



REST 實作詳細資料

ONTAP automation

NetApp
January 13, 2026

目錄

| | |
|--------------------------|----|
| REST實作詳細資料 | 1 |
| ONTAP REST API 的操作特性 | 1 |
| 要求及回應API交易 | 1 |
| 支援CRUD作業 | 1 |
| 物件識別碼 | 1 |
| 物件執行個體和集合 | 1 |
| 同步與非同步作業 | 1 |
| 安全性 | 2 |
| ONTAP REST API 要求的輸入變數 | 2 |
| HTTP方法 | 2 |
| 路徑變數 | 3 |
| 要求標頭 | 3 |
| 申請本文 | 3 |
| 篩選物件 | 3 |
| 要求特定物件欄位 | 4 |
| 排序輸出集中的物件 | 5 |
| 擷取集中的物件時分頁 | 5 |
| 大小屬性 | 6 |
| 解讀 ONTAP REST API 回應 | 6 |
| HTTP狀態代碼 | 6 |
| 回應標頭 | 7 |
| 回應本文 | 7 |
| HAL連結 | 7 |
| 錯誤 | 8 |
| 與 ONTAP REST API 進行非同步處理 | 8 |
| 控制處理要求的方式 | 8 |
| 查詢與API要求相關聯的工作物件 | 9 |
| 發出非同步要求的一般程序 | 9 |
| ONTAP REST API 物件參考與存取 | 10 |
| 物件存取路徑 | 10 |
| 使用UUID存取物件 | 10 |
| 使用物件內容存取物件 | 10 |
| 叢集與SVM內容 | 11 |
| 在物件集合上使用修補程式和刪除 | 11 |
| 使用 ONTAP REST API 存取效能指標 | 11 |

REST實作詳細資料

ONTAP REST API 的操作特性

REST建立一套通用的技術和最佳實務做法、但每個API的詳細資料可能會因設計選項而異。

要求及回應API交易

每次REST API呼叫都會以HTTP要求的形式執行ONTAP、以產生與用戶端相關的回應。此要求/回應配對被視為API交易。在使用API之前、您應該先熟悉可用於控制要求的輸入變數、以及回應輸出的內容。

支援CRUD作業

透過「REST API」提供的每項資源ONTAP、都是根據CRUD模式來存取：

- 建立
- 讀取
- 更新
- 刪除

對於某些資源、只支援一部分作業。您應該檢閱ONTAP 位於您的叢集上的「ENetApp API」文件頁面ONTAP、以取得每項資源的詳細資訊。

物件識別碼

每個資源執行個體或物件在建立時都會指派一個唯一的識別碼。在大多數情況下、識別碼為128位元UUID。這些識別碼在特定ONTAP 的支援中心叢集內是全域唯一的。發出API呼叫以建立新的物件執行個體之後、會在HTTP回應的位置標頭中、傳回具有相關ID值的URL給呼叫者。您可以擷取識別碼、並在參照資源執行個體時用於後續通話。



物件識別碼的內容和內部結構可隨時變更。當您參照相關的物件時、只能視需要在適用的API呼叫上使用識別碼。

物件執行個體和集合

根據資源路徑和HTTP方法、API呼叫可套用至特定物件執行個體或物件集合。

同步與非同步作業

執行從用戶端收到的HTTP要求有兩種方式ONTAP。

同步處理

如果HTTP狀態代碼為200或201（成功）、則會立即執行要求、並以HTTP狀態代碼回應。ONTAP

使用方法Get、head和options的每個要求都會一律同步執行。此外、使用POST、修補及刪除的要求、設計

為在預期不到兩秒內完成時同步執行。

非同步處理

如果非同步要求有效、ONTAP 則會建立背景工作來處理要求、並建立工作物件來固定工作。這會將202 HTTP狀態連同工作物件傳回給呼叫者。若要判斷最終成功或失敗、您必須擷取工作狀態。

使用POST、修補和刪除方法的要求、是設計成在預期需要兩秒以上才能完成時、非同步執行。



◦ `return_timeout` 查詢參數可用於非同步 API 呼叫、並可將非同步呼叫轉換為同步完成。請參閱 ["使用工作物件進行非同步處理"](#) 以取得更多資訊。

