Introduction

Custom Commands allow OpenLM users to create functions for personalizing their OpenLM experience. This is an extremely flexible tool which allows users to run scripts for managing specialized situations for licenses and permissions that occur with changing usage status. The opportunities include, but are not at all limited to:

- Displaying any message such as a warning
- Change registry entries to allow creative management of available licenses
- Launch any program or process automatically

Custom Commands are created in OpenLM's Administration controls through OpenLM EasyAdmin using Agent Procedures, Alert Management or App Manager. Agent Procedures can initiate execution of script or executable in a defined path on the agent machine. Alert Management can run a program on OpenLM Server or run a command on the license management server through OpenLM Broker. OpenLM App Manager provides customized scripting functionality for rules created in the App Manager during allowing or denying license consumption. Scripts run as an application (user owned) or as a service (system owned).

Creating a custom command requires a series of steps to define the processes and how the processes are initiated. As the variations of what can be accomplished are potentially endless, the setup here will look at an overview of creating commands. *[NOTE: Custom Commands was not intended to initiate long running processes. There is a 2 minute limitation on execution of any command. After 120 seconds wait, command execution will respond with a failure if it is not completed (an initiated application launch is considered a success.)]*

The feature requires separate installation of component parts (e.g., OpenLM Server and OpenLM Agent), creation and identification of processes, and launch. OpenLM Broker enables additional functionality.

This document covers:

- Creating Custom Commands with App Manager

Please see dedicated documentation for more information on other components not contained in this document:

- OpenLM Server
- OpenLM Agent
- OpenLM Broker
- OpenLM App Manager

Custom Command License Download and Installation

The additional functionality for Custom Commands is free to use but requires downloading an additional license and acknowledgement of the OpenLM usage agreement.

1. **Request Custom Command access** from OpenLM by sending an email to OpenLM sales (sales@openlm.com). After the request, an email will be sent to the address provided. Please be sure to add the sending address (**license@openlm.com**) to your list of approved contacts if you use spam filters.

2. Open the email and respond by clicking the **License Download Link**. This will open a web page to login (see **Figure 1**). This page can only be used one time as the one-time access token will expire.

Figure 1: The login page.

3. Enter the email address used to obtain the link.

4. Click **[Next]** to advance to the **Terms and Conditions** form (see **Figure 2**).

Figure 2: The Terms and Conditions form.

5. Read the agreement form.

6. Check the **"I agree to these terms and conditions"** only if all of the terms and

conditions are acceptable.

7. Click the **[Download Now]** button. This will advance you to the download page where there are options for downloading the license and the support agreement (see **Figure 3**).

Figure 3: The download page.

8. Locate the license file in the download folder used by your browser (e.g., C:\Users\user\Downloads).

9. Copy the license file to the OpenLM Licenses folder (e.g., C:\Program Files (x86)\OpenLM\OpenLM Server\License).

10. Restart the OpenLM Server for the license to take effect and activate Custom Commands.

OpenLM Services Check

You will need to check that services are running in order to use Custom Commands successfully. This checks includes the following described in the steps below:

1. Check the connection to OpenLM Server
2. Check on availability of OpenLM SoftwareLocker

A. Check the Connection to the OpenLM Server

1. Open the OpenLM Agent Configuration tool. This is located in the Show Hidden Icons menu at the right of the Windows Taskbar. See **Step #1.a** to **Step #1.c** below along with **Figure 4**.

a. Click the arrow (1) to display the hidden icons panel.

Figure 4: Steps for opening the OpenLM Agent Configuration tool.

b. Right-click the OpenLM icon (2) and a pop-up menu will appear with a link to the **OpenLM Agent Configuration** screen (3).

c. Click the link to open the Agent Configuration screen (see **Figure 5**).

Figure 5: The OpenLM Agent Configuration screen.

2. Click the **[Check Connectivity to OpenLM Server]** button. A screen should appear confirming the connection to the server (see **Figure 6**). If the check is successful skip to **Step #3**. If the attempt to connect is not successful, a warning screen will appear (see **Figure 7**). If the warning appears, check that the server is running, that the configuration is correct, and repeat this step. If the connection is still not successful, contact OpenLM support (support@openlm.com).

Figure 6: The server connection check success screen.

Figure 7: The server connection warning screen.

3 . Click **[OK]** to close the success screen.

