes" or a "No".
	Users must acknowledge receipt of a query by responding to it in either the affirmative or negative. A non-acknowledgement ("nack") results in a retry after a default time of 60 seconds has elapsed. The waiting time for a non-acknowledgement is much shorter than the waiting time for a response of "No", which is determined by `<ack-reminder-delay/>`.

`<configuration>`
`  <voice>`
`    <completion-query>`
`      <ack-timeout/>`
	Use the `<ack-timeout/>` tag to specify the timeout value (in seconds) when a user has not responded to a completion query and has responded to the previous query with a "No". If the specified timeout value is reached, the room cleaning request is escalated to a manager group in the Vocera Voice Server and the task is removed. By default, the timeout value is 7200 seconds.

`<configuration>`
`  <voice>`
`    <completion-query>`
`      <nack-timeout/>`
	Use the `<nack-timeout/>` tag to specify the timeout value (in seconds) when a user has not responded to a completion query and has failed to acknowledge the previous query with either a "Yes" or a "No". If the specified timeout value is reached, the room cleaning request is escalated to a manager group in the Vocera Voice Server and the task is removed. By default, the timeout value is 3600 seconds.

`<configuration>`
`  <voice>`
`    <completion-query>`
`      <delay/>`
	Use the `<delay/>` tag to specify the amount of time (in seconds) to wait before sending a completion query, after a user indicates that a cleaning request is in progress. By default, this time interval is 10 seconds.

`<configuration>`
`  <voice>`
`    <completion-query>`
`      <ack-reminder-delay/>`
	Use the `<ack-reminder-delay/>` tag to specify the amount of time (in seconds) to wait before sending a follow-up completion query when a user responds to the previous query with a "No". By default, the system waits 600 seconds before sending a new completion query to the user.
	Users must acknowledge receipt of a query by responding to it in either the affirmative or negative. The waiting time for a non-acknowledgement, which is determined by the `<nack-reminder-delay/>`, is much shorter than the waiting time for a response of "No".

`<configuration>`
`  <voice>`
`    <completion-query>`
`      <message/>`
	Use the `<message/>` tag to specify a yes/no question that asks the Vocera device user if a cleaning request is complete. The Vocera device auto-enunciates this question for the user based on the `<delay>, <ack-reminder-delay>`, and `<nack-reminder-delay>` timers. By default, this query is, "Would you like to mark the request completed?".

3. Customize the behavior of the in-progress query by setting values for the tags in the `<configuration><voice><inprogress-query>` section of the `evs.xml` file.

   Specify a value within each pair of begin and end tags as follows:

Tag	Value
<configuration> <voice> <inprogress-query> <ack-timeout/>	Use the <ack-timeout/> tag to specify the timeout value (in seconds) when a user has not responded to an in-progress query and has responded to the previous query with a "No". If the specified timeout value is reached, the room cleaning request is escalated to a manager group in the Vocera Voice Server and the task is removed. By default, the timeout value is 1000 seconds.
<configuration> <voice> <inprogress-query> <message/>	Use the <message/> tag to specify a yes/no question that asks the Vocera device user if a cleaning request is in progress. The Vocera device auto-enunciates this question for the user immediately after announcing a cleaning request. The device also enunciates reminders based on the <ack-reminder-delay> and <nack-reminder-delay> timers. By default, this query is, "Would you like to mark the request in progress?".
<configuration> <voice> <inprogress-query> <nack-reminder-delay/>	Use the <nack-reminder-delay/> tag to specify the amount of time (in seconds) to wait before sending a follow-up in-progress query when a user fails to acknowledge the previous query with either a "Yes" or a "No". Users must acknowledge receipt of a query by responding to it in either the affirmative or negative. A non-acknowledgement ("nack") results in a retry after a default time of 60 seconds has elapsed. The waiting time for a non-acknowledgement is much shorter than the waiting time for a response of "No", which is determined by <ack-reminder-delay/>.
<configuration> <voice> <inprogress-query> <ack-reminder-delay/>	Use the <ack-reminder-delay/> tag to specify the amount of time (in seconds) to wait before sending a follow-up in-progress query when a user responds to the previous query with a "No". By default, the system waits 600 seconds before sending a new in-progress query to the user. Users must acknowledge receipt of a query by responding to it in either the affirmative or negative. The waiting time for a non-acknowledgement, which is determined by the <nack-reminder-delay/>, is much shorter than the waiting time for a response of "No".
<configuration> <voice> <inprogress-query> <nack-timeout/>	Use the <nack-timeout/> tag to specify the timeout value (in seconds) when a user has not responded to an in-progress query and has failed to acknowledge the previous query with either a "Yes" or a "No". If the specified timeout value is reached, the room cleaning request is escalated to a manager group in the Vocera Voice Server and the task is removed. By default, the timeout value is 600 seconds.

4. Specify how often Vocera retries the deletion of a message from a user's device (if the first deletion attempt fails) by setting values for the tags in the <configuration><voice><delete><retry> section of the evs.xml file.

   Specify a value within each pair of begin and end tags as follows:

Tag	Value
<configuration> <voice> <delete> <retry> <timeout/>	Use the <timeout/> tag to specify the total duration of time (in seconds) that may elapse during message delete retries before a timeout occurs. By default, the value is 600 seconds.

Tag	Value

`<configuration>`
`<voice>`
`<delete>`
`<retry>`
`<interval/>`
	Use the `<interval/>` tag to specify the total duration of time (in seconds) that Vocera will wait before attempting to delete a message, if the previous deletion attempt failed. By default, the value is 30 seconds.

The `<configuration><voice><delete>` section of `evs.xml` specifies system-level defaults for the delete retry behavior. You can optionally override these settings in the `pluginproperties.xml` file for each vendor.

5. Specify how often Vocera re-sends a message to a user's device (if the first delivery attempt fails) by setting values for the tags in the `<configuration><voice><send><retry>` section of the `evs.xml` file.

Specify a value within each pair of begin and end tags as follows:

Tag	Value

`<configuration>`
`<voice>`
`<send>`
`<retry>`
`<timeout/>`
	Use the `<timeout/>` tag to specify the total duration of time (in seconds) that may elapse during message send retries before a timeout occurs. By default, the value is 600 seconds.

`<configuration>`
`<voice>`
`<send>`
`<retry>`
`<interval/>`
	Use the `<interval/>` tag to specify the amount of time (in seconds) that Vocera will wait before attempting to re-send a message, if the previous delivery has failed. By default, the value is 30 seconds.

The `<configuration><voice><send>` section of `evs.xml` specifies system-level defaults for message send retry behavior. You can optionally override these settings in the `pluginproperties.xml` file for each vendor.

6. Specify tone, priority, and recipient customizations for an escalation message in the `<configuration><voice><escalation><message>` section of the `evs.xml` file. Vocera initiates an escalation if a message send or delete attempt results in an unrecoverable failure.

Customize the escalation routine by specifying a value within each pair of begin and end tags as follows:

Tag	Value

`<configuration>`
`<voice>`
`<escalation>`
`<message>`
`<priority/>`
	Use the `<priority/>` tag to specify one of the following VMI message priorities:
	• `0` for normal priority
	• `1` for high priority
	• `2` for urgent priority

`<configuration>`
`<voice>`
`<escalation>`
`<message>`
`<voceraid/>`
	Use the `<voceraid/>` tag to specify the name of a Vocera group or a Vocera User ID as the recipient of the escalation message. If you do not specify a value, Vocera EVS discards escalation messages.

Tag	Value

`<configuration>`
`<voice>`
`<escalation>`
`<message>`
`<tone/>`
	Optionally use the `<tone/>` tag to specify the filename of a custom audio tone to play when an escalation arrives on a Vocera device. If you do not specify a custom tone, the Vocera device plays the default message "clunk" sound. The custom tone file has the following requirements:
	• **Location:** `\vocera\data\prompts\custom`
	• **Audio Format:** 16 bit Monophonic WAV PCM
	• **Sampling Rate:** 8000 samples/second

The `<configuration><voice><escalation>` section of `evs.xml` specifies system-level defaults for message escalations. You can optionally override these settings in the `pluginproperties.xml` file for each vendor.

7. Specify tone and priority customizations for all Vocera-initiated messages (except escalations) in the `<configuration><voice><message>` section of the `evs.xml` file.

   Customize the message by specifying a value within each pair of begin and end tags as follows:

Tag	Value

`<configuration>`
`<voice>`
`<message>`
`<priority/>`
	Use the `<priority/>` tag to specify one of the following VMI message priorities:
	• `0` for normal priority
	• 1 for high priority
	• 2 for urgent priority

`<configuration>`
`<voice>`
`<message>`
`<tone/>`
	Optionally use the `<tone/>` tag to specify the filename of a custom audio tone to play when a cleaning request arrives on a Vocera device. If you do not specify a custom tone, the Vocera device plays the default message "clunk" sound. The custom tone file has the following requirements:
	• **Location:** `\vocera\data\prompts\custom`
	• **Audio Format:** 16 bit Monophonic WAV PCM
	• **Sampling Rate:** 8000 samples/second

The `<configuration><voice><message>` section of `evs.xml` specifies system-level defaults for message priorities and tones. You can optionally override these settings in the `pluginproperties.xml` file for each vendor.

8. Save the `evs.xml` file and continue with the system-level plugin integration.

The voice configuration is now complete. Following is an example of a typical system-level voice configuration.

```
<voice>
    <completion-query>
        <nack-reminder-delay>60</nack-reminder-delay>
        <ack-timeout>7200</ack-timeout>
        <nack-timeout>3600</nack-timeout>
        <delay>10</delay>
        <ack-reminder-delay>600</ack-reminder-delay>
        <message>Would you like to mark the request completed?</message>
    </completion-query>
    <inprogress-query>
        <ack-timeout>1800</ack-timeout>
        <message>Would you like to mark the request in progress?</message>
        <nack-reminder-delay>60</nack-reminder-delay>
        <ack-reminder-delay>300</ack-reminder-delay>
        <nack-timeout>600</nack-timeout>
    </inprogress-query>
    <delete>
        <retry>
            <timeout>600</timeout>
            <interval>30</interval>
        </retry>
    </delete>
    <send>
        <retry>
            <timeout>600</timeout>
            <interval>30</interval>
        </retry>
    </send>
    <escalation>
        <message>
            <priority>0</priority>
            <voceraid/>
            <tone/>
        </message>
    </escalation>
    <message>
        <priority>0</priority>
        <tone/>
    </message>
</voice>
```

Configuring the System-Level Plugin Integration

The system-level plugin integration allows EVS to retry the completion and in-progress status updates that it sends to the integration partner (for example, to the Epic Envifonmental Services module).

Specify retry handlers for the web services completion and in-progress updates in the `<plugin>` section of the `evs.xml` file. You can customize the following main features of the web services integration:

- Web services status updates for complete cleaning requests
  Vocera initiates a web services call to send a status update to the Environmental Services vendor when a user indicates that a cleaning request is complete. If the vendor does not acknowledge the update, Vocera waits a configurable amount of time and then retries the update.
  These configuration parameters allow you to customize the retry and timeout intervals for the web services completion update.
- Web services status updates for in-progress cleaning requests
  Vocera initiates a web services call to send a status update to the Environmental Services vendor when a user indicates that a cleaning request is in progress. If the vendor does not acknowledge the update, Vocera waits a configurable amount of time and then retries the update.
  These configuration parameters allow you to customize the retry and timeout intervals for the web services in-progress update.

How to Customize the System-Level Plugin Integration

Configure the system-level Environmental Services vendor integration by editing the `<plugin>` section of the `evs.xml` file, a structured XML document.

In most situations, the default values Vocera provides in the `<plugin>` section of `evs.xml` are sufficient. You do not need to modify this section of `evs.xml` unless you want to customize the default behavior of the web services retry handlers. In all other cases, the default values provided for the parameters in `evs.xml` are sufficient.

To customize the system-level web services integration for status updates:

1. On the Vocera Environmental Services Integration server, open the `\vocera\evs\resources\evs.xml` file in a text editor, if it is not open already.
2. Specify how often Vocera retries the web services update for cleaning request completion (if the first update attempt fails) by setting values for the tags in the `<configuration><plugin><complete><retry>` section of the `evs.xml` file.

   Specify a value within each pair of begin and end tags as follows:

Tag	Value

`<configuration>`
`<plugin>`
`<complete>`
`<retry>`
`<timeout/>`
	Use the `<timeout/>` tag to specify the total duration of time (in seconds) that may elapse during completion update retries before a timeout occurs. By default, the value is 60 seconds.

`<configuration>`
`<plugin>`
`<complete>`
`<retry>`
`<interval/>`
	Use the `<interval/>` tag to specify the total duration of time (in seconds) that Vocera will wait before attempting to update cleaning request completion status, if the previous update attempt failed. By default, the value is 5 seconds.

3. Specify how often Vocera retries the web services in-progress update for cleaning requests (if the first update attempt fails) by setting values for the tags in the `<configuration><plugin><inprogress><retry>` section of the `evs.xml` file.

   Specify a value within each pair of begin and end tags as follows:

Tag	Value

`<configuration>`
`<plugin>`
`<inprogress>`
`<retry>`
`<timeout/>`
	Use the `<timeout/>` tag to specify the total duration of time (in seconds) that may elapse during in-progress update retries before a timeout occurs. By default, the value is 60 seconds.

`<configuration>`
`<plugin>`
`<inprogress>`
`<retry>`
`<interval/>`
	Use the `<interval/>` tag to specify the total duration of time (in seconds) that Vocera will wait before attempting to update cleaning request in-progress status, if the previous update attempt failed. By default, the value is 5 seconds.