安全性

REST API所提供的安全性主要是以ONTAP 現有的資訊安全功能為基礎。API使用下列安全性：

傳輸層安全性

透過網路在用戶端和 ONTAP LIF 之間傳送的所有流量、通常會根據 ONTAP 組態設定、使用 TLS 進行加密。

用戶端驗證

使用相同的驗證選項ONTAP、也可以搭配使用搭配使用《Rrest ONTAP API》（《Rrest System Manager》）和《Network Manageability SDK》（網路管理SDK）。

HTTP驗證

在 HTTP 層級、例如直接存取 ONTAP REST API 時、有兩個驗證選項、如下所述。在每種情況下、您都需要建立 HTTP 授權標頭、並將其納入每個要求中。

| 選項 | 說明 |
|-----------|--|
| HTTP 基本驗證 | ONTAP 使用者名稱和密碼會與冒號串連在一起。字串會轉換為 base64、並包含在要求標頭中。 |
| OAuth 2.0 | 從 ONTAP 9.14 開始、您可以從外部授權伺服器要求存取權杖、並將其作為承載權杖包含在要求標頭中。 |

如需 OAuth 2.0 及其如何在 ONTAP 中實作的詳細資訊、請參閱 ["ONTAP OAuth 2.0 實作總覽"](#)。另請參閱 ["準備好使用工作流程"](#) 請參閱本網站下方的。

驗證ONTAP

執行角色型授權模式。ONTAP您在存取ONTAP「靜態API」或API文件頁面時所使用的帳戶、應有適當的權限。

ONTAP REST API 要求的輸入變數

您可以透過HTTP要求中設定的參數和變數來控制API呼叫的處理方式。

HTTP方法

下表顯示了由REST API支援的HTTP方法ONTAP。



並非所有的HTTP方法都可在每個REST端點上使用。此外、修補程式和刪除功能也可用於集合。如需詳細資訊、請參閱 [_Object](#) 參考資料和 [access_](#)。

| HTTP方法 | 說明 |
|--------|-----------------------|
| 取得 | 擷取資源執行個體或集合上的物件屬性。 |
| 貼文 | 根據提供的輸入建立新的資源執行個體。 |
| 修補程式 | 根據提供的輸入內容更新現有的資源執行個體。 |
| 刪除 | 刪除現有的資源執行個體。 |
| 標題 | 有效發出GET要求、但僅傳回HTTP標頭。 |
| 選項 | 判斷特定端點支援哪些HTTP方法。 |

路徑變數

每個 REST API 呼叫所使用的端點路徑可以包含各種識別碼。每個 ID 對應特定的資源執行個體。範例包括叢集 ID 和 SVM ID。

要求標頭

您必須在HTTP要求中包含多個標頭。

內容類型

如果申請本文包含Json、則此標頭必須設定為「application/json」。

接受

此標頭應設定為「application/hal+json」。如果改為「application/json」、則除了擷取下一批記錄所需的連結之外、不會傳回任何的HAL連結。如果標頭是這兩個值以外的其他值、則回應中「content-type」標頭的預設值為「application/hal+json」。

授權

基本驗證必須使用編碼為基礎64字串的使用者名稱和密碼來設定。例如：

```
Authorization: Basic YWRtaW46cGV0ZXJzb24=。
```

申請本文

申請本文的內容會因特定通話而有所不同。HTTP要求本文包含下列其中一項：

- 具有輸入變數的Json物件
- 清空Json物件

篩選物件

使用 GET 方法發出 API 呼叫時、您可以使用查詢參數、根據任何屬性來限制或篩選傳回的物件。

剖析及解譯查詢參數

一組或多個參數可以附加到 URL 字串、從開始於 ? 字元。如果提供多個參數、則會根據分割查詢參數 & 字元。參數中的每個機碼和值都會在分割 = 字元。

例如、您可以使用等號來指定完全符合的值：

```
<field>=<value>
```

對於較複雜的查詢、額外的運算子會置於等號之後。例如、若要根據大於或等於某個值的特定欄位來選取物件集、查詢將是：

```
<field>=>=<value>
```