4. Click **[Apply]** to save the changes. The OpenLM Agent Configuration Tool will close.

The connection to the server is confirmed and Custom Command functionality should be enabled.

B. Check on Availability of OpenLM SoftwareLocker

To run scripts as a service from OpenLM Server on the workstation, you will need to be sure OpenLM SoftwareLocker is running on the workstation. This check is only necessary for those planning to customize Agent Procedures using OpenLM Server and the agent service (i.e., OpenLM Agent installed with the 'install service' option at a minimum. See the OpenLM Agent Installation documentation for more information). If this functionality is not desired, it does not affect other capabilities; skip to Creating Custom Commands with Agent Procedures. To check availability, follow the steps below.

1 .Open Windows Services (e.g., C:\WINDOWS\system32\services.msc or type “Services” in the Windows search on the taskbar and choose the Services App from the result).

2 .To do this, locate OpenLM SoftwareLocker in the Name listing and right-click. The popup menu will reveal the status (e.g., if the service is running, the “Start” item will not be available.).

3. If the service is not running, right-click the service name in the Windows Services screen and choose “Start” from the pop-up menu.

4. When the service is running (see **Step #2**), close the Services window by clicking the **[X]** in the upper right of the screen.

With services running and the server connection validated, it is time to create and configure custom commands.

Creating Custom Commands with Agent Procedures

Procedures initiated by Custom Commands may take multiple criteria into account as well as initiating the steps in a productive workflow. The basic steps for setup of Agent Procedures is completed in EasyAdmin using the following outline:

1. Define Custom Agent Procedures
2. Define Unmanaged Processes
3. Configure Unmanaged Processes to Enable Custom Procedures

A. Define Custom Agent Procedures

1. Open EasyAdmin **[Start] > [OpenLM] > [OpenLM EasyAdmin]**. This will launch a web browser session for OpenLM EasyAdmin. The session will open with the Alerts, Feature Usage Status, and License Servers screens visible by default.

2. Click **[Start]** on the EasyAdmin start bar and select **[Administration]** from the

menu (see **Figure 8**). This will open the Administration screen (see **Figure 9**).

Figure 8: Select Administration from the EasyAdmin Start menu.

Figure 9: The EasyAdmin Administration screen.

3. Click the **Agent Procedures** icon on the Administration screen (highlighted in **Figure 10**) to open the **Administration — Agent Procedures** screen (see **Figure 11**).

Figure 10: The Administration screen with Agent Procedures highlighted.

Figure 11: The Agent Procedures screen.

4. Click on the **[Add]** button. This will open the **Agent Procedure Editor** screen (see **Figure 12**). The **Enabled** box will be checked by default.

Figure 12: Agent Procedure Editor screen.

5. Enter a **Name** for the procedure in the Name field. This is a free text field. It is helpful to use a name that is easy to identify as the name will appear in the Administration — Agent Procedures screen when later adding, editing or deleting items in the Custom Command setup. Standardizing naming schemes is recommended.

6. Click on the **[+Add Action]** button on the Agent Procedure Editor screen. This will add a row to the bottom panel of the screen with default values (see **Figure 13**). Default values are listed below.

Action Type = License Manager

Script Info = <*Empty*>

Execute Condition = NoWait

Run At = <*Empty*>

Active = [Checked]

Figure 13: Agent Procedure Editor screen with new row added.

7. Choose an **Action Type**. To do this follow **Step #7.a** to **Step #7.c**.

a. Double-click the default value for the **Action Type** in the new row to activate the field.

b. Click the arrow that appears on the right of the field to open the Action Types options (see **Figure 14**).

Figure 14: View of the Action Type drop menu.

c. Select “**AgentScript**” by clicking on it the drop menu. *[NOTE: “AgentScript” is the only value applicable to Custom Commands.]*

8. Enter the **Script Info**. To do this follow **Step #8.a** to **Step #8.c**.

a. Double-click the Script Info field.

b. Add the script. Script options are available in the interface by clicking the **[?]** icon. The following list summarizes these options. **Table 1** includes example commands.

- The Script Info field should contain commands using Windows Shell scripting.
- Any executable file or application needs to be defined by a full path.
- Selecting ‘Wait Success’ as an Execution Condition executes a command only after the previous command is successfully completed.
- Prefixing any script line with ‘@’ executes the command line and moves to the next action without waiting for the action to complete (e.g., if a script prefixed with ‘@’ initiates copying files, the OpenLM Server will not wait for the files to copy before initiating the command that follows).
- Script lines without an ‘@’ prefix execute the command line and moves to the next action after waiting for the action to complete (e.g., if a script without ‘@’ prefix

initiates copying files, the OpenLM Server will wait for the files to copy before initiating the command that follows).