4. Save and close the `evs.xml` file.

The web services status udpate configuration is now complete. Following is an example of a typical web services configuration.

```
<plugin>
    <complete>
        <retry>
            <timeout>60</timeout>
            <interval>5</interval>
        </retry>
    </complete>
    <inprogress>
        <retry>
            <timeout>60</timeout>
            <interval>5</interval>
        </retry>
    </inprogress>
</plugin>
```

Configuring the Epic-EVS Integration

The Vocera Environmental Services Integration uses an XML file that defines parameters for the required email and web services integrations between the Epic system and the Vocera Environmental Services Integration server. You can also optionally use it to override system-level defaults established in the `evs.xml` file.

The XML file, `pluginproperties.xml`, is located in the `\vocera\evs\resources\plugin\epic` directory on the Vocera Environmental Services Integration server.

The content of `pluginproperties.xml` is structured XML with begin tags (for example `<ServerType>`) and end tags (for example `</ServerType>`). You enter values between the tags to configure the integration parameters.

The `pluginproperties.xml` file contains the following main sections:

Section	Description
name	Specifies the name of the Environmental Services software vendor you are integrating Vocera software with.
emailConfig	Specifies the configuration parameters for the email integration between the EVS server and the Epic system.
voiceOther	Customizes features of an email message that Vocera EVS server sends to a Vocera device upon receipt of an email for anything other than a start shift, end shift, or cleaning request.
voice	Specifies the message that the Vocera EVS server sends to a Vocera device upon receiving a cleaning request from Epic, as well as related configuration parameters for the message.
epicRestWebService	Specifies the configuration parameters for the web services integration between the EVS server and the Epic system.
timestamp	Specifies the date and time `pluginproperties.xml` was most recently updated. The value in this field is updated automatically by the EVS server when the file is saved.

Configuring the Email Integration

The Epic Environmental Services module uses an email integration to send cleaning requests to Vocera EVS. EVS uses information in the email to construct a cleaning request which it delivers as an auto-enunciated message for a user of a Vocera device. EVS requires a dedicated email service account on your mail server. If you already have a Vocera Voice Server, do not re-use its existing email account.

How to Configure the Email Integration

Configure the email integration by editing the `<emailConfig>` section of the `pluginproperties.xml` file, a structured XML document. All the settings in this section are required.

To configure the email integration:

1. On the Vocera Environmental Services Integration server, open the `\vocera\evs\resources\plugin\epic\pluginproperties.xml` file in a text editor.
2. Specify a value within each pair of begin and end tags to configure the email integration:

Tag	Value

`<emailConfig>`
`<serverType/>`
	Enter the type of mail server based on the protocol you are using. Possible values include `pop3` (Post Office Protocol 3) or `imap` (Internet Message Access Protocol).

`<emailConfig>`
`<host/>`
	Enter the DNS name of your mail server (`exchange.yourcompany.local`, for example).

`<emailConfig>`
`<userName/>`
	Enter the user name for accessing the mailbox that your IT administrator reserved for EVS email (`VoceraEVS@yourcompany.com`, for example).

`<emailConfig>`
`<password/>`
	Enter the obfuscated password the EVS server must use to log in to the mailbox. This password must be obfuscated manually by Vocera and supplied to you.
	Note: When providing the password, make sure that any special characters are encoded using the proper escape sequences. See *Appendix: Special Characters in XML* on page 47 for more details.

`<emailConfig>`
`<checkIntervalMillis/>`
	Enter the time in milliseconds that the system waits between mail checks. During a mail check, the EVS server connects with the mail host to check for new mail. If you do not specify a value, the default is 10000 (the minimum). If you specify anything smaller, EVS treats it as 10000.

`<emailConfig>`
`<additionalProperties/>`
	Specify the port number used by the mail server and whether the EVS uses SSL to connect. This field takes two comma-separated properties in the following format:
	• *serverType*`.port=`*portNumber*, where *serverType* is the value you specified in the `serverType` field and *portNumber* is the port your mail server uses.
	• *serverType*`.ssl=`*trueFalse*, where *serverType* is the value you specified in the `serverType` field and *trueFalse* is either `true` or `false`.
	For example, a typical value for this field is `pop3.port=445,pop3.ssl.enable=false`.

3. Save the `pluginproperties.xml` file and continue with the email pass-through integration.

The email configuration is now complete. Following is an example of a typical email configuration.

```
<EmailConfig>
    <serverType>imap</ServerType>
    <host>exchange.voceraclinic.local</Host>
    <userName>vcts@voceraclinic.com</UserName>
    <password>vcts2ADT</Password>
    <checkIntervalMillis>7000</CheckIntervalMillis>
    <additionalProperties>imap.port=445,imap.ssl.enable=true</AdditionalProperties>
</EmailConfig>
```

Configuring Email Pass-Through Handling

The email integration allows Epic to send emails to Vocera announcing the start of a shift, the end of a shift, or a cleaning request for a device user that is specified in the subject of the email. If Epic sends any other type of email, the EVS integration passes it through as a message to the Vocera device of the person specified in the email subject. The email pass-through configuration allows you to customize features of this message, such as a tone to play on the device, a priority, and so forth.

Specify any parameters for the email pass-through integration in the `<voiceOther>` section of the `pluginproperties.xml` file. You can customize the following main features of the email pass-through integration:

- Retry settings for message pass-through attempts

  Vocera attempts to pass-through email messages other than start shift, end shift, or cleaning requests to a user's device. If the first delivery attempt for a message fails:

  - Vocera escalates immediately when the failure is due to an invalid user ID.
  - Vocera re-sends the message when the failure is potentially recoverable; for example, when delivery fails because a user is not logged in.

  The values in the `<send><retry>` section determine how often Vocera attempts to re-send the message.
- Message escalation settings

  In the pass-through integration, an escalation is a message that is sent to an administrator's Vocera device when an unrecoverable error occurs. The following events result in an escalation during the pass-through integration:

  - Sending a pass-through message to an invalid user ID
  - A timeout that occurs as a result of a pass-through message retry failure

  If you want Vocera to send an escalation notification when a message pass-through fails, you must specify a value for `<voceraID>` in the message escalation parameters. If `<voceraID>` is not specified, Vocera discards the escalation message.
- Message customizations

  In the pass-through integration, a message is an email that Vocera is passing on to an Environmental Services employee. You typically do not need to specify any content for the message, since Vocera passes the content of the email through to the user; however, you may want to customize the message tone or priority.

How to Configure Email Pass-Through Handling

Customize the email pass-through integration by editing the `<voiceOther>` section of the `pluginproperties.xml`, a structured XML document.

In most situations, the default values Vocera provides in the `<voiceOther>` section of `pluginproperties.xml` are sufficient. However, if you want Vocera to send an escalation notification when an email pass-through attempt fails, you must specify a value for `<voceraID>` in the message escalation parameters. When `<voceraID>` is not specified, Vocera discards the escalation message.

In all other cases, the default values Vocera provides for the parameters in the `voiceOther` section of `pluginproperties.xml` are sufficient, unless you want to customize the integration.

To configure the email pass-through integration:

1. On the Vocera Environmental Services Integration server, open the `\vocera\evs\resources\plugin\epic\pluginproperties.xml` file in a text editor, if it is not already open.
2. Specify how often Vocera re-sends an email pass-through message to a user's device (if the first delivery attempt fails) by setting values for the tags in the `<voiceOther><send><retry>` section of the `pluginproperties.xml` file.

   Specify a value within each pair of begin and end tags as follows:

Tag	Value

`<voiceOther>`
`  <send>`
`    <retry>`
`      <timeout/>`
	Use the `<timeout/>` tag to specify the total duration of time (in seconds) that may elapse during email pass-through retries before a timeout occurs. By default, the value is 3600 seconds.

Tag	Value
<voiceOther> <send> <retry> <interval/>	Use the `<interval/>` tag to specify the amount of time (in seconds) that Vocera will wait before attempting to re-send an email pass-through message, if the previous delivery has failed. By default, the value is 600 seconds.

3. Specify the text for an escalation message template as well as other customizations such as a tone, priority, and recipient in the `<voiceOther><escalation><message>` section of the `pluginproperties.xml` file.

   Vocera initiates an escalation if a message pass-through attempt results in an unrecoverable failure.

   Customize the escalation routine by specifying a value within each pair of begin and end tags as follows:

Tag	Value
<voiceOther> <escalation> <message> <voceraID/>	Use the `<voceraID/>` tag to specify the name of a Vocera group name or User ID as the recipient of the failure message. If you do not specify a value, Vocera EVS discards failure messages.
<voiceOther> <escalation> <message> <messageTemplate/>	Use the `<messageTemplate/>` tag to specify a message that the Vocera device will auto-enunciate when it receives an escalation. You may use any combination of plain text and the `${HKRID}` variable, which is auto-enunciated as the Environmental Services employee ID.
<voiceOther> <escalation> <message> <priority/>	Use the `<priority/>` tag to specify one of the following VMI message priorities: • `0` for normal priority • `1` for high priority • `2` for urgent priority
<voiceOther> <escalation> <message> <tone/>	Optionally use the `<tone/>` tag to specify the filename of a custom audio tone to play when an escalation arrives on a Vocera device. If you do not specify a custom tone, the Vocera device plays the default message "clunk" sound. The custom tone file has the following requirements: • **Location:** `\vocera\data\prompts\custom` • **Audio Format:** 16 bit Monophonic WAV PCM • **Sampling Rate:** 8000 samples/second

4. Specify any tone and priority customizations for the pass-through message, as well as custom content for the message (if necessary) in the `<voiceOther><message>` section of the `pluginproperties.xml` file.

   Customize the message by specifying a value within each pair of begin and end tags as follows:

Tag	Value
<voiceOther> <message> <messageTemplate/>	Use the `<messageTemplate/>` tag to specify a message that the Vocera device will auto-enunciate in a pass-through email message. **Important:** You typically do not want to provide custom content for this message template. Vocera automatically passes through the content of the email. If you provide a custom message template, Vocera sends it instead of the email content.
<voiceOther> <message> <priority/>	Use the <priority/> tag to specify one of the following VMI message priorities: • `0` for normal priority • `1` for high priority • `2` for urgent priority

Tag	Value

`<voiceOther>`
`<message>`
`<tone/>`
	Optionally use the <tone/> tag to specify the filename of a custom audio tone to play when a pass-through message arrives on a Vocera device. If you do not specify a custom tone, the Vocera device plays the default message "clunk" sound. The custom tone file has the following requirements:
	• **Location:** `\vocera\data\prompts\custom`
	• **Audio Format:** 16 bit Monophonic WAV PCM
	• **Sampling Rate:** 8000 samples/second

5. Save the `pluginproperties.xml` file and continue with the voice integration.

The email pass-through configuration is now complete. Following is an example of a typical email pass-through configuration.

```
<voiceOther>
    <send>
        <retry>
            <timeout>3600</timeout>
            <interval>600</interval>
        </retry>
    </send>
    <escalation>
        <message>
            <voceraID></voceraID>
            <messageTemplate></messageTemplate>
            <priority>0</priority>
            <tone></tone>
        </message>
    </escalation>
    <message>
        <messageTemplate></messageTemplate>
        <priority>0</priority>
        <tone></tone>
    </message>
</voiceOther>
```

Configuring the Epic Voice Integration

The voice integration allows a Vocera device to auto-enunciate cleaning requests for Environmental Services employees. You configure the voice integration by specifying a template that provides the message text used for cleaning requests and escalations; you can also optionally override default parameters specified in `evs.xml`.

Specify any parameters for the voice integration in the `<voice>` section of the `pluginproperties.xml` file. You can customize the following main features of the voice integration:

- Retry settings for message deletion attempts
  Vocera attempts to delete a message from a user's device after every successful state transition. Device users initiate a state transition by acknowledging messages that they receive. For example, when a user responds with a "Yes" to the prompt "Do you want to mark this request in progress", Vocera attempts to delete this message from the user's device. If the deletion fails, the values in the `<delete><retry>` section determine how often Vocera attempts to retry the deletion.
  Values in the this section (`<voice><delete>`) of `pluginproperties.xml` override settings made in the same section of `evs.xml`.
- Retry settings for message send attempts

Vocera sends cleaning request and reminder messages to a user's device as part of the Epic Environmental Services integration. If the first delivery attempt for a message fails:

- Vocera escalates immediately when the failure is due to an invalid user ID.
- Vocera re-sends the message when the failure is potentially recoverable; for example, when delivery fails because a user is not logged in.

The values in the `<send><retry>` section determine how often Vocera attempts to re-send the message.

Values in the this section (`<voice><send>`) of `pluginproperties.xml` override settings made in the same section of `evs.xml`.

- Message escalation settings

  In the EVS integration, an escalation is a message that is sent to an administrator's Vocera device when an unrecoverable error occurs. The following events result in an escalation during the voice integration:

  - Sending a message to an invalid user ID
  - A timeout that occurs as a result of a message retry failure
  - A timeout that occurs as a result of a message delete failure

  If you want Vocera to send an escalation notification when a message send or deletion fails, you must specify a value for `<voceraID>` in the message escalation parameters. If `<voceraID>` is not specified, Vocera discards the escalation message.

  Values in the `<voice><escalation><message>` section of `pluginproperties.xml` override settings made in the same section of `evs.xml`. The `<messageTemplate>` property, however, is unique to `pluginproperties.xml`.

- Message template settings

  In the EVS integration, a message is a cleaning request or reminder that is sent to the Vocera device of an Environmental Services employee. You do not need to specify the Vocera ID of a recipient for a message; the Epic system specifies it in the Subject line of the request email it sends to Vocera.

  Values in the `<voice><message>` section of `pluginproperties.xml` override settings made in the same section of `evs.xml`. The `<messageTemplate>` property, however, is unique to `pluginproperties.xml`.

How to Configure the Voice Integration

Configure the voice integration by editing the `<voice>` section of the `pluginproperties.xml`, a structured XML document.

In most situations, the default values Vocera provides in the `<voice>` section of `pluginproperties.xml` are sufficient. However, if you want Vocera to send an escalation notification when a cleaning request delivery fails, you must specify a value for `<voceraID>` in the message escalation parameters. When `<voceraID>` is not specified, Vocera discards the escalation message.

