

# 使用適用於 vRealize Automation 7.3 的 vRealize Orchestrator 外掛程式

vRealize Orchestrator 7.3  
vRealize Automation 7.3  
vRealize Automation 7.3

您可以在 VMware 網站上找到最新的技術說明文件，網址為：

<https://docs.vmware.com/tw/>

VMware 網站還提供了最新的產品更新。

如果您對此文件有何想法，請將您的回應意見提交至：

[docfeedback@vmware.com](mailto:docfeedback@vmware.com)

Copyright © 2012–2017 VMware, Inc. 版權所有。 [版權和商標資訊](#)。

**VMware, Inc.**  
3401 Hillview Ave.  
Palo Alto, CA 94304  
[www.vmware.com](http://www.vmware.com)

# 內容

使用適用於 vRealize Automation 的 vRealize Orchestrator 外掛程式	5
<b>1 適用於 vRealize Automation 的 VMware vRealize Orchestrator 外掛程式簡介</b>	<b>7</b>
vRealize Orchestrator 搭配 vRealize Automation 外掛程式的角色	7
<b>2 設定 vRealize Automation 外掛程式</b>	<b>9</b>
組態工作流程	9
新增 vRealize Automation 主機	10
新增 IaaS 主機	10
<b>3 使用 vRealize Automation 外掛程式工作流程</b>	<b>13</b>
移除作業限制	13
受限制的作業	14
使用 vRealize Automation 外掛程式詳細目錄	15
使用 vRealize Automation 外掛程式管理工作流程	15
使用 vRealize Automation 外掛程式基礎結構管理工作流程	21
建立 vRealize Automation IaaS 模型實體	22
讀取 vRealize Automation IaaS 模型實體	23
使用 vRealize Automation 外掛程式要求工作流程	23
使用 vRealize Automation 外掛程式範例工作流程	24
存取 vRealize Automation 外掛程式 API	24
<b>4 vRealize Automation 外掛程式指令碼範例</b>	<b>27</b>
CRUD 基礎結構管理工作指令碼範例	27
尋找 vRealize Automation 實體指令碼範例	31
取得 vRealize Automation 範例指令碼佈建的資源	32
一般工作指令碼範例	33
索引	37



# 使用適用於 vRealize Automation 的 vRealize Orchestrator 外掛程式

---

〈使用適用於 vRealize Automation 的 vRealize Orchestrator 外掛程式〉提供了有關設定和使用適用於 VMware vRealize Automation 之 VMware® vRealize Orchestrator 外掛程式的資訊和指示。

## 適合對象

〈使用適用於 vRealize Automation 的 vRealize Orchestrator 外掛程式〉中的資訊專門為熟悉虛擬機器技術、Orchestrator 工作流程開發和 VMware vRealize Automation 的有經驗使用者而撰寫。

如需有關 Orchestrator 的詳細資訊，請參閱

[http://www.vmware.com/support/pubs/orchestrator\\_pubs.html](http://www.vmware.com/support/pubs/orchestrator_pubs.html)。

如需有關 vRealize Automation 的詳細資訊，請參閱

<http://www.vmware.com/support/pubs/vrealize-automation.html>。

## VMware 技術出版品詞彙表

VMware 技術出版品提供您可能不熟悉的專有詞彙表。如需 VMware 技術說明文件中所用專有詞彙的定義，請前往 <http://www.vmware.com/support/pubs>。



# 適用於 vRealize Automation 的 VMware vRealize Orchestrator 外掛程式簡介

# 1

適用於 vRealize Automation 的 VMware vRealize Orchestrator 外掛程式可讓 vRealize Orchestrator 與 vRealize Automation 進行互動。

您可以使用 vRealize Automation 外掛程式建立和執行下列 vRealize Automation 功能的工作流程：

- XaaS 自訂資源和藍圖管理
- 目錄項目和資源管理與要求
- 權利組態
- 核准原則組態
- 工作項目互動
- vSphere 和 vCloud Director 虛擬機器佈建與佈建後動作
- vRealize Automation IaaS 模型中的建立、讀取、更新和刪除 (CRUD) 作業

## vRealize Orchestrator 搭配 vRealize Automation 外掛程式的角色

您使用 Orchestrator 用戶端執行及建立工作流程和存取外掛程式 API。您可以使用 vRealize Automation 安裝的內嵌式 vRealize Orchestrator 執行個體，也可以使用外部 vRealize Orchestrator 伺服器。

vRealize Orchestrator 增強 vRealize Automation 外掛程式。vRealize Orchestrator 是一個開發和程序自動化平台，提供管理 VMware 雲端堆疊和第三方技術的可延伸工作流程。

vRealize Orchestrator 允許透過開放式外掛程式架構整合管理及系統管理解決方案。



## 設定 vRealize Automation 外掛程式

您可以新增 vRealize Automation 主機和 IaaS 主機來設定外掛程式。

### 組態工作流程

您可以使用**組態**工作流程類別中的工作流程管理 vRealize Automation 主機。

#### vRealize Automation 主機

您可以從 Orchestrator 用戶端的工作流程視圖 (在外掛程式庫的**組態**子目錄中) 存取這些工作流程。

工作流程名稱	說明
新增 vRA 主機	將 vRealize Automation 主機新增至外掛程式詳細目錄。對於承租人管理和系統管理工作，您可以使用 <b>詳細目錄</b> 視圖對每個承租人執行工作流程。若要對承租人使用外掛程式的完整功能，請針對每個承租人建立一個專用 vRealize Automation 主機。
使用元件登錄新增 vRA 主機	使用每一使用者工作階段連線，將 vRealize Automation 主機新增至外掛程式詳細目錄。您必須使用 vRealize Automation 系統管理員的認證登入 Orchestrator 用戶端。 若要將此選項與外部 vRealize Orchestrator 伺服器搭配使用，您必須在 vRealize Automation 元件登錄中登錄 Orchestrator 伺服器。
新增 vRA 主機的 IaaS 主機	將所選 vRealize Automation 主機的 IaaS 主機新增至外掛程式詳細目錄。
移除 vRA 主機	從外掛程式詳細目錄移除 vRealize Automation 主機。
更新 vRA 主機	更新外掛程式詳細目錄中的 vRealize Automation 主機。
驗證 vRA 主機	驗證 vRealize Automation 主機及其連線。

**備註** 如果您的 vRealize Orchestrator 伺服器登錄於 vRealize Automation 元件登錄中，將自動新增名稱為 Default 的 vRealize Automation 主機。預設主機使用預設承租人的每一使用者工作階段連線。依預設，vRealize Automation 安裝的內嵌式 Orchestrator 伺服器登錄於 vRealize Automation 元件登錄中。

#### vRealize Automation IaaS 主機

您可以從 Orchestrator 用戶端的工作流程視圖 (在外掛程式庫的**基礎結構管理 > 組態**子目錄中) 存取這些工作流程。

依預設，vRealize Automation 安裝的內嵌式 vRealize Orchestrator 伺服器登錄於 vRealize Automation 元件登錄中。

工作流程名稱	說明
新增 IaaS 主機	將 vRealize Automation IaaS 主機新增至外掛程式詳細目錄。這個工作流程在功能上與 vRA 主機的新增 IaaS 主機相同，但是不需要 vRealize Automation 主機。
移除 IaaS 主機	從外掛程式詳細目錄移除 vRealize Automation IaaS 主機。
更新 IaaS 主機	更新外掛程式詳細目錄中的 vRealize Automation IaaS 主機。
驗證 IaaS 主機	驗證 vRealize Automation IaaS 主機及其連線。