篩選運算子

除了上述範例之外、還有其他運算子可用於傳回超過某個值範圍的物件。ONTAP REST API 支援的篩選運算子摘要如下表所示。



任何未設定的欄位通常都會排除在相符查詢之外。

| 營運者 | 說明 |
|-----|---------|
| = | 等於 |
| < | 小於 |
| > | 大於 |
| <= | 小於或等於 |
| >= | 大於或等於 |
| ! | 不等於 |
| * | 貪婪的萬用字元 |

您也可以使用「null」關鍵字或其否定詞「!null」作為查詢的一部分、根據是否設定特定欄位來傳回物件集合。

工作流程範例

以下是本網站 REST API 工作流程的一些範例。

- ["列出磁碟"](#)

根據篩選 state 變數以選取備用磁碟。

要求特定物件欄位

根據預設、使用Get發出API呼叫時、只會傳回唯一識別物件的屬性、以及可用的HAL自我連結。這組最小欄位可做為每個物件的金鑰、而且會根據物件類型而有所不同。您可以使用「功能變數」查詢參數、以下列方式選取其他物件內容：

- 一般或標準欄位

指定「Fields =*」以擷取最常用的物件欄位。這些欄位通常會保留在本機伺服器記憶體中、或只需少量處理即可存取。使用Get搭配URL路徑金鑰 (UUID) 之後、傳回的物件內容相同。

- 所有欄位

指定「Fields =」以擷取所有物件欄位、包括需要額外伺服器處理才能存取的欄位。

- 自訂欄位選擇

使用「field=<field_name>」來指定所需的確切欄位。要求多個欄位時、必須使用不含空格的逗號分隔值。



最佳實務做法是、務必找出您想要的特定欄位。您只能在需要時擷取一組通用欄位或所有欄位。哪些欄位屬於一般欄位、並使用「Fields =*」傳回、由NetApp根據內部效能分析來決定。欄位的分類可能會在未來的版本中變更。

排序輸出集中的物件

資源集中的記錄會以物件定義的預設順序傳回。您可以使用「order_by」查詢參數、以下列欄位名稱和排序方向來變更順序：

```
「order_by=<欄位名稱> asc|desc」
```

例如、您可以依遞增順序、以遞減順序排序類型欄位、然後依ID排序：

```
「order_by=type desc 、 id asc」
```

請注意下列事項：

- 如果您指定排序欄位但未提供方向、則會以遞增順序排序這些值。
- 包含多個參數時、您必須以逗號分隔欄位。

擷取集中的物件時分頁

使用Get存取同一類型物件的集合時發出API呼叫ONTAP、根據兩個限制、嘗試傳回盡可能多的物件。您可以使用要求上的其他查詢參數來控制這些限制。針對特定Get要求所達成的第一個限制會終止要求、因此會限制傳回的記錄數目。



如果要求在重複所有物件之前結束、回應會包含擷取下一批記錄所需的連結。

限制物件數量

根據預設、ONTAP 針對Get要求、最多可傳回10、000個物件。您可以使用「max_Records」查詢參數來變更此限制。例如：

```
"Marax_Records=20"
```

實際傳回的物件數目可能會低於有效的最大值、取決於相關的時間限制、以及系統中的物件總數。

限制擷取物件所用的時間

根據預設、ONTAP 在允許的取得要求時間內、將盡可能多的物件傳回。預設的逾時時間為15秒。您可以使用「RETON_Timeout」查詢參數來變更此限制。例如：

```
"RETON_Timeout=5"
```

實際傳回的物件數目可能會低於有效的最大值、這是根據物件數目的相關限制、以及系統中的物件總數而定。

縮小結果集

如有需要、您可以將這兩個參數與其他查詢參數結合、以縮小結果集範圍。例如、下列項目最多會傳回指定時間之後產生的10個EMS事件：

```
time=> 2018-04-04T15:41:29.140265Z&max_records=10
```