In all other cases, the default values Vocera provides for the parameters in the `<voice>` section of `pluginproperties.xml` are sufficient, unless you want to customize the integration.

To configure the voice integration:

1. On the Vocera Environmental Services Integration server, open the `\vocera\evs\resources\plugin\epic\pluginproperties.xml` file in a text editor, if it is not already open.
2. Optionally specify how often Vocera retries the deletion of a message from a user's device (if the first deletion attempt fails) by setting values for the tags in the `<voice><delete><retry>` section of the `pluginproperties.xml` file.

   Specify a value within each pair of begin and end tags as follows:

Tag	Value
<voice> <delete> <retry> <timeout/>	Use the <timeout/> tag to specify the total duration of time (in seconds) that may elapse during message delete retries before a timeout occurs. By default, the value is 600 seconds.
<voice> <delete> <retry> <interval/>	Use the <interval/> tag to specify the total duration of time (in seconds) that Vocera will wait before attempting to delete a message, if the previous deletion attempt failed. By default, the value is 30 seconds.

Values in this section override values in the same section of `evs.xml`.

3. Optionally specify how often Vocera re-sends a message to a user's device (if the first delivery attempt fails) by setting values for the tags in the `<voice><send><retry>` section of the `pluginproperties.xml` file.

   Specify a value within each pair of begin and end tags as follows:

Tag	Value
<voice> <send> <retry> <timeout/>	Use the <timeout/> tag to specify the total duration of time (in seconds) that may elapse during message send retries before a timeout occurs. By default, the value is 600 seconds.
<voice> <send> <retry> <interval/>	Use the <interval/> tag to specify the amount of time (in seconds) that Vocera will wait before attempting to re-send a message, if the previous delivery has failed. By default, the value is 30 seconds.

   Values in this section override values in the same section of `evs.xml`.

4. Specify the text for an escalation message template as well as other customizations such as a tone, priority, and recipient in the `<voice><escalation><message>` section of the `pluginproperties.xml` file. Vocera initiates an escalation if a message send or delete attempt results in an unrecoverable failure.

   Customize the escalation routine by specifying a value within each pair of begin and end tags as follows:

Tag	Value
<voice> <escalation> <message> <voceraID/>	Use the <voceraID/> tag to specify the name of a Vocera group name or User ID as the recipient of the failure message. If you do not specify a value, Vocera EVS discards failure messages. The value for this property overrides the value specified for the same property in evs.xml.
<voice> <escalation> <message> <messageTemplate/>	Use the <messageTemplate/> tag to specify a message that the Vocera device will auto-enunciate when it receives an escalation. You may use any combination of plain text and the following variables: • ${BEVID}, which is auto-enunciated as the Epic Environmental Services request ID. • ${HKRID}, which is auto-enunciated as the Environmental Services employee ID. • ${Bed}, which is auto-enunciated as the name of a Vocera role-based bed group.

Tag	Value

`<voice>`
`<escalation>`
`<message>`
`<priority/>`
	Use the <priority/> tag to specify one of the following VMI message priorities:
	• `0` for normal priority
	• `1` for high priority
	• `2` for urgent priority
	The value for this property overrides the value specified for the same property in `evs.xml`.

`<voice>`
`<escalation>`
`<message>`
`<tone/>`
	Optionally use the <tone/> tag to specify the filename of a custom audio tone to play when an escalation arrives on a Vocera device. If you do not specify a custom tone, the Vocera device plays the default message "clunk" sound. The custom tone file has the following requirements:
	• **Location:** `\vocera\data\prompts\custom`
	• **Audio Format:** 16 bit Monophonic WAV PCM
	• **Sampling Rate:** 8000 samples/second
	The value for this property overrides the value specified for the same property in `evs.xml`.

5. Specify the text for the message template as well as tone and priority customizations in the `<voice><message>` section of the `pluginproperties.xml` file.

   Customize the message by specifying a value within each pair of begin and end tags as follows:

Tag	Value

`<voice>`
`<message>`
`<messageTemplate/>`
	Use the `<messageTemplate/>` tag to specify a message that the Vocera device will auto-enunciate when it receives a cleaning request. You may use any combination of plain text and the following variables:
	• `${Bed}`, which is auto-enunciated as the name of a Vocera role-based bed group. For example, the default value Clean request for Bed ${Bed} may result in the Vocera device user hearing the message, "Clean request for Bed 401 A".
	• `${Room}`, which is auto-enunciated as the name of a Vocera role-based room group.

`<voice>`
`<message>`
`<priority/>`
	Use the <priority/> tag to specify one of the following VMI message priorities:
	• `0` for normal priority
	• `1` for high priority
	• `2` for urgent priority
	The value for this property overrides the value specified for the same property in `evs.xml`.

`<voice>`
`<message>`
`<tone/>`
	Optionally use the <tone/> tag to specify the filename of a custom audio tone to play when a cleaning request arrives on a Vocera device. If you do not specify a custom tone, the Vocera device plays the default message "clunk" sound. The custom tone file has the following requirements:
	• **Location:** `\vocera\data\prompts\custom`
	• **Audio Format:** 16 bit Monophonic WAV PCM
	• **Sampling Rate:** 8000 samples/second
	The value for this property overrides the value specified for the same property in `evs.xml`.

6. Save the `pluginproperties.xml` file and continue with the web services integration.

The voice configuration is now complete. Following is an example of a typical voice configuration.

```
<voice>
   <delete>
```

```
        <retry>
           <timeout>600</timeout>
           <interval>30</interval>
        </retry>
     </delete>
     <send>
        <retry>
           <timeout>600</timeout>
           <interval>30</interval>
        </retry>
     </send>
     <escalation>
        <message>
           <voceraID>Epic Administrator</voceraID>
           <messageTemplate>Clean request ${BEVID} for housekeeper ${HKRID} Bed ${Bed}
   failed, please investigate.... </messageTemplate>
           <priority>0</priority>
           <tone></tone>
        </message>
     </escalation>
     <message>
        <messageTemplate>Clean request for Bed ${Bed}</messageTemplate>
        <priority>0</priority>
        <tone></tone>
     </message>
  </voice>
```

Configuring the Web Services Integration

The web services integration allows EVS to send status updates to the Epic Environmental Services module when Vocera device users engage in voice interactions. For example, device users may issue commands such as "Begin break" or respond to a cleaning request with an "In progress" status.

Specify any parameters for the web services integration in the `<epicRestWebService>` section of the `pluginproperties.xml` file, a structured XML document. You can customize the following main features of the web services integration:

- Web services endpoint URL

  The web services integration requires you to provide a web services end point dedicated to the Vocera Environmental Services Integration in your Epic system.
- The type of authentication and the user name / password that the web service must provide to log in to the Epic system
- Retry settings for "Cleaning request complete" status updates

  If a Vocera device user states that a cleaning request is complete, the Vocera Environmental Services Integration attempts to update the cleaning request status in the Epic Environmental services module. That is, when a user responds with a "Yes" to the prompt "Do you want to mark this request complete", Vocera initiates a web service request to notify Epic of the change in status. If the initial web service request fails, the values in the `<complete><retry>` section determine how often Vocera attempts to retry the web service request.
- Retry settings for "Cleaning request in progress" status updates

  If a Vocera device user states that a cleaning request is in progress, the Vocera Environmental Services Integration attempts to update the cleaning request status in the Epic Environmental services module. That is, when a user responds with a "Yes" to the prompt "Do you want to mark this request in progress", Vocera initiates a web service request to notify Epic of the change in status. If the initial web service request fails, the values in the `<inprogress><retry>` section determine how often Vocera attempts to retry the web service request.

How to Configure Web Services

Configure the web services integration by editing the `<epicRestWebService>` section of the `pluginproperties.xml` file, a structured XML document.

You must specify a value for `<baseURL>` in the web services configuration parameters. In all other cases, the default values provided for the parameters in `pluginproperties.xml` are sufficient unless you want to customize the web services integration.

To configure the web services integration:

1. On the Vocera Environmental Services Integration server, open the `\vocera\evs\resources\plugin\epic\pluginproperties.xml` file in a text editor, if it is not already open.
2. Specify the URL of the Epic system's web services end point as follows.

Tag	Value

`<epicRestWebService>`
`  <baseUrl/>`
	Specify the URL of the Epic system's web services end point between the `<baseUrl>` and `</baseUrl>` tags to configure the integration.
	For example, the entry for an Open Epic staging system may look like `<baseUrl>https://open-ic.epic.com/Interconnect-FHIR/wcf/Epic.Access.GeneratedServices/EVS.svc/rest/</baseUrl>`.

3. Specify the credentials the web service uses to log into the Epic system as follows.

Tag	Value

`<epicRestWebService>`
`  <username/>`
	Enter the type of authentication and the user name that the web service uses to log in to the Epic system.
	EVS supports the following types of authentication for Epic:
	• `windows$`
	• `local$`
	• `EMP$`
	For example, if the Epic system uses `windows$` authentication and the user name is `jblair`, enter `windows$jblair` in the `username` field.

`<epicRestWebService>`
`  <password/>`
	Enter the obfuscated password the EVS web service uses to log in to the Epic server. This password must be obfuscated manually by Vocera and supplied to you. The password Vocera provides begins with "OBF:" to indicate that it is obfuscated; the "OBF:" value is a literal part of the password you must enter.
	Note: When providing the password, make sure that any special characters are encoded using the proper escape sequences. See *Appendix: Special Characters in XML* on page 47 for more details.

If your Epic system does not require any authentication, leave these fields blank.

Note: If you have upgraded from a version of EVS earlier than 1.1.1, the `<username/>` and `<password/>` fields will not exist in the `pluginproperties.xml` file. Add these fields manually after the `<baseUrl/>` section, so the `pluginproperties.xml` file appears similar to the example at the end of this section.

4. Specify how often Vocera retries updating the Epic cleaning request status to "Complete" (if the first attempt fails) by setting values for the tags in the `<epicRestWebService><complete><retry>` section of the `pluginproperties.xml` file.

   Specify a value within each pair of begin and end tags as follows:

Tag	Value
<epicRestWebService> <complete> <retry> <timeout/>	Use the <timeout/> tag to specify the total duration of time (in seconds) that may elapse during web service retries when Vocera EVS attempts to update the Epic cleaning request to "Complete". When this time is exceeded, a timeout occurs. By default, the value is 60 seconds. If a timeout occurs, Vocera EVS sends an escalation message as specified in the <voice><escalation><message> routine.
<epicRestWebService> <complete> <retry> <interval/>	Use the <interval/> tag to specify the amount of time (in seconds) that Vocera will wait before attempting to retry updating the Epic cleaning request status to "Complete", if the previous request failed. By default, the value is 5 seconds.

5. Specify how often Vocera retries updating the Epic cleaning request status to "In progress" (if the first attempt fails) by setting values for the tags in the `<epicRestWebService><inprogress><retry>` section of the `pluginproperties.xml` file.

   Specify a value within each pair of begin and end tags as follows:

Tag	Value
<epicRestWebService> <inprogress> <retry> <timeout/>	Use the <timeout/> tag to specify the total duration of time (in seconds) that may elapse during web service retries when Vocera EVS attempts to update the Epic cleaning request to "In progress". When this time is exceeded, a timeout occurs. By default, the value is 60 seconds. If a timeout occurs, Vocera EVS sends an escalation message as specified in the <voice><escalation><message> routine.
<epicRestWebService> <inprogress> <retry> <interval/>	Use the <interval/> tag to specify the amount of time (in seconds) that Vocera will wait before attempting to retry updating the Epic cleaning request status to "In progress", if the previous request failed. By default, the value is 5 seconds.

6. Save and close the `pluginproperties.xml` file.

The web services configuration is now complete. Following is an example of a typical web services configuration.

```
<epicRestWebService>
  <baseUrl>https://open-ic.epic.com/Interconnect-FHIR/wcf/
Epic.Access.GeneratedServices/EVS.svc/rest/</baseUrl>
  <username>windows$jblair</username>
  <password>OBF:1ytm1nqg1f1c1miu1mm01f321nrq1yt0</password>
  <complete>
      <retry>
          <timeout>60</timeout>
          <interval>5</interval>
      </retry>
  </complete>
  <inprogress>
      <retry>
          <timeout>60</timeout>
          <interval>5</interval>
      </retry>
  </inprogress>
</epicRestWebService>
```

Configuring the VS-EVS Integration

The Vocera Environmental Services Integration uses a VMI connection to allow the Vocera Environmental Services Integration server and the Vocera Voice Server to exchange information.

Important: Do not connect more than one EVS server to a VS or VS cluster. Configuring more than one EVS connection will cause an interruption in service.