## 新增 vRealize Automation 主機

您可以執行工作流程來新增 vRealize Automation 主機，並設定主機連線參數。

### 程序

- 1 從 Orchestrator 用戶端的下拉式功能表中選取**執行或設計**。
- 2 按一下**工作流程**視圖。
- 3 展開**程式庫 > vRealize Automation > 組態**。
- 4 在**新增 vRA 主機**工作流程上按一下滑鼠右鍵，然後選取**啟動工作流程**。
- 5 在**主機名稱**文字方塊中輸入主機的唯一名稱。
- 6 在**主機 URL**文字方塊中輸入主機的 URL 地址。  
例如：*https://hostname*。
- 7 在**承租人**文字方塊中輸入承租人的名稱。  
若要對承租人使用外掛程式的完整功能，請針對每個承租人建立一個專用 vRealize Automation 主機。
- 8 選取是否要在未經使用者確認的情況下自動安裝 SSL 憑證。
- 9 (選擇性) 若要設定 vRealize Orchestrator 等待連線或來自 vRealize Automation 的回應的時間長度，請在**連線逾時 (秒)**和**作業逾時 (秒)**文字方塊中輸入逾時間隔。
- 10 從**工作階段模式**下拉式功能表中選取主機的連線類型。

選項	動作
<b>共用工作階段</b>	在 <b>驗證使用者名稱</b> 和 <b>驗證密碼</b> 文字方塊中，輸入 vRealize Automation 使用者的認證。
<b>每一使用者工作階段</b>	使用目前已登入的使用者的認證進行連線。您必須使用 vRealize Automation 系統管理員的認證登入 Orchestrator 用戶端。 若要將此選項與外部 vRealize Orchestrator 伺服器搭配使用，您必須在 vRealize Automation 元件登錄中登錄 Orchestrator 伺服器。

- 11 按一下**提交**。

### 下一個

新增 vRealize Automation 基礎結構管理主機。

## 新增 IaaS 主機

您可以執行工作流程來新增 vRealize Automation 主機的 IaaS 主機，並設定連線參數。

### 程序

- 1 從 Orchestrator 用戶端的下拉式功能表中選取**執行或設計**。

- 2 按一下**工作流程**視圖。
- 3 展開**程式庫 > vRealize Automation > 基礎結構管理 > 組態**。
- 4 在**新增 IaaS 主機**上按一下滑鼠右鍵，然後選取**啟動工作流程**。
- 5 從 **vCAC 主機**下拉式功能表中選取您要為其設定 IaaS 主機的 vRealize Automation 主機。
- 6 在**主機名稱**文字方塊中輸入主機的唯一名稱。
- 7 輸入 Model Manager 安裝所在機器的 URL。  
例如：`https://model_manager_machine.com`。
- 8 若要安裝 SSL 憑證，請選取**是**。
- 9 若要使用 Proxy 存取 Model Manager 機器，請選取**是**。  
如果您選取此選項，則必須在以下頁面上提供 Proxy 主機和 Proxy 連接埠。
- 10 按**下一步**。
- 11 如果您要設定明確的 Proxy，請提供 Proxy 主機 URL 和連接埠。
- 12 按**下一步**。
- 13 若要設定自己的逾時值，請按一下**否**。
- 14 (選擇性) 若要設定 vRealize Orchestrator 等待連線或來自 vRealize Automation 的回應的時間長度，請在**連線逾時 (秒)**和**作業逾時 (秒)**文字方塊中輸入逾時間隔。
- 15 按**下一步**。
- 16 選取主機的驗證類型。

選項	說明
<b>SSO</b>	選取此選項可使用 vCenter Single Sign-On。
<b>NTLM</b>	僅在 Active Directory 基礎結構依賴於 NTLM 驗證時，選取此選項以啟用 NT LAN Manager (NTLM) 通訊協定式驗證。 如果選取此選項，您必須選取其他 NTLM 認證和驗證選項。

- 17 如果您已選取 NTLM，請按**下一步**並輸入工作站機器的名稱和 NetBIOS 網域名稱。
- 18 按一下**提交**。



# 使用 vRealize Automation 外掛程式工作流程

# 3

vRealize Automation 外掛程式工作流程詳細目錄包含您可於一般工作 (如與目錄互動、管理基礎結構與建立承租人和服務) 中使用的工作流程。

您可以使用自訂 HTTP 標頭，如 vRealize Automation 特定標頭工作與身分識別，並將其套用至 CRUD、佈建與佈建後工作流程。

本章節討論下列主題：

- “[移除作業限制](#),” 第 13 頁
- “[使用 vRealize Automation 外掛程式詳細目錄](#),” 第 15 頁
- “[使用 vRealize Automation 外掛程式管理工作流程](#),” 第 15 頁
- “[使用 vRealize Automation 外掛程式基礎結構管理工作流程](#),” 第 21 頁
- “[使用 vRealize Automation 外掛程式要求工作流程](#),” 第 23 頁
- “[使用 vRealize Automation 外掛程式範例工作流程](#),” 第 24 頁
- “[存取 vRealize Automation 外掛程式 API](#),” 第 24 頁

## 移除作業限制

部分建立、讀取、更新和刪除作業僅限 7.0 版以後的版本才能進行。如果您是在較早的版本中使用工作流程的作業，這些作業無法在 7.0 版及更新版本中使用。您可以將工作流程更新為受支援的作業，也可以重新啟用您需要的作業。

若要重新啟用作業，必須從 `operations.properties` 檔案中移除要啟用的作業。如需檔案中的作業清單，請參閱“[受限制的作業](#),” 第 14 頁。

### 程序

- 1 從 vRealize Orchestrator 的下拉式功能表中選取**設計**。
- 2 按一下**資源**視圖。
- 3 在資源階層中，展開**程式庫 > VCAC > Util**。
- 4 建立備份，並修改 `operations.properties` 檔案。
  - a 在 `operations.properties` 上按一下滑鼠右鍵，並選取**儲存至檔案**。
  - b 將複本另存為備份。
  - c 建立新的複本，並刪除要重新啟用的作業。
  - d 儲存新的檔案。

- 5 取代 vRealize Orchestrator 中既有的檔案。
  - a 在 vRealize Orchestrator 的 **Util** 資料夾上按一下滑鼠右鍵，並且按一下**匯入資源**。
  - b 瀏覽至新版的 `operations.properties` 檔案，並且按一下**開啟**。
  - c 按一下**取代一次**儲存修改的版本。
- 6 重新啟動 vRealize Orchestrator 伺服器。
- 7 選取 `operations.properties` 檔案，並且按一下**檢視器**索引標籤。
- 8 確認您啟用的作業不再出現在檔案中。

