Sun Java System Access Manager 7.1 管理指南

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

文件號碼：820–0840

070428@17039

目錄

前言

「Sun Java System Access Manager 7.1 管理指南」描述如何使用 Sun Java™ System Access Manager 主控台，以及如何透過指令行介面管理使用者和服務資料。

Access Manager 是 Sun Java Enterprise System (Java ES) 的元件，它是一組軟體元件，提供支援分散於整個網路或網際網路環境之企業應用程式所需的服務。

本書的適用對象

本書的適用對象爲使用 Sun Java System 伺服器與軟體實作網路存取平台的 IT 管理員與軟體開發人員。

閱讀本書之前

讀者應熟悉下列元件與概念：

- 「Sun Java System Access Manager 7.1 Technical Overview」中描述之 Access Manager 技術方面的概念。
- 部署平台：Solaris™ 或 Linux 作業系統
- 可執行 Access Manager 的 Web 容器：Sun Java System Application Server、Sun Java System Web Server、BEA WebLogic 或 IBM WebSphere Application Server
- 技術方面的概念：簡易目錄存取協定 (Lightweight Directory Access Protocol, LDAP)、Java 技術、JavaServer Pages™ (JSP) 技術、超文字傳輸協定 (HyperText Transfer Protocol, HTTP)、超文字標記語言 (HyperText Markup Language, HTML) 及可延伸標記語言 (eXtensible Markup Language, XML)

相關書籍

可用相關文件如下：

Access Manager 核心文件

「Access Manager 核心文件集」包含下列標題：

- 「Sun Java System Access Manager 7.1 版本說明」，可在產品發行後於線上取得。其匯集了各類最新資訊，包括目前版本中新功能的描述、已知問題和限制、安裝注意事項及如何報告軟體或文件的問題。
- 「Sun Java System Access Manager 7.1 Technical Overview」簡介 Access Manager 元件如何一同運作以整合存取控制功能，及保護企業資產和基於 Web 的應用程式。它還說明了 Access Manager 的基本概念與詞彙。
- 「Sun Java System Access Manager 7.1 Deployment Planning Guide」以解決方案生命週期爲根據，提供 Sun Java System Access Manager 的規劃和部署解決方案。
- 「Sun Java System Access Manager 7.1 Postinstallation Guide」提供安裝後配置 Access Manager 的相關資訊。
- 「Sun Java System Access Manager 7.1 Performance Tuning Guide」提供有關如何調校 Access Manager 及其相關元件以取得最佳效能的資訊。
- 「Sun Java System Access Manager 7.1 管理指南」描述如何使用 Access Manager 主控台，及如何透過指令行介面管理使用者和服務資料。
- 「Sun Java System Access Manager 7.1 Federation and SAML Administration Guide」提供以 Liberty Alliance Project 規格爲基礎的聯合模組之相關資訊。它包含以這些規格爲基礎的整合性服務之相關資訊、啓用基於 Liberty 環境的指示及用於延伸架構的應用程式程式設計介面 (application programming interface, API) 之摘要。
- 「Sun Java System Access Manager 7.1 Developer's Guide」提供如何自訂 Access Manager 及整合其功能至組織的現行技術基礎架構之相關資訊。它還包含有關此產品及其 API 之程式設計方面的詳細資訊。
- 「Sun Java System Access Manager 7.1 C API Reference」提供組成公用 Access Manager C API 的資料類型、結構及函數之摘要。
- 「Java API Reference」提供於 Access Manager 中實作 Java 套裝軟體的相關資訊。
- 「Sun Java System Access Manager Policy Agent 2.2 User's Guide」簡介 Access Manager 可用的策略功能和策略代理程式。

「版本說明」的更新內容與和核心文件修正之連結，可在 Sun Java Enterprise System 文件網站的 Access Manager 頁面中找到。已更新的說明文件標示有修訂日期。

Sun Java Enterprise System 產品文件

可在下列產品的文件中找到有用的資訊：

- Directory Server
- Web Server
- Application Server
- Web Proxy Server

相關的協力廠商網站參考

本文件提供了協力廠商的 URL 及其他相關資訊做爲參考。

備註 – Sun 對於本文件中所提及之協力廠商網站的使用不承擔任何責任。Sun 對於此類網站或資源中的 (或透過它們所取得的) 任何內容、廣告、產品或其他材料不做背書，也不承擔任何責任。對於因使用或依靠此類網站或資源中的 (或透過它們所取得的) 任何內容、產品或服務而造成的或連帶產生的實際或名義上之損壞或損失，Sun 概不負責，也不承擔任何責任。

文件、支援與訓練

Sun 網站支援關於下列額外資源的資訊：

- 文件 (`http://www.sun.com/documentation/`)
- 支援 (`http://www.sun.com/support/`)
- 訓練 (`http://www.sun.com/training/`)

印刷排版慣例

下表說明本書使用的印刷排版慣例。

表 P–1 印刷排版慣例

字體	意義	範例
`AaBbCc123`	指令、檔案及目錄的名稱；螢幕畫面輸出	請編輯您的 `.login` 檔案。 請使用 `ls -a` 列出所有檔案。 `machine_name% you have mail.`

表 P–1 印刷排版慣例 *(續)*

字體	意義	範例
AaBbCc123	您所鍵入的內容 (與螢幕畫面輸出相區別)	`machine_name%` **`su`** `Password:`
術語強調變數	新的字彙或術語、要強調的詞。將用實際的名稱或數值取代的指令行變數。	移除檔案的指令是 `rm` *filename*。
AaBbCc123	保留未譯的新的字彙或術語、要強調的詞。	**快取**是儲存在本機的副本。 請**不要**儲存此檔案。
「AaBbCc123」	用於書名及章節名稱。	請閱讀「使用者指南」的第 6 章。 **注意**：有些強調的項目在線上以粗體顯示。

指令範例中的 Shell 提示符號

下表顯示預設的 UNIX® 系統提示符號，以及 C shell、Bourne shell 和 Korn shell 的超級使用者提示符號。

表 P–2 Shell 提示符號

Shell	提示符號
C shell	**電腦名稱**%
C shell 超級使用者	**電腦名稱**#
Bourne shell 與 Korn shell	$
Bourne shell 與 Korn shell 超級使用者	#

Sun 歡迎您提出寶貴意見

Sun 致力於提高文件品質，因此誠心歡迎您提出意見與建議。

請至下列網址提出您對本文件的意見：`http://docs.sun.com`，並按一下 [Send Comments] (傳送您的意見)。請在線上表單中提供文件標題以及文件號碼。文件號碼可以在文件的標題頁或文件頂部找到，通常是一個七位或九位數的數字。

例如，本書標題爲「Sun Java System Access Manager 7.1 管理指南」，文件號碼是 820-0840。提出意見時您還需要在表格中輸入此文件的英文標題和文件號碼。例如，本文件的英文文件號碼爲 819-4670-10，完整標題爲「Sun Java System Access Manager 7.1 Administration Guide」。

第 1 部分

存取控制

這是「Sun Java System Access Manager™ 7.1 管理指南」的第一部分。「存取控制」介面提供建立與管理認證與授權服務的方法，以保護並管理基於範圍的資源。當企業使用者請求資訊時，Access Manager 將驗證使用者識別並授權使用者存取其所請求的特定資源。該部分包含以下章節：

◆ ◆ ◆ 第 1 章

1 Access Manager 主控台

Access Manager 主控台爲 Web 介面，允許具不同層級存取權限的管理員執行作業。比如建立範圍和組織、在範圍中建立使用者或從範圍刪除使用者以及建立用以保護和限制對範圍資源之存取的強制策略。此外，管理員可檢視和終止目前的使用者階段作業，管理其聯合配置 (建立、刪除和修改認證網域與提供者)。另一方面，不具管理權限的使用者可以管理個人資訊 (名稱、電子郵件位址、電話號碼等)、變更密碼、訂閱和取消訂閱群組以及檢視其角色。Access Manager 主控台有兩個主要檢視：

- 第 17 頁的 「管理檢視」
- 第 20 頁的 「使用者設定檔檢視」

管理檢視

當具有管理角色的使用者通過 Access Manager 認證後，預設檢視爲 [管理] 檢視。在此檢視中，管理員可執行大部份與 Access Manager 相關的管理工作。Access Manager 可用兩種不同的模式安裝；「範圍」模式和「舊有」模式。每個模式都有自己的主控台。如需有關「範圍」和「舊有」模式的更多資訊，請參閱「Sun Java System Access Manager 7.1 Technical Overview」。

備註 – 如果您以「範圍」模式安裝 Access Manager 7.1，則無法回復到「舊有」模式。如果您以「舊有」模式安裝 Access Manager，則可使用 amadmin 指令切換爲「範圍」模式。如需更多資訊，請參閱「Access Manager Administration Reference」中的「Changing from Legacy Mode to Realm Mode」。

範圍模式主控台

管理員可在「範圍」模式中使用管理主控台來管理基於範圍的存取控制、預設服務配置、Web 服務和聯合。若要存取管理員登入畫面，請在您的瀏覽器中使用以下位址語法：

protocol://*servername* /amserver/UI/Login

protocol 可爲 http 或 https，依您的部署而定。

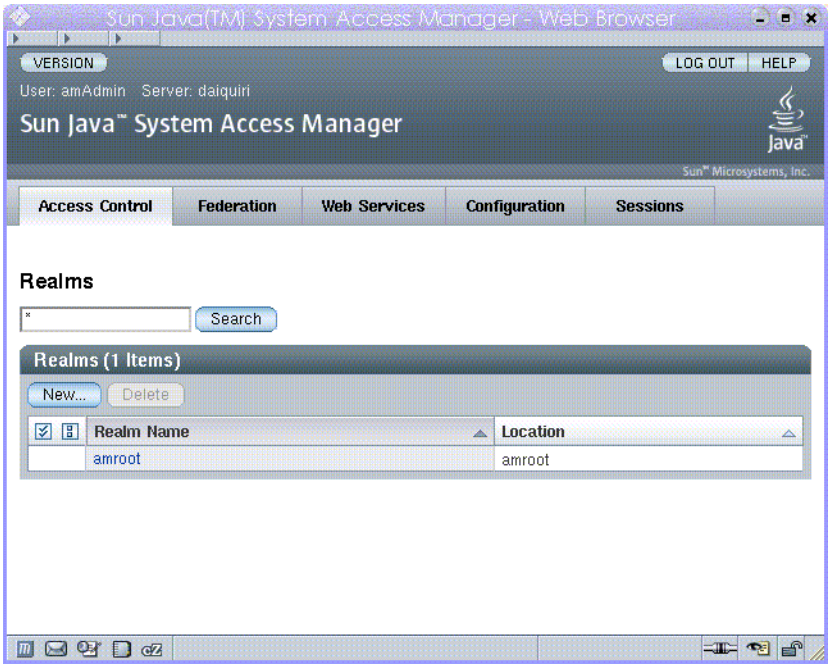

圖1–1 範圍模式管理檢視

舊有模式主控台

「舊有模式」主控台是以 Access Manager 6.3 的架構爲基礎。此舊有 Access Manager 架構使用 Sun Java System Directory Server 內的 LDAP 目錄資訊樹狀結構 (DIT) 。在「舊有」模式中，使用者資訊和存取控制資訊都是儲存在 LDAP 組織中。選擇「舊有」模式時，LDAP 組織相當於存取控制範圍。範圍資訊會整合在 LDAP 組織中。在「舊有」模式中，[目錄管理] 標籤可用於基於 Access Manager 的識別管理。

若要存取管理員登入畫面，請在您的瀏覽器中使用以下位址語法：

protocol://*servername* /amserver/console

protocol 可爲 http 或 https，依您的部署而定。

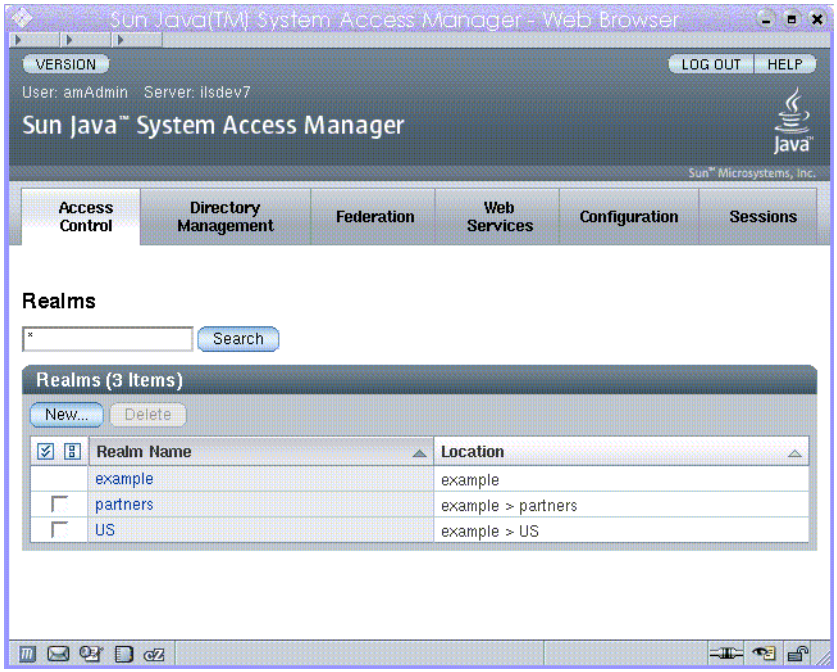

圖 1–2 舊有模式管理檢視

舊有模式 6.3 主控台

Access Manager 6.3 的部份功能不能在 Access Manager 7.1 主控台中使用。因此，管理員可透過 7.1 舊有部署登入 6.3 主控台。若 Access Manager 是建立在 Sun Java System Portal Server 或其他需使用 Sun Java System Directory Server 做爲中央識別儲存庫的 Sun Java System 通訊產品上時，通常是使用此主控台。其他功能，如「委託管理」和「服務類別」，只能透過此主控台存取。

備註 – 請勿互換使用 6.3 和 7.1 舊有模式主控台。

若要存取 6.3 主控台，請在您的瀏覽器中使用以下位址語法：

protocol://*servername* /amconsole

`protocol` 可爲 http 或 https，依您的部署而定。

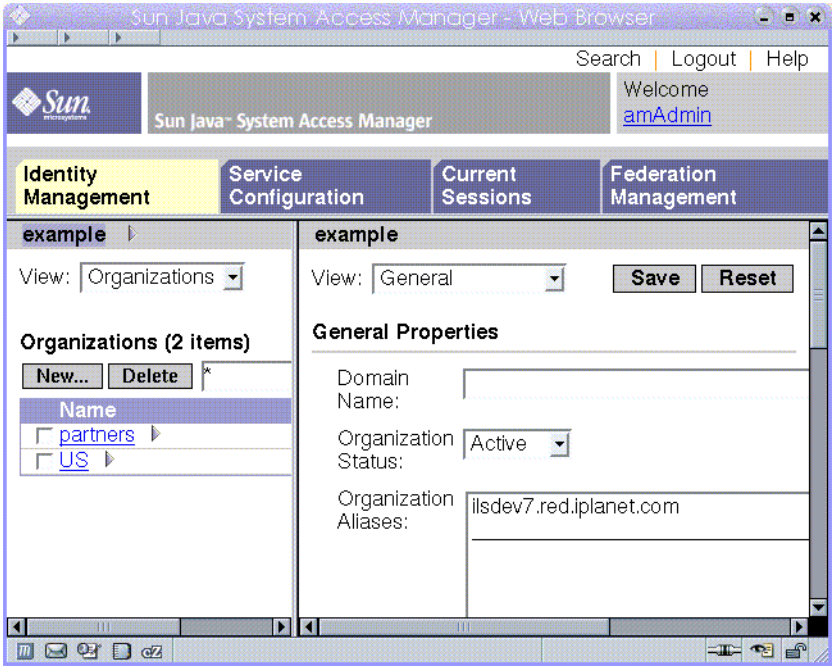

圖 1–3 舊有 6.3 主控台

使用者設定檔檢視

沒有指定管理角色的使用者認證 Access Manager 時，預設的檢視爲使用者本身的使用者設定檔。[使用者設定檔] 檢視可從「範圍」或「舊有」模式存取。使用者必須在 [登入] 頁面輸入使用者自己的使用者名稱和密碼才可存取此檢視。