您可以針對物件發出多個分頁要求。每次後續的API呼叫都應根據最後結果集中的最新事件、使用新的時間值。

大小屬性

某些API呼叫所使用的輸入值以及某些查詢參數均為數值。您可以選擇使用下表所示的字尾、而不是提供以位元組為單位的整數。

| 後置 | 說明 |
|----|----------------------------------|
| KB | KB千位元組（1024位元組）或Kibibibyte |
| MB | MB MB（KB x 1024位元組）或百萬位元組 |
| GB | GB GB GB（MB x 1024位元組）或GB |
| TB | TB TB TB（GB x 1024位元組）或TB位元組 |
| PB | PB PB PB（TB x 1024位元組）或pibibytes |

相關資訊

- ["物件參考與存取"](#)

解讀 ONTAP REST API 回應

每個API要求都會對用戶端產生回應。您應該檢查回應、判斷回應是否成功、並視需要擷取其他資料。

HTTP狀態代碼

以下說明了由靜止API使用的HTTP狀態代碼ONTAP。

| 程式碼 | 原因詞彙 | 說明 |
|--------|------|-----------------------------|
| 200 | 好的 | 表示未建立新物件的通話成功。 |
| 201. | 已建立 | 已成功建立物件。回應中的位置標頭包含物件的唯一識別碼。 |
| 202.02 | 已接受 | 已開始執行要求的背景工作、但尚未完成。 |
| 400 | 錯誤要求 | 無法辨識或不適當的要求輸入。 |
| 401. | 未獲授權 | 使用者驗證失敗。 |
| 403. | 禁止 | 由於授權錯誤、存取遭拒。 |

| 程式碼 | 原因詞彙 | 說明 |
|-------|---------|------------------------------------|
| 404.. | 找不到 | 要求中提及的資源不存在。 |
| 405 | 不允許使用方法 | 資源不支援要求中的HTTP方法。 |
| 409. | 衝突 | 建立物件的嘗試失敗、因為必須先建立不同的物件、或要求的物件已經存在。 |
| 500 | 內部錯誤 | 伺服器發生一般內部錯誤。 |

回應標頭

許多標頭都包含在ONTAP 由該功能產生的HTTP回應中。

位置

建立物件時、位置標頭會包含新物件的完整URL、包括指派給物件的唯一識別碼。

內容類型

這通常是「application/hal+json」。

回應本文

API要求所產生的回應本文內容、會因物件、處理類型、以及要求的成功或失敗而有所不同。回應一律以Json呈現。

- 單一物件

單一物件可根據要求傳回一組欄位。例如、您可以使用「Get」（取得）、使用唯一識別碼擷取叢集的選定內容。

- 多個物件

可從資源集合傳回多個物件。在所有情況下、都會使用一致的格式、其中「nm_Records」表示包含物件執行個體陣列的記錄和記錄數目。例如、您可以擷取在特定叢集中定義的節點。

- 工作物件

如果API呼叫以非同步方式處理、則會傳回工作物件、以固定背景工作。例如、用於更新叢集組態的修補程式要求會以非同步方式處理、並傳回工作物件。

- 錯誤物件

如果發生錯誤、一律會傳回錯誤物件。例如、當您嘗試變更未為叢集定義的欄位時、會收到錯誤訊息。

- 清空Json物件

在某些情況下、不會傳回任何資料、而且回應本文包含空白的Json物件。

HAL連結

支援HyperMedia做為應用程式狀態引擎（HATEOAS）的機制是使用HAL。ONTAP當傳回識別特定資源的物件

或屬性時、也會隨附一個由Hal-encoded編碼的連結、讓您輕鬆找到並判斷有關資源的其他詳細資料。

錯誤

如果發生錯誤、回應本文會傳回錯誤物件。

格式

錯誤物件的格式如下：

```
"error": {
  "message": "<string>",
  "code": <integer>[,
  "target": "<string>"]
}
```