您從檔案中移除的作業此時即可在舊的工作流程中使用。

## 下一個

建立新的工作流程時，避免使用受限制的作業。

## 受限制的作業

`Operations.properties` 檔的內容包含受限制的作業。若要重新啟用作業，必須從檔案移除作業。

下列文字是 `operations.properties` 檔的預設版本。若要重新啟用作業，請參閱“[移除作業限制](#)”第 13 頁。

```
#Blueprints
operation.create=ManagementModelEntities.svc@VirtualMachineTemplates
operation.update=ManagementModelEntities.svc@VirtualMachineTemplates
operation.delete=ManagementModelEntities.svc@VirtualMachineTemplates
#Blueprint properties
operation.create=ManagementModelEntities.svc@VirtualMachineProperties
operation.read=ManagementModelEntities.svc@VirtualMachineProperties
operation.update=ManagementModelEntities.svc@VirtualMachineProperties
operation.delete=ManagementModelEntities.svc@VirtualMachineProperties
#Global profiles
operation.create=ManagementModelEntities.svc@GlobalProfiles
operation.read=ManagementModelEntities.svc@GlobalProfiles
operation.update=ManagementModelEntities.svc@GlobalProfiles
operation.delete=ManagementModelEntities.svc@GlobalProfiles
#Global profile properties
operation.create=ManagementModelEntities.svc@GlobalProfileProperties
operation.read=ManagementModelEntities.svc@GlobalProfileProperties
operation.update=ManagementModelEntities.svc@GlobalProfileProperties
operation.delete=ManagementModelEntities.svc@GlobalProfileProperties
#PropertySetXml
operation.create=ManagementModelEntities.svc@PropertySetXml
operation.read=ManagementModelEntities.svc@PropertySetXml
operation.update=ManagementModelEntities.svc@PropertySetXml
operation.delete=ManagementModelEntities.svc@PropertySetXml
#Property definitions
operation.create=ManagementModelEntities.svc@PropertyDefinitions
operation.read=ManagementModelEntities.svc@PropertyDefinitions
operation.update=ManagementModelEntities.svc@PropertyDefinitions
operation.delete=ManagementModelEntities.svc@PropertyDefinitions
#Property attributes
operation.create=ManagementModelEntities.svc@PropertyAttributes
operation.read=ManagementModelEntities.svc@PropertyAttributes
operation.update=ManagementModelEntities.svc@PropertyAttributes
```

```

operation.delete=ManagementModelEntities.svc@PropertyAttributes
#Property Attribute Types
operation.create=ManagementModelEntities.svc@PropertyAttributeTypes
operation.read=ManagementModelEntities.svc@PropertyAttributeTypes
operation.update=ManagementModelEntities.svc@PropertyAttributeTypes
operation.delete=ManagementModelEntities.svc@PropertyAttributeTypes
#Control layouts
operation.create=ManagementModelEntities.svc@ControlLayouts
operation.read=ManagementModelEntities.svc@ControlLayouts
operation.update=ManagementModelEntities.svc@ControlLayouts
operation.delete=ManagementModelEntities.svc@ControlLayouts
#Amazon Virtual Machine Templates
operation.create=AmazonWSModelEntities.svc@AmazonVirtualMachineTemplates
operation.read=AmazonWSModelEntities.svc@AmazonVirtualMachineTemplates
operation.update=AmazonWSModelEntities.svc@AmazonVirtualMachineTemplates
operation.delete=AmazonWSModelEntities.svc@AmazonVirtualMachineTemplates
#Openstack Virtual Machine Templates
operation.create=OpenStackModelEntities.svc@OpenstackVirtualMachineTemplates
operation.read=OpenStackModelEntities.svc@OpenstackVirtualMachineTemplates
operation.update=OpenStackModelEntities.svc@OpenstackVirtualMachineTemplates
operation.delete=OpenStackModelEntities.svc@OpenstackVirtualMachineTemplates
#Endpoint credentials
operation.create=ManagementModelEntities.svc@ConnectionCredentials
operation.update=ManagementModelEntities.svc@ConnectionCredentials
operation.delete=ManagementModelEntities.svc@ConnectionCredentials
#Management endpoints
operation.create=ManagementModelEntities.svc@ManagementEndpoints
operation.update=ManagementModelEntities.svc@ManagementEndpoints
operation.delete=ManagementModelEntities.svc@ManagementEndpoints
#Management endpoint properties
operation.create=ManagementModelEntities.svc@ManagementEndpointProperties
operation.read=ManagementModelEntities.svc@ManagementEndpointProperties
operation.update=ManagementModelEntities.svc@ManagementEndpointProperties
operation.delete=ManagementModelEntities.svc@ManagementEndpointProperties

```

## 使用 vRealize Automation 外掛程式詳細目錄

您可使用**詳細目錄**視圖在 vRealize Automation 物件上執行工作流程。

若要顯示可供詳細目錄物件使用的工作流程，請導覽至**工具 > 使用者喜好設定 > 詳細目錄**，然後選取在**詳細目錄**中使用**關聯式功能表**核取方塊。啟用此選項後，當您以滑鼠右鍵按一下 Orchestrator 詳細目錄中的物件時，會顯示此物件的所有可用工作流程。

## 使用 vRealize Automation 外掛程式管理工作流程

您可以使用管理工作流程來管理 vRealize Automation 服務、承租人、核准原則、權利、業務群組、目錄項目，以及進階服務元件。

某些工作流程包括 vRealize Automation 主機的輸入參數 vCACCAFE:VCACHost。您設定 vRealize Automation 主機連線的方式會決定使用者執行工作流程時角色的套用方式。

- 若您將連線設定為共用工作階段，則共用工作階段的使用者帳戶必須擁有執行工作流程所需的角色。

- 若您將連線設定為每一使用者工作階段，則每位執行工作流程的使用者都必須如同他們在 vRealize Automation 使用者介面中一樣，擁有所需要的角色。

您可以在 vRealize Orchestrator 用戶端的工作流程視圖內找到這些工作流程，路徑為**程式庫 > vRealize Automation > 管理**子目錄。

您可以在**核准原則**子目錄中使用工作流程來建立並管理核准原則。

**表格 3-1. 核准原則**

工作流程	說明
啟用核准原則	啟用核准原則。啟用核准原則後，該原則變為唯讀。
新增核准層級	在某項核准中新增永遠需要的核准層級。您必須選取特定使用者與群組作為核准者。
複製核准原則	複製核准原則。
建立核准原則	建立不具備層級或核准者的草稿核准原則。若要針對您的原則建立核准層級並指定核准者，請執行 [新增核准層級] 工作流程。
停用核准原則	停用核准原則。您也可以刪除所有與核准原則相關聯的現有權利。
刪除核准原則	刪除仍在草稿狀態的核准原則。作用中的核准原則為唯讀。

您可以在**業務群組**子目錄中使用工作流程，以建立並管理業務群組和業務群組自訂內容。

**表格 3-2. 業務群組**

工作流程	說明
新增自訂內容	新增自訂內容至業務群組。
建立業務群組	建立業務群組。
刪除業務群組	刪除業務群組。
刪除自訂內容	從業務群組中移除自訂內容。
更新業務群組	更新業務群組詳細資訊，如預設機器前置詞、Active Directory 容器與使用者角色。
更新自訂內容	更新業務群組內的自訂內容。

管理子目錄中包括一個搭配 vRealize Automation 7.0 執行的**業務群組 (已過時)**子目錄。在主資料夾中，請使用名稱相同的工作流程。

您可以在**目錄項目**子目錄中使用該工作流程來管理目錄項目。

**表格 3-3. 目錄項目**

工作流程	說明
啟用目錄項目	啟用目錄項目。您必須啟用某個目錄項目，並在使用者可以要求該項目前將其指派至某個服務。
指派目錄項目至服務	指派目錄項目至服務。您必須啟用某個目錄項目，並在使用者可以要求該項目前將其指派至某個服務。
停用目錄項目	停用目錄項目，並將其從服務目錄中移除，讓使用者無法要求。

您可以在**複合藍圖**子目錄中使用工作流程來管理於設計畫布中建立的複合藍圖。

**表格 3-4. 複合藍圖**

工作流程	說明
刪除複合藍圖	從設計藍圖清單中刪除已解除發佈的藍圖。
匯入複合藍圖	從 YAML 檔匯入複合藍圖。
發佈複合藍圖	發佈仍處於草稿狀態的複合藍圖。
解除發佈複合藍圖	解除發佈已發佈的複合藍圖。

內容子目錄工作流程已過時。請使用 Cloud Client 來執行匯入與匯出動作。您可在 <https://developercenter.vmware.com/tool/cloudclient> 取得 Cloud Client 下載與說明文件。

**表格 3-5. 內容**

工作流程	說明
匯出內容 (已過時)	請使用 Cloud Client 來執行匯入與匯出動作。您可在 <a href="https://developercenter.vmware.com/tool/cloudclient">https://developercenter.vmware.com/tool/cloudclient</a> 取得 Cloud Client 下載與說明文件。
匯入內容 (已過時)	請使用 Cloud Client 來執行匯入與匯出動作。您可在 <a href="https://developercenter.vmware.com/tool/cloudclient">https://developercenter.vmware.com/tool/cloudclient</a> 取得 Cloud Client 下載與說明文件。
傳輸內容 (已過時)	請使用 Cloud Client 來執行匯入與匯出動作。您可在 <a href="https://developercenter.vmware.com/tool/cloudclient">https://developercenter.vmware.com/tool/cloudclient</a> 取得 Cloud Client 下載與說明文件。
驗證內容 (已過時)	請使用 Cloud Client 來執行匯入與匯出動作。您可在 <a href="https://developercenter.vmware.com/tool/cloudclient">https://developercenter.vmware.com/tool/cloudclient</a> 取得 Cloud Client 下載與說明文件。