在此檢視中，使用者可以修改其個人設定檔的特定屬性值。這包括但不僅限於名稱、家庭住址和密碼。[使用者設定檔] 檢視中顯示的屬性可以延伸。

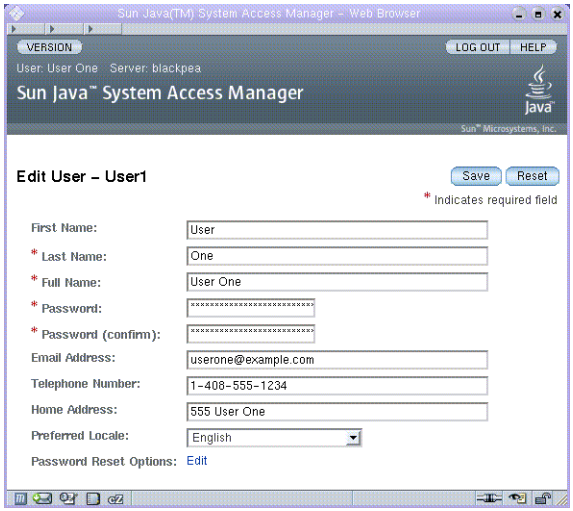

圖 1–4 使用者設定檔檢視

◆ ◆ ◆ 第 2 章

2 管理範圍

存取控制範圍是一組可與使用者或使用者群組關聯的認證特性與授權策略。範圍資料儲存於一個專用資訊樹狀結構中，該樹狀結構由 Access Manager 在您指定的資料存放區中建立。Access Manager 框架於 Access Manager 資訊樹狀結構中聚集每一個範圍中的策略與特性。依預設，Access Manager 會將 Access Manager 資訊樹狀結構做爲特殊分支插入到 Sun Java Enterprise System Directory Server 中，但使用者資料除外。當使用任何 LDAPv3 資料庫時，您可以使用存取控制範圍。

如需有關範圍的更多資訊，請參閱「Sun Java System Access Manager 7.1 Technical Overview」。

於 [範圍] 標籤中，您可爲存取控制配置下列特性：

- 第 24 頁的 「認證」
- 第 25 頁的 「服務」
- 第 26 頁的 「權限」

建立及管理範圍

本節描述如何建立及管理範圍。

▼ 建立新的範圍

1 **從 [存取控制] 標籤下的 [範圍] 清單中選取 [新增]。**

2 **定義下列一般屬性：**

名稱　輸入範圍的名稱。

父系　定義您正在建立的範圍位置。選取新範圍將存在處的父系範圍。

3 **定義下列範圍屬性：**

範圍狀態	選擇 [使用中] 或 [非使用中] 狀態。預設值爲 [使用中]。在範圍存在期間，可以透過選取 [特性] 圖示隨時變更該狀態。登入時，選擇 [非使用中] 以停用使用者存取。
範圍/DNS 別名	允許增加範圍 DNS 名稱的別名。此屬性僅接受「實際的」網域別名 (不允許使用隨機字串)。

4 **按一下 [確定] 以儲存，或按一下 [取消] 以返回前一個頁面。**

一般特性

[一般特性] 頁面顯示範圍的基本屬性。若要修改這些特性，於 [存取控制] 標籤之下按一下 [範圍名稱] 清單中的範圍。然後，編輯下列特性：

範圍狀態	選擇 [使用中] 或 [非使用中] 狀態。預設值爲 [使用中]。在範圍存在期間，可以透過選取 [特性] 圖示隨時變更該狀態。登入時，選擇 [非使用中] 以停用使用者存取。
範圍/DNS 別名	允許增加範圍 DNS 名稱的別名。此屬性僅接受「實際的」網域別名 (不允許使用隨機字串)。

一旦您編輯了特性，請按一下 [儲存]。

備註 – `AMAdmin.dtd` 中的 `recursive=true` 旗標對於以範圍模式在子範圍內搜尋物件不起作用。這個旗標只在舊有模式中有效，因爲所有子組織都位在相同根尾碼之下。而在範圍模式，每個子範圍能有不同的根尾碼，甚至可能位於不同的伺服器。若要在子範圍中搜尋物件 (例如群組)，必須在 XML 資料檔案內指定要在其中進行搜尋的子範圍。

認證

一般認證服務必須先註冊爲某個範圍的服務，使用者才能使用其他認證模組登入。核心認證服務可讓 Access Manager 管理員定義範圍認證參數的預設值。若未於特定認證模組中定義置換值，則稍後可以使用這些值。核心認證服務的預設值定義於 `amAuth.xml` 檔案中，並於安裝後儲存於 Directory Server 內。

如需更多資訊，請參閱管理認證

服務

在 Access Manager 中，服務是一組由 Access Manager 主控台一起管理的屬性。屬性可以只是一些相關資訊，如員工名稱、職稱與電子郵件地址。但屬性通常做爲軟體模組 (如郵件應用程式或發薪服務) 的配置參數。

經由 [服務] 標籤，您可爲範圍增加並配置大量 Access Manager 預設服務。您可以增加下列服務：

- 管理
- 探索服務
- 全域化設定
- 密碼重設
- 階段作業
- 使用者

備註 – Access Manager 會強制服務 .xml 檔案中的必需屬性必須具備一些預設值。若服務的必需屬性不具有任何值，則需要增加預設值並重新載入服務。

▼ 將服務增加至範圍

1. **按一下您要增加服務的範圍之名稱。**
2. **選取 [服務] 標籤。**
3. **按一下 [服務] 清單中的 [增加]。**
4. **選取您要為範圍增加的服務。**
5. **按 [下一步]。**
6. **定義範圍屬性以配置服務。請參閱線上說明中的「配置」以取得服務屬性的說明。**
7. **按一下 [完成]。**
8. **若要編輯服務的特性，請按一下 [服務] 清單中的名稱。**

權限

在 Access Manager 中，委託模型以指定給管理員的權限 (或資格) 爲基礎。權限是一種可對資源執行的作業 (或動作)，例如對「策略」物件執行的「讀取」作業。定義的作業集爲「讀取」、「修改」及「委託」。資源是可對其執行動作的物件，可以是配置物件或識別物件。

配置物件的範例有「認證配置」、「策略」、「資料存放區」等等。識別物件的範例有「使用者」、「群組」、「角色」及「代理程式」。可以在 Access Manager 中動態建立和動態增加一組權限，但在安裝期間，會在 Access Manager 中增加一小組權限，以使 Access Manager 正確地執行。一旦載入權限，即可將其指定給角色及群組。屬於這些角色及群組的使用者會成爲受委託的管理員，並可執行所指定的作業。一般來說，管理員是被指定一組或更多權限的角色或群組之成員。

Access Manager 7.1 可讓您爲下列管理員類型配置權限：

- 範圍管理員 — 範圍管理員具有對所有物件 (包括配置及識別物件) 執行「讀取」、「修改」及「委託」作業的所有權限。可將範圍管理員視爲 Unix 系統中的「超級使用者」。範圍管理員可以建立子範圍、修改所有服務的配置，也可建立、修改及刪除「使用者」、「群組」、「角色」及「代理程式」。
- 策略管理員 — 策略管理員僅具有管理策略及策略服務配置的權限。他們可以建立、修改及刪除包含「規則」、「主體」、「條件」及「回應」屬性的策略。但是，若要管理策略，這些管理員需要具有「識別儲存庫主體」及「認證」配置的讀取權限。這些管理員可以檢視識別及認證配置。
- 記錄管理員 — 記錄管理員具有讀取及/或寫入記錄的權限，這些權限可用來防止稽核記錄遭受惡意應用程式惡意濫用。因爲記錄介面是公用的，任何經過認證的使用者可能都可以讀寫記錄，因此增加此權限可以防止發生這種濫用情形。記錄介面的主要使用者是 J2EE 及 Web Agents，它們只需要「修改」權限，不應具有「讀取」權限。同樣地，檢視記錄的管理員只應具有「讀取」權限，不應具有「修改」權限。爲了配合這些類型的用法，記錄權限進一步細分如下：
  - 具有「寫入存取權」的記錄管理員 – 這些管理員具有寫入所有記錄檔的權限。
  - 具有「讀取存取權」的記錄管理員 – 這些管理員具有讀取所有記錄檔的權限。
  - 具有「讀取與寫入存取權」的記錄管理員 – 這些管理員具有讀寫所有記錄檔的權限。

定義 Access Manager 7.1 的權限

新的 Access Manager 7.1 安裝實例爲策略管理員、範圍管理員 (或「舊有」模式中的組織管理員) 及記錄管理員提供存取權限。若要指定或修改權限，按一下您要編輯的角色或群組名稱。您可以選擇下列任一選項：

對所有記錄檔的讀取和寫入存取　　爲記錄管理員定義讀取及寫入存取權限。

對所有記錄檔的寫入存取	只爲記錄管理員定義寫入存取權限。
對所有記錄檔的讀取存取	只爲記錄管理員定義讀取存取權限。
僅針對策略特性的讀取與寫入存取權	爲策略管理員定義讀取及寫入存取權限。
所有範圍與策略特性的讀取與寫入存取權	爲範圍管理員定義讀取及寫入存取權限。

為從 Access Manager 7.0 升級到 7.1 定義權限

如果您將 Access Manager 從 7.0 升級到 7.1，其權限配置會與新的 Access Manager 7.1 安裝的權限配置有所不同，但仍支援策略管理員、範圍管理員及記錄管理員的權限。若要指定或修改權限，按一下您要編輯的角色或群組名稱。您可以選擇下列任一選項：

對資料存放區唯讀存取	爲策略管理員定義對資料存放區的讀取存取權限。
對所有記錄檔的讀取和寫入存取	爲記錄管理員定義讀取及寫入存取權限。
對所有記錄檔的寫入存取	只爲記錄管理員定義寫入存取權限。
對所有記錄檔的讀取存取	只爲記錄管理員定義讀取存取權限。
僅針對策略特性的讀取與寫入存取權	爲策略管理員定義讀取及寫入存取權限。
所有範圍與策略特性的讀取與寫入存取權	爲範圍管理員定義讀取及寫入存取權限。
所有特性與服務的唯讀存取權	爲策略管理員定義對所有特性及服務的讀取存取權限。

對於下列定義，無論是單獨使用還是一同使用，皆不受 Access Manager 支援：

- 資料存放區唯讀存取權
- 所有特性與服務的唯讀存取權

這些權限定義必須與「僅針對策略特性的讀取與寫入存取權」定義一起使用，以定義策略管理員的委託控制。

◆ ◆ ◆ 第 3 章

3 資料存放區

資料存放區是一個資料庫，您可在其中儲存使用者屬性與使用者配置資料。Access Manager 提供可連線至 LDAPv3 識別儲存庫框架的識別儲存庫外掛程式。這些外掛程式可讓您檢視並擷取 Access Manager 使用者資訊，而無需對您現有的使用者資料庫進行變更。Access Manager 框架整合識別儲存庫外掛程式的資料與其他 Access Manager 外掛程式的資料以形成每位使用者的虛擬識別。Access Manager 稍後可在多個識別儲存庫間的認證與授權程序中使用通用識別。當使用者階段作業結束時，將銷毀虛擬使用者識別。

Access Manager 資料存放區類型

本節描述您可配置的資料存放區類型，也提供建立新資料存放區類型的步驟以及配置它們的方法。

您可以針對下列任一資料存放區類型建立新的資料存放區實例：

Access Manager 儲存庫外掛程式

此資料存放區類型位於 Sun Java System Directory Server 實例中，並保存 Access Manager 資訊樹狀結構。此資料存放區類型使用不屬於 LDAP 版本 3 規格的 Directory Server 功能 (如角色及服務類別)，並與先前版本的 Access Manager 相容。

Active Directory

此資料存放區類型使用 LDAP 版本 3 規格向 Microsoft Active Directory 的實例寫入識別資料。

平面檔案儲存庫

此儲存庫可讓您在 Access Manager 的本機安裝實例上以平面 DIT 結構儲存資料及識別，而不必建立個別的資料存放區。它通常用來測試或驗證概念部署。

通用 LDAPv3

此資料存放區類型可讓您向任何與 LDAPv3 相容的資料庫寫入識別資料。如果您使用的 LDAPv3 資料庫不支援持續搜尋，則無法使用快取功能。

具有 Access Manager 模式的 Sun Directory Server

此資料存放區類型位於 Sun Java System Directory Server 實例中，並保存 Access Manager 資訊樹狀結構。它與 Access Manager 儲存庫外掛程式的不同之處在於，後者有更多配置屬性可讓您更好地自訂資料存放區。

▼ 建立新的資料存放區

下節將描述連線資料存放區的步驟。

1. **選取要增加資料存放區的範圍**。
2. **按一下 [資料存放區] 標籤**。
3. **按一下 [資料存放區] 清單中的 [新建]**。
4. **輸入資料存放區的名稱**。
5. **選取要建立的資料存放區類型**。
6. **按 [下一步]**。
7. **輸入適當的屬性值以配置資料存放區**。
8. **按一下 [完成]**。

資料存放區屬性

本節定義用來配置每個新 Access Manager 資料存放區的屬性。資料存放區屬性爲：

備註 – Active Directory、通用 LDAPv3 以及具有 Access Manager 模式的 Sun Directory Server 資料存放區類型共用相同的基礎外掛程式，因此配置屬性都一樣。但是，對每一種資料存放區類型而言，其中有些屬性的預設值會不同，這些值會對應顯示在 Access Manager 主控台中。

Access Manager 儲存庫屬性

下列屬性用於配置 Access Manager 儲存庫外掛程式：

類別名稱

指定實作 Access Manager 儲存庫外掛程式的類別檔案位置。

Access Manager 支援的類型和作業

指定 LDAP 伺服器允許的或可執行的作業。預設作業是僅限於此 LDAPv3 儲存庫外掛程式支援的作業。以下是「LDAPv3 儲存庫外掛程式」支援的作業：

- 群組 — 讀取、建立、編輯、刪除
- 使用者 — 讀取、建立、編輯、刪除、服務
- 代理程式 — 讀取、建立、編輯、刪除

可根據您的 LDAP 伺服器設定及作業從上述清單中移除權限，但您不能增加更多權限。

如果所配置的 LDAPv3 儲存庫外掛程式指向 Sun Java Systems Directory Server 的實例，則可以增加**角色**類型的權限。否則，由於其他資料存放區可能不支援角色，或許不能增加此權限。「角色」類型的權限爲：