Example commands:	
Activate an application and launch the next action	@<path>\[filename].exe
Activate an application and wait for it to end before launching the next action	<path>\[filename].exe
Activate a batch command and launch the next action	@<path>\[filename].bat
Activate a batch command and wait for it to complete before launching the next action	<path>\[filename].bat

Table 1: Command examples with controlled execution.

c. Press **[Return]** or **[Enter]** on the keyboard to accept the changes to the script field.

9. [Optional] Change the **Execute Condition** which will be prepopulated. Execute Condition allows users to define how an action is implemented as per the options listed in **Table 2**. To change the condition follow **Step #9.a** to **Step #9.d**.

a. Double-click the Execute Condition field to activate it.

b. Click the down arrow to the right of the field to open the drop list of selections.

c. Choose from the drop list. *[NOTE: The items in the drop list are context sensitive.]*

Condition	Description
No Wait	This condition is the default for the first action. The script will be executed without waiting.
Wait Complete	Executes only after the previous action is complete.
Wait Success	Executes only after the previous action completes successfully. If a previous action completes with error, the procedure is terminated.

Table 2: Execution Condition options.

d. Press **[Return]** or **[Enter]** on the keyboard to accept the changes to the field.

10. [Optional] Choose a **Run At** option. Run At allows users to define how the script step should be run — by the Agent Application domain or the Agent Service domain (see **Table 3** for condition descriptions). To choose a Run At option, follow **Steps #10.a** to **Step #10.d**.

a. Double-click the **Run At** field.

b. Click the down-arrow to the right of the field to reveal options for the field.

c. Select an option from the drop list by clicking directly on it in the list.

Condition	**Description**
Application	• Run the script in Agent Application domain, which is intended for on-screen viewing and user interaction (e.g., show messages or display applications).. • Runs as the current user's UserName. • Cannot influence System-owned tasks.
Service	• Run the script in Agent Service domain, which is intended for silent procedures (e.g., manage the registry or configure security). • Runs as the System user. • Cannot influence UserName-owned tasks

Table 3: Run At options.

d. Press **[Return]** or **[Enter]** on the keyboard to enter the changes.

11. Repeat **Step #6** to **Step #10** for other procedures that will be part of the process.

12. When the process is assembled as desired, click **[OK]** to commit the changes and close the Agent Procedure Editor window. This will return the user to the Administration — Agent Procedures screen. *[NOTE: Any errors in entry will be displayed at the bottom of the entry screen. If there is an error, address the error message and repeat this step.]*

13. [Optional] Repeat **Steps #4** to **Step #12** to add additional procedures. All additional procedures should have a unique Name.

14. Close the Administration — Agent Procedures screen by clicking the **[x]** in the upper right of the screen window. This will display the Administration screen again.

B. Define Unmanaged Processes

Unmanaged Processes allows the creation of vendors and processes where customized behaviors created in the Agent Procedure setup are assigned. These unmanaged processes must exist in the system or be created to configure them in section C. Configure Unmanaged Processes to Enable Custom Procedures. Please refer to the documentation dedicated to this setup for more information (Working with Unmanaged Licenses).

C. Configure Unmanaged Processes to Enable Custom Procedures

Managing created processes requires editing to establish desired runtime behaviors to control such things as frequency of behavioral checks, idle-time behaviors, and initiating or enabling the processes. *[NOTE: The details here are specific to set-up of custom procedures. For example, this document assumes Vendors and Features have already been configured. For more information on adding Vendors and Features or additional configuration information, please read the dedicated documentation for Working with Unmanaged Licenses.]*

1. Open the Administration — Unmanaged Processes screen (**[Start] > [Administration] > [Unmanaged Processes]**).

2. Click the **Vendor** name which includes the process to manage. This will display the available processes defined for the Vendor in the lower half of the screen.

3. Double-click the process to manage. This will open the Edit Process screen (see **Figure 15**).

Figure 15: The Edit Process screen.

4. *[Optional]* Check the **Enable Automatic Process Release Functionality** box to activate the options in the panel (see **Figure 16**).