您可以在**權利**子目錄中使用工作流程來建立並管理權利。

**表格 3-6. 權利**

工作流程	說明
啟用權利	啟用權利。
指派目錄項目至權利	指派一或多個目錄項目至權利。您也可以使用此工作流程指派核准原則。
指派立即動作至權利	指派一或多個立即動作至權利。立即動作不會建立申請。
指派資源動作至權利	指派一或多個資源動作至權利。您也可以使用此工作流程指派核准原則。
指派服務至權利	指派一或多個服務至權利。您也可以使用此工作流程指派核准原則。
指派使用者與群組至權利	指派一或多個使用者與群組至權利。
建立權利 (已過時)	建立權利。使用建立轉租人權利。
建立轉租人權利	建立權利。
停用權利	停用權利。
從權利取消指派使用者與群組	從權利使用者清單中移除使用者與群組。

您可以在**內容**子目錄中使用工作流程來管理內容定義與內容群組。為避免與 vRealize Automation 內容相衝突，請針對所有自訂內容名稱使用前置詞，例如公司名稱或功能名稱後接圓點。

**表格 3-7. 內容定義**

工作流程	說明
建立內容定義	建立自訂內容。
刪除內容定義	刪除自訂內容。

內容群組是內容定義的集合。

**表格 3-8. 內容群組**

工作流程	說明
新增內容至群組	新增已定義的自訂內容至群組。
建立內容群組	建立您可新增已定義自訂內容的內容群組。
刪除內容群組	刪除內容群組。
從群組移除內容	從內容群組中移除已定義的自訂內容。
更新內容群組	修改內容群組的名稱或描述。
更新群組內的內容	修改內容群組中內容的名稱、數值與行為。

您可以在**服務**子目錄中使用工作流程來管理服務。

**表格 3-9. 服務**

工作流程	說明
啟用服務	啟用服務。
指派目錄項目至服務	指派一或多個目錄項目至服務。
複製服務	複製服務。
建立服務	建立服務。
停用服務	停用服務。
刪除服務	刪除服務。

您可以在**承租人**子目錄中使用工作流程來建立並管理承租人。

身分識別存放區工作流程已過時。取代工作流程會搭配 vRealize Automation 針對 Directories Management API 所作的變更來執行。

**表格 3-10. 承租人**

工作流程	說明
新增管理員	新增一或多位承租人管理員與基礎結構管理員至承租人。
將身分識別存放區新增至承租人。	將身分識別存放區新增至 vRealize Automation 主機的管理員。若您不是設定承租人的系統管理員，就不能執行這個工作流程。
將身分識別存放區新增至承租人 (已過時)	使用 [將身分識別存放區新增至承租人] 工作流程。
將身分識別存放區新增至 vCAC 主機	將身分識別存放區新增至設定為 vRealize Automation 主機的管理員。若您不是為承租人設定身分識別存放區的管理員，就不能執行這個工作流程。
將身分識別存放區新增至 vCAC 主機 (已過時)	使用 [將身分識別存放區新增至 vCAC 主機] 工作流程。
建立承租人	建立承租人。您必須選取連同系統管理員認證新增的 vRealize Automation 主機。
從承租人刪除身分識別存放區	從 vRealize Automation 主機的管理員刪除身分識別存放區。若您不是設定承租人的系統管理員，就不能執行這個工作流程。

表格 3-10. 承租人 (繼續)

工作流程	說明
從 vCAC 主機刪除身分識別存放區	從設定為 vRealize Automation 主機的承租人刪除身分識別存放區。若您不是為承租人設定身分識別存放區的承租人管理員，就不能執行這個工作流程。
刪除承租人	刪除承租人。
移除管理員	從承租人移除一或多位承租人管理員與基礎結構管理員。
更新承租人身分識別存放區	更新 vRealize Automation 主機承租人的現有身分識別存放區。若您不是設定承租人的系統管理員，就不能執行這個工作流程。
更新承租人身分識別存放區 (已過時)	使用 [更新承租人身分識別存放區] 工作流程。
更新 vCAC 主機身分識別存放區	更新設定為 vRealize Automation 主機之承租人的身分識別存放區。若您不是為承租人設定身分識別存放區的承租人管理員，就不能執行這個工作流程。
更新 vCAC 主機身分識別存放區 (已過時)	使用 [更新 vCAC 主機身分識別存放區] 工作流程。
更新承租人	更新現有承租人的名稱、描述與聯絡電子郵件地址。

您可以在[工作流程訂閱](#)子目錄中使用工作流程來管理事件工作流程訂閱。

表格 3-11. 工作流程訂閱

工作流程	說明
刪除工作流程訂閱	刪除解除發佈的工作流程訂閱。此工作流程可套用至系統與承租人工作流程訂閱。
匯出系統工作流程訂閱	匯出系統工作流程訂閱，並將其儲存為 JSON 格式的 vRealize Orchestrator 資源元素。 系統工作流程訂閱是回應系統事件和所有承租人中的事件的專用工作流程訂閱。
匯出承租人工作流程訂閱	匯出承租人工作流程訂閱，並將其儲存為 JSON 格式的資源元素。 執行承租人特定工作流程的專門工作流程訂閱。
匯入系統工作流程訂閱	從 JSON 檔案匯入系統工作流程訂閱。系統工作流程訂閱會在發生系統事件時觸發，且可跨承租人。
匯入承租人工作流程訂閱	從 JSON 檔案匯入已匯出的工作流程訂閱。這些工作流程訂閱均為承租人特定。
發佈工作流程訂閱	發佈處於草稿或解除發佈狀態的工作流程訂閱。此工作流程可套用至系統與承租人工作流程訂閱。
登錄系統工作流程訂閱	建立系統工作流程訂閱，包括逾時與優先順序數值。
登錄承租人工作流程訂閱	建立承租人特定工作流程訂閱，包括逾時與優先順序數值。
解除發佈工作流程訂閱	解除發佈已發佈的工作流程訂閱。此工作流程可套用至系統與承租人工作流程訂閱。
更新工作流程訂閱	變更名稱、描述、vRealize Orchestrator 工作流程、訂閱條件、逾時數值、狀態數值與優先順序數值。您不可更新事件主題或封鎖狀態。

您可以在[XaaS 自訂資源](#)子目錄中使用工作流程來建立和刪除 XaaS 自訂資源。

**表格 3-12. XaaS 自訂資源**

工作流程	說明
建立自訂資源	建立自訂資源。
刪除自訂資源	移除自訂資源。

您可以在 **XaaS 資源動作** 子目錄中使用工作流程來建立和管理 XaaS 資源動作。

**表格 3-13. XaaS 資源動作**

工作流程	說明
複製資源動作	建立現有資源動作的複本。
建立資源動作	建立資源動作。
刪除資源動作	刪除資源動作。
發佈資源動作	發佈資源動作。
解除發佈資源動作	解除發佈資源動作。

您可以在 **XaaS 資源對應** 子目錄中使用工作流程來建立和管理 XaaS 對應至非 XaaS 資源。

**表格 3-14. XaaS 資源對應**

工作流程	說明
建立資源對應	將目錄資源類型對應至 vRealize Orchestrator 類型。
刪除資源對應	刪除資源對應。
設定目標準則	指定會決定資源對應可用性的條件。

您可以在 **XaaS 伺服器組態** 子目錄中使用工作流程來管理目標 Orchestrator 執行個體。

**表格 3-15. XaaS 伺服器組態**

工作流程	說明
更新 Orchestrator 伺服器組態	修改伺服器設定，包括連接埠、主機、使用者名稱與密碼。
驗證 Orchestrator 伺服器組態	驗證 vRealize Orchestrator 設定是否有效。如果組態有效，工作流程會傳回數值 TRUE；組態無效則傳回 FALSE。