- 角色 — 讀取、建立、編輯、刪除

如果將**使用者**做爲 LDAPv3 儲存庫支援的類型，則該使用者可以進行讀取、建立、編輯及刪除服務作業。換句話說，如果**使用者**是受支援的類型，則讀取、編輯、建立及刪除作業可讓您讀取、編輯、建立及刪除識別儲存庫中的使用者項目。`user=service` 作業可讓 Access Manager 服務存取使用者項目中的屬性。此外，如果爲使用者所屬的範圍或角色指定了動態服務，則使用者也可以存取動態服務屬性。

使用者還能管理所有指定服務的使用者屬性。如果使用者將 `service` 做爲作業 (`user=service`)，則它會指定所有服務相關的作業均受支援。這些作業包括 `assignService`、`unassignService`、`getAssignedServices`、`getServiceAttributes`、`removeServiceAttributes` 及 `modifyService`。

組織 DN 值

定義指向 Access Manager 管理的 Directory Server 中組織的 DN。此將做爲於資料存放區中執行之所有作業的基底 DN。

使用者容器命名屬性

使用者存在於使用者容器中時，指定使用者容器的命名屬性。若使用者並未位於使用者容器中，此欄位應爲空白。

使用者容器值

指定使用者容器值。預設值爲 `people`。

代理程式容器命名屬性

若代理程式位於一個代理程式容器中，則爲代理程式容器的命名屬性。若代理程式並未位於代理程式容器中，此欄位應爲空白。

代理程式容器值

指定代理程式容器值。預設值爲 `agents`。

遞迴搜尋

若啓用，則在 Access Manager 儲存庫中執行的搜尋會針對指定的識別進行遞迴搜尋。例如，對下列資料結構執行遞迴搜尋：

root
realm1
  subrealm11
    user5
  subrealm12
    user6
realm2
  user1
  user2
  subrealm21
    user3
    user4

會產生下列結果：

- 如果從 root 開始執行搜尋，而且該層級上沒有定義任何使用者 (amadmin 及 anonymous 除外)，則搜尋會傳回 user 1–6。
- 如果從 realm1 開始執行搜尋，而且沒有定義任何使用者，則搜尋會傳回 user5 及 user6。
- 如果從 realm2 開始執行搜尋 (定義了 2 位使用者)，則搜尋會傳回 user 1–4。

複製範圍配置

如果在範圍模式安裝中啓用了此屬性，則 Access Manager 會爲每個存在於儲存庫中的範圍及子範圍建立等同的組織及子組織。此外，在範圍/子範圍中註冊的服務也會在新建立的組織/子組織中進行註冊。範圍 DIT 及組織 DIT 均存在於資料存放區內。

平面檔案儲存庫屬性

下列屬性用於配置平面檔案儲存庫：

檔案儲存庫外掛程式類別名稱

此屬性指定可提供平面檔案實作的 Java 類別檔案。此屬性不得修改。

檔案儲存庫目錄

定義存放識別及其屬性的基底目錄。

快取

若啓用 (預設值)，則會快取識別及其屬性。從而後續請求將不會存取檔案系統。

更新快取的時間

若啓用了快取，則此屬性決定檢查快取的時間間隔 (以分鐘爲單位)，即在該時間間隔之後檢查快取中的項目，以確定有無對檔案系統進行過任何變更。檢查機制基於時間戳記。

檔案使用者物件類別

定義建立使用者時要自動增加至使用者的物件類別。

密碼屬性

提供包含用於認證之密碼的屬性名稱。若啓用了「資料存放區」認證模組，則此屬性用於認證使用者。

狀態屬性

提供儲存識別狀態的屬性名稱。狀態屬性的值是**使用中**或**非使用中**。認證識別期間會使用狀態屬性。如果識別是**非使用中**，則不會認證使用者。

雜湊的屬性

提供一份屬性值將雜湊且儲存在檔案內的屬性清單。一旦雜湊，就無法取得原始值。只能擷取雜湊的值。這可用來確保特定屬性不應永久儲存 (但需用於驗證) 情況下的私密性。識別的密碼屬性即爲此類屬性的範例。

已加密的屬性

提供一份屬性值將加密且儲存在檔案內的屬性清單。雖然屬性經過加密及儲存，但呼叫識別存放區 API 仍會傳回原來未加密的值。這樣可防止使用者直接存取檔案系統並讀取機密屬性。

LDAPv3 屬性

下列屬性用於配置 LDAPv3 儲存庫外掛程式：

LDAP 伺服器

輸入您要連線的 LDAP 伺服器名稱。格式應爲 `hostname.domainname:portnumber`。

如果輸入了多個 `host:portnumber` 項目，則會嘗試連線清單中的第一個主機。僅當連線至目前主機失敗時，才會嘗試清單中的下一個項目。

LDAP 連結 DN

指定 Access Manager 將用來向您目前所連線之 LDAP 伺服器認證的 DN 名稱。具有連結所用之 DN 名稱的使用者應具有您於第 35 頁的「LDAPv3 外掛程式支援的類型和作業」屬性中配置之正確的增加/修改/刪除權限。

LDAP 連結密碼

指定 Access Manager 將用來向您目前所連線之 LDAP 伺服器認證的 DN 密碼。

LDAP 連結密碼 (確認)

確認密碼。

LDAP 組織 DN

此資料存放區將對映的 DN 。此將爲於此資料存放區中執行之所有作業的基底 DN。

LDAP SSL

當啓用時，Access Manger 將使用 HTTPS 協定連線至主伺服器。

LDAP 連線池最小大小

指定連線池中的初始連線數目。使用連線池就不必每次都建立新的連線。

LDAP 連線池最大大小

指定允許的最大連線數目。

從搜尋傳回的最多結果

指定搜尋作業傳回項目的最大數目。若已達到上限，Directory Server 會傳回任何符合搜尋請求的項目。

搜尋逾時

指定爲搜尋請求配置的最大秒數。若已達到上限，Directory Server 會傳回任何符合搜尋請求的搜尋項目。

LDAP 依照參照

若啓用，此選項指定自動跟隨其他 LDAP 伺服器的參照。

LDAPv3 儲存庫外掛程式類別名稱

指定實作 LDAPv3 儲存庫的類別檔案的位置。

一般屬性名稱對映

使框架所知的通用屬性對映至本機資料存放區。例如，若框架使用 `inetUserStatus` 來決定使用者狀態，則本機資料存放區可能實際使用 `userStatus`。屬性定義區分大小寫。

LDAPv3 外掛程式支援的類型和作業

指定 LDAP 伺服器允許的或可執行的作業。預設作業是僅限於此 LDAPv3 儲存庫外掛程式支援的作業。以下是「LDAPv3 儲存庫外掛程式」支援的作業：

- 群組 — 讀取、建立、編輯、刪除
- 使用者 — 讀取、建立、編輯、刪除、服務
- 代理程式 — 讀取、建立、編輯、刪除

可根據您的 LDAP 伺服器設定及作業從上述清單中移除權限，但您不能增加更多權限。

如果所配置的 LDAPv3 儲存庫外掛程式指向 Sun Java Systems Directory Server 的實例，則可以增加**角色**類型的權限。否則，由於其他資料存放區可能不支援角色，或許不能增加此權限。「角色」類型的權限爲：

- 角色 — 讀取、建立、編輯、刪除

如果將**使用者**做爲 LDAPv3 儲存庫支援的類型，則該使用者可以進行讀取、建立、編輯及刪除服務作業。換句話說，如果**使用者**是受支援的類型，則讀取、編輯、建立及

刪除作業可讓您讀取、編輯、建立及刪除識別儲存庫中的使用者項目。user=service 作業可讓 Access Manager 服務存取使用者項目中的屬性。此外，如果爲使用者所屬的範圍或角色指定了動態服務，則使用者也可以存取動態服務屬性。

使用者還能管理所有指定服務的使用者屬性。如果使用者將 service 做爲作業 (user=service)，則它會指定所有服務相關的作業均受支援。這些作業包括 assignService、unassignService、getAssignedServices、getServiceAttributes、removeServiceAttributes 及 modifyService。

LDAPv3 外掛程式搜尋範圍

定義用於尋找 LDAPv3 外掛程式項目的範圍。此範圍必須爲以下一種：

- SCOPE_BASE
- SCOPE_ONE
- SCOPE_SUB (預設)

LDAP 使用者搜尋屬性

此欄位定義對使用者進行搜尋的屬性類型。例如，如果使用者的 DN 是 uid=user1,ou=people,dc=iplanet,dc=com，則命名屬性是 uid。

LDAP 使用者搜尋篩選器

指定用於尋找使用者項目的搜尋篩選器。

LDAP 使用者物件類別

指定使用者的物件類別。建立使用者時，本使用者物件類別清單將增加至使用者的屬性清單。

LDAP 使用者屬性

定義與使用者相關聯的屬性清單。不允許讀取/寫入任何不在此清單上的使用者屬性。這些屬性區分大小寫。於此處定義物件類別與屬性模式之前，必須在 Directory Server 中定義物件類別與屬性模式。

LDAP 使用者建立屬性對映

指定建立使用者時需要哪些屬性。此屬性使用下列語法：

```
DestinationAttributeName=SourceAttributeName
```

如果缺少來源屬性名稱，則預設值是使用者 ID (uid)。例如：

cn
sn=givenName

若要建立使用者設定檔，則 cn 和 sn 都是必要的。cn 取得屬性 uid 的值，而 sn 取得屬性 givenName 的值。

使用者狀態屬性

指定可指示使用者狀態的屬性名稱。

使用者狀態使用中的值

指定使用中使用者狀態的屬性名稱。預設值爲**使用中**。

使用者狀態非使用中的值

指定非使用中使用者狀態的屬性名稱。預設值爲**非使用中**。

LDAP 群組搜尋屬性

此欄位定義對群組進行搜尋的屬性類型。預設值爲 cn。

LDAP 群組搜尋篩選器

指定用於尋找群組項目的搜尋篩選器。預設值爲 (objectclass=groupOfUniqueNames)。

LDAP 群組容器命名屬性

若群組存在於容器中，指定群組容器的命名屬性。否則，此屬性將爲空白。例如，若 cn=group1,ou=groups,dc=iplanet,dc=com 的群組 DN 位於 ou=groups 之中，則群組容器命名屬性爲 ou。

LDAP 群組容器值

指定群組容器值。例如，如果 cn=group1,ou=groups,dc=iplanet,dc=com 的群組 DN 位於容器名稱 ou=groups 之中，則群組容器值應爲 groups。

LDAP 群組物件類別

指定群組的物件類別。建立群組時，本群組物件類別清單將增加至群組的屬性清單。

LDAP 群組屬性

定義與群組相關聯的屬性清單。不允許讀取/寫入任何不在此清單上的群組屬性。這些屬性區分大小寫。於此處定義物件類別與屬性模式之前，必須在 Directory Server 中定義物件類別與屬性模式。

群組成員身份屬性

指定屬性名稱，其值爲 DN 所屬之所有群組的名稱。預設值爲 memberOf。

唯一成員屬性

指定屬性名稱，其值爲屬於此群組的 DN。預設值爲 uniqueMember。

群組成員 URL 屬性

指定屬性名稱，其值爲一個 LDAP URL，可解析爲此群組的成員。預設值爲 memberUrl。

LDAP 使用者容器命名屬性

使用者存在於使用者容器中時，指定使用者容器的命名屬性。若使用者並未位於使用者容器中，此欄位應爲空白。

LDAP 使用者容器值

指定使用者容器值。預設值爲 people。

LDAP 代理程式搜尋屬性

此欄位定義對代理程式進行搜尋的屬性類型。預設值爲 uid。

LDAP 代理程式容器命名屬性

若代理程式位於一個代理程式容器中，則爲代理程式容器的命名屬性。若代理程式並未位於代理程式容器中，此欄位應爲空白。

LDAP 代理程式容器值

指定代理程式容器值。預設值爲 agents。

LDAP 代理程式搜尋篩選器

定義用來搜尋代理程式的篩選器。LDAP 代理程式搜尋屬性置於此欄位之前以形成實際代理程式搜尋篩選器。

例如，若 LDAP 代理程式搜尋屬性爲 uid，而 LDAP 使用者搜尋篩選器爲 (objectClass=sunIdentityServerDevice)，則實際使用者搜尋篩選器將爲：(&(uid=*)(objectClass=sunIdentityServerDevice))

LDAP 代理程式物件類別

定義代理程式的物件類別。建立代理程式時，本使用者物件類別清單將增加至代理程式的屬性清單

LDAP 代理程式屬性

定義與代理程式相關聯的屬性清單。不允許讀取/寫入任何不在此清單上的代理程式屬性。這些屬性區分大小寫。於此處定義物件類別與屬性模式之前，必須在 Directory Server 中定義物件類別與屬性模式。

可認證的識別類型

指定當範圍的認證模組模式設定爲「資料存放區」時，資料存放區可以認證使用者及/或代理程式識別類型。

持續搜尋基底 DN

定義用於持續搜尋的基底 DN。某些 LDAPv3 伺服器僅在根尾碼層級上支援持續搜尋。

持續搜尋篩選器

定義可傳回目錄伺服器項目之特定變更的篩選器。資料存放區只會接收與定義的篩選器相符的變更。

重新啓動前持續搜尋最長閒置時間

定義重新啓動持續搜尋之前的最大閒置時間。此值必須大於 1。若值小於或等於 1，則無論連線的閒置時間爲何，皆將重新啓動搜尋。

若 Access Manager 與負載平衡器同時部署，則某些負載平衡器將在閒置一段特定時間后逾時。在此情況下，您應該將 [重新啓動前持續搜尋最長閒置時間] 設定爲一個小於負載平衡器之特定時間的值。

出現錯誤碼後的最大重試次數

定義遇到 [需要重試的 LDAPException 錯誤碼] 中指定的錯誤碼時，持續搜尋作業的最大重試次數。

重試之間的延遲時間

指定每次重試前的等待時間。僅適用於持續搜尋連線。

需要重試的 LDAPException 錯誤碼

指定需要重新啓動持續搜尋作業的錯誤碼。此屬性僅適用於持續搜尋，並不適用於所有 LDAP 作業。

快取

若啓用，則可讓 Access Manager 快取從資料存放區擷取的資料。

快取項目的最長保留時間

指定資料在被移除之前要儲存在快取中的最長時間。以秒爲單位定義值。

快取的最大大小

指定快取的最大大小。值越大，可儲存的資料越多，但也需要更多記憶體。以位元組爲單位定義值。

◆ ◆ ◆ 第 4 章

管理認證

認證服務提供一項基於 Web 的使用者介面給所有安裝於 Access Manager 部署中的預設認證模組。該介面提供動態和可自訂的方式，透過爲使用者請求的存取顯示登入需求螢幕 (基於呼叫的認證模組) 來匯集認證憑證。該介面使用 Sun Java System™ Application Framework (有時稱爲 *JATO*，它是一種 Java 2 Enterprise Edition (J2EE) 表示框架，用於協助開發者建立實用的 Web 應用程式) 建立。

配置認證

本節描述如何配置您部署的認證。第一部分略述預設認證模組類型並提供任何所需的預先配置指示。您可爲範圍、使用者、角色等的相同認證模組類型配置多重配置實例。此外，您可增加認證鏈接，如此一來，在成功認證之前，認證必須通過多重實例的條件。本節包含：