Figure 16: The Enable Automatic Process Release Functionality panel activated.

5. Check the **Track Process Idle/Active Periods** box (see **Figure 17**).

Figure 17: The Track Process Idle/Active Periods box checked.

6. Click the **[+]** button to the right of the **Procedure Name** field. This will open the **Administration — Agent Procedures** screen (see **Figure 18**). *[NOTE: The procedures are those created in the Define Custom Agent Procedures section of this document.]*

Figure 18: The Administration — Agent Procedures screen

7. Click on the desired procedure to highlight it.

8. Click the **[Select]** button. This will close the Agent Procedures screen and populate the Procedure Name field with the selected procedure (see **Figure 19**).

Figure 19: Edit Process screen with added procedure.

9. Click the **[Save]** button to commit the changes and exit the Edit Processes screen.

At this point setup is complete and the process will be enabled. It may be desirable to disable processes for a variety of reasons. The following steps allow users to temporarily disable the process.

To Disable a Procedure

1. Open the Unmanaged Processes screen (**[Start] > [Administration] > [Unmanaged Processes]**).

2. Click the Vendor name where the process resides. This opens the Edit Process screen.

3. Double click the process which contains the procedure to suspend.

4. Uncheck the **Enable Process Release Functionality** box by clicking it. The panel will gray out and be disabled. All the settings will be retained.

5. Click **[Save]** to commit the changes.

The process will be disabled and can be reinstated by checking the Enable Process Release Functionality box again.

Creating Custom Commands with Alert Management

System managers can define alert conditions which trigger behaviors that present alert messages or run custom scripts. When Custom Commands are enabled several features are added to Alert Management functionality:

1. Run a Program on OpenLM Server
2. Run a Broker Command on a License Manager Server

Both commands are managed from the Alert Management module in OpenLM EasyAdmin. Additional information about the specifics of Alert Management configuration can be found in the dedicated OpenLM Alerts Configuration document.

A. Run a Program on OpenLM Server

1. Open the Administration screen in OpenLM EasyAdmin (**[Start] > [Administration]**).

2. Click on the Alerts Management icon (see **Figure 20**). This will open the Alert Rules screen (see **Figure 21**).

Figure 20: Choose Alerts Management from the Administration screen.

Figure 21: The Alert Rules screen.

3. Click **[+ Add Rule]**. This opens the Add Alert Rule screen (see **Figure 22**). Required fields will be highlighted with an exclamation point ( ).

Figure 22: The Add Alert Rule screen.

4. Enter a name for the rule in the **Rule Name** field.

5. Click the down arrow on the **Severity** drop list and choose a level of severity from the list of choices (Alert, Warning or Notice).

6. Configure the **Frequency**. *[NOTE: Consult the Alert Management documentation for details on configuring Frequency.]*

7. Click the **[Save]** button. The rule will be stored and the Add Alert Rule screen will close. The Alert Rules screen will populate with the new rule and additional fields for configuration will become available (see **Figure 23**).

Figure 23: The Alert Rules screen populated with the added rule.

8. Set a **Condition**. To do this follow **Step #8.a** to **Step#8.e**.

   a. Choose a **Type** from the Type drop list on the Conditions panel.

   b. Click the **[+Add]** button to the right of the selected Type to open the Add Condition screen (See **Figure 24**).

Figure 24: The Add Condition screen for Feature Expiration Date.

[NOTE: The Condition panel on the Add Condition screen is context sensitive to the selection made for Type.]

   c. Enter the required **Condition** parameters.

   d. Enter the required **Filter** parameters. When all the required parameters are entered, the **[OK]** button will become available.

   e. Click **[OK]** to save the condition and close the screen. A row will be added to the Conditions panel on the Alert Rules screen (see **Figure 25**).

Figure 25: A condition added to the Alert Rules screen.

9. Set an Action. To do this follow **Step #9.a** to **Step#9.f**.

a. Choose **Run a Program on OpenLM Server** from the **Type** drop list on the Actions panel.

b. Click the **[+Add]** button to the right of the Type field to open the Add Action screen (See **Figure 26**).

Figure 26: The Add Action screen.

c. Enter a **Command** to run on the server.

d. [Optional] Configure the **Notification Mode** by choosing from the available options. The default is “Notify on Every Occurance.”

e. [Optional] Configure the **Limit Sending Times**. The default will be any time of day.

f. Click **[OK]** to commit the changes and close the Add Actions screen. The Alert Rules screen will populate with the action (see **Figure 27**).