您可以在 **XaaS 服務藍圖** 子目錄中使用工作流程來建立和管理 XaaS 藍圖。

**表格 3-16. XaaS 藍圖**

工作流程	說明
複製服務藍圖	建立服務藍圖複本。
建立服務藍圖	建立服務藍圖。
刪除服務藍圖	刪除服務藍圖。
發佈服務藍圖	發佈服務藍圖。
解除發佈服務藍圖	解除發佈服務藍圖。

## 使用 vRealize Automation 外掛程式基礎結構管理工作流程

您可以使用基礎結構管理工作流程來執行基本作業。您能使用可延伸性套件 (或使用自訂作業功能表) 來自訂 vRealize Automation 使其具備呼叫 vRealize Orchestrator 工作流程的能力，以作為佈建程序的一部分。

您可以在 Orchestrator 用戶端的工作流程視圖 (在外掛程式庫的基礎結構管理子目錄中) 找到 [基礎結構管理] 工作流程。

您可以使用 [基礎結構管理] 工作流程來佈建虛擬機器，並執行基本的建立、讀取、更新或刪除作業。

**表格 3-17. 基礎結構管理**

工作流程名稱	說明
等待虛擬機器狀態變更	<p>等待一系列虛擬機器的狀態變更。如果所有虛擬機器都處於成功狀態，則會呼叫觸發器，且工作流程會成功結束。如果任一部虛擬機器進入失敗狀態，或已經不存在，則工作流程失敗。您必須從以下選取的選項中輸入成功與失敗狀態。</p> <ul style="list-style-type: none"> <li>■ Requested</li> <li>■ AwaitingApproval</li> <li>■ RegisterMachine</li> <li>■ BuildingMachine</li> <li>■ AddingDisks</li> <li>■ MachineProvisioned</li> <li>■ MachineActivated</li> <li>■ InstallTools (僅 VMware)</li> <li>■ On</li> <li>■ Off</li> <li>■ TurningOn</li> <li>■ TurningOff</li> <li>■ ShuttingDown</li> <li>■ Suspending</li> <li>■ Resetting</li> <li>■ Rebooting</li> <li>■ Expired</li> <li>■ DeactivateMachine</li> <li>■ UnprovisionMachine</li> <li>■ Disposing</li> <li>■ Finalized</li> </ul>
建立 IaaS 模型實體	針對指定的 vRealize Automation 模型建立並保留實體。
刪除 IaaS 模型實體	刪除指定的 vRealize Automation 模型實體。
叫用佈建後動作 (已過時)	使用 [要求資源動作] 工作流程。
經由藍圖佈建虛擬機器 (於 vRealize Automation 7.0 中移除)	由 [要求目錄項目] 或 [以佈建要求要求目錄項目] 取代。
透過自訂篩選器讀取 IaaS 實體	使用自訂篩選器讀取 vRealize Automation 實體清單。若您並未指定篩選器，則會回傳所有實體。
透過系統查詢讀取 IaaS 實體	使用 OData 系統篩選器讀取 vRealize Automation 實體清單。將系統篩選器套用於 OData URI 慣例。
讀取 IaaS 模型實體	經由模型實體識別碼讀取 vRealize Automation 模型實體。
更新 IaaS 模型實體	經由模型實體識別碼更新 vRealize Automation 模型實體。

您能使用可延伸性子目錄中的工作流程 (或透過自訂作業功能表) 來自訂 vRealize Automation 使其具備呼叫 vRealize Orchestrator 工作流程的能力，以作為佈建程序的一部分。

子目錄同時也包括管理 IaaS 認證、端點、企業群組、機器前置詞與其他實體的工作流程。

**表格 3-18. 可延伸性**

工作流程名稱	說明
安裝 vCO 自訂項目	安裝 Orchestrator 自訂項目，包括自訂狀態變更工作流程與功能表作業工作流程。
解除安裝 vCO 自訂項目	解除安裝 Orchestrator 自訂項目，包括自訂狀態變更工作流程與功能表作業工作流程。
變更 IaaS 虛擬機器的保留區	變更受管理的虛擬機器屬性，例如保留區和業務群組。
匯入 IaaS 虛擬機器 (已過時)	使用 Cloud Client。您可在 <a href="https://developercenter.vmware.com/tool/cloudclient">https://developercenter.vmware.com/tool/cloudclient</a> 取得 Cloud Client 下載與說明文件。
匯入 vCenter 虛擬機器 (已過時)	使用 Cloud Client。您可在 <a href="https://developercenter.vmware.com/tool/cloudclient">https://developercenter.vmware.com/tool/cloudclient</a> 取得 Cloud Client 下載與說明文件。
解除登錄虛擬機器 (於 vRealize Automation 7.0 中移除)	不提供取代工作流程。
將功能表作業指派至藍圖與其虛擬機器 (已過時)	在虛擬機器上新增或更新功能表作業。 替代的未過時工作流程包括 [指派資源動作至權利] 和 [匯入複合藍圖]。
將功能表作業指派至虛擬機器 (已過時)	經由模型實體識別碼更新 vRealize Automation 模型實體。 替代的未過時工作流程包括 [指派資源動作至權利] 和 [匯入複合藍圖]。
指派狀態變更工作流程至藍圖與其虛擬機器 (已過時)	由 vRealize Automation 中的事件代理訂閱取代。
自訂功能表作業 (於 vRealize Automation 7.0 中移除)	不提供取代工作流程。
從藍圖與其虛擬機器移除功能表作業 (於 vRealize Automation 7.0 中移除)	不提供取代工作流程。
從藍圖與其虛擬機器移除狀態變更工作流程	從藍圖與其虛擬機器移除狀態變更工作流程。

## 建立 vRealize Automation IaaS 模型實體

您可以執行工作流程，建立簡單或複雜 vRealize Automation IaaS 實體，例如使用者的虛擬機器參考。

### 程序

- 1 從 Orchestrator 用戶端的下拉式功能表中選取**執行或設計**。
- 2 按一下**工作流程**視圖。
- 3 展開**程式庫 > vRealize Automation > 基礎結構管理**。
- 4 在**建立 IaaS 模型實體**工作流程上按一下滑鼠右鍵，並選取**啟動工作流程**。
- 5 選取 vRealize Automation 主機。
- 6 在**機型名稱**文字方塊中輸入機型的名稱。
- 7 在**實體集名稱**文字方塊中輸入實體集的名稱。

您可以使用指令碼或 REST API 設定簡單內容、複雜內容的連結、以及 HTTP 標頭內容。

- 8 按一下**提交**執行工作流程。

## 讀取 vRealize Automation IaaS 模型實體

您可以執行工作流程以讀取 vRealize Automation IaaS 模型實體。

### 程序

- 1 從 Orchestrator 用戶端的下拉式功能表中選取**執行或設計**。
- 2 按一下**工作流程**視圖。
- 3 展開**程式庫 > vRealize Automation > 基礎結構管理**。
- 4 在**讀取 IaaS 模型實體**上按一下滑鼠右鍵，並選取**啟動工作流程**。
- 5 選取 vRealize Automation 主機。
- 6 在**機型名稱**文字方塊中輸入機型的名稱。
- 7 在**實體集名稱**文字方塊中輸入實體集的名稱。  
您可以使用指令碼或 REST API 設定 HTTP 標頭內容。
- 8 按一下**提交**執行工作流程。