認證模組類型

認證模組是一個收集使用者資訊 (如使用者 ID 和密碼) 並根據資料庫中項目檢查資訊的外掛程式。若使用者提供的資訊符合認證條件，則將對使用者授予所請求資源的存取權。若使用者提供的資訊不符合認證條件，則將拒絕使用者存取所請求的資源。Access Manager 安裝時附有下列認證模組類型：

- 第 44 頁的 「HTTP Basic」
- 第 44 頁的 「JDBC」
- 第 44 頁的 「LDAP」
- 第 44 頁的 「成員身份」
- 第 44 頁的 「MSISDN」
- 第 45 頁的 「RADIUS」
- 第 46 頁的 「SafeWord」
- 第 47 頁的 「SAML」
- 第 47 頁的 「SecurID」
- 第 47 頁的 「UNIX」
- 第 48 頁的 「Windows Desktop SSO」
- 第 51 頁的 「Windows NT」

備註 – 用作認證實例之前，某些認證模組類型需要進行預先配置。如需要，配置步驟將列於模組類型描述之中。

核心

依預設，Access Manager 提供十五種不同的認證模組，以及核心認證模組。核心認證模組爲認證模組提供總體配置。增加及啓用 Active Directory、匿名、基於憑證的認證、HTTP Basic、JDBC、LDAP 等任何認證模組之前，必須先增加和啓用核心認證。對預設範圍自動啓用核心和 LDAP 認證兩種模組。

按一下 [進階特性] 按鈕顯示可爲範圍定義的核心認證屬性。全域屬性不適用於範圍，因此將不顯示。

Active Directory

Active Directory 認證模組執行認證的方式與 LDAP 模組相似，但使用的是 Microsoft 的 Active Directory™ 伺服器 (而 LDAP 認證模組使用的 Directory Server)。雖然可爲 Active Directory 伺服器配置 LDAP 認證模組，但此模組可讓您在相同範圍下同時擁有 LDAP 和 Active Directory 兩種認證。

備註 – 在此發行版本中，Active Directory 認證模組僅支援使用者認證。只有 LDAP 認證模組會支援密碼策略。

匿名

依預設，啓用此模組時，使用者能以 *anonymous* 使用者的身份登入 Access Manager。藉由配置 [有效匿名使用者清單] 屬性，亦可定義此模組的匿名使用者清單。授予匿名存取權意味著無需提供密碼即可進行存取。可以將匿名存取權限制爲特定類型的存取權 (例如，讀取存取權或搜尋存取權)，或限制在目錄內的子樹狀結構或個別項目中。

憑證

基於憑證的認證需要使用個人數位憑證 (personal digital certificate, PDC) 來識別和認證使用者。可以將 PDC 配置爲需要與儲存在 Directory Server 中的 PDC 相符，並要根據憑證撤銷清單進行驗證。

在對範圍加入基於憑證的認證模組之前，需要完成許多工作。首先，需要確保與 Access Manager 一同安裝之 Web 容器的安全，並對其進行配置，以用於基於憑證的認證。

備註 – 若要以由啓用了 SSL 的 Sun Java System Web Server 6.1 實例來配置 Access Manager 憑證認證，並希望將 WebServer 定義成接受基於憑證以及不基於憑證的認證請求，您必須在 WebServer 的 `obj.conf` 檔案中設定下列值：

```
PathCheck fn="get-client-cert" dorequest="1" require="0"
```

這是因爲設定這種行爲的選擇性屬性時，在 WebServer 主控台中有限制。

於啓用基於憑證的模組之前，請參閱「*Sun ONE Web Server 6.1* **管理員指南**」中的第 6 章「使用證書和金鑰」，以取得 Web Server 的初始配置步驟。此文件位於以下位置：

```
http://docs.sun.com/db/prod/s1websrv#hic
```

或者，參閱位於下列位置的「*Sun ONE Application Server Administrator's Guide to Security*」：

http://docs.sun.com/db/prod/s1appsrv#hic (`http://docs.sun.com/db/prod/s1appsrv#hic`)

備註 – 每一位要使用基於憑證的模組進行認證的使用者，必須請求用於使用者瀏覽器的 PDC。根據所使用的瀏覽器不同，會有不同的說明。請參閱您瀏覽器的說明文件，以取得更多資訊。

爲了加入此模組，您必須以範圍管理員的身份登入 Access Manager，並配置 Access Manager 和 Web 容器，以使用 SSL 並啓用用戶端認證。如需更多資訊，請參閱「*Access Manager Post Installation Guide*」中的「Configuring Access Manager in SSL Mode」。

資料存放區

「資料存放區」認證模組也允許使用範圍的「識別儲存庫」登入，以認證使用者。如果要根據相同的資料存放區儲存庫進行認證，則使用「資料存放區」模組可以不必撰寫認證外掛程式模組、載入及配置認證模組。此外，您不需要撰寫範圍中對應儲存庫需要純文字檔認證的自訂認證模組。

此認證類型在配置 Access Manager 認證時提供某些程度的方便性。在 Access Manager 7.1 之前的發行版本中，如果您想讓 LDAPv3 資料存放區中的使用者能夠認證到他們的範圍，您必須：

- 配置 LDAPv3 資料存放區
- 配置 LDAP 認證模組實例，以參照相同的範圍主體

「資料存放區」認證模組可讓定義於範圍識別儲存庫中的使用者進行認證。不需要配置 LDAP 認證。例如，假設有個範圍的識別儲存庫包含一個 LDAPv3 資料存放區，且同一個範圍使用資料存放區認證。在此情況下，定義於識別儲存庫中的任何使用者都可以認證到該範圍。

HTTP Basic

此模組使用基本認證，它是 HTTP 通訊協定內建的認證支援。Web 伺服器發出要求提供使用者名稱和密碼的用戶端請求，並將這些資訊作爲授權請求的一部分傳回伺服器。Access Manager 會擷取該使用者名稱和密碼，並從內部將使用者認證至 LDAP 認證模組。爲使 HTTP Basic 正常工作，必須加入 LDAP 認證模組 (僅加入 HTTP Basic 模組將不起作用)。一旦使用者認證成功，其無需提供使用者名稱和密碼即可重新進行認證。

JDBC

Java Database Connectivity (JDBC) 認證模組提供一種機制，可讓 Access Manager 經由提供 JDBC 技術啓用驅動程式的 SQL 資料庫來認證使用者。SQL 資料庫的連線可以直接透過 JDBC 驅動程式或透過 JNDI 連線池。

備註 – 此模組已在 MySQL4.0 和 Oracle 8i 上通過測試。

LDAP

如果使用 LDAP 認證模組，當使用者登入時，他們必須以特定的使用者 DN 和密碼連結至 LDAP Directory Server。此爲所有基於範圍的認證之預設認證模組。若使用者提供 Directory Server 中的使用者 ID 和密碼，系統將允許此使用者存取有效的 Access Manager 階段作業，並使用該階段作業進行設定。對預設範圍自動啓用核心和 LDAP 認證兩種模組。

成員身份

成員身份認證的實作類似於個人化網站，例如：`my.site.com` 或 `mysun.sun.com`。啓用此模組時，使用者無需借助管理員，即可建立帳號並對其進行個人化設定。對於這個新帳號，使用者能以已加入使用者的身份來存取它。還可以存取檢視器介面，此介面作爲授權資料和使用者偏好設定儲存在使用者設定檔資料庫中。

MSISDN

Mobile Station Integrated Services Digital Network (MSISDN) 認證模組會使用如行動電話等裝置相關的行動用戶 ISDN 來啓用認證。這是非互動式模組。此模組擷取用戶 ISDN 並根據 Directory Server 進行驗證，以找到符合該號碼的使用者。

RADIUS

Access Manager 可以配置爲搭配已安裝的 RADIUS 伺服器使用。如果您的企業使用舊有的 RADIUS 伺服器進行認證，這會很有用。啓用 RADIUS 認證模組需要執行兩個步驟：

1. 配置 RADIUS 伺服器。

   如需詳細說明，請參閱 RADIUS 伺服器的文件。

2. 註冊和啓用 RADIUS 認證模組。

與 Sun Java System Application Server 一起配置 RADIUS

RADUIS 用戶端與其伺服器形成通訊端連線時，依預設，Application Server 的 `server.policy` 檔案中，只允許有 SocketPermissions 的連線權限。爲了使 RADUIS 認證正常工作，需要爲以下動作授予權限：

- 接受
- 連線
- 偵聽
- 解析

若要授予通訊端連線的權限，您必須在 Application Server 的 `server.policy` 檔案中加入一個項目。SocketPermission 由主機規格和一組指定與該主機連線方式的動作組成。主機依如下方式指定：

```
host = hostname | IPaddress:portrange:portrange = portnumber
| -portnumberportnumber-portnumber
```

主機表示爲 DNS 名稱、數字 IP 位址或本地主機 (針對本端機器)。DNS 名稱主機規格中可使用一次萬用字元「`*`」。如果包含萬用字元，它必須位於最左側，
如：`*.example.com`。

連接埠 (或連接埠範圍) 爲選擇性的。形式爲 `N-` 的連接埠規格 (其中 `N` 爲連接埠號碼)，表示號碼爲 `N` 及 `N` 以上的所有連接埠。形式爲 `-N` 的連接埠規格則表示號碼爲 `N` 及 `N` 以下的所有連接埠。

偵聽動作僅在與本地主機搭配使用時才有意義。如果存在任何其他動作，則暗含**解析** (解析主機/IP 名稱服務查找) 動作。

例如，建立 SocketPermissions 時請注意，如果將以下權限授予某程式碼，則該權限可讓程式碼與 `machine1.example.com` 上的 `port 1645` 連線，並接受該連接埠上的連線：

```
permission java.net.SocketPermission machine1.example.com:1645, "connect,accept";
```

同樣，如果將以下權限授予某程式碼，則該權限可讓程式碼接受本地主機上 1024 至 65535 之間任一連接埠上的連線，並可連線或偵聽這些連接埠：

```
permission java.net.SocketPermission "machine1.example.com:1645", "connect,accept";
permission java.net.SocketPermission "localhost:1024-", "accept,connect,listen";
```

備註 – 因爲有害的程式碼可以更容易在不擁有資料存取權的多方中傳輸和共用這些資料，所以將接受或建立與遠端主機連線的權限授予程式碼可能會引發問題。請確保僅透過指定精確的連接埠號 (而不是指定連接埠號範圍) 授予適當的權限。

SafeWord

可配置 Access Manager 以處理對安全運算的 SafeWord™ 或 SafeWord PremierAccess™ 認證伺服器的 SafeWord 認證請求。Access Manager 會提供 SafeWord 認證的用戶端。SafeWord 伺服器可以存在於安裝有 Access Manager 的系統，或是單獨的系統上。

與 Sun Java System Application Server 一起配置 SafeWord

SafeWord 用戶端與其伺服器形成通訊端連線時，依預設，Application Server 的 `server.policy` 檔案中，只允許有 SocketPermissions 的 `connect` 權限。爲了使 SafeWord 認證正常工作，需要爲以下動作授予權限：

- 接受
- 連線
- 偵聽
- 解析

若要授予通訊端連線的權限，您必須在 Application Server 的 `server.policy` 檔案中加入一個項目。SocketPermission 由主機規格和一組指定與該主機連線方式的動作組成。主機依如下方式指定：

```
host = (hostname | IPaddress)[:portrange] portrange =
portnumber | -portnumberportnumber-[portnumber]
```

主機表示爲 DNS 名稱、數字 IP 位址或本地主機 (針對本端機器)。DNS 名稱主機規格中可使用一次萬用字元「`*`」。如果包含萬用字元，它必須位於最左側，如：`*.example.com`。

連接埠 (或 portrange) 爲選擇性的。形式爲 `N-` 的連接埠規格 (其中 `N` 爲連接埠號碼)，表示號碼爲 `N` 及 `N` 以上的所有連接埠。形式爲 `-N` 的規格則表示號碼爲 `N` 及 `N` 以下的所有連接埠。

偵聽動作僅在與本地主機搭配使用時才有意義。如果存在任何其他動作，則暗含**解析** (解析主機/IP 名稱服務查找) 動作。

例如，建立 SocketPermissions 時請注意，如果將以下權限授予某程式碼，則該權限可讓程式碼與 `machine1.example.com` 上的**連接埠** `1645` 連線，並接受該連接埠上的連線：

```
permission java.net.SocketPermission machine1.example.com:5030, "connect,accept";
```

同樣，如果將以下權限授予某程式碼，則該權限可讓程式碼接受本地主機上 1024 至 65535 之間任一連接埠上的連線，並可連線或偵聽這些連接埠：

```
permission java.net.SocketPermission "machine1.example.com:5030", "connect,accept";
permission java.net.SocketPermission "localhost:1024-", "accept,connect,listen";
```

備註 – 因爲有害的程式碼可以更容易在不擁有資料存取權的多方中傳輸和共用這些資料，所以將接受或建立與遠端主機連線的權限授予程式碼可能會引發問題。請確保僅透過指定精確的連接埠號 (而不是指定連接埠號範圍) 授予適當的權限。

SAML

安全指定標記語言 (Security Assertion Markup Language, SAML) 認證模組接收並驗證目標伺服器上的 SAML 指定。只有在此模組是配置於目標機器上時 (包括升級後，例如：Access Manager 2005Q4 升級至 Access Manager 7.1)，SAML SSO 才有作用。

SecurID

Access Manager 可以配置爲能處理對 RSA 的 ACE/Server 認證伺服器提出之「SecurID 認證」請求。Access Manager 會提供 SecurID 認證的用戶端。ACE/Server 可以存在於安裝有 Access Manager 的系統，或是單獨的系統上。若要對本機管理的使用者 ID 進行認證 (請參閱 `admintool` (1M))，則需要超級使用者存取權限。

「SecurID 認證」使用認證**輔助程式** `amsecuridd`，它是 Access Manager 主程序以外的單獨程序。此輔助程式會在啓動時偵聽某連接埠，以取得配置資訊。若安裝了 Access Manager 並以 `nobody` 的身份或超級使用者以外的使用者 ID 執行，則仍必須以超級使用者的身份執行 *AccessManager-base*`/SUNWam/share/bin/amsecuridd` 程序。如需關於 `amsecuridd` 輔助程式的更多資訊，請參閱「Access Manager Administration Reference」中的「The amSecurID Helper」。

備註 – 在此版本的 Access Manager 中，「SecurID 認證」模組不適用於 Linux 或 Solaris x86 平台，且不應在這兩個平台上註冊、配置或啓用。它僅適用於 SPARC 系統。

UNIX

Access Manager 可以配置爲根據安裝有 Access Manager 的 Solaris 或 Linux 系統上已知的 Unix 使用者 ID 和密碼，處理認證請求。雖然 Unix 認證只有一個範圍屬性和幾個全域屬性，但仍有一些針對系統的考量。若要對在本機管理的使用者 ID 進行認證 (請參閱 `admintool` (1M))，需要超級使用者存取權限。

「Unix 認證」使用認證**輔助程式** `amunixd`，它是 Access Manager 主程序以外的單獨程序。此輔助程式會在啓動時偵聽某連接埠，以取得配置資訊。每個 Access Manager 只有一個 Unix 輔助程式以供其所有範圍使用。