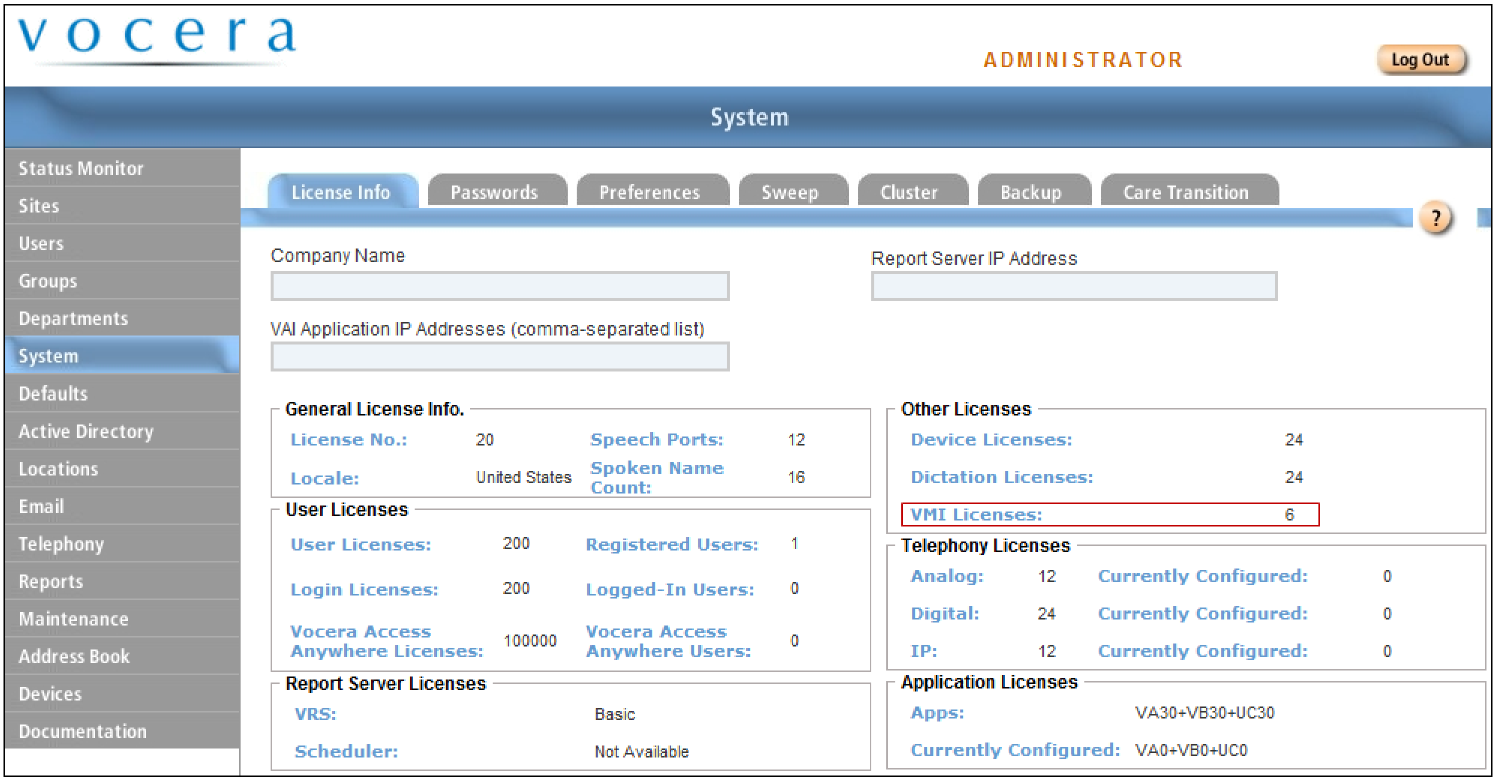
How to Check the Number of VMI Licenses

The Vocera Environmental Services Integration requires an available VMI license to allow the VS to connect to the Vocera Environmental Services Integration server.

To check the total number of VMI licenses assigned to your system:

1. Log into the Administration Console on the Vocera Voice Server.
   See "Using the Administration Console" in the *Vocera Voice Server Administration Console Guide*.
2. Navigate to the System|License Info page.
3. Confirm that you have a sufficient number of licenses to assign one to EVS.
   The Vocera Environmental Services Integration requires a single VMI license.

Note: Vocera provides a VMI license as part of the purchase of the EVS solution. See the Vocera Voice Server *Vocera Voice Server Administration Console Guide* for information on updating your license key.



Company Name

Report Server IP Address

VAI Application IP Addresses (comma-separated list)

General License Info.

License No.:	20	Speech Ports:	12
Locale:	United States	Spoken Name Count:	16

User Licenses

User Licenses:	200	Registered Users:	1
Login Licenses:	200	Logged-In Users:	0
Vocera Access Anywhere Licenses:	100000	Vocera Access Anywhere Users:	0

Report Server Licenses

VRS:	Basic
Scheduler:	Not Available

Other Licenses

Device Licenses:	24
Dictation Licenses:	24
VMI Licenses:	6

Telephony Licenses

Analog:	12	Currently Configured:	0
Digital:	24	Currently Configured:	0
IP:	12	Currently Configured:	0

Application Licenses

Apps:	VA30+VB30+UC30
Currently Configured:	VA0+VB0+UC0

Figure 1: The System|License Info page

How to Set Up the VMI Connection

An environment variable on the Vocera Environmental Services Integration server specifies the IP address of the Vocera Voice Server or cluster. The IP address list is written into the EVS database, and the system uses it to direct VMI requests to the appropriate Vocera Voice Server.

To set up the VMI connection:

1. On the Vocera Environmental Services Integration server, display the **Advanced** tab of the Windows **System Properties** dialog box.

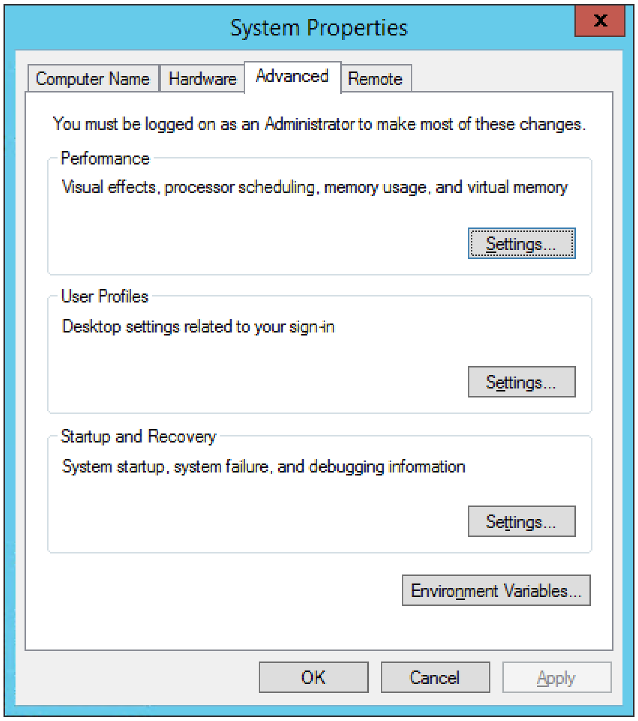
   For example, on Windows 2012 Server do the following:

   a. Right-click the **Windows** icon and select **System**.

      The Windows **System** dialog box appears.

   b. Select **Advanced System Settings**.

      The **Advanced** tab of the Windows **System Properties** dialog box appears.



You must be logged on as an Administrator to make most of these changes.

Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your sign-in

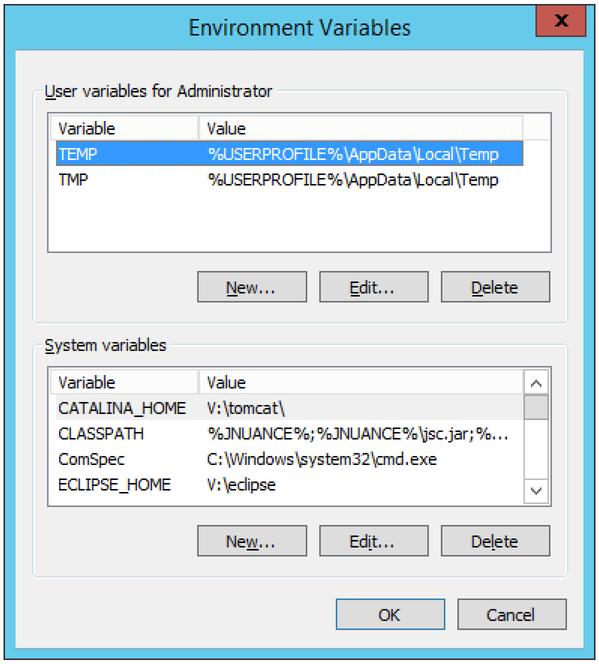
System startup, system failure, and debugging information

Figure 2: Windows Advanced System Properties

2. Create the new environment variable as follows:

   a. On the **Advanced** tab of the **System Properties** dialog box, select **Environment Variables**.

      The Windows **Environment Variables** dialog box appears.



User variables for Administrator

Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

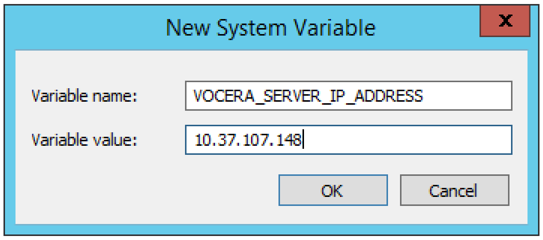
System variables

Variable	Value
CATALINA_HOME	V:\tomcat\
CLASSPATH	%JNUANCE%;%JNUANCE%\jsc.jar;%...
ComSpec	C:\Windows\system32\cmd.exe
ECLIPSE_HOME	V:\eclipse

Figure 3: Environment Variables

b. In the **System Variables** section of the **Environment Variables** dialog box, select **New...**.

The Windows **New System Variable** dialog box appears.



Variable name: VOCERA_SERVER_IP_ADDRESS

Variable value: 10.37.107.148

Figure 4: New System Variable

c. Enter the following values in the **New System Variable** dialog box:

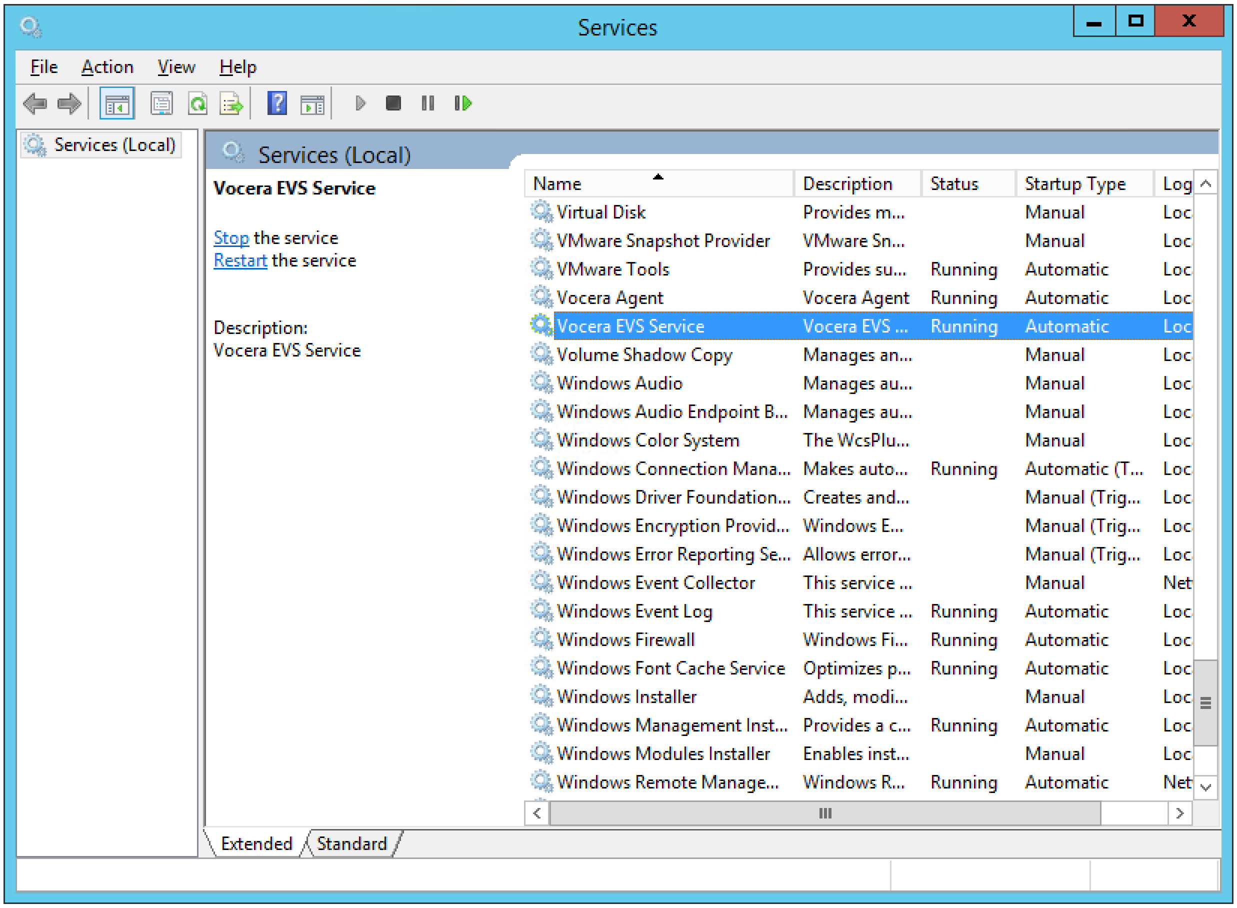
Field	Description
Variable Name	VOCERA_SERVER_IP_ADDRESS
Variable Value	The IP address of the VS. If you are using a VS cluster, enter a comma-separated list of IP addresses with no spaces between them.

d. Click OK to close the **New System Variable**, the **Environment Variables**, and the **System Properties** dialog boxes.

3. Restart the Vocera EVS Service as follows:

a. Open the **Windows Services** dialog box.

b. Navigate to the **Vocera EVS Service**.