Figure 27: The Alert Rules screen populated with the newly added Action.

10. Click **[Save]** to commit the changes.

B. Run a Broker Command on a License Manager Server

1. Open the Administration screen in OpenLM EasyAdmin (**[Start] > [Administration]**).

2. Click on the **Alerts Management** icon (see **Figure 28**). This will open the Alert Rules screen (see **Figure 29**).

Figure 28: Choose Alerts Management from the Administration screen.

Figure 29: The Alert Rules screen.

3. Click **[+ Add Rule]**. This opens the Add Alert Rule screen (see **Figure 30**).

Figure 30: The Add Alert Rule screen.

4. Enter a name for the rule in the **Rule Name** field.

5. Click the down arrow on the **Severity** drop list and choose a level of severity from the list of choices (Alert, Warning or Notice). When the required fields are populated, the **[Save]** button will become available.

6. *[Optional]* Configure the **Frequency**. The Default will be “Any time of day.” *[NOTE: Consult the Alert Management documentation for details on configuring Frequency.]*

7. Click the **[Save]** button. The rule will be stored and the Add Alert Rule screen will close. The Alert Rules screen will populate with the new rule and additional fields for configuration will become available (see **Figure 31**).

Figure 31: The Alert Rules screen populated with the added rule.

8. Set a **Condition**. To do this follow **Step #8.a** to **Step#8.e**.

a. Choose a **Type** from the Type drop list on the Conditions panel.

b. Click the **[+Add]** button to the right of the selected Type to open the Add Condition screen (See **Figure 32**).

Figure 32: The Add Condition screen for Feature Expiration Date.

[NOTE: The Condition panel on the Add Condition screen is context sensitive to the selection made for Type.]

c. Enter the required **Condition** parameters.

d. Enter the required **Filter** parameters. When all the required parameters are entered, the **[OK]** button will become available.

e. Click **[OK]** to save the condition and close the screen. A row will be added to the Conditions panel on the Alert Rules screen (see **Figure 33**).

Figure 33: A condition added to the Alert Rules screen.

9. Choose **Run a Broker Command on LM Server** from the Type drop list in the Actions panel. This will open the Add Action screen (see **Figure 34**).

Figure 34: The Add Action screen.

10. Click the down arrow on the **Command** drop list and choose **Custom**. When custom is selected Command Data and Servers fields become available (see **Figure 35**).

Figure 35: Add Actions with Command Data and Servers fields revealed.

11. Enter the command to be executed in the **Command Data** field.

12. Click the down arrow on the Servers drop list to reveal the configured servers.

13. Choose the server(s) from the drop list by clicking on a selection.

14. *[Optional]* Configure the **Notification Mode** by choosing from the available options.

15. *[Optional]* Configure the **Limit Sending Times**. The default will be any time of day.

16. Click **[OK]** to commit the changes and close the Add Actions screen. The Alert Rules screen will populate with the action (see **Figure 36**).

Figure 36: The Alert Rules screen populated with the newly added Action.

17. Click **[Save]** to commit the changes.

Creating Custom Commands with App Manager

Custom Commands provide customized scripting functionality for rules created in the App Manager module. Rules that allow or deny license consumption can trigger scripts that run as an application or as a service on the end user’s workstation.

Commands are managed from the App Manager module in OpenLM EasyAdmin. The commands are configured in the Action Editor while adding rules to the License Consumption Rules Table. This section of the documentation deals specifically with how rules are implemented as part of Custom Commands. Additional information about the specifics of App Manager configuration can be found in the dedicated OpenLM App Manager Configuration documentation.

1. Open the Administration screen in OpenLM EasyAdmin (**[Start] > [Administration]**).

2. Click on the App Manager icon in the Administration screen (see **Figure 37**). This will open the OpenLM App Manager administration screen (see **Figure 38**). The screen will display products configured in OpenLM Agent Configuration screen under the OpenLM App Manager tab (accessed via the **[Configure]** button).

Figure 37: The OpenLM App Manager icon, highlighted.

Figure 38: The OpenLM App Manager administration screen.