如果安裝了 Access Manager 並以 nobody 身份或非超級使用者的使用者 ID 執行，必須仍以超級使用者身份執行 *AccessManager-base*/SUNWam/share/bin/amunixd 程序。Unix 認證模組透過開啓 localhost:58946 的通訊端來呼叫 amunixd 常駐程式，以偵聽 Unix 認證請求。若要在預設連接埠上執行 amunixd 輔助程式程序，請輸入以下指令：

```
./amunixd
```

若要在非預設連接埠上執行 amunixd，請輸入以下指令：

```
./amunixd [-c portnm] [ipaddress]
```

IP 位址與連接埠埠號位於 AMConfig.properties 的 UnixHelper.ipadrs 屬性 (IPV4 格式) 和 UnixHelper.port 屬性中。您可透過 amserver 指令行公用程式執行 amunixd (amserver 會自動執行此程序，從 AMConfig.properties 擷取連接埠號和 IP 位址)。

/etc/nsswitch.conf 檔案中的 passwd 項目會決定是參考 /etc/passwd 和 /etc/shadow 檔案，還是參考 NIS 來進行認證。

Windows Desktop SSO

「Windows Desktop SSO 認證」模組是以 Kerberos 爲基礎的認證外掛程式模組，用於 Windows 2000™。它可讓通過 Kerberos 分發中心 (Kerberos Distribution Center, KDC) 認證的使用者，毋需再次提交登入條件便可通過 Access Manager 的認證 (單次登入)。

使用者透過 SPNEGO (Simple and Protected GSS-API Negotiation Mechanism，簡單和受保護的 GSS-API 協商機制) 通訊協定向 Access Manager 提供 Kerberos 記號。爲了經由此認證模組來執行基於 Kerberos 的單次登入 Access Manager，在用戶端的使用者必須支援 SPNEGO 通訊協定，才能自我認證。通常，任何支援此通訊協定的使用者皆應可使用這個模組以進行 Access Manager 認證。視用戶端記號的可用性而定，此模組會提供 SPENGO 記號或 Kerberos 記號 (不論那一個，通訊協定都相同)。於 Windows 2000 (或更新版本) 上執行的 Microsoft Internet Explorer (5.01 或更新版本) 目前支援此通訊協定。此外，Solaris (9 和 10) 上的 Mozilla 1.4 具有 SPNEGO 支援，但僅會傳回 KERBEROS 記號，因爲 Solaris 不支援 SPNEGO。

備註 – 您必須使用 JDK 1.4 或更新版本，才能利用 Kerberos V5 認證模組的新功能和 Java GSS API，在此 SPNEGO 模組中執行基於 Kerberos 的 SSO。

Internet Explorer 的已知限制

若在 WindowsDesktopSSO 認證時使用的是 Microsoft Internet Explorer 6.x，且瀏覽器不具使用者的 kerberos/SPNEGO 記號 (符合 WindowsDesktopSSO 模組中配置的 (KDC) 範圍) 之存取權，則在瀏覽器向 WindowsDesktopSSO 模組的認證失敗後，瀏覽器對其他模組的運作也會不正確。導致此問題的直接原因在於當 Internet Explorer 向 WindowsDesktopSSO 模組認證失敗後，即使出現回呼的提示，瀏覽器也無法將回呼 (屬於其他模組) 傳遞至 Access Manager，除非瀏覽器重新啓動。由於使用者憑證爲空，因此 WindowsDesktopSSO 之後的所有模組都將失敗。

請參閱下列文件以取得相關資訊：

http://support.microsoft.com/default.aspx?scid=kb;en-us;308074
(http://support.microsoft.com/default.aspx?scid=kb;en-us;308074)

http://www.wedgetail.com/jcsi/sso/doc/guide/troubleshooting.html#ieNTLM
(http://support.microsoft.com/default.aspx?scid=kb;en-us;308074)

備註 – 截至此 Access Manager 發行版本，Microsoft 已修正這個限制。如需更多資訊，請參閱下列文件：

http:// http://www.microsoft.com/technet/security/bulletin/ms06-042.mspx

---

配置 Windows Desktop SSO

啓用 Windows Desktop SSO 認證是一個具有兩個步驟的程序：

1. 在 Windows 2000 網域控制器中建立一個使用者
2. 設定 Internet Explorer。

▼ 要在 Windows 2000 網域控制器中建立一個使用者

1 在網域控制器中，為 Access Manager 認證模組建立使用者帳號。

a. 從 [開始] 功能表移至 [程式集] > [管理工具]。

b. 選取 [使用者與電腦]。

c. 移至 [電腦] > [新增] > [電腦]，並增加用戶端電腦的名稱。如果您使用 Windows XP，則會在配置網域控制站帳號時自動執行這個步驟。

d. 移至 [使用者] > [新增] > [使用者]，並以 Access Manager 主機名稱做為使用者 ID (登入名稱) 建立新使用者。Access Manager 主機名稱不應包含網域名稱。

2 將使用者帳號與服務提供者名稱產生關聯，並將 keytab 檔案匯出至安裝 Access Manager 的系統。若要進行上述動作，請執行下列指令：

```
ktpass -princ host/hostname.domainname@DCDOMAIN -pass password -mapuser userName-out
hostname.host.keytab
ktpass -princ HTTP/hostname.domainname@DCDOMAIN -pass
password -mapuser userName-out hostname
.HTTP.keytab
```

備註 – ktpass 公用程式不作爲 Windows 2000 伺服器的一部分安裝。您必須從安裝光碟將它安裝到 c:\program files\support 工具目錄。

ktpass 指令接受下列參數：

hostname。執行 Access Manager 的主機名稱 (不含網域名稱)。

domainname。Access Manager 網域名稱。

DCDOMAIN。網域控制器的網域名稱。此名稱可能與 Access Manager 的網域名稱不同。

password。使用者帳號的密碼。請確定密碼正確，因爲 ktpass 不會驗證密碼。

userName。使用者帳號 ID。它應該與 hostname 相同。

備註 – 請確保兩個 keytab 檔案均已做好安全措施。

服務範本值應類似於以下範例：

服務主體： HTTP/machine1.EXAMPLE.COM@ISQA.EXAMPLE.COM

Keytab 檔案名稱：/tmp/machine1.HTTP.keytab

Kerberos 範圍： ISQA.EXAMPLE.COM

Kerberos 伺服器名稱：machine2.EXAMPLE.com

使用網域名稱傳回委託人：false

認證層級：22

備註 – 如果您使用 Windows 2003 或 Windows 2003 Service Pack，請使用下列 ktpass 指令語法：

```
ktpass /out filename /mapuser username /princ HTTP/hostname.domainname
     /crypto encryptiontype /rndpass /ptype principaltype /target domainname
```

例如：

```
ktpass /out demo.HTTP.keytab /mapuser http
/princ HTTP/demo.identity.sun.com@IDENTITY.SUN.COM /crypto RC4-HMAC-NT
/rndpass /ptype KRB5_NT_PRINCIPAL /target IDENTITY.SUN.COM
```

如需語法定義，請參閱 http://technet2.microsoft.com/WindowsServer/en/library/64042138-9a5a-4981-84e9-d576a8db0d051033.mspx?mfr=true網站。

3 **重新啓動伺服器。**

▼ 設定 Internet Explorer

上述步驟適用於 Microsoft Internet Explorer™ 6 及更高版本。若您是使用較早的版本，請確定 Access Manager 在瀏覽器的網際網路區域內，並啓用本機 Windows 認證。

1 **在 [工具] 功能表中，移至 [網際網路選項] > [進階/安全性] > [安全性]。**

2 **選取 [整合 Windows 認證] 選項。**

3 **移至 [安全性] > [本機網際網路]。**

a. **選取 [自訂層級]。在 [使用者驗證/登入] 面板中，選取 [只在企業內部網路區域自動登入] 選項。**

b. **前往 [網站] 並選取所有選項。**

c. **按一下 [進階]，並將 Access Manager 加入至本機區域 (若尚未加入的話)。**

Windows NT

Access Manager 可以配置爲搭配已安裝的 Windows NT/Windows 2000 伺服器使用。Access Manager 會提供 NT 認證的用戶端部分。

1. 配置 NT 伺服器。如需詳細說明，請參閱 Windows NT 伺服器的文件。
2. 加入和啓用 Windows NT 認證模組之前，您必須先取得和安裝 Samba 用戶端，以便與 Solaris 系統上的 Access Manager 進行通訊。

安裝 Samba Client

若要啓動 Windows NT 認證模組，必須下載 Samba Client 2.2.2，並將之安裝至下列目錄：

```
AccessManager-base/SUNWam/bin
```

Samba Client 是一種檔案與列印伺服器，用於將 Windows 和 UNIX 機器結合在一起，而無需單獨的 Windows NT/2000 Server。如需更多資訊及下載該軟體，請移至：http://wwws.sun.com/software/download/products/3e3af224.html。

Red Hat Linux 隨附 Samba 用戶端，位於下列目錄：

```
/usr/bin
```

若要使用 Linux 的 Windows NT 認證模組進行認證，將用戶端二進位檔案複製到下列 Access Manager 目錄中：

AccessManager-base/sun/identity/bin

備註 – 如果您有多個介面，則需要額外的配置。多重介面可以透過 smb.conf 檔案中的配置設定，以傳遞到 mbclient。

認證模組實例

根據預設認證模組，可爲範圍建立多重認證模組實例。您可增加相同認證模組之個別配置的多重實例。

▼ 建立新的認證模組實例

1. **按一下您要增加認證模組實例的範圍名稱。**
2. **選取 [認證] 標籤。**

備註 – [管理員認證配置] 按鈕僅定義管理員的認證服務。如果需要管理員的認證模組與一般使用者的認證模組有所不同，則可以使用此屬性。在訪問 Access Manager 主控台時，將使用該屬性中配置的模組。

---

3. **按一下 [模組實例] 清單中的 [新建]。**
4. **輸入認證模組實例的名稱。名稱必須唯一。**
5. **選取範圍之認證模組類型的 [類型]。**
6. **按一下 [建立]。**
7. **按一下剛建立的模組實例名稱並編輯該模組的特性。請參閱線上說明中的「認證」一節，以取得每個模組類型特性的定義。**
8. **重複這些步驟以增加多重模組實例。**

認證鏈接

可以配置一個以上的認證模組，使用者必須傳送認證憑證給這些模組。這稱爲**認證鏈接**。Access Manager 中的認證鏈接是使用整合於認證服務中的 JAAS 框架來達成。

▼ 建立新的認證鏈接

1. 按一下您要增加認證鏈接的範圍名稱。
2. 選取 [認證] 標籤。
3. 按一下 [認證鏈接] 清單中的 [新建]。
4. 輸入此認證鏈接的名稱。
5. 按一下 [建立]。
6. 按一下 [增加] 以定義您要包括於鏈接中的認證模組實例。若要這麼做，請由 [實例] 清單中選取模組實例名稱。 此清單中顯示的模組實例名稱於模組實例屬性中建立。
7. 選取鏈接的條件。這些旗標為其定義的認證模組建立實施條件。此實施具有階層結構。[必要的] 為最高階層而 [可選的] 為最低階層：

必要條件	模組實例必須成功。若成功，認證將繼續進行至 [認證鏈接] 清單中的下一個選項。如果失敗，立即返回應用程式 (不會繼續認證鏈接清單中的下一個選項)。
必要的	此模組的認證過程必須成功。若鏈接中任一必要的模組失敗了，則整個認證鏈接將最終失敗。然而，無論必要的模組成功與否，將繼續進行至鏈接中的下一個模組。
充足的	模組實例不必要成功。若其確實成功，立即返回到應用程式 (認證將不進行至模組實例清單的下一個選項)。若失敗，認證將繼續進行至 [認證鏈接] 清單中的下一個選項。
可選的	模組實例不必要成功。無論成功或失敗，認證都將繼續進行至 [認證鏈接] 清單中的下一個選項。

8. 輸入鏈接的選項。允許此模組使用的其他選項，格式為「鍵=值」對。多重選項由空格分隔。
9. 定義下列屬性：

成功登入 URL	指定認證成功後將使用者重新導向至的 URL。
登入失敗 URL	指定認證失敗後將使用者重新導向至的 URL。
認證發佈處理類別	定義在登入成功或失敗後用來自訂認證後程序的 Java 類別名稱。

10. 按 [儲存]。

認證類型

認證服務提供不同的方式讓認證套用。這些不同的認證方法可藉由指定登入 URL 參數或透過認證 API 來獲取 (請參閱開發者指南「Sun Java System Access Manager 7.1 Developer's Guide」中的第 2 章「Using Authentication APIs and SPIs」以取得更多資訊)。配置認證模組之前，必須先修改核心認證服務屬性 [範圍認證模組]，使之包括特定的認證模組名稱。

認證配置服務用於爲以下任一認證類型定義認證模組：

- 第 56 頁的「基於範圍的認證」
- 第 58 頁的「基於組織的認證」
- 第 60 頁的「基於角色的認證」
- 第 63 頁的「基於服務的認證」
- 第 65 頁的「基於使用者的認證」
- 第 67 頁的「基於認證層級的認證」
- 第 70 頁的「基於模組的認證」

爲這些認證類型之一定義認證模組後，便可以將此模組配置爲根據認證程序成敗提供重新導向 URL 以及處理後的 Java 類別規格。

認證類型決定存取的方式

這些方法的每一種，使用者認證都可能通過或失敗。一旦確定，每種方法都會依照此程序。步驟 1 至步驟 3 在成功認證後執行；步驟 4 在認證成功或失敗後執行。

1. Access Manager 會確認所認證的使用者是否在 Directory Server 資料存放區中定義，以及設定檔是否在使用中。

   核心認證模組中的使用者設定檔屬性可以定義爲**必需**、**動態**、**隨使用者別名動態變化**或**忽略**。認證成功之後，Access Manager 會確認 Directory Server 資料存放區中是否定義了所認證的使用者，並且如果使用者設定檔值爲**必需**，再確認設定檔是否在使用中。這是預設情形。如果使用者設定檔爲**動態配置**，認證服務將會在 Directory Server 資料存放區中建立使用者設定檔。若使用者設定檔設定爲**忽略**，將不會完成使用者驗證。

2. 認證處理後 SPI 的執行完成。

   核心認證模組包含認證處理後類別屬性，其中可能包含認證處理後類別名稱作爲其值。`AMPostAuthProcessInterface` 是處理後介面。它可以在成功或失敗的認證或登出時執行。

3. 下列特性會加入階段作業記號，或在階段作業記號中更新，而使用者的階段作業會啓動。

   realm。這是使用者歸屬的範圍 DN。

   Principal。這是使用者的 DN。

Principals。這是使用者已認證過的名稱清單。該特性可以有多個值，值之間以管道符號分隔。

UserId。這是模組傳回的使用者 DN，在非 LDAP 或成員模組的情況下，爲使用者名稱。(所有主體都必須對映到相同的使用者。UserID 爲它們所對映之使用者 DN。)

備註 – 此特性可爲非 DN 值。

UserToken。這是使用者名稱。(所有主體都必須對映到相同的使用者。UserToken 爲它們所對映之使用者名稱。)