## 使用 vRealize Automation 外掛程式要求工作流程

您可以使用要求工作流程來要求目錄項目與資源動作，並藉此完成或取消工作項目。

工作項目要求使用者輸入或動作，例如工作流程互動、核准動作或回應某個回收要求。

您可以從 vRealize Orchestrator 用戶端的**工作流程**視圖 (在外掛程式庫詳細目錄的**要求**子目錄中) 存取這些工作流程。

工作流程	說明
取消工作項目	取消作用中工作項目。若您不是系統管理員，即無法使用這個工作流程。
完成工作項目	根據所提供的使用者輸入來完成工作項目。
要求目錄項目	針對執行工作流程的使用者要求目錄項目。 若您需要工作流程來要求複合藍圖，請使用 [以佈建要求要求目錄] 工作流程。
代表使用者要求目錄項目	代表使用者傳送目錄項目要求此工作流程只能用於您以及您代表其寄送要求之使用者同時有權存取的目錄項目。
以佈建要求要求目錄	針對執行工作流程的使用者要求複合藍圖作為目錄項目。若您為要求提供自訂輸入，您必須自訂工作流程。針對複合藍圖使用此工作流程。
要求資源動作	針對執行工作流程之使用者所擁有的目錄項目要求資源動作。
代表使用者要求資源動作	代表使用者傳送資源動作要求。此工作流程只能用於您以及您代表其寄送要求之使用者同時有權存取的資源動作。

工作流程	說明
以要求範本要求資源動作	<p>要求包括複雜參數的資源動作。最佳做法是複製工作流程，然後針對動作自訂工作流程。您可以使用工作流程來傳遞複雜參數，或隱藏您不希望出現在要求表單上的參數。此一工作流程的其中一項主要應用是自訂 IaaS 重新設定虛擬機器動作。</p> <p>若要在虛擬機器上建立重新設定作業，您必須先建立工作流程的複本，然後修改指令碼。設定 vRealize Orchestrator 中出現的參數，然後設定 <code>Cafe.Shim.VirtualMachine.Reconfigure.Requestor</code> 參數。此參數用於記錄，且不得為空白。請參閱下列範例。</p> <pre>var requestTemplate = vCACCAFERequestsHelper.getRequestForResourceAction(operation)  var jsonData = vCACCAFERequestsHelper.getResourceActionRequestData(requestTemplate); var json = JSON.parse(jsonData); //Change cpu example json.cpu = 2;  //This is a property needed for the Reconfigure IaaS operation: json["Cafe.Shim.VirtualMachine.Reconfigure.Requestor"] = 1; vCACCAFERequestsHelper.setResourceActionRequestData(requestTemplate, JSON.stringify(json));  request = System.getModule("com.vmware.library.vcaccafe.request").requestResourceAction WithRequestTemplate(operation, requestTemplate);</pre>
等待目錄項目要求	等待目錄項目要求完成。
等待資源動作要求	等待資源動作要求完成。
等待工作項目	等待工作項目完成。

## 使用 vRealize Automation 外掛程式範例工作流程

您可使用範例工作流程作為範例，或作為建立自訂工作流程的起點。

您可以在 vRealize Orchestrator 用戶端的工作流程視圖 (在外掛程式庫詳細目錄的範例子目錄中) 找到這些工作流程。

工作流程名稱	說明
建立權限	提供能與授權用戶端互動的範例指令碼，以及可在 vRealize Automation 內建立權限的權限服務。
建立承租人	建立具備相同 vRealize Automation 主機與 Active Directory 組態的承租人作為預設承租人。若要執行此工作流程，請選取透過您的系統管理員認證新增的 vRealize Automation 主機。執行工作流程前，您可以變更 Active Directory 設定。
列出目錄項目	針對選取的承租人傳回目錄項目清單。
將目錄項目佈建要求列印為 JSON	針對某個目錄項目擷取預設要求表單，並以 JSON 格式將其新增至主控台紀錄。您可以使用資料來自訂佈建要求。您可以使用此資訊來修改以佈建要求要求目錄項目工作流程。

## 存取 vRealize Automation 外掛程式 API

Orchestrator 提供 API Explorer，以便您搜尋 vRealize Automation 外掛程式 API，並參閱您可在命令元素中使用的 JavaScript 物件有關的說明文件。

如需更新的 vRealize Automation API 說明文件，請參閱 <https://www.vmware.com/support/pubs/vcac-pubs.html>。

### 程序

- 1 以管理員身分登入 Orchestrator 用戶端。

- 2 選取工具 > API Explorer。
- 3 按兩下左窗格中的 vCAC 和 VCACCAFE 模組，以便展開 vRealize Automation 外掛程式 API 物件的階層清單。

#### 下一個

您可從 API 元素中複製代碼並將其貼至指令碼方塊中。如需有關 API 指令碼的詳細資訊，請參閱《*使用 VMware vRealize Orchestrator 進行開發*》。

如需開發最佳做法的詳細資訊，請參閱 [vRealize Orchestrator 說明文件](#)。



# vRealize Automation 外掛程式指令碼範例

# 4

您可以剪下、貼上和編輯我們提供的 JavaScript 範例，來開發您自己的自動執行 vRealize Automation 工作自訂指令碼。

本章節討論下列主題：

- “[CRUD 基礎結構管理工作指令碼範例](#),” 第 27 頁
- “[尋找 vRealize Automation 實體指令碼範例](#),” 第 31 頁
- “[取得 vRealize Automation 範例指令碼佈建的資源](#),” 第 32 頁
- “[一般工作指令碼範例](#),” 第 33 頁

## CRUD 基礎結構管理工作指令碼範例

您可以剪下、貼上和編輯 JavaScript 範例來寫入執行 CRUD vRealize Automation 工作所需的指令碼。

如需有關 vRealize Orchestrator 中指令碼處理的詳細資訊，請參閱《[使用 VMware vRealize Orchestrator 進行開發](#)》。

### 範例 4-1. 建立 vRealize Automation 模型實體

此指令碼範例可執行下列動作：

- 1 定義模型名稱和實體集名稱。
- 2 定義主機前置詞的內容。
- 3 儲存主機前置詞實體。
- 4 定義佈建群組的內容。
- 5 將佈建群組定義為連結。
- 6 將佈建群組實體與主機名稱前置詞連結，來儲存該佈建群組實體。

**表格 4-1. 輸入變數**

變數	類型
host	vCAC:VcacHost

```
var modelName = 'ManagementModelEntities.svc';
var entitySetName = 'HostNamePrefixes';
var links = null;
var headers = null;
//Create properties for prefix entity
```

```

var prefixInputProperties = {
    MachinePrefix:'test-prefix',
    NextMachineNo:1,
    MachineNumberLength:3
};
//Save the prefix
var prefixEntity = vCACEntityManager
    .createModelEntity(host.id, modelName, entitySetName, prefixInputProperties, links, headers);
entitySetName = 'ProvisioningGroups';
//Create properties for the provisioning group entity
inputProperties = {
    GroupName:'TestGroupName',
    GroupDescription:'This group was generated with a vCO workflow',
    AdministratorEmail:'test@test.com',
    AdContainer:'AD',
    IsTestGroup:false,
    Flags:2,
    GroupType:1};
//Add a reference to the newly created prefix entity
links = {
    HostNamePrefix:prefixEntity
};
//Save the provisioning group
var entity = vCACEntityManager.createModelEntity(host.id, modelName, entitySetName,
inputProperties, links, headers);

```

## 範例 4-2. 更新 vRealize Automation 模型實體

此指令碼範例可執行下列動作：

- 1 從提供的實體取得主機識別碼。
- 2 從提供的實體取得模型名稱。
- 3 從提供的實體取得實體集名稱。
- 4 從提供的實體取得實體識別碼。
- 5 定義將進行更新的內容組合。
- 6 開始執行更新實體所需的動作。

**表格 4-2. 輸入變數**

變數	類型
entity	vCAC:Entity
updatedDescription	字串

```

var hostId = entity.hostId;
var modelName = entity.modelName;
var entitySetName = entity.entitySetName;
var entityIdString = entity.keyString;
var links = null;
var headers = null;
var updateProperties = new Properties();
updateProperties.put("UserNameDescription", updatedDescription);

```

```
//Update the user description
System.getModule("com.vmware.library.vcac")
    .updateVCACEntity(hostId, modelName, entitySetName, entityIdString, updateProperties, links,
headers);
```