您可以使用程式碼值來判斷一般錯誤類型或類別、以及判斷特定錯誤的訊息。如果可用、目標欄位會包含與錯誤相關的特定使用者輸入。

常見錯誤代碼

下表說明常見的錯誤代碼。特定API呼叫可能包含其他錯誤代碼。

| 程式碼 | | 說明 |
|-----|------|-----------------------|
| 1. | 409. | 具有相同識別碼的物件已存在。 |
| 2. | 400 | 欄位的值有無效值或遺失、或是提供額外欄位。 |
| 3. | 400 | 不支援此作業。 |
| 4. | 405 | 找不到具有指定識別碼的物件。 |
| 6. | 403. | 拒絕執行要求的權限。 |
| 8. | 409. | 資源正在使用中。 |

與 ONTAP REST API 進行非同步處理

在發出設計為非同步執行的API要求之後、一律會建立工作物件並傳回呼叫者。工作會說明並固定處理要求的背景工作。視HTTP狀態代碼而定、您必須擷取工作狀態、以判斷要求是否成功。

請參閱 ["API 參考"](#) 以判斷哪些API呼叫是設計為非同步執行。

控制處理要求的方式

您可以使用「RETON_Timeout」查詢參數來控制處理非同步API呼叫的方式。使用此參數有兩種可能的結果。

定時器會在要求完成之前過期

若為有效要求、ONTAP 則將傳回202HTTP狀態代碼及工作物件。您必須擷取工作狀態、以判斷要求是否成功完成。

要求會在定時器過期之前完成

如果申請有效且在到期前成功完成、ONTAP 則會連同工作物件一起傳回200個HTTP狀態代碼。由於要求已同步完成（如200所示）、因此您不需要擷取工作狀態。



"RETON_Timeout"參數的預設值為零秒。因此、如果您不包含此參數、系統會針對有效的要求傳回202 HTTP狀態代碼。

查詢與API要求相關聯的工作物件

HTTP回應中傳回的工作物件包含數個內容。您可以在後續的API呼叫中查詢狀態內容、以判斷要求是否成功完成。工作物件永遠處於下列其中一種狀態：

非終端機狀態

- 已佇列
- 執行中
- 已暫停

終端機狀態

- 成功
- 故障

發出非同步要求的一般程序

您可以使用下列高階程序來完成非同步API呼叫。此範例假設不使用「recnute_timeout」參數、或是背景工作完成之前的時間過期。

1. 發出設計為非同步執行的API呼叫。
2. 接收HTTP回應202、表示已接受有效的要求。
3. 從回應本文擷取工作物件的識別碼。
4. 在定時迴圈內、在每個週期中執行下列步驟：
 - a. 取得工作的目前狀態。
 - b. 如果工作處於非終端機狀態、請再次執行迴圈。
5. 當工作達到終端機狀態（成功、失敗）時停止。

相關資訊

- ["更新叢集連絡人"](#)
- ["取得工作執行個體"](#)

ONTAP REST API 物件參考與存取

透過REST API曝光的資源執行個體或物件ONTAP、可透過多種不同的方式進行參考和存取。

物件存取路徑

在較高層級、存取物件時有兩種路徑類型：

- 主要

物件是API呼叫的主要或直接目標。

- 國外

物件不是API呼叫的主要參考、而是從主要物件連結至。因此、它是外部或下游物件、並透過主要物件的欄位加以參照。

使用UUID存取物件

每個物件在建立時都會指派一個唯一的識別碼、在大多數情況下是128位元UUID。指派的UUID值不可改變、可在ONTAP 內部用於存取及管理資源。因此、UUID通常是存取物件的最快且最穩定的方法。

對於許多資源類型、可以在URL中提供UUID值作為路徑金鑰的一部分、以存取特定物件。例如、您可以使用下列項目來存取節點執行個體：「/叢集/節點/ {uuid} 」

使用物件內容存取物件

除了UUID之外、您也可以使用物件內容來存取物件。在大多數情況下、使用name屬性很方便。例如、您可以在URL字串中使用下列查詢參數、以名稱存取節點執行個體：「/cluster / nodes?name=node_on」。除了查詢參數之外、外部物件也可透過主要物件的屬性存取。