Host。這是用戶端的主機名稱或是 IP 位址。

authLevel。這是使用者已認證過的最高層級。

AuthType。這是已認證使用者的認證模組清單，以管道符號分隔 (例如，`module1|module2|module3`)。

clientType。這是用戶端瀏覽器的裝置類型。

Locale。這是用戶端的語言環境。

CharSet。這是爲用戶端確定的字元集。

Role。僅適用於基於角色的認證，此爲使用者歸屬的角色。

Service。僅適用於基於服務的認證，此爲使用者歸屬的服務。

4. 在成功或失敗認證後，Access Manager 會尋找資訊以確定將使用者重新導向至何處。

   URL 重新導向的位置可以是 Access Manager 頁面或 URL。重新導向會依據優先順序進行，Access Manager 會根據認證方法及認證是否已成功或已失敗，依此優先順序尋找重新導向。此順序詳述於下列認證方法章節的 URL 重新導向部分。

URL 重新導向

在認證配置服務中，您可以爲成功或失敗的認證指定 URL 重新導向。URL 本身在此服務的登入成功 URL 和登入失敗 URL 屬性中定義。爲了啓用 URL 重新導向，您必須將認證配置服務加入您的範圍，使之可用於角色、範圍或使用者的配置。在加入認證配置服務時，請確定您加入的是認證模組，例如 LDAP - REQUIRED。

基於範圍的認證

此認證方法可讓使用者向範圍或子範圍進行認證。此爲 Access Manager 的預設認證方法。範圍的認證方法是透過對範圍註冊核心認證模組，並定義範圍認證配置屬性來設定的。

基於範圍的認證登入 URL

藉由定義 realm 參數或 domain 參數，可於使用者介面登入 URL 中指定認證的範圍。由下列項目決定認證請求的範圍，其優先順序爲：

1. domain 參數。
2. realm 參數。
3. 管理服務中的 DNS 別名屬性值。

   於呼叫正確的範圍後，將從核心認證服務的範圍認證配置屬性擷取將認證使用者的認證模組。用來指定並啓動基於範圍的認證的登入 URL 爲：

   ```
   http://server_name.domain_name:port/amserver/UI/Login
   http://server_name.domain_name:port/amserver/UI/Login?domain=domain_name
   http://server_name.domain_name:port/amserver/UI/Login?realm=realm_name
   ```

   若無定義的參數，則將由登入 URL 中指定的伺服器主機和網域決定範圍。

備註 – 如果使用者是特定範圍的成員，且經過該範圍認證，接著嘗試向其他範圍認證，則只傳送 realm 和 module 兩個參數。例如，如果 User1 是 realmA 的成員，並認證至該範圍，後來想嘗試切換或認證至 realmB，則使用者會收到警告頁面，要求他們使用針對 realmB 指定的模組實例啓動向 realmB 的新認證，或返回到現有經過驗證的 realmA 階段作業。如果使用者認證到 realmB，則只會傳送及利用範圍名稱及模組名稱 (若有指定) 來判斷新的認證程序。

基於範圍的認證重新導向 URL

在基於組織的認證成功或失敗後，Access Manager 會尋找資訊以確定將使用者重新導向至何處。以下爲應用程式尋找此資訊的優先順序。

成功的基於範圍的認證重新導向 URL

成功的基於範圍的認證重新導向 URL 是依優先順序檢查下列位置來決定：

1. 認證模組設定的 URL。
2. goto 登入 URL 參數設定的 URL。
3. clientType 自訂檔案中爲使用者設定檔 (amUser.xml) 的 iplanet-am-user-success-url 屬性設定的 URL。

4. clientType 自訂檔案中爲使用者角色項目的 iplanet-am-auth-login-success-url 屬性設定的 URL。
5. clientType 自訂檔案中爲使用者範圍項目的 iplanet-am-auth-login-success-url 屬性設定的 URL。
6. clientType 自訂檔案中爲 iplanet-am-auth-login-success-url 屬性做爲全域預設值設定的 URL。
7. 設定於使用者設定檔 (amUser.xml) 之 iplanet-am-user-success-url 屬性中的 URL。
8. 使用者角色項目之 iplanet-am-auth-login-success-url 屬性中設定的 URL。
9. 設定於使用者範圍項目之 iplanet-am-auth-login-success-url 屬性中的 URL。
10. iplanet-am-auth-login-success-url 屬性中設定的 URL，作爲全域預設值。

失敗的基於範圍的認證重新導向 URL

失敗的基於範圍的認證重新導向 URL 是以下列順序檢查下列位置來決定：

1. 認證模組設定的 URL。
2. gotoOnFail 登入 URL 參數設定的 URL。
3. clientType 自訂檔案中爲使用者項目 (amUser.xml) 的 iplanet-am-user-failure-url 屬性設定的 URL。
4. clientType 自訂檔案中爲使用者角色項目的 iplanet-am-auth-login-failure-url 屬性設定的 URL。
5. clientType 自訂檔案中爲使用者範圍項目的 iplanet-am-auth-login-failure-url 屬性設定的 URL。
6. clientType 自訂檔案中爲 iplanet-am-auth-login-failure-url 屬性做爲全域預設值設定的 URL。
7. 於使用者項目 (amUser.xml) 中設定 iplanet-am-user-failure-url 屬性的 URL。
8. 針對使用者角色項目之 iplanet-am-auth-login-failure-url 屬性設定的 URL。
9. 設定使用者範圍項目之 iplanet-am-auth-login-failure-url 屬性的 URL。
10. 針對 iplanet-am-auth-login-failure-url 屬性設定的 URL，作爲全域預設值。

若要配置基於範圍的認證

要爲範圍設定認證模組，先爲範圍增加核心認證服務。

▼ 若要配置範圍的認證屬性

1 **瀏覽至您要增加認證鏈接的範圍。**

2 **按一下 [認證] 標籤。**

3. **選取 [預設認證鏈接]。**

4. **由下拉式功能表選取 [管理員認證鏈接]。如果需要管理員的認證模組與一般使用者的認證模組有所不同，則可以使用此屬性。預設認證模組為 LDAP。**

5. **定義了認證鏈接之後，按一下 [儲存]。**

基於組織的認證

此認證類型僅可套用至以「舊有」模式安裝的 Access Manager 部署。

此認證方法可讓使用者向一個組織或子組織認證。它是 Access Manager 的預設認證方法。組織的認證方法是透過對組織註冊核心認證模組，並定義組織認證配置屬性來設定的。

基於組織的認證登入 URL

藉由定義 `org` 參數或 `domain` 參數，可以在使用者介面登入 URL 中指定認證的組織。由下列項目決定認證請求的組織，優先順序爲：

1. `domain` 參數。
2. `org` 參數。
3. 管理服務中 `DNS` **別名** (組織別名) 屬性的值。

   在呼叫正確的組織後，會從核心認證服務的組織認證配置屬性擷取將認證使用者的認證模組。用來指定並啓動基於組織的認證的登入 URL 爲：

   ```
   http://server_name.domain_name:port/amserver/UI/Login
   http://server_name.domain_name:port/amserver/UI/Login?domain=domain_name
   http://server_name.domain_name:port/amserver/UI/Login?org=org_name
   ```

   若無定義的參數，則將由登入 URL 中指定的伺服器主機和網域決定組織。

備註 – 如果使用者是特定組織的成員，且經過該組織認證，接著嘗試向其他組織認證，則只傳送 `org` 和 `module` 兩個參數。例如，如果 `User1` 是 `orgA` 的成員，並認證至該組織，後來想嘗試切換或認證至 `orgB`，則使用者會收到警告頁面，要求他們使用針對 `orgB` 指定的模組實例啓動向 `orgB` 的新認證，或返回到現有經過驗證的 `orgA` 階段作業。如果使用者認證到 `orgB`，則只會傳送及利用組織名稱及模組名稱 (若有指定) 來判斷新的認證程序。

基於組織的認證重新導向 URL

在基於組織的認證成功或失敗後，Access Manager 會尋找資訊以確定將使用者重新導向至何處。以下爲應用程式尋找此資訊的優先順序。

成功的基於組織的認證重新導向 URL

成功的基於組織的認證，其重新導向 URL 是以此優先順序檢查下列位置決定的：

1. 認證模組設定的 URL。
2. `goto` 登入 URL 參數設定的 URL。
3. `clientType` 自訂檔案中爲使用者設定檔 (`amUser.xml`) 的 `iplanet-am-user-success-url` 屬性設定的 URL。
4. `clientType` 自訂檔案中爲使用者角色項目的 `iplanet-am-auth-login-success-url` 屬性設定的 URL。
5. `clientType` 自訂檔案中爲使用者組織項目的 `iplanet-am-auth-login-success-url` 屬性設定的 URL。
6. `clientType` 自訂檔案中爲 `iplanet-am-auth-login-success-url` 屬性做爲全域預設值設定的 URL。
7. 設定於使用者設定檔 (`amUser.xml`) 之 `iplanet-am-user-success-url` 屬性中的 URL。
8. 使用者角色項目之 `iplanet-am-auth-login-success-url` 屬性中設定的 URL。
9. 使用者組織項目之 `iplanet-am-auth-login-success-url` 屬性中設定的 URL。
10. `iplanet-am-auth-login-success-url` 屬性中設定的 URL，作爲全域預設值。

失敗的基於組織的認證重新導向 URL

失敗的基於組織的認證，其重新導向 URL 是以此順序檢查下列位置決定的：

1. 認證模組設定的 URL。
2. `gotoOnFail` 登入 URL 參數設定的 URL。
3. `clientType` 自訂檔案中爲使用者項目 (`amUser.xml`) 的 `iplanet-am-user-failure-url` 屬性設定的 URL。
4. `clientType` 自訂檔案中爲使用者角色項目的 `iplanet-am-auth-login-failure-url` 屬性設定的 URL。
5. `clientType` 自訂檔案中爲使用者組織項目的 `iplanet-am-auth-login-failure-url` 屬性設定的 URL。
6. `clientType` 自訂檔案中爲 `iplanet-am-auth-login-failure-url` 屬性做爲全域預設值設定的 URL。
7. 於使用者項目 (`amUser.xml`) 中設定 `iplanet-am-user-failure-url` 屬性的 URL。
8. 針對使用者角色項目之 `iplanet-am-auth-login-failure-url` 屬性設定的 URL。

9. 針對使用者組織項目之 iplanet-am-auth-login-failure-url 屬性設定的 URL。
10. 針對 iplanet-am-auth-login-failure-url 屬性設定的 URL，作爲全域預設值。

若要配置基於組織的認證

要爲組織設定認證模組，先爲組織加入核心認證服務。

▼ 若要配置組織的認證屬性

1 **瀏覽至您要增加認證鏈接的組織。**

2 **按一下 [認證] 標籤。**

3 **選取 [預設認證鏈接]。**

4 **由下拉式功能表選取 [管理員認證鏈接]。如果需要管理員的認證模組與一般使用者的認證模組有所不同，則可以使用此屬性。預設認證模組為 LDAP。**

5 **定義了認證鏈接之後，按一下 [儲存]。**

基於角色的認證

此認證方法可讓使用者向組織或是子組織之中的角色 (靜態或篩選) 進行認證 。

備註 – 在認證配置服務可做爲實例註冊到角色之前，必須先註冊至範圍中。

若要成功認證，使用者必須屬於該角色，並且必須認證到爲該角色配置的認證配置服務實例中定義的每個模組。對每個基於角色的認證之實例，可指定下列屬性：

衝突解決層級。這爲認證配置服務實例 (針對可能包含相同使用者的不同角色所定義) 設定優先層級。例如，若將 User1 指定給 Role1 和 Role2 兩者，則可對 Role1 設定較高的衝突解決層級；因此當使用者嘗試認證時，Role1 將具有較高的優先順序以處理成功或失敗的重新導向及後認證程序。

認證配置。這會定義針對角色的認證程序配置之認證模組。

登入成功 URL。此項定義在成功認證上重新導向使用者的 URL。

登入失敗 URL。此項定義在失敗認證上重新導向使用者的 URL。

認證處理後類別。此將定義後認證介面。

基於角色的認證登入 URL

透過定義角色參數，可以在使用者介面登入 URL 中指定基於角色的認證。在呼叫正確的角色後，會從爲角色定義的認證配置服務實例擷取將認證使用者的認證模組。

用於指定和啓動基於角色的認證的登入 URL 爲：

```
http://server_name.domain_name:port/amserver/UI/Login?role=role_name
http://server_name.domain_name:port/amserver/UI/Login?realm=realm_name&role=role_name
```

若無配置的範圍參數，則將由登入 URL 中指定的伺服器主機和網域決定角色所屬的範圍。

基於角色的認證重新導向 URL

在基於角色的認證成功或失敗後，Access Manager 會尋找資訊以確定將使用者重新導向至何處。以下爲應用程式尋找此資訊的優先順序。

成功的基於角色的認證重新導向 URL

成功的基於角色的認證，其重新導向 URL 是以此順序檢查下列位置決定的：

1. 認證模組設定的 URL。
2. `goto` 登入 URL 參數設定的 URL。
3. `clientType` 自訂檔案中爲使用者設定檔 (`amUser.xml`) 的 `iplanet-am-user-success-url` 屬性設定的 URL。
4. `clientType` 自訂檔案中爲使用者已認證至的角色之 `iplanet-am-auth-login-success-url` 屬性設定的 URL。
5. `clientType` 自訂檔案中爲已認證使用者的另一個角色項目之 `iplanet-am-auth-login-success-url` 屬性設定的 URL。(如果前一個重新導向 URL 失敗，此選項爲備用。)
6. `clientType` 自訂檔案中爲使用者範圍項目的 `iplanet-am-auth-login-success-url` 屬性設定的 URL。
7. `clientType` 自訂檔案中爲 `iplanet-am-auth-login-success-url` 屬性做爲全域預設值設定的 URL。
8. 設定於使用者設定檔 (`amUser.xml`) 之 `iplanet-am-user-success-url` 屬性中的 URL。
9. 已對其認證使用者的角色之 `iplanet-am-auth-login-success-url` 屬性中設定的 URL。
10. 已認證使用者另一個角色項目之 `iplanet-am-auth-login-success-url` 屬性中設定的 URL。(如果前一個重新導向 URL 失敗，此選項爲備用。)
11. 設定於使用者範圍項目之 `iplanet-am-auth-login-success-url` 屬性中的 URL。
12. `iplanet-am-auth-login-success-url` 屬性中設定的 URL，作爲全域預設值。

失敗的基於角色的認證重新導向 URL

失敗的基於角色的認證，其重新導向 URL 是以此順序檢查下列位置決定的：

1. 認證模組設定的 URL。
2. `goto` 登入 URL 參數設定的 URL。
3. `clientType` 自訂檔案中爲使用者設定檔 (`amUser.xml`) 的 `iplanet-am-user-failure-url` 屬性設定的 URL。
4. `clientType` 自訂檔案中爲使用者已認證至的角色之 `iplanet-am-auth-login-failure-url` 屬性設定的 URL。
5. `clientType` 自訂檔案中爲已認證使用者的另一個角色項目之 `iplanet-am-auth-login-failure-url` 屬性設定的 URL。(如果前一個重新導向 URL 失敗，此選項爲備用。)
6. `clientType` 自訂檔案中爲使用者範圍項目的 `iplanet-am-auth-login-failure-url` 屬性設定的 URL。
7. `clientType` 自訂檔案中爲 `iplanet-am-auth-login-failure-url` 屬性做爲全域預設值設定的 URL。
8. 於使用者設定檔 (`amUser.xml`) 之 `iplanet-am-user-failure-url` 屬性中設定的 URL。
9. 已對其認證使用者的角色之 `iplanet-am-auth-login-failure-url` 屬性中設定的 URL。
10. 已認證使用者另一個角色項目之 `iplanet-am-auth-login-failure-url` 屬性中設定的 URL。(如果前一個重新導向 URL 失敗，此選項爲備案。)
11. 設定於使用者範圍項目之 `iplanet-am-auth-login-failure-url` 屬性中的 URL。
12. 於 `iplanet-am-auth-login-failure-url` 屬性中設定的 URL，作爲全域預設值。