### 範例 4-3. 讀取 vRealize Automation 模型實體

此指令碼範例可執行下列動作：

- 1 定義模型名稱和實體集名稱。
- 2 以內容物件定義藍圖識別碼。
- 3 讀取實體。

**表格 4-3. 輸入變數**

變數	類型
host	vCAC:VcacHost
blueprintID	字串

```
var modelName = 'ManagementModelEntities.svc';
var entitySetName = 'VirtualMachineTemplates';
var links = null;
var headers = null;
//Create properties for the prefix entity
var blueprintId = {
    VirtualMachineTemplateID:blueprintId,
};
//Read the blueprint
var entity = vCACEntityManager
    .readModelEntity(host.id, modelName, entitySetName, blueprintId, headers);
```

### 範例 4-4. 刪除 vRealize Automation 模型實體

此指令碼範例可執行下列動作：

- 1 從提供的實體取得主機識別碼。
- 2 從提供的實體取得模型名稱。
- 3 從提供的實體取得實體集名稱。
- 4 從提供的實體取得實體識別碼。
- 5 開始進行刪除實體所需的動作。

**表格 4-4. 輸入變數**

變數	類型
entity	vCAC:Entity

```
var hostId = entity.hostId;
var modelName = entity.modelName;
var entitySetName = entity.entitySetName;
var entityKeyString = entity.keyString;
var headers = null;
//Delete the entity
System.getModule("com.vmware.library.vcac")
    .deleteVCACEntity(hostId, modelName, entitySetName, entityKeyString, headers);
```

## 範例 4-5. 讀取 vRealize Automation 實體 (按自訂篩選器)

此指令碼範例可執行下列動作：

- 1 定義模型名稱和實體集名稱。
- 2 定義篩選實體所用的內容。
- 3 讀取實體的清單。

**表格 4-5. 輸入變數**

變數	類型
host	vCAC:VcacHost
templateName	字串

```
var modelName = 'ManagementModelEntities.svc';
var entitySetName = 'VirtualMachineTemplates';
var headers = null;
//Create properties for prefix entity
var properties = {
    VirtualMachineTemplateName:templateName,
};
//Read a list of entities
var entities = vCACEntityManager
    .readModelEntitiesByCustomFilter(host.id, modelName, entitySetName, properties, headers);
```

## 範例 4-6. 讀取 vRealize Automation 實體 (按系統查詢)

此指令碼範例可執行下列動作：

- 1 定義模型名稱和實體集名稱。
- 2 定義篩選實體所用的系統查詢，並選取按機器狀態和元件旗標篩選後，所產生的前十項虛擬機器結果。
- 3 讀取實體的清單。

**表格 4-6. 輸入變數**

變數	類型
host	vCAC:VcacHost

```
var modelName = 'ManagementModelEntities.svc';
var entitySetName = 'VirtualMachines';
var filter = "VirtualMachineState eq 'Off' and IsComponent eq true";
var orderBy = 'VirtualMachineName asc';
var top = 10; {
var skip = 0;,
var headers = null;
var select = null;
var entities = vCACEntityManager
    readModelEntitiesBySystemQuery(host.id, modelName, entitySetName, filter, orderBy, select,
top, skip, headers);
```

## 尋找 vRealize Automation 實體指令碼範例

您可以使用 `vCACCAFEEntitiesFinder` 指令碼公用程式物件，剪下、貼上和編輯 JavaScript 範例來寫入尋找 vRealize Automation 實體所需的指令碼。

如需有關 vRealize Orchestrator 中指令碼處理的詳細資訊，請參閱《*使用 VMware vRealize Orchestrator 進行開發*》。

### 範例 4-7. 尋找目錄資源 (按名稱篩選)

表格 4-7. 輸入變數

變數	類型
host	vCACCAFE:VcacHost

您可以使用下列其中一個範例：

- 此指令碼範例能夠按名稱和描述為符合 `name_of_the_resource` 查詢的目標主機，取得所有目錄資源。  

```
var items = vCACCAFEEntitiesFinder.findCatalogResources(host, "name_of_the_resource");
```
- 此指令碼範例可執行下列動作：
  - a 取得取用者資源服務，並叫用 `get` 方法，使其做為 `Pageable` 參數 (vCACCAFEPageOdataRequest 物件的執行個體) 傳遞。
  - b 提供 OData 查詢做為符合 `name_of_the_resource` 字串之 `name` 屬性的單一篩選器來建立 vCACCAFEPageOdataRequest 物件。

```
var service = host.createCatalogClient().getCatalogConsumerResourceService();

var filter = new Array();
filter[0] = vCACCAFEFilterParam.equal("name",
vCACCAFEFilterParam.string("name_of_the_resource"));
var query = vCACCAFEOdataQuery.query().addFilter(filter);

var items = service.getResourcesList(new vCACCAFEPageOdataRequest(query));
```

### 範例 4-8. 尋找目錄資源 (按擁有者篩選)

此指令碼範例可執行下列動作：

- 1 取得取用者資源服務，並叫用 `get` 方法，使其做為 `Pageable` 參數 (vCACCAFEPageOdataRequest 物件的執行個體) 傳遞。
- 2 提供 OData 查詢做為符合 `user@domain.com` 字串之 `owner/ref` 屬性的單一篩選器來建立 vCACCAFEPageOdataRequest 物件。

`owners/ref` 屬性是內部結構和目錄資源欄位組成的組合。vCACCAFECatalogResource 實體具有 `owners` 屬性，該屬性為 vCACCAFECatalogPrincipal 實體的集合。vCACCAFECatalogPrincipal 實體具有 `ref` 內容，該內容為使用者主體識別碼的字串表示法。

```
var filter = new Array();
filter[0] = vCACCAFEFilterParam.substringOf("owners/ref",
vCACCAFEFilterParam.string("user@domain.com"));
var query = vCACCAFEOdataQuery.query().addFilter(filter);

var items = service.getResourcesList(new vCACCAFEPageOdataRequest(query));
```

## 範例 4-9. 尋找目錄資源 (按名稱和擁有者篩選)

此指令碼範例使用 `vCACCAFEFilterParam.and(array of conditions)` 邏輯運算子，將前兩個範例中的 OData 查詢合併為一個條件。

```
var conditions = new Array();
conditions[0] = vCACCAFEFilterParam.equal("name",
vCACCAFEFilterParam.string("name_of_the_resource_here"));
conditions[1] = vCACCAFEFilterParam.substringOf("owners/ref",
vCACCAFEFilterParam.string("user@domain.com"));

var filter = new Array();
filter[0] = vCACCAFEFilterParam.and(conditions);
var query = vCACCAFE0dataQuery.query().addFilter(filter);

var items = service.getResourcesList(new vCACCAFEPage0dataRequest(query));
```

您可以使用其他邏輯運算子 (例如 `vCACCAFEFilterParam.group(array of parameters)`、`vCACCAFEFilterParam.not(parameter)`、`vCACCAFEFilterParam.startsWith(id, string)`、`vCACCAFEFilterParam.endsWith(id, string)`、`vCACCAFEFilterParam.greaterThan(id, number)`、`vCACCAFEFilterParam.lessThan(id, number)` 等) 定義其他條件。

## 取得 vRealize Automation 範例指令碼佈建的資源

您可以剪下、貼上和編輯 JavaScript 範例來撰寫指令碼，用於擷取 vRealize Automation 佈建的資源實際實體。

`CatalogResource` 類型代表 vRealize Automation 中的佈建資源。這個類型有 `ProviderBinding` 類型的屬性，代表目錄資源與其提供者之間的關係，其屬性如下：

- `bindingId` - 代表提供者唯一的實體識別碼
- `providerRef` - 識別直接與 vRealize Automation 元件登錄中登錄的服務相對應的目錄提供者

如需有關 vRealize Orchestrator 中指令碼處理的詳細資訊，請參閱《[使用 VMware vRealize Orchestrator 進行開發](#)》。

## 範例 4-10. 取得做為 vRealize Automation 目錄資源佈建的虛擬機器

這個範例使用 vRealize Automation 主機及其 IaaS 主機做為輸入參數，而且，對於提供的資源識別碼，傳回相對應的 IaaS 虛擬機器。指令碼代碼僅取用 `iaas-service` 提供者佈建的 `Virtual Machine` 類型目錄資源。