3. Click the **[License Consumption Rules Table]** button to show the existing rules in the License Consumption Rules Table screen (see **Figure 39**). The default rule are:

Rule Name = Everyone

Action = Deny

[NOTE: This configuration means that no one can run a process managed by App Manager as configured in the OpenLM App Manager tab on the OpenLM Agent configuration screen. Additional configuration is required to allow users to initiate actions.]

Figure 39: The License Consumption Rules Table screen.

4. Assign a Custom Command behavior. Adjusting control of behaviors. A few examples covered in **Step #4.a** to **Step #4.c** include editing an existing action, adding an action, and defining a condition.

   a. Edit the behavior of an existing action.

i. Double-click directly on the **Deny** label. This will open the action in the Action Editor screen with the current parameters for the Deny rule (see **Figure 40**).

Figure 40: The Action Editor screen populated with parameters for Deny.

ii. Enter a valid script into the **Script** field.

iii. Click **[Save]**. The script will now apply to all denied licenses configured in the App Manager.

b. Add a new action.

i. Click **[+New Action]** to open the Action Editor screen (see **Figure 41**).

Figure 41: The Action Editor screen.

ii. Click the arrow at the right of the **Name** field to reveal the drop list of choices for the action name.

iii. Select one of the choices from the drop list (**Allow**, **Deny**) by clicking on it in the menu. See descriptions for the options in **Table 4**.

Name	Action
Allow	The Action will be associated with successful grant of a license.
Deny	The Action will be associated with a license denial.

Table 4: Action name choices.

iv. Click the arrow at the right of the **Run At** to reveal the drop list of choices for run at options.

v. Select one of the choices from the drop list (**Application**, **Service**) by clicking on it in the menu. See descriptions for the options in **Table 5**.

Option	Description

Application	• The script runs in the Agent Application domain, which is intended for on-screen viewing and user interaction (e.g., show messages or display applications). • The script runs as the current user’s UserName. • These scripts cannot influence System-owned tasks.
Service	• The script run in the Agent Service domain, which is intended for silent procedures (e.g., manage the registry or configure security). • The script runs as the System user. • These scripts cannot influence UserName-owned tasks

Table 5: Run At choices.

vi. Enter a valid script in the **Script** field.

vii. Click **[Save]**. A new Deny column will appear on the right **License Consumption Rules Table** screen.

*[NOTE: It is possible to have multiple actions for a rule name as in cases where it is desirable to run both an application and a service. To delete undesired columns, double-click the label (e.g., Allow or Deny) and click the **[Delete]** button on the Edit Action screen.]*

.

Figure 42: New Deny column in the License Consumption Rules Table.

viii. Click the checkbox under **Deny** in the added column. The script will now apply to all denied licenses configured in the App Manager.

c. Add a condition for the rule.

i. Click on the **[+Add Condition]** button. This will open the Condition Editor screen.

Figure 43: The Condition Editor screen.

ii. Click the down arrow to the right of the **Argument** field. This will reveal a list of argument options (see **Figure 44**).

Figure 44: The choices in the Argument drop list.

iii. Click on an argument from the **Argument** drop list to populate the field. *[Note: **Vendor** was selected in this example.]*

iv. Click the down arrow to the right of the **Select an Operation** field. This will reveal a list of argument options (see **Figure 44**). The options are context sensitive to the selection made in the Argument field.

Figure 44: The choices in the Argument drop list.

v. Click on an option from the **Select an Operation** drop list to populate the field. *[Note: **Equals** was selected in this example.]*

vi. Enter a parameter in the **Compared Value** field. This should logically complete the condition. *[Note: **Adobe** was selected in this example.]*

vii. Click **[Save]**. A new Conditions column will appear on the **License Consumption Rules Table** screen.

Figure 45: New Deny column in the License Consumption Rules Table.

viii. Click the checkbox under **Conditions** in the added column (see **Figure 46**). The condition will now apply to all licenses configured in the App Manager which use Adobe as the vendor and will work in tandem with the active Allow actions as defined. If the condition is met, the action will execute (an allowed license will trigger the script). Messages will appear for success or failure results (see **Figure 47** and **Figure 48**).

Figure 46: A license consumption rule with multiple functions.

Figure 47: License denial Error screen.

Figure 48: The message returned by the Allow script.

By building consumption rules it is possible to handle complex licensing situations that enable creative solutions to help end users and customize license handling.

https://www.openlm.com/wp-content/uploads/2018/04/Configuring-Custom-Commands-Agent-App-Manager.mp4