Vocera EVS Service

Stop the service
Restart the service

Description:
Vocera EVS Service

Name	Description	Status	Startup Type	Log
Virtual Disk	Provides m...		Manual	Loc
VMware Snapshot Provider	VMware Sn...		Manual	Loc
VMware Tools	Provides su...	Running	Automatic	Loc
Vocera Agent	Vocera Agent	Running	Automatic	Loc
Vocera EVS Service	Vocera EVS ...	Running	Automatic	Loc
Volume Shadow Copy	Manages an...		Manual	Loc
Windows Audio	Manages au...		Manual	Loc
Windows Audio Endpoint B...	Manages au...		Manual	Loc
Windows Color System	The WcsPlu...		Manual	Loc
Windows Connection Mana...	Makes auto...	Running	Automatic (T...	Loc
Windows Driver Foundation...	Creates and...		Manual (Trig...	Loc
Windows Encryption Provid...	Windows E...		Manual (Trig...	Loc
Windows Error Reporting Se...	Allows error...		Manual (Trig...	Loc
Windows Event Collector	This service ...		Manual	Net
Windows Event Log	This service ...	Running	Automatic	Loc
Windows Firewall	Windows Fi...	Running	Automatic	Loc
Windows Font Cache Service	Optimizes p...	Running	Automatic	Loc
Windows Installer	Adds, modi...		Manual	Loc
Windows Management Inst...	Provides a c...	Running	Automatic	Loc
Windows Modules Installer	Enables inst...		Manual	Loc
Windows Remote Manage...	Windows R...	Running	Automatic	Net

Figure 5: Vocera EVS Service

c. Right-click **Vocera EVS Service** and choose **Stop**.

d. Right-click **Vocera EVS Service** and choose **Start**.
In the **Status** column, **Vocera EVS Service** displays **Running**.

The value of the `VOCERA_SERVER_IP_ADDRESS` environment variable is written into the database.

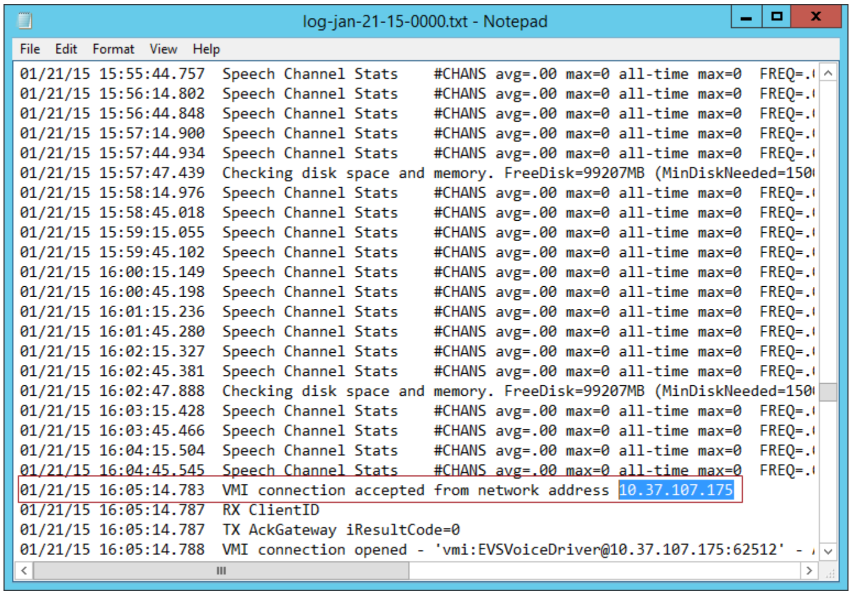
How to Confirm the VMI Connection is Established Successfully

Examine the logs on both the Vocera Voice Server and the Vocera Environmental Services Integration server to confirm that you have correctly established the VMI connection.

To confirm the VMI connection:

1. Confirm the connection in the Vocera Voice Server logs.
   a. On the VS machine, use Windows Explorer to navigate to the `\vocera\logs` directory.
   b. Open the most recent server log in a text editor.
   Vocera Voice Server log files have the following syntax, where **<Prefix>** indicates the type of log file, **<Mon>** is the three letter abbreviation of the month, **<DD>** is the day, **<YY>** is the two-digit year, and **<HHMM>** is the time in 24-hour format:
   <Prefix>-<Mon>-<DD>-<YY>-<HHMM>.txt
   For example, the following log file was created on 4-July-2014 at 11:30 PM to record system events:
   log-jul-04-14-2330.txt
   c. Search for the IP address of the Vocera Environmental Services Integration server in the logs.
   For example, search for `10.37.107.175`.

d. Confirm that you see the statement "VMI connection accepted from network address" followed by the IP address of the Vocera Environmental Services Integration server.



```
01/21/15 15:55:44.757  Speech Channel Stats    #CHANS avg=.00 max=0 all-time max=0  FREQ=.
01/21/15 15:56:14.802  Speech Channel Stats    #CHANS avg=.00 max=0 all-time max=0  FREQ=.
01/21/15 15:56:44.848  Speech Channel Stats    #CHANS avg=.00 max=0 all-time max=0  FREQ=.
01/21/15 15:57:14.900  Speech Channel Stats    #CHANS avg=.00 max=0 all-time max=0  FREQ=.
01/21/15 15:57:44.934  Speech Channel Stats    #CHANS avg=.00 max=0 all-time max=0  FREQ=.
01/21/15 15:57:47.439  Checking disk space and memory. FreeDisk=99207MB (MinDiskNeeded=150
01/21/15 15:58:14.976  Speech Channel Stats    #CHANS avg=.00 max=0 all-time max=0  FREQ=.
01/21/15 15:58:45.018  Speech Channel Stats    #CHANS avg=.00 max=0 all-time max=0  FREQ=.
01/21/15 15:59:15.055  Speech Channel Stats    #CHANS avg=.00 max=0 all-time max=0  FREQ=.
01/21/15 15:59:45.102  Speech Channel Stats    #CHANS avg=.00 max=0 all-time max=0  FREQ=.
01/21/15 16:00:15.149  Speech Channel Stats    #CHANS avg=.00 max=0 all-time max=0  FREQ=.
01/21/15 16:00:45.198  Speech Channel Stats    #CHANS avg=.00 max=0 all-time max=0  FREQ=.
01/21/15 16:01:15.236  Speech Channel Stats    #CHANS avg=.00 max=0 all-time max=0  FREQ=.
01/21/15 16:01:45.280  Speech Channel Stats    #CHANS avg=.00 max=0 all-time max=0  FREQ=.
01/21/15 16:02:15.327  Speech Channel Stats    #CHANS avg=.00 max=0 all-time max=0  FREQ=.
01/21/15 16:02:45.381  Speech Channel Stats    #CHANS avg=.00 max=0 all-time max=0  FREQ=.
01/21/15 16:02:47.888  Checking disk space and memory. FreeDisk=99207MB (MinDiskNeeded=150
01/21/15 16:03:15.428  Speech Channel Stats    #CHANS avg=.00 max=0 all-time max=0  FREQ=.
01/21/15 16:03:45.466  Speech Channel Stats    #CHANS avg=.00 max=0 all-time max=0  FREQ=.
01/21/15 16:04:15.504  Speech Channel Stats    #CHANS avg=.00 max=0 all-time max=0  FREQ=.
01/21/15 16:04:45.545  Speech Channel Stats    #CHANS avg=.00 max=0 all-time max=0  FREQ=.
01/21/15 16:05:14.783  VMI connection accepted from network address 10.37.107.175
01/21/15 16:05:14.787  RX ClientID
01/21/15 16:05:14.787  TX AckGateway iResultCode=0
01/21/15 16:05:14.788  VMI connection opened - 'vmi:EVSVoiceDriver@10.37.107.175:62512' -
```

Figure 6: Confirming the VMI connection on the VS

If the logs do not indicate a successful VMI connections, see *How to Troubleshoot the VMI Connection* on page 36.

2. Confirm the connection in the EVS logs.

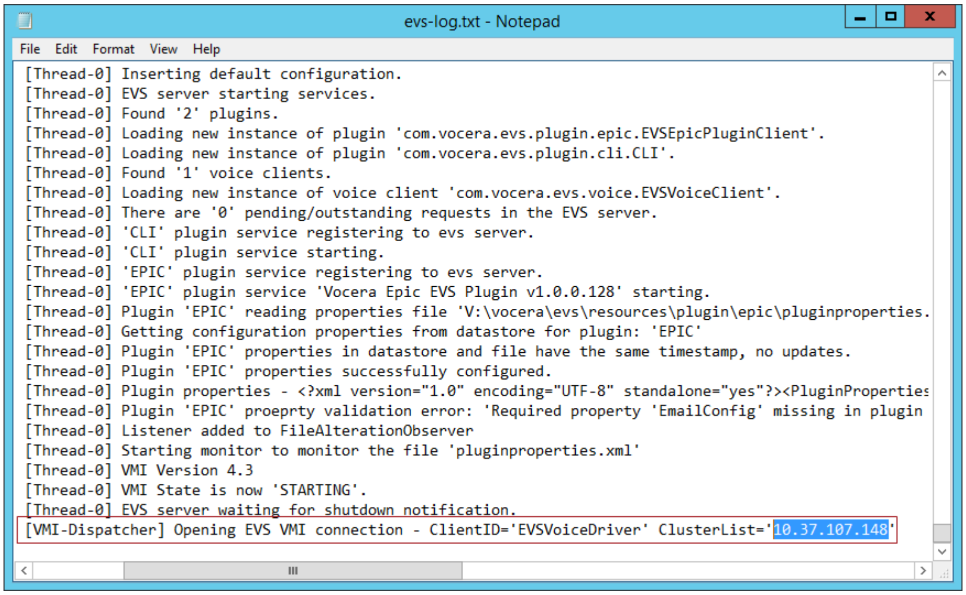
a. On the Vocera Environmental Services Integration machine, use Windows Explorer to navigate to the `\vocera\logs` directory.

b. Open the `evs-log.txt` file in a text editor.

c. Search for the IP address of the VS machine in the logs.

For example, search for `10.37.107.148`.

d. Confirm that you see the statement "Opening EVS VMI connection - ClientID='EVSVoiceDriver' ClusterList='" followed by the IP address of the VS.



```
[Thread-0] Inserting default configuration.
[Thread-0] EVS server starting services.
[Thread-0] Found '2' plugins.
[Thread-0] Loading new instance of plugin 'com.vocera.evs.plugin.epic.EVSEpicPluginClient'.
[Thread-0] Loading new instance of plugin 'com.vocera.evs.plugin.cli.CLI'.
[Thread-0] Found '1' voice clients.
[Thread-0] Loading new instance of voice client 'com.vocera.evs.voice.EVSVoiceClient'.
[Thread-0] There are '0' pending/outstanding requests in the EVS server.
[Thread-0] 'CLI' plugin service registering to evs server.
[Thread-0] 'CLI' plugin service starting.
[Thread-0] 'EPIC' plugin service registering to evs server.
[Thread-0] 'EPIC' plugin service 'Vocera Epic EVS Plugin v1.0.0.128' starting.
[Thread-0] Plugin 'EPIC' reading properties file 'V:\vocera\evs\resources\plugin\epic\pluginproperties.
[Thread-0] Getting configuration properties from datastore for plugin: 'EPIC'
[Thread-0] Plugin 'EPIC' properties in datastore and file have the same timestamp, no updates.
[Thread-0] Plugin 'EPIC' properties successfully configured.
[Thread-0] Plugin properties - <?xml version="1.0" encoding="UTF-8" standalone="yes"?><PluginPropertie
[Thread-0] Plugin 'EPIC' proeprty validation error: 'Required property 'EmailConfig' missing in plugin
[Thread-0] Listener added to FileAlterationObserver
[Thread-0] Starting monitor to monitor the file 'pluginproperties.xml'
[Thread-0] VMI Version 4.3
[Thread-0] VMI State is now 'STARTING'.
[Thread-0] EVS server waiting for shutdown notification.
[VMI-Dispatcher] Opening EVS VMI connection - ClientID='EVSVoiceDriver' ClusterList='10.37.107.148'
```

Figure 7: Confirming the VMI connection on the Vocera Environmental Services Integration Server

If the logs do not indicate a successful VMI connections, see *How to Troubleshoot the VMI Connection* on page 36.

How to Troubleshoot the VMI Connection

Most problems with the VMI connection are due to entering the IP address of the Vocera Voice Server or cluster incorrectly. You cannot solve this problem by editing the value of the environment variable alone. To troubleshoot this issue, you must also delete the IP address list from the database and set up the VMI connection again.

Note: Before you begin, double-check to make sure that your system has an available VMI license. An unavailable VMI license also prevents you from establishing the VMI connection.

To troubleshoot an incorrect IP address:

1. Update the value of the `VOCERA_SERVER_IP_ADDRESS` environment variable to the correct IP address or cluster list.

   If necessary, review the information in *How to Set Up the VMI Connection* on page 32 to understand how to edit the environment variable.
2. Disable the MySQL "Safe Updates" preference to allow you to update the database easily:

   a. Use the Windows **Start** menu to launch MySQL Workbench.