**表格 4-8. 輸入變數**

變數	類型
<code>vcacHost</code>	<code>vCACCAFE:VCACHost</code>
<code>iaasHost</code>	<code>vCAC:VCACHost</code>

```
// Id of the catalog resource (or vCACCAFECatalogResource_instance.getId())
var resourceId = "c222629c-6f90-4458-8c92-8ece0ba06173";

var resource = vCACCAFEEntitiesFinder.getCatalogResource(vcacHost, resourceId);

var resourceType = resource.getResourceTypeRef().getLabel();
System.log("resource type: " + resourceType);

var providerBinding = resource.getProviderBinding();
```

```

var bindingId = providerBinding.getBindingId();
System.log("provider binding id: " + bindingId);

var provider = providerBinding.getProviderRef();
System.log("provider id: " + provider.getId());
System.log("provider name: " + provider.getLabel());

if ((resourceType == "Virtual Machine") && (provider.getLabel() == "iaas-service")) {
    System.log("It is an IaaS VM!");

    // IaaS virtual machine
    var vm = Server.findForType("vCAC:VirtualMachine", bindingId);
    System.log("IaaS VM id: " + vm.virtualMachineID);
    System.log("IaaS VM name: " + vm.displayName);

    // IaaS Entity
    var entity =
System.getModule("com.vmware.library.vcac").getVirtualMachineEntityFromId(iaasHost, bindingId);
    System.log("IaaS entity id: " + entity.keyString);
}

```

## 一般工作指令碼範例

您可以剪下、貼上和編輯 JavaScript 範例，也可以利用這些範例來學習開發您自己的 vRealize Automation 一般工作指令碼。

如需有關 vRealize Orchestrator 中指令碼處理的詳細資訊，請參閱《*使用 VMware vRealize Orchestrator 進行開發*》。

### 範例 4-11. 建立 vRealize Automation 進階服務藍圖

此指令碼範例可執行下列動作：

- 1 設定用於建置服務藍圖的 vRealize Orchestrator 工作流程。
- 2 根據工作流程產生服務藍圖的內容。
- 3 產生服務藍圖實體。
- 4 發佈服務藍圖。

**表格 4-9. 輸入變數**

變數	類型
host	vCACCAFE:VCACHost

```

//ID of the workflow used to create the service blueprint
var workflowId = "44e42047-2fa0-4e4a-ba0c-12086540b28b";

var name = "MyBlueprint"
var description = "Blueprint description";
var workflowClient = host.createAdvancedDesignerClient().getAdvancedDesignerWorkflowService();

//Generate a service blueprint based on the workflow ID
var blueprint = workflowClient.generateServiceBlueprintByWorkflowId(workflowId);
blueprint.setTenant(host.tenant);
blueprint.setName(name);

```

```

blueprint.setDescription(description);

//Create the service blueprint
var blueprintService =
host.createAdvancedDesignerClient().getAdvancedDesignerServiceBlueprintService();
var uri = blueprintService.createServiceBlueprint(host.tenant , blueprint);

//Publish the service blueprint
var createdBlueprint = blueprintService.getServiceBlueprintByUri(uri);
blueprintService.updateServiceBlueprintStatus(host.tenant, createdBlueprint.getId(),
vCACCAFEDesignerPublishStatus.PUBLISHED);

```

## 範例 4-12. 建立 vRealize Automation 核准原則

此指令碼範例可執行下列動作：

- 1 取得核准原則類型。
- 2 設定需要核准的使用者和群組。
- 3 設定核准層級。
- 4 定義佈建前核准階段。
- 5 定義佈建後核准階段。
- 6 定義核准原則規格，例如名稱、描述和類型。
- 7 建立核准原則。
- 8 發佈核准原則。核准原則發佈後，即變成唯讀檔案。

**表格 4-10. 輸入變數**

變數	類型
host	vCACCAFE:VCACHost

```

// Get the type of approval policy by ID
var typeService = host.createApprovalClient().getApprovalApprovalPolicyTypeService();
var type = typeService.getApprovalPolicyType("com.vmware.cafe.catalog.request");

// Set the user and group required to complete the approval
var user = new vCACCAFEApprovalPrincipal();
user.setValue("user@domain.com");
user.setType(vCACCAFEApprovalPrincipalType.USER);

var group = new vCACCAFEApprovalPrincipal();
group.setValue("group@domain.com");
group.setType(vCACCAFEApprovalPrincipalType.GROUP);

// Set the level of the approval
var level = new vCACCAFEApprovalLevel();
level.setName("IT Approval Level");
level.setDescription("IT Approval Level description");
level.setApprovalMode(vCACCAFEApprovalMode.ALL);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(level, "getApprovers",
user);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(level, "getApprovers",
group);

```

```

level.setLevelNumber(1);

// Set pre-provisioning phase type and the phase of the approval
var phase1Type = new vCACCAFEApprovalPhaseType();
phase1Type.setId("com.vmware.cafe.catalog.request.pre");
phase1Type.setName("Pre-Provisioning type");
phase1Type.setDescription("Pre-Provisioning type description");
phase1Type.setPhaseOrder(1);

var phase1 = new vCACCAFEPhase();
phase1.setName("Pre-Provisioning");
phase1.setDescription("Pre provisioning phase");
phase1.setPhasetype(phase1Type);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(phase1, "getLevels",
level);

// Set post-provisioning phase type and the phase of the approval
var phase2Type = new vCACCAFEApprovalPhaseType();
phase2Type.setId("com.vmware.cafe.catalog.request.post");
phase2Type.setName("Post-Provisioning type");
phase2Type.setDescription("Post-Provisioning type description");
phase2Type.setPhaseOrder(1);

var phase2 = new vCACCAFEPhase();
phase2.setName("Post-Provisioning");
phase2.setDescription("Post provisioning phase");
phase2.setPhasetype(phase2Type);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(phase2, "getLevels",
level);

// Create the approval policy specifications
var spec = new vCACCAFEApprovalPolicy();
spec.setName("New Policy");
spec.setDescription("New Policy description");
spec.setPolicyType(type);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(spec, "getPhases", phase1);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(spec, "getPhases", phase2);

// Create the approval policy
var approvalPolicyService = host.createApprovalClient().getApprovalApprovalPolicyService();
var approvalPolicy = approvalPolicyService.createPolicy(spec);

// Publish the approval policy
approvalPolicy.setState(vCACCAFEApprovalPolicyState.PUBLISHED);
approvalPolicy = approvalPolicyService.update(approvalPolicy);
System.log("New approval policy id: " + approvalPolicy.getId());

```



# 索引

## 字母

API 存取 **24**  
CRUD 作業, vRealize Automation **13, 14**  
IaaS 主機, 設定 **10**  
vRealize Automation, CRUD 作業 **13, 14**  
vRealize Automation 主機, 設定 **10**  
vRealize Automation 外掛程式  
    設定 **9**  
    簡介 **7**  
vRealize Automation 模型實體  
    新增 **22**  
    讀取 **23**  
vRealize Orchestrator **7**

## 三劃

工作流程  
    CRUD **15, 21, 24**  
    IaaS **21**  
    可延伸性工作流程 **21**  
    要求工作流程 **23**  
    設定 **9**  
    標準工作流程 **15, 21, 24**  
    模型實體 **15, 21, 24**  
工作流程程式庫 **13, 27**

## 五劃

主機  
    設定 **9**  
    管理 **9**  
可編寫指令碼工作元素 **27, 33**

## 八劃

使用 vCACCAFEEntitiesFinder 物件 **31**  
取得佈建的虛擬機器 **32**

## 九劃

指令碼 **27, 33**

## 十二劃

尋找目錄資源 **31**

## 十三劃

新增 vRealize Automation 模型實體 **22**  
詳細目錄 **15**

## 十四劃

對象 **5**

## 十五劃

範例 **31, 32**

## 二十二劃

讀取 vRealize Automation 模型實體 **23**