▼ 若要配置基於角色的認證

1. **瀏覽至您將增加認證配置服務的範圍 (或組織)。**
2. **按一下 [主體] 標籤。**
3. **篩選的角色或角色。**
4. **選取要設定認證配置的角色。**
5. **選取您想啓用的「預設認證鏈接」。**
6. **按 [儲存]。**

備註 – 如果您要建立新的角色，系統不會自動爲此角色指定認證配置服務。請確定先選取角色設定檔頁面頂部的 [認證配置服務] 選項，然後再建立角色。

啓用基於角色的認證後，可以保留 LDAP 認證模組做爲預設方式，因爲無需配置成員身份。

基於服務的認證

此認證方法允許使用者向在範圍或子範圍中註冊的特定服務或應用程式進行認證。服務配置爲認證配置服務中的服務實例並且與一個實例名稱相關。若要成功認證，使用者必須向爲服務配置的認證配置服務實例中定義的每個模組進行認證。對每個基於服務的認證之實例，可指定下列屬性：

認證配置。這會定義爲服務的認證程序配置之認證模組。

登入成功 URL。此項定義在成功認證上重新導向使用者的 URL。

登入失敗 URL。此項定義在失敗認證上重新導向使用者的 URL。

認證處理後類別。此將定義後認證介面。

基於服務的認證登入 URL

透過定義服務參數，可以在使用者介面登入 URL 中指定基於服務的認證。在呼叫服務後，會從爲服務定義的認證配置服務實例擷取將認證使用者的認證模組。

用於指定和啓動基於服務的認證的登入 URL 爲：

```
http://server_name.domain_name:port/amserver/UI/
Login?service=auth-chain-name
```

和

```
http://server_name.domain_name:port/amserver
     /UI/Login?realm=realm_name&service=auth-chain-name
```

若無配置的 `org` 參數，將由登入 URL 中指定的伺服器主機和網域決定範圍。

基於服務的認證重新導向 URL

在基於服務的認證成功或失敗後，Access Manager 會尋找資訊以確定將使用者重新導向至何處。以下爲應用程式尋找此資訊的優先順序。

成功的基於服務的認證重新導向 URL

成功的基於服務的認證，其重新導向 URL 是以此順序檢查下列位置決定的：

1. 認證模組設定的 URL。
2. goto 登入 URL 參數設定的 URL。
3. clientType 自訂檔案中爲使用者設定檔 (amUser.xml) 的 iplanet-am-user-success-url 屬性設定的 URL。
4. clientType 自訂檔案中爲使用者已認證至的服務之 iplanet-am-auth-login-success-url 屬性設定的 URL。
5. clientType 自訂檔案中爲使用者角色項目的 iplanet-am-auth-login-success-url 屬性設定的 URL。
6. clientType 自訂檔案中爲使用者範圍項目的 iplanet-am-auth-login-success-url 屬性設定的 URL。
7. clientType 自訂檔案中爲 iplanet-am-auth-login-success-url 屬性做爲全域預設值設定的 URL。
8. 設定於使用者設定檔 (amUser.xml) 之 iplanet-am-user-success-url 屬性中的 URL。
9. 已對其認證使用者的服務之 iplanet-am-auth-login-success-url 屬性中設定的 URL。
10. 使用者角色項目之 iplanet-am-auth-login-success-url 屬性中設定的 URL。
11. 設定於使用者範圍項目之 iplanet-am-auth-login-success-url 屬性中的 URL。
12. iplanet-am-auth-login-success-url 屬性中設定的 URL，作爲全域預設值。

失敗的基於服務的認證重新導向 URL

失敗的基於服務的認證，其重新導向是以此順序檢查下列位置決定的：

1. 認證模組設定的 URL。
2. goto 登入 URL 參數設定的 URL。
3. clientType 自訂檔案中爲使用者設定檔 (amUser.xml) 的 iplanet-am-user-failure-url 屬性設定的 URL。
4. clientType 自訂檔案中爲使用者已認證至的服務之 iplanet-am-auth-login-failure-url 屬性設定的 URL。
5. clientType 自訂檔案中爲使用者角色項目的 iplanet-am-auth-login-failure-url 屬性設定的 URL。
6. clientType 自訂檔案中爲使用者範圍項目的 iplanet-am-auth-login-failure-url 屬性設定的 URL。
7. clientType 自訂檔案中爲 iplanet-am-auth-login-failure-url 屬性做爲全域預設值設定的 URL。

8. 設定於使用者設定檔 (amUser.xml) 之 iplanet-am-user-failure-url 屬性中的 URL。
9. 已對其認證使用者的服務之 iplanet-am-auth-login-failure-url 屬性中設定的 URL。
10. 於使用者角色項目之 iplanet-am-auth-login-failure-url 屬性中設定的 URL。
11. 設定於使用者範圍項目之 iplanet-am-auth-login-failure-url 屬性中的 URL。
12. 於 iplanet-am-auth-login-failure-url 屬性中設定的 URL，作爲全域預設值。

▼ 若要配置基於服務的認證

加入認證配置服務後，爲服務設定認證模組。若要如此，請：

1 **選擇您要配置基於服務的認證的範圍。**

2 **按一下 [認證] 標籤。**

3 **建立認證模組實例。**

4 **建立認證鏈接。**

5 **按 [儲存]。**

6 **若要存取範圍的基於服務的認證，請輸入下列位址：**

`http://`*server_name.domain_name:port*`/amserver/UI/Login?realm=`*realm_name*`&service=`*auth-chain-name*

基於使用者的認證

此認證方法可讓使用者向特別爲使用者配置的認證程序進行認證。該程序被配置爲使用者設定檔中使用者認證配置屬性的值。若要成功認證，使用者必須認證到每個定義的模組。

基於使用者的認證登入 URL

透過定義使用者參數，可以在使用者介面登入 URL 中指定基於使用者的認證。在呼叫正確的使用者後，將從爲使用者定義的使用者認證配置服務實例擷取將認證使用者的認證模組。

用於指定和啓動基於角色的認證的登入 URL 爲：

```
http://server_name.domain_name:port/amserver/UI/Login?user=user_name
http://server_name.domain_name:port/amserver/UI/Login?org=org_name&user=user_name
```

若無配置的 realm 參數，則將由登入 URL 中指定的伺服器主機和網域決定角色所屬的範圍。

使用者別名清單屬性

在接收基於使用者的認證請求時，認證服務會先驗證使用者是有效的使用者，然後爲其擷取認證配置資料。在有一個以上有效使用者設定檔與使用者登入 URL 參數有關的情形時，所有的設定檔必須對映到指定的使用者。使用者設定檔中的使用者別名屬性 (`iplanet-am-user-alias-list`) 是用於定義其他屬於該使用者的設定檔之位置。如果對映失敗，則使用者會受到有效階段作業的拒絕。例外的情況是：若其中一個使用者爲頂層管理員，則不會執行使用者對映驗證並給予使用者頂層管理員權限。

基於使用者的認證重新導向 URL

在基於使用者的認證成功或失敗後，Access Manager 會尋找資訊以確定將使用者重新導向至何處。以下爲應用程式尋找此資訊的優先順序。

成功的基於使用者的認證重新導向 URL

成功的基於使用者的認證，其重新導向 URL 是以此優先順序檢查下列位置決定的：

1. 認證模組設定的 URL。
2. `goto` 登入 URL 參數設定的 URL。
3. `clientType` 自訂檔案中爲使用者設定檔 (`amUser.xml`) 的 `iplanet-am-user-success-url` 屬性設定的 URL。
4. `clientType` 自訂檔案中爲使用者角色項目的 `iplanet-am-auth-login-success-url` 屬性設定的 URL。
5. `clientType` 自訂檔案中爲使用者範圍項目的 `iplanet-am-auth-login-success-url` 屬性設定的 URL。
6. `clientType` 自訂檔案中爲 `iplanet-am-auth-login-success-url` 屬性做爲全域預設值設定的 URL。
7. 設定於使用者設定檔 (`amUser.xml`) 之 `iplanet-am-user-success-url` 屬性中的 URL。
8. 使用者角色項目之 `iplanet-am-auth-login-success-url` 屬性中設定的 URL。
9. 設定於使用者範圍項目之 `iplanet-am-auth-login-success-url` 屬性中的 URL。
10. `iplanet-am-auth-login-success-url` 屬性中設定的 URL，作爲全域預設值。

失敗的基於使用者的認證重新導向 URL

失敗的基於使用者的認證，其重新導向 URL 是以此順序檢查下列位置決定的：

1. 認證模組設定的 URL。

2. gotoOnFail 登入 URL 參數設定的 URL。
3. clientType 自訂檔案中爲使用者項目 (amUser.xml) 的 iplanet-am-user-failure-url 屬性設定的 URL。
4. clientType 自訂檔案中爲使用者角色項目的 iplanet-am-auth-login-failure-url 屬性設定的 URL。
5. clientType 自訂檔案中爲使用者範圍項目的 iplanet-am-auth-login-failure-url 屬性設定的 URL。
6. clientType 自訂檔案中爲 iplanet-am-auth-login-failure-url 屬性做爲全域預設值設定的 URL。
7. 於使用者項目 (amUser.xml) 中設定 iplanet-am-user-failure-url 屬性的 URL。
8. 針對使用者角色項目之 iplanet-am-auth-login-failure-url 屬性設定的 URL。
9. 設定使用者範圍項目之 iplanet-am-auth-login-failure-url 屬性的 URL。
10. 針對 iplanet-am-auth-login-failure-url 屬性設定的 URL，作爲全域預設值。

▼ 若要配置基於使用者的認證

1 **瀏覽至您要為使用者配置認證的範圍**。

2 **按一下 [主體] 標籤並按一下 [使用者]**。

3 **按一下您要修改的使用者名稱**。
[使用者設定檔] 隨即顯示。

備註 – 如果您要建立新的使用者，系統不會自動爲此使用者指定認證配置服務。請確定先於服務設定檔中選取 [認證配置] 服務選項，然後再建立使用者。如果未選取此選項，使用者將無法繼承爲角色定義的認證配置。

---

4 **於 [使用者認證配置] 屬性中，選取您要套用的認證鏈接**。

5 **按 [儲存]**。

基於認證層級的認證

每個認證模組均可與其**認證層級**的整數值相關聯。變更模組 [認證層級] 屬性的對應值，即可指定認證層級。使用者在一個或多個認證模組中經過認證後，較高的認證層級爲使用者定義較高的信任層級。

在模組中成功認證使用者後，將在使用者的 SSO 記號上設定認證層級。如果使用者被要求在多個認證模組中認證，並且成功完成認證，則最高的認證層級值將標記在使用者的 SSO 記號上。

若使用者嘗試存取服務，服務可檢查使用者的 SSO 記號中之認證層級，來決定是否允許使用者進行存取。然後，它將重新導向使用者，使之以設定的認證層級通過認證模組。

使用者還可以使用特定的認證層級存取認證模組。例如，某使用者使用以下語法執行登入：

```
http://hostname:port/deploy_URI/UI/Login?authlevel=
auth_level_value
```

其認證層級大於或等於 *auth_level_value* 的所有模組將顯示爲認證功能表，供使用者選擇。如果僅找到一個相符的模組，則會直接顯示此認證模組的登入頁面。

此認證方法可讓管理員指定可認證身份的模組的安全層級。每個認證模組都有個別的認證層級屬性，而此屬性的值可以被定義爲任何有效的整數。藉由基於認證層級的認證，「認證服務」會顯示一個模組登入頁面，其中有一個功能表，包含認證層級等於或大於登入 URL 參數所指定的值之認證模組。使用者可從現有的清單選取一個模組。一旦使用者選取模組後，剩餘的程序則視基於模組的認證而定。

基於認證層級的認證登入 URL

透過定義 authlevel 參數，可以在使用者介面登入 URL 中指定基於認證層級的認證。在呼叫含有相關模組清單的登入螢幕後，使用者必須選擇一項要認證至的模組。用於指定和啓動基於認證層級的認證登入 URL 爲：

```
http://server_name.domain_name:port/amserver/UI/Login?authlevel=authentication_level
```

和

```
http://server_name.domain_name:port/amserver/UI/
Login?realm=realm_name&authlevel=authentication_level
```

若無配置的 `realm` 參數，則將由登入 URL 中指定的伺服器主機和網域決定使用者所屬的範圍。

基於認證層級的認證重新導向 URL

在基於認證層級的認證成功或失敗後，Access Manager 會尋找資訊以確定將使用者重新導向至何處。以下爲應用程式尋找此資訊的優先順序。

成功的基於認證層級的認證重新導向 URL

成功的基於認證層級的認證，其重新導向 URL 是以此優先順序檢查下列位置決定的：

1. 認證模組設定的 URL。
2. `goto` 登入 URL 參數設定的 URL。
3. `clientType` 自訂檔案中爲使用者設定檔 (`amUser.xml`) 的 `iplanet-am-user-success-url` 屬性設定的 URL。
4. `clientType` 自訂檔案中爲使用者角色項目的 `iplanet-am-auth-login-success-url` 屬性設定的 URL。
5. `clientType` 自訂檔案中爲使用者範圍項目的 `iplanet-am-auth-login-success-url` 屬性設定的 URL。
6. `clientType` 自訂檔案中爲 `iplanet-am-auth-login-success-url` 屬性做爲全域預設值設定的 URL。
7. 於使用者設定檔 (`amUser.xml`) 中的 `iplanet-am-user-success-url` 屬性中設定一個 URL。
8. 使用者角色項目之 `iplanet-am-auth-login-success-url` 屬性中設定的 URL。
9. 設定於使用者範圍項目之 `iplanet-am-auth-login-success-url` 屬性中的 URL。
10. `iplanet-am-auth-login-success-url` 屬性中設定的 URL，作爲全域預設值。

失敗的基於認證層級的認證重新導向 URL

失敗的基於認證層級的認證重新導向 URL 是以此順序檢查下列位置決定的：

1. 認證模組設定的 URL。
2. `gotoOnFail` 登入 URL 參數設定的 URL。
3. `clientType` 自訂檔案中爲使用者項目 (`amUser.xml`) 的 `iplanet-am-user-failure-url` 屬性設定的 URL。
4. `clientType` 自訂檔案中爲使用者角色項目的 `iplanet-am-auth-login-failure-url` 屬性設定的 URL。
5. `clientType` 自訂檔案中爲使用者範圍項目的 `iplanet-am-auth-login-failure-url` 屬性設定的 URL。
6. `clientType` 自訂檔案中爲 `iplanet-am-auth-login-failure-url` 屬性做爲全域預設值設定的 URL。
7. 於使用者項目 (`amUser.xml`) 中設定 `iplanet-am-user-failure-url` 屬性的 URL。
8. 針對使用者角色項目之 `iplanet-am-auth-login-failure-url` 屬性設定的 URL。
9. 設定使用者範圍項目之 `iplanet-am-auth-login-failure-url` 屬性的 URL。
10. 針對 `iplanet-am-auth-login-failure-url` 屬性設定的 URL，作爲全域預設值。