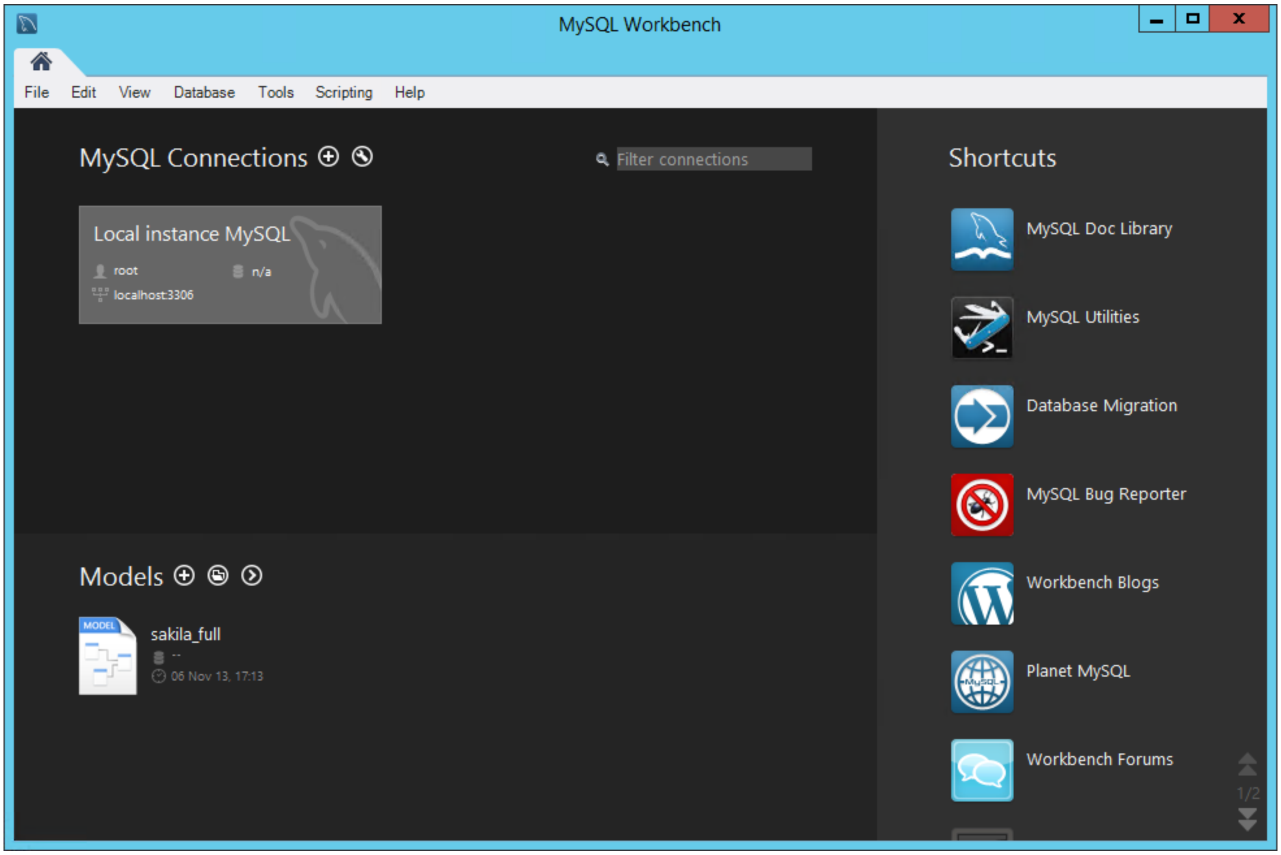
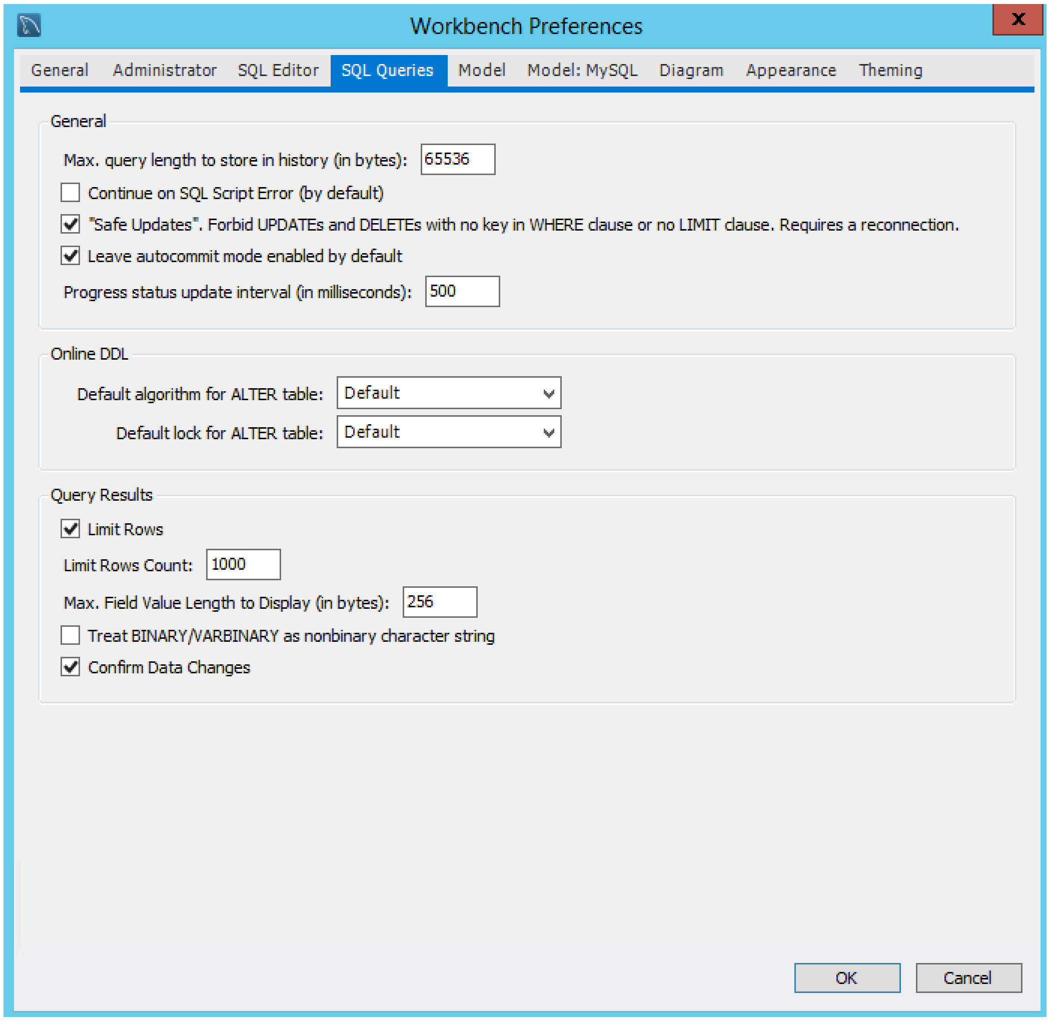


Figure 8: The MySQL Workbench Main Window

   b. Choose **Edit|Preferences** from the menu.

   The **Workbench Preferences** dialog box appears.

   c. Select **SQL Queries** in the menu of the **Workbench Preferences** dialog box.



General

Max. query length to store in history (in bytes): 65536

□ Continue on SQL Script Error (by default)

☑ "Safe Updates". Forbid UPDATEs and DELETEs with no key in WHERE clause or no LIMIT clause. Requires a reconnection.

☑ Leave autocommit mode enabled by default

Progress status update interval (in milliseconds): 500

Online DDL

Default algorithm for ALTER table: Default

Default lock for ALTER table: Default

Query Results

☑ Limit Rows

Limit Rows Count: 1000

Max. Field Value Length to Display (in bytes): 256

□ Treat BINARY/VARBINARY as nonbinary character string

☑ Confirm Data Changes

Figure 9: The SQL Queries Preferences

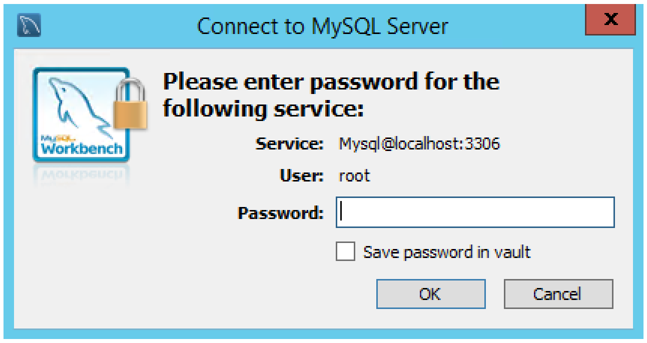
d. Uncheck **Safe Updates** and click **OK**.

   The **Workbench Preferences** dialog box closes.

3. Delete the incorrect IP address from the database as follows:

   a. In the MySQL Workbench Main Window, double-click **Local Instance MySQL**.

      The **Connect to MySQL Server** dialog box appears.



Please enter password for the following service:

Service: Mysql@localhost:3306

User: root

Password:

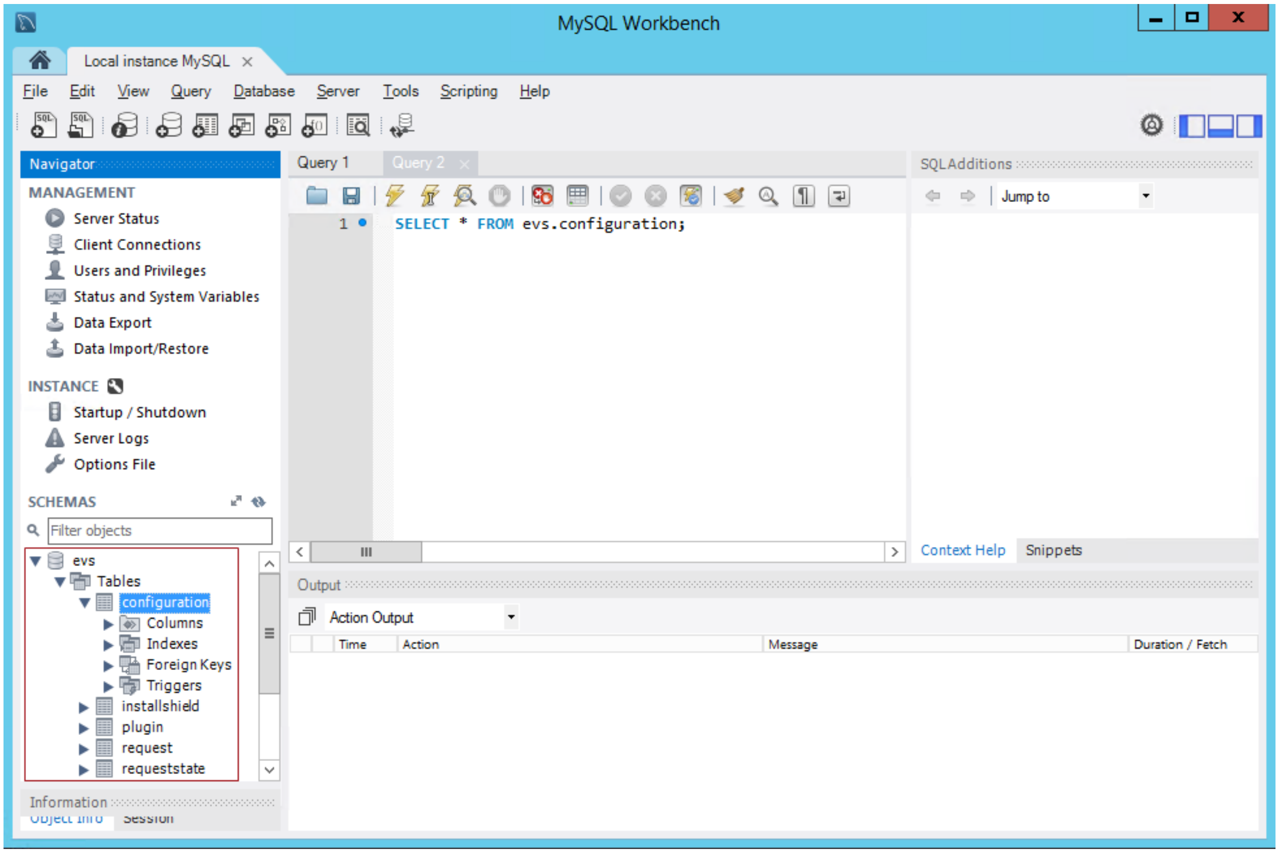
□ Save password in vault

Figure 10: The Connect to MySQL Server dialog box

   b. Enter the password `vocera` and click **OK**.

      The **Local Instance MySQL** dialog box appears.

   c. In the **Schemas** section of the dialog box, navigate to **evs|Tables|configuration**.

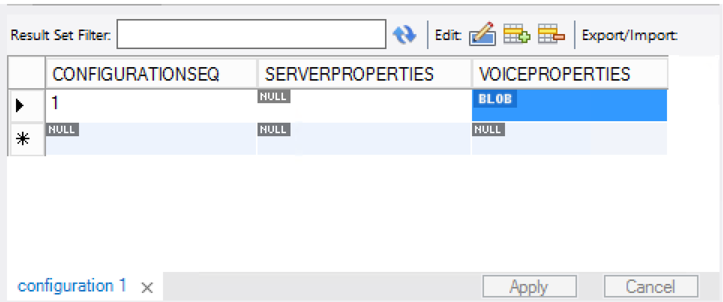


```
1 SELECT * FROM evs.configuration;
```

Figure 11: Editing the Database Schema

d. Right-click **configuration** and choose **Select Rows**.

The first row in the **configuration** table appears. This row contains the value of the Vocera Voice Server IP address.



	CONFIGURATIONSEQ	SERVERPROPERTIES	VOICEPROPERTIES
▶	1	NULL	BLOB
✳	NULL	NULL	NULL

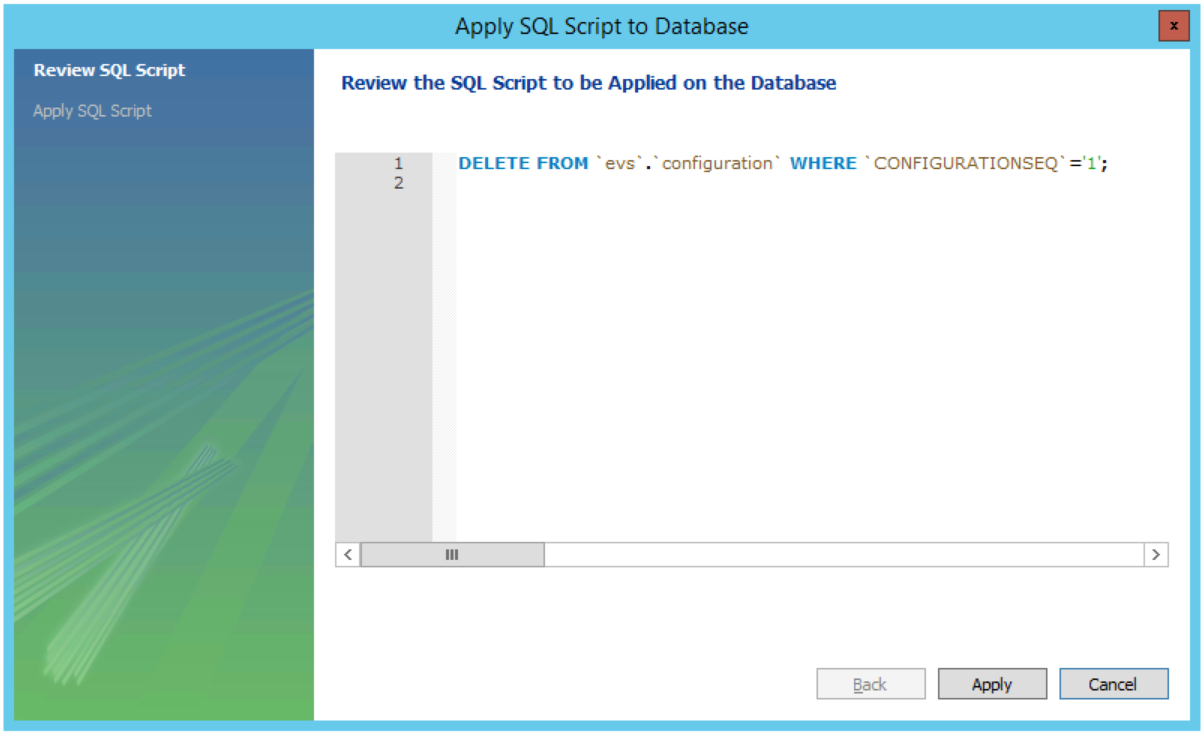
Figure 12: Editing the Configuration Table

e. Right-click the first row and choose **Delete Row(s)**.