雖然您可以使用名稱或其他屬性來存取物件、而非UUID、但仍有幾個可能的缺點：

- 名稱欄位並非不可變更的欄位、可以變更。如果在存取物件之前變更物件名稱、將會傳回錯誤的物件、否則物件存取錯誤將會失敗。



此問題可能發生於外部物件上的POST或修補方法、或是主要物件上的Get方法。

- 必須將名稱欄位轉譯為對應的UUID。ONTAP這是一種間接存取、可能會成為效能問題。

尤其是當下列一項或多項條件成立時、效能可能會降低：

- 使用Get方法
- 系統會存取大量物件集合
- 使用複雜或精細的查詢

叢集與SVM內容

有幾個REST端點同時支援叢集和SVM。使用其中一個端點時、您可以透過「shscope=[SVM|cluster]」（叢集）值來指出API呼叫的內容。支援雙內容的端點範例包括IP介面和安全角色。



範圍值會根據每個API呼叫所提供的內容、提供預設值。

在物件集合上使用修補程式和刪除

支援資源執行個體上修補或刪除的每個REST端點、也支援物件集合上的相同方法。唯一的需求是必須透過URL字串中的查詢參數提供至少一個欄位。在發佈修補程式或刪除集合時、這相當於在內部執行下列動作：

- 以查詢為基礎的Get來擷取集合
- 修補或刪除集合中每個物件上的呼叫的序列順序

作業的逾時時間可設定為預設為15秒的「恢復逾時」。如果在逾時之前未完成、回應會包含下一個物件的連結。您必須使用下一個連結重新核發相同的HTTP方法、才能繼續作業。

使用 ONTAP REST API 存取效能指標

此功能可收集所選SVM儲存物件和傳輸協定的效能指標、並透過REST API回報此資訊。ONTAP您可以使用此資料來監控ONTAP 整個系統的效能。

對於特定的儲存物件或傳輸協定、效能資料分為三類：

- IOPS
- 延遲
- 處理量

在每個類別中、您可以使用下列一種或多種資料類型：

- 讀取 (R)
- 寫入 (W)
- 其他 (O)
- 總計 (T)

下表摘要說明ONTAP 透過「REST API」提供的效能資料、包括新增時的版本。如需詳細資訊、請參閱ONTAP 您的系統資訊的REST API線上文件頁面。

| 儲存物件或傳輸協定 | IOPS | 延遲 | 處理量 | 發行版ONTAP |
|-----------|------|------|-----|----------|
| 乙太網路連接埠 | 不適用 | 不適用 | RWT | 9.8 |
| FC連接埠 | RWOT | RWOT | RWT | 9.8 |
| IP介面 | 不適用 | 不適用 | RWT | 9.8 |
| FC介面 | RWOT | RWOT | RWT | 9.8 |

| 儲存物件或傳輸協定 | IOPS | 延遲 | 處理量 | 發行版ONTAP |
|------------------|------------|------------|------------|----------|
| NVMe命名空間 | RWOT | RWOT | RWOT | 9.8 |
| qtree統計資料 | 原始RWOT | 不適用 | 原始RWOT | 9.8 |
| Volume FlexCache | RWOT | RWOT | RWT | 9.8 |
| 節點-程序使用率 | 以數值形式處理使用率 | 以數值形式處理使用率 | 以數值形式處理使用率 | 9.8 |
| 雲端磁碟區 | RWOT | RWOT | 無法應用 | 9.7 |
| LUN | RWOT | RWOT | RWOT | 9.7 |
| Aggregate | RWOT | RWOT | RWOT | 9.7 |
| SVM NFS傳輸協定 | RWOT | RWOT | RWT | 9.7 |
| SVM CIFS傳輸協定 | RWOT | RWOT | RWT | 9.7 |
| SVM FCP傳輸協定 | RWOT | RWOT | RWT | 9.7 |
| SVM iSCSI傳輸協定 | RWOT | RWOT | RWT | 9.7 |
| SVM NVMe傳輸協定 | RWOT | RWOT | RWT | 9.7 |
| 叢集 | RWOT | RWOT | RWOT | 9.6 |
| 磁碟區 | RWOT | RWOT | RWOT | 9.6 |

版權資訊

Copyright © 2026 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變更本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

商標資訊

NETAPP、NETAPP 標誌及 <http://www.netapp.com/TM> 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。