The editor marks the first row of the **configuration** table for deletion and removes it from the display.

f. Click **Apply**.

The **Apply SQL Script to Database** dialog box appears.

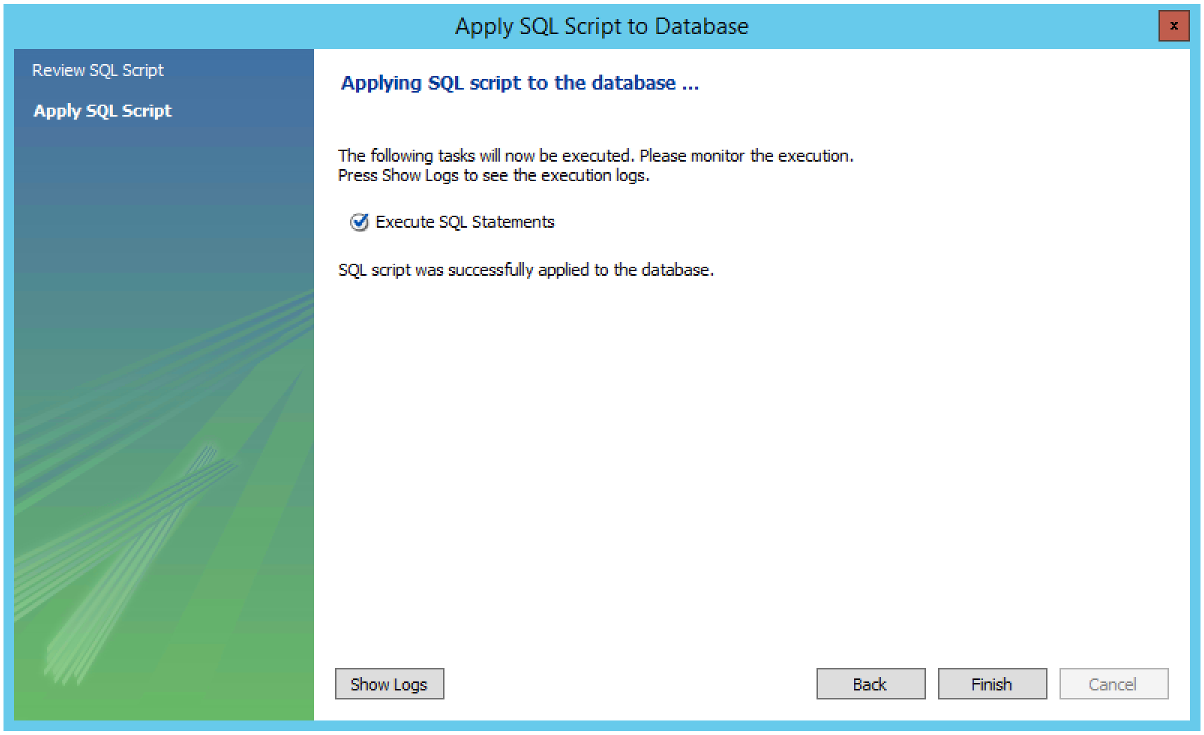


Review the SQL Script to be Applied on the Database

```
DELETE FROM `evs`.`configuration` WHERE `CONFIGURATIONSEQ`='1';
```

Figure 13: Apply SQL Script to Database dialog box

g. Review the SQL statement in the **Apply SQL Script to Database** dialog box and click **Apply**.
   The editor displays the message "Applying SQL script to database..." and deletes the row from the **configuration** table.



Applying SQL script to the database ...

The following tasks will now be executed. Please monitor the execution.
Press Show Logs to see the execution logs.

☑ Execute SQL Statements

SQL script was successfully applied to the database.

Figure 14: Applying SQL Script

h. Click **Finish**.
   The **Apply SQL Script to Database** dialog box closes.

i. Close MySQL Workbench.

The value of the `VOCERA_SERVER_IP_ADDRESS` environment variable is removed from the database.

4. Restart the Vocera EVS Service.
   The EVS application writes the value of the Vocera Voice Server IP address list to the database again.

   Note: If necessary, review the information in *How to Set Up the VMI Connection* on page 32 to understand how to restart the EVS service.

Backing Up and Restoring the EVS Server

EVS provides a script-based backup and restore process for the EVS server. Vocera recommends that you perform a full backup every day and enable incremental backups.

Backing Up the EVS Server

Follow these steps to perform a full backup of the EVS server and to optionally enable incremental logging.

Note: To back up the EVS server, you will need the following files: `evs-backup.bat`, `evs-restore-incremental.bat`, and `my-backup.ini`. Contact Vocera technical support to obtain these files.

1. Locate the drive on which you have installed EVS. In the `vocera\evs\tools` subfolder, copy the following files:
   - `evs-backup.bat`: A batch file script that executes the MySQL dump command (`mysqldump`) on the EVS database. If the total number of backup files is greater than 10, old backup files are removed.
   - `evs-restore-incremental.bat`: A batch file script that joins MySQL binary logs together and loads them into MySQL.
2. Copy the `my-backup.ini` file to the `%APPDATA%\MySQL` folder. Create this folder if it does not exist. Access to this file should be restricted to the owner and possibly to the system administrators.

   Make sure you select a user account that is local to the server and has a password which does not expire.
3. On the drive on which you have installed EVS, in the `vocera\evs` folder, create a new subfolder named `backup`.
4. Use the Windows Task Scheduler to create a backup task. The account that runs this backup task must be a local account with administrator privileges and a non-expiring password. See *Creating the Backup Task* on page 42 for more information on creating this backup task.
5. Optionally enable incremental logging:

   a. Stop the EVS and MySQL services.

   b. Using Notepad or your favorite text editor, open the file `%MYSQL_HOME%\my.ini`, where `%MYSQL_HOME%` is the value of the `MYSQL_HOME` environment variable.

   c. At the bottom of this file, paste the following text (replace `C:` with the location of your hard drive):

   ```
   #Enable binary logs
   log-bin = "C:/vocera/evs/backup/evs-bin"
   #Set the number of days to preserve binary logs
   expire_logs_days = 1
   ```

   d. Save and close the `my.ini` file.

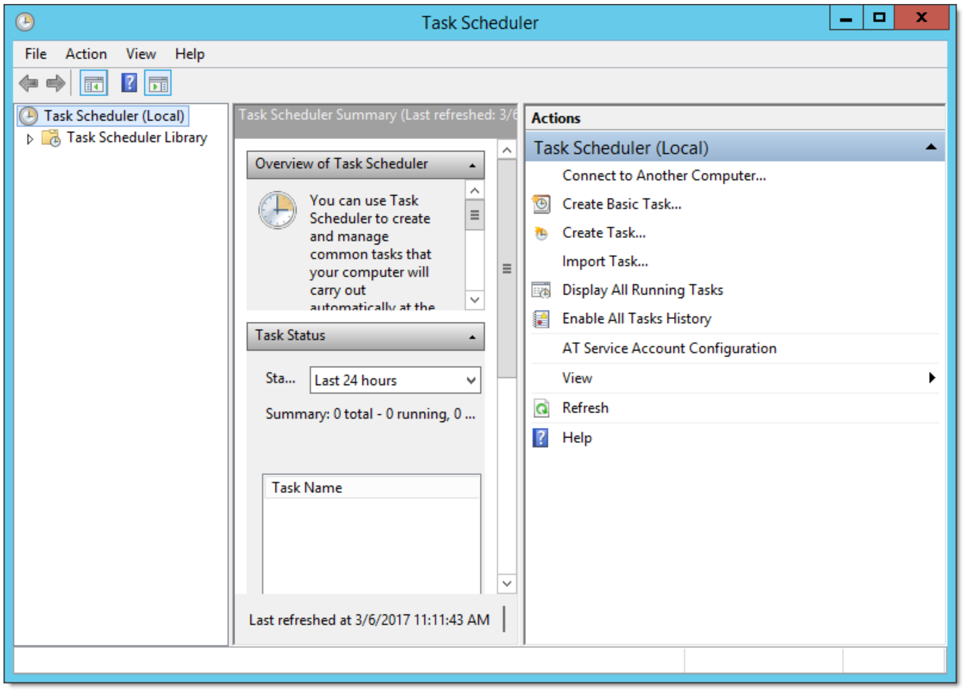
   e. Restart the MySQL and EVS services.

Creating the Backup Task

To perform a backup of the EVS server, you must use the Windows Task Scheduler to set up the daily backup.

To schedule the daily backup using the Windows Task Scheduler:

1. Open the **Windows Administrative Tools Control Panel** and choose **Task Scheduler**.
   The **Task Scheduler** window opens.

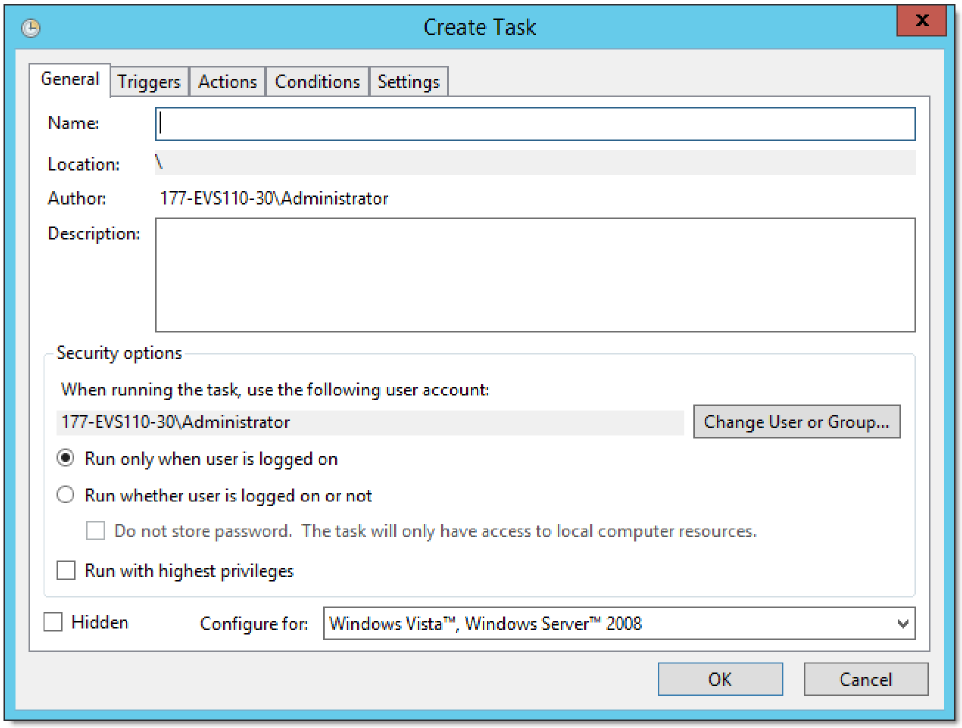


You can use Task Scheduler to create and manage common tasks that your computer will carry out automatically at the

Summary: 0 total - 0 running, 0 ...

Last refreshed at 3/6/2017 11:11:43 AM

2. In the menu system, choose **Action > Create Task**.
   The **Create Task** window opens.



Name:

Location: \

Author: 177-EVS110-30\Administrator

Description:

Security options

When running the task, use the following user account:

177-EVS110-30\Administrator

● Run only when user is logged on

○ Run whether user is logged on or not

□ Do not store password. The task will only have access to local computer resources.

□ Run with highest privileges

□ Hidden

Configure for: Windows Vista™, Windows Server™ 2008

3. On the **General** tab, specify the following options:

a. In the **Name** field, enter `evs-backup.bat`.

b. Click **Change User or Group**.
   The **Select User or Group** dialog box appears.

c. Use the **Object Types** control to specify the name of an account with administrative privileges.

> **Note:** This must be the user whose `AppData\MySQL` folder contains the file `my-backup.ini`. See *Backing Up the EVS Server* on page 41 for more information on this file.

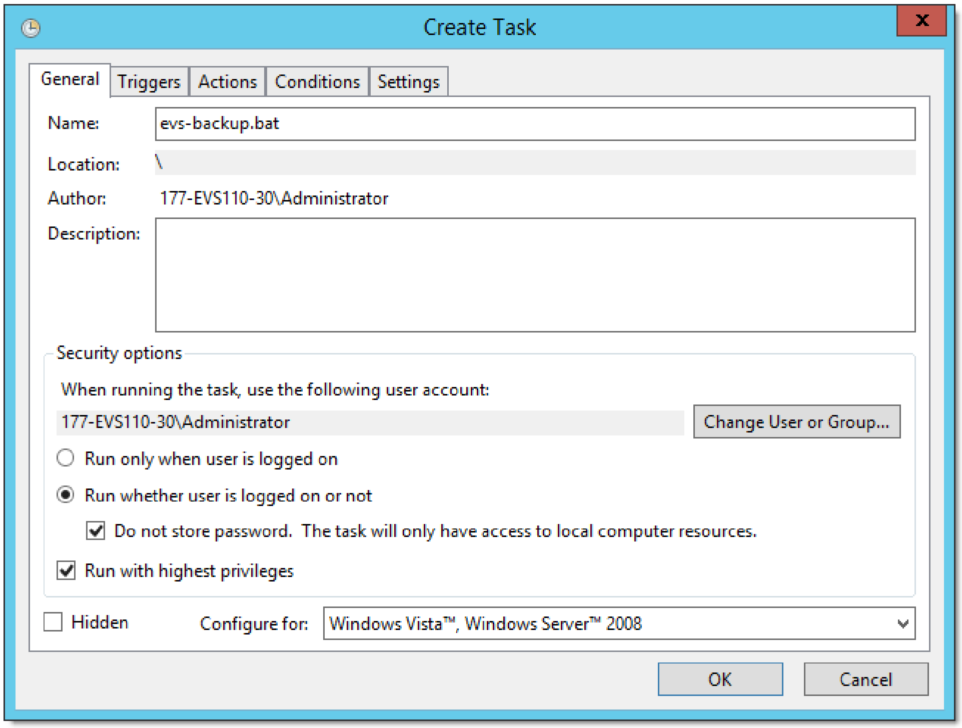
d. After you specify the account name, click **OK** to close the **Select User or Group** dialog box.

e. Select **Run whether user is logged on or not**.

f. Select **Do not store password**.

g. Select **Run with highest privileges**.

When you are finished, the **General** tab of the **Create Task** dialog box should look similar to the following:



Name: evs-backup.bat

Location: \

Author: 177-EVS110-30\Administrator

Description:

Security options

When running the task, use the following user account:

177-EVS110-30\Administrator

○ Run only when user is logged on

● Run whether user is logged on or not

☑ Do not store password. The task will only have access to local computer resources.

☑ Run with highest privileges

□ Hidden

Configure for: Windows Vista™, Windows Server™ 2008

4. Navigate to the **Triggers** tab and set the following options:

a. Click **New** at the bottom of the window.
   The **New Trigger** window opens.

b. In the **Begin the task** list, make sure that **On a schedule** is selected.

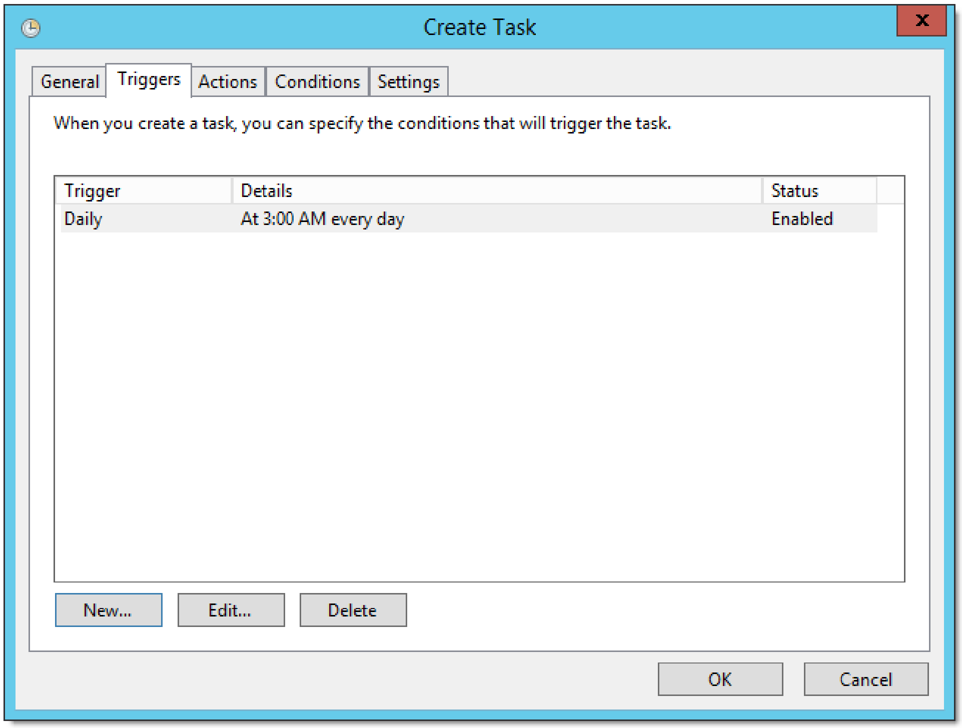
c. Select **Daily**.

d. In the **Start** section, leave the start date as the current date and choose a start time. Choose a time that is least active.

e. Click **OK** to close the **New Trigger** window.

f. Make sure **Enabled** is checked.

Click **OK** to close the **Edit Trigger** dialog box. The **Triggers** tab of the **Create Task** dialog box should look similar to the following:



When you create a task, you can specify the conditions that will trigger the task.

Trigger	Details	Status
Daily	At 3:00 AM every day	Enabled

5. Navigate to the **Actions** tab and set the following options:

   a. Click **New** at the bottom of the window.

      The **New Action** window opens.

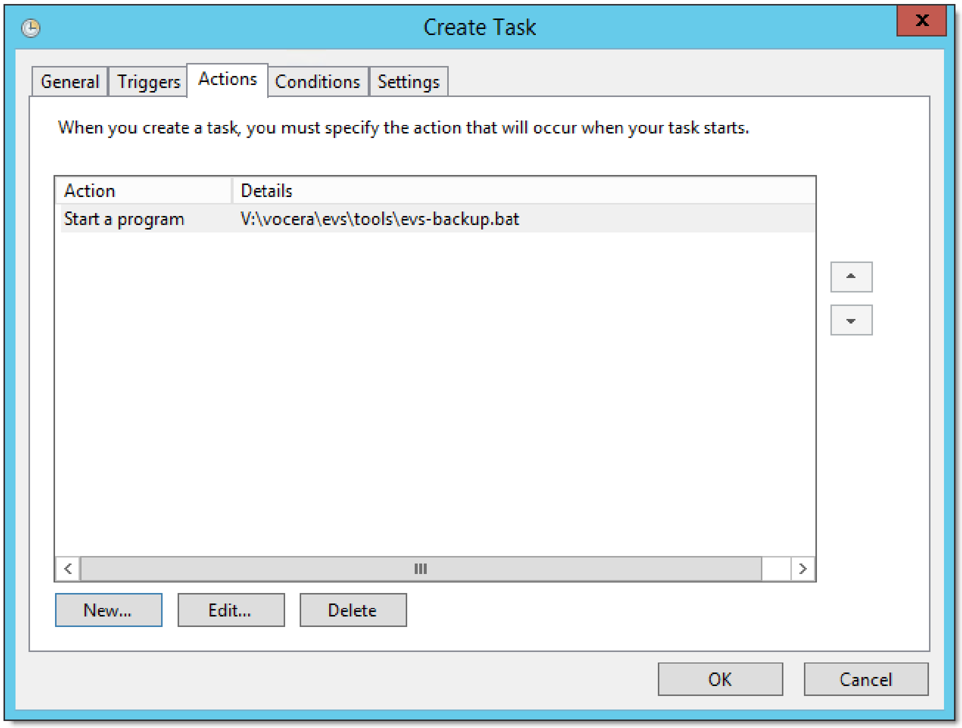
   b. In the **Action** list, make sure that **Start a program** is selected.

   c. Click **Browse**.

   d. On the drive on which EVS is installed, navigate to `vocera\evs\tools` and select `evs-backup.bat`.

   e. Click **OK** to close the **New Action** dialog box.

   The **Actions** tab of the **Create Task** dialog box should look similar to the following:

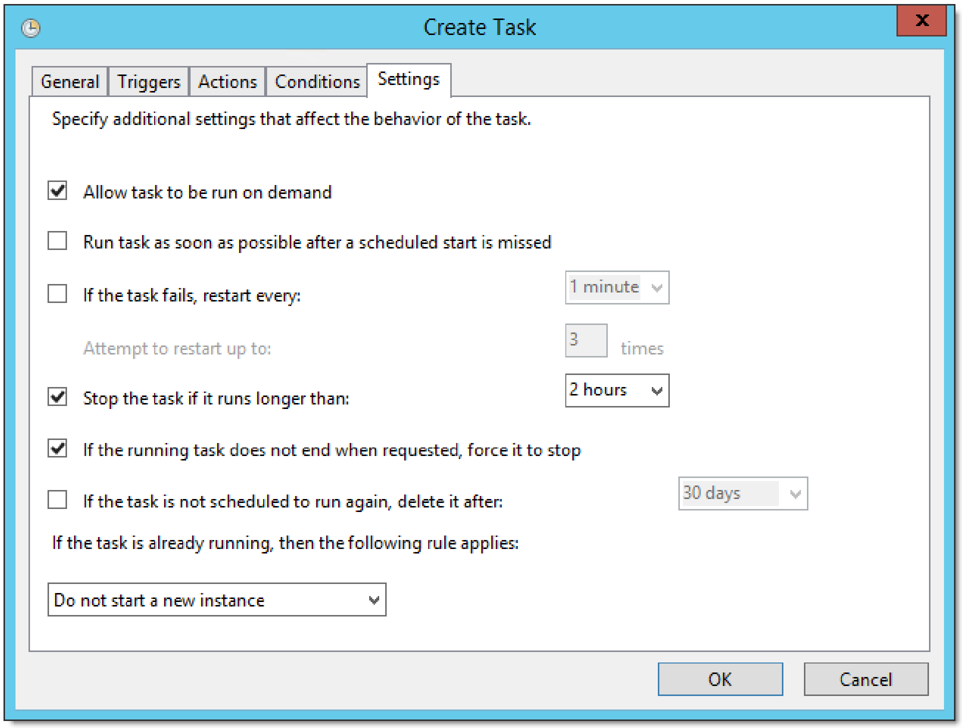


When you create a task, you must specify the action that will occur when your task starts.

Action	Details
Start a program	V:\vocera\evs\tools\evs-backup.bat

6. Navigate to the **Settings** tab and set the following options:
   a. In the **Stop the task if it runs longer than** list, select **2 hours**.

   The **Settings** tab of the **Create Task** dialog box should look similar to the following:



Specify additional settings that affect the behavior of the task.

☑ Allow task to be run on demand

□ Run task as soon as possible after a scheduled start is missed

□ If the task fails, restart every: 1 minute

Attempt to restart up to: 3 times

☑ Stop the task if it runs longer than: 2 hours

☑ If the running task does not end when requested, force it to stop

□ If the task is not scheduled to run again, delete it after: 30 days

If the task is already running, then the following rule applies:

Do not start a new instance

7. Click **OK** to close the **Create Task** window. Windows saves the task you created.

Restoring the EVS Server

If you have created a backup of the EVS server, follow these steps to restore it.

To perform the restore itself, you can either use the Windows command line or use the MySQL workbench.

1. Stop the EVS service.
2. If you have enabled incremental logging:
   a. Restart the MySQL service.
   b. Execute the command `mysqladmin flush-logs`.
3. To perform the restore from the Windows command line:
   a. Open a Command Prompt window.
   b. Run the command `%MYSQL_HOME%\mysql -u root -v`.
   c. When prompted for a password, provide the password for the root user. The default password is `vocera`.
   d. Type the command `\. path`, where `path` is the path to the desired backup file.
4. To perform the restore from the MySQL workbench:
   a. Open the MySQL workbench. This is included with EVS.
   b. Connect to the EVS database (`localhost:3306`).
   c. Use **Open SQL Script** to open the backup file (or type `Ctrl+Shift+O`).
   d. Run the SQL script.
5. To optionally perform an incremental restore:
   a. On the drive on which you have installed EVS, go to the `evs\backup` folder.

b. Locate all files with prefix `evs-bin.` whose date is newer than the time of the error or reinstall. Copy these files to a temporary directory.

c. Run the batch script `evs-restore-incremental.bat`, passing the location of the temporary directory as its argument. For example, if you have copied files to the `C:\vocera\temp\recover` folder, run the following command:

```
evs-restore-incremental.bat C:\vocera\temp\recover
```

d. If the most recent of your incremental recovery files contains transactions that you want to exclude from the start or end, you can modify the restore command in the `evs-restore-incremental.bat` file to include either of the following:

- `--start-datetime=datetime`: Exclude transactions before the time specified in `datetime`.
- `--stop-datetime=datetime`: Exclude transactions after the time specified in `datetime`.

For example, to restore transactions prior to 9:00 on April 1, 2017 only, edit the `evs-restore-incremental.bat` file. Change the line

```
for %%i in (".\evs-bin*") do (%MYSQL_HOME%\bin\mysqlbinlog %%i --database=evs >>
 recovery.sql)
```

to

```
for %%i in (".\evs-bin*") do (%MYSQL_HOME%\bin\mysqlbinlog %%i --stop-
datetime="2017-04-01 09:00:00" --database=evs >> recovery.sql)
```

6. Start the EVS service.

Appendix: Special Characters in XML

When providing field values in an XML file, make sure that special characters are encoded using the proper escape sequences. A character is a special character if it has a specific meaning in XML.

The following table lists the special characters and their escape sequences.

Special Character	Description	Escape Sequence
"	Double quotation mark	&quot;
&	Ampersand	&amp;
'	Apostrophe (single quotation mark)	&apos;
<	Open angle bracket (less-than sign)	&lt;
>	Close angle bracket (greater-than sign)	&gt;