基於模組的認證

使用者可以使用以下語法存取特定認證模組：

```
http://hostname:port/deploy_URI/UI/Login?module=
module_name
```

存取認證模組之前，必須先修改核心認證服務屬性 [範圍認證模組] ，使之包括此認證模組名稱。如果該屬性中未包括此認證模組名稱，使用者嘗試認證時，系統將顯示 [認證模組被拒絕] 頁面。

此認證方法可讓使用者指定進行認證的模組。指定的模組必須註冊至使用者存取的範圍或子範圍。這是在範圍核心認證服務的範圍認證模組屬性中配置。在接收此項基於模組的認證請求時，認證服務會驗證模組是否依據說明正確配置，如果未定義模組，使用者會被拒絕存取。

基於模組的認證登入 URL

透過定義模組參數，可以在使用者介面登入 URL 中指定基於模組的認證。用於指定和啓動基於模組的認證登入 URL 爲：

```
http://server_name.domain_name:port/amserver/UI/Login?module=authentication_module_name
http://server_name.domain_name:port/amserver/UI/
Login?org=org_name&module=authentication_module_name
```

若無配置的 `org` 參數，則將由登入 URL 中指定的伺服器主機和網域決定使用者所屬的範圍。

基於模組的認證重新導向 URL

在基於模組的認證成功或失敗後，Access Manager 會尋找資訊以確定將使用者重新導向至何處。以下爲應用程式尋找此資訊的優先順序。

成功的基於模組的認證重新導向 URL

成功的基於模組的認證，其重新導向 URL 是以此優先順序檢查下列位置決定的：

1. 認證模組設定的 URL。
2. `goto` 登入 URL 參數設定的 URL。
3. `clientType` 自訂檔案中爲使用者設定檔 (`amUser.xml`) 的 `iplanet-am-user-success-url` 屬性設定的 URL。
4. `clientType` 自訂檔案中爲使用者角色項目的 `iplanet-am-auth-login-success-url` 屬性設定的 URL。

5. clientType 自訂檔案中爲使用者範圍項目的 iplanet-am-auth-login-success-url 屬性設定的 URL。
6. clientType 自訂檔案中爲 iplanet-am-auth-login-success-url 屬性做爲全域預設值設定的 URL。
7. 於使用者設定檔 (amUser.xml) 中的 iplanet-am-user-success-url 屬性中設定一個 URL。
8. 使用者角色項目之 iplanet-am-auth-login-success-url 屬性中設定的 URL。
9. 設定於使用者範圍項目之 iplanet-am-auth-login-success-url 屬性中的 URL。
10. iplanet-am-auth-login-success-url 屬性中設定的 URL，作爲全域預設值。

失敗的基於模組的認證重新導向 URL

失敗的基於模組的認證，其重新導向 URL 是以此順序檢查下列位置決定的：

1. 認證模組設定的 URL。
2. gotoOnFail 登入 URL 參數設定的 URL。
3. clientType 自訂檔案中爲使用者項目 (amUser.xml) 的 iplanet-am-user-failure-url 屬性設定的 URL。
4. clientType 自訂檔案中爲使用者角色項目的 iplanet-am-auth-login-failure-url 屬性設定的 URL。
5. clientType 自訂檔案中爲使用者範圍項目的 iplanet-am-auth-login-failure-url 屬性設定的 URL。
6. clientType 自訂檔案中爲 iplanet-am-auth-login-failure-url 屬性做爲全域預設值設定的 URL。
7. 針對使用者角色項目之 iplanet-am-auth-login-failure-url 屬性設定的 URL。
8. 設定使用者範圍項目之 iplanet-am-auth-login-failure-url 屬性的 URL。
9. 針對 iplanet-am-auth-login-failure-url 屬性設定的 URL，作爲全域預設值。

使用者介面登入 URL

輸入登入 URL 到網路瀏覽器的 [位置列] 可存取 [認證服務] 使用者介面。此 URL 爲：

http://*AccessManager-root*/.*domain_name*:*port* /*service_deploy_uri* /UI/Login

備註 – 於安裝期間，將 *service_deploy_uri* 配置爲 amserver。本文件中將使用此預設的服務部署 URI。

使用者介面登入 URL 也可以附加登入 URL 參數，以定義特定的認證方法或是成功/失敗的認證重新導向 URL。

登入 URL 參數

URL 參數是附加到 URL 尾端的「名稱/值」對。參數以問號開頭 (？)，形式爲 `name=value`。一個登入 URL 可以包含多個參數，例如：

```
http://server_name.domain_name:port/amserver/UI/
Login?module=LDAP&locale=ja&goto=http://www.sun.com
```

如果有一個或多個參數，會以 & 符號做爲分隔符號。不過組合必須遵守下列指導方針：

- 每個參數在一個 URL 中只能出現一次。例如：`module=LDAP&module=NT` 是不可以計算的。
- `org` 參數與 `domain` 參數兩者皆可決定登入範圍。在這種情形下，登入 URL 中只能使用其中一個參數。如果兩者都使用了而且未指定優先順序，只有其中一個會生效。
- 參數 `user`、`role`、`service`、`module` 及 `authlevel` 用於定義認證模組 (根據其各自的條件)。因此，只應於登入 URL 中使用其中之一。如果使用了一個以上而且未指定優先順序，只有其中一個會生效。

下節說明各個參數，這些參數在附加到使用者介面登入 URL 並鍵入網路瀏覽器的位置列時，可達到多種認證功能。

備註 – 若要簡化認證 URL 和參數以便在範圍內分發，管理員可配置一個具備簡單 URL 的 HTML 網頁，該頁面可連結到更複雜的登入 URL 以獲取所有已配置的認證方法。

---

goto 參數

`goto=`*successful_authentication_URL* 參數會覆寫認證配置服務之 [登入成功 URL] 中定義的值。當成功認證時，會連結到指定的 URL。`goto=logout_URL` 參數也可用於在使用者登出時連結到指定的 URL。以下是成功認證 URL 的範例：

```
http://server_name.domain_name:port/amserver/
UI/Login?goto=http://www.sun.com/homepage.html
```

`goto` 登出 URL 的範例：

```
http://server_name.domain_name:port/amserver/
UI/Logout?goto=http://www.sun.com/logout.html.
```

備註 – Access Manager 使用優先順序尋找成功的認證重新導向 URL。因爲這些重新導向 URL 及其順序是以認證方法爲基礎，此順序 (及相關資訊) 於「認證類型」一節中有詳細說明。

gotoOnFail 參數

`gotoOnFail=failed_authentication_URL` 參數會覆寫認證配置服務之 [登入失敗 URL] 中定義的值。如果使用者認證失敗，將會連結到指定的 URL。`gotoOnFail` URL 的範例爲：`http://`*`server_name.domain_name`*`:`*`port`*`/amserver/UI/Login?gotoOnFail=http://www.sun.com/auth_fail.html`。

備註 – Access Manager 使用優先順序尋找失敗的認證重新導向 URL。因爲這些重新導向 URL 及其順序是以認證方法爲基礎，此順序 (及相關資訊) 於「認證類型」一節中有詳細說明。

realm 參數

`org=`*`realmName`* 參數允許使用者做爲指定範圍中的使用者進行認證。

備註 – 當使用者嘗試以 `realm` 參數認證時，若其不是指定範圍的成員，就會收到錯誤訊息。如果以下全部皆爲 TRUE，可以於 Directory Server 中動態建立使用者設定檔：

- 核心認證服務中的使用者設定檔屬性必須設定爲**動態**或**隨使用者別名動態變化**。
- 使用者必須成功認證到需要的模組。
- Directory Server 中還沒有使用者的設定檔。

使用這項參數，將顯示正確的登入頁 (根據範圍及其語言環境設定)。若未設定此參數，預設值爲頂層範圍。例如：`org` URL 可以是：

```
http://server_name.domain_name:port/amserver/UI/Login?realm=sun
```

org 參數

`org=`*`orgName`* 參數可讓使用者做爲指定組織中的使用者進行認證。

備註 – 當使用者嘗試以 org 參數認證時，若其不是指定組織的成員，就會收到錯誤訊息。如果以下全部皆爲 TRUE，可以於 Directory Server 中動態建立使用者設定檔：

- 核心認證服務中的使用者設定檔屬性必須設定爲**動態**或**隨使用者別名動態變化**。
- 使用者必須成功認證到需要的模組。
- Directory Server 中還沒有使用者的設定檔。

使用這項參數，將顯示正確的登入頁 (根據組織及其語言環境設定)。如果未設定此參數，預設值爲頂層組織。例如：org URL 可以是：

```
http://server_name.domain_name:port/amserver/UI/Login?org=sun
```

user 參數

user=*userName* 參數基於使用者設定檔之 [使用者認證配置] 屬性中配置的模組進行強制認證。例如，可以將一個使用者的設定檔配置爲使用「憑證」模組進行認證，同時可以將另一個使用者配置爲使用 LDAP 模組進行認證。增加此參數會將使用者傳送到其配置的認證程序，而非爲其組織配置的方法。例如：

```
http://server_name.domain_name:port/amserver/UI/Login?user=jsmith
```

role 參數

role=roleName 參數會將使用者傳送到爲指定角色配置的認證程序。尚未成爲指定角色成員的使用者如果嘗試用此參數進行認證，會收到一條錯誤訊息。例如：

```
http://server_name.domain_name:port/amserver/UI/Login?role=manager.
```

locale 參數

Access Manager 具有爲認證程序及主控台本身顯示本土化螢幕 (譯爲非英語的語言) 的功能。locale=*localeName* 參數指定的語言環境較任何其他定義的語言環境的優先權更高。在下列位置中按指定順序搜尋配置後，用戶端會顯示登入語言環境：

1. 登入 URL 中的語言環境參數值

   locale=*localeName* 參數的值優先於所有其他定義的語言環境。

2. 使用者設定檔中定義的語言環境

   如果沒有 URL 參數，會根據在使用者設定檔的 [使用者偏好的語言] 屬性中設定的值顯示語言環境。

3. 在標頭中定義的語言環境

   語言環境由 Web 瀏覽器設定。

4. [核心認證服務] 中定義的語言環境

這是在 [核心認證] 模組中 [預設認證語言環境] 屬性的值。

5. 在 [平台服務] 中定義的語言環境

   這是在 [平台] 服務中 [平台語言環境] 屬性的值。

作業系統語言環境

由此等級順序得到的語言環境儲存於使用者的階段作業記號中，Access Manager 僅用它來載入本土化的認證模組。成功認證後，會使用於使用者設定檔之「使用者偏好的語言」屬性中定義的語言環境。如果都沒有設定，將繼續使用認證所使用的語言環境。例如：

```
http://server_name.domain_name:port/amserver/UI/Login?locale=ja.
```

備註 – 如何本土化螢幕文字和錯誤訊息的資訊可於 Access Manager 中找到。

module 參數

`module=`*moduleName* 參數允許經由指定的認證模組進行認證。可指定任何模組，但模組必須首先在使用者所屬範圍下註冊且做爲「核心認證」模組中該範圍的認證模組之一被選取。例如：

```
http://server_name.domain_name:port/amserver/UI/Login?module=Unix.
```

備註 – 在 URL 參數中使用認證模組名稱時要區分大小寫。

service 參數

`service=`*serviceName* 參數允許經由服務已配置的認證方案來認證使用者。使用 [認證配置] 服務可以爲不同的服務配置不同的認證方案。例如，線上薪資應用程式可能需要使用更安全「憑證認證」模組進行認證，而範圍的員工目錄應用程式可能只需要「LDAP 認證」模組。可以爲這些服務中的每一個配置並命名認證方案。例如：

```
http://server_name.domain_name:port/amserver/UI/Login?service=sv1.
```

備註 – 「認證配置」服務用來爲基於服務的認證定義方案。

arg 參數

`arg=`*newsession* 參數用於結束使用者的目前階段作業，並開始新的階段作業。認證服務會透過一個請求銷毀使用者現有的階段作業記號，並執行新的登入。此選項通常用於 [匿名認證] 模組中。使用者先以匿名階段作業認證，然後點一下註冊或登入連結。例如：

`http://server_name.domain_name:port/amserver/UI/Login?arg=newsession`。

authlevel 參數

`authlevel=value` 參數會指示認證服務呼叫認證層級等於或大於指定認證層級值的模組。每個認證模組定義有一個固定的整數認證層級。例如：

```
http://server_name.domain_name:port/amserver/UI/Login?authlevel=1.
```

備註 – 認證層級是於每個模組的特定設定檔中設定。

domain 參數

此參數可讓使用者登入到識別爲指定網域的範圍。指定的網域必須符合定義於範圍設定檔之網域名稱屬性中的值。例如：

`http://server_name.domain_name:port/amserver/UI/Login?domain=sun.com`。

備註 – 當使用者嘗試以 `org` 參數認證時，若其不是指定網域/範圍的成員，就會收到錯誤訊息。如果以下各點全部皆爲 TRUE，可以於 Directory Server 中動態建立使用者設定檔：

- 核心認證服務中的使用者屬性必須設定爲**動態**或**隨使用者別名動態變化** 。
- 使用者必須成功認證到需要的模組。
- Directory Server 中還沒有使用者的設定檔。

iPSPCookie 參數

`iPSPCookie=yes` 參數可讓使用者以永久性的 cookie 登入。永久性的 cookie 在瀏覽器視窗關閉後仍然繼續存在。要使用此參數，使用者登入的範圍必須在其核心認證模組中啓用永久性 Cookie。一旦使用者認證及瀏覽器關閉，使用者可以新的瀏覽器階段作業登入並被導向至控制台，而不需重新認證。在核心服務的 [永久性 Cookie 最長時間] 屬性指定之時間消逝前，該功能都有效。例如：

`http://server_name.domain_name:port/amserver/UI/Login?org=example&iPSPCookie=yes`

IDTokenN 參數

此參數可讓使用者藉由 URL 或 HTML 表單傳送認證憑證。利用 `IDTokenN=` *value* 參數，使用者毋須存取認證服務使用者介面便可被認證。此程序稱爲**零頁登入**。零頁登入只適用於使用單一登入頁的認證模組。`IDToken0`、`IDToken1`、 ...、`IDTokenN` 的值會對映至認證模組的登入頁面之欄位。例如，LDAP 認證模組可能將 `IDToken1` 用於 `userID` 資訊、將 `IDToken2` 用於密碼資訊。在這種情形下，LDAP 模組 IDTokenN URL 將是：

```
http://server_name.domain_name:port/amserver/UI/
Login?module=LDAP&IDToken1=userID&IDToken2=password
```

(若預設認證模組爲 LDAP，就可以省略 `module=LDAP`。)

就匿名認證而言，登入 URL 參數會是：

`http://server_name.domain_name:port/amserver/UI/Login?module=Anonymous&IDToken1=anonymousUserID`。

備註 – 名稱爲 `Login.Token0`、`Login.Token1`、...、`Login.TokenN` 的記號 (來自之前的版本) 仍受支援，但將於未來版本中停用。建議您使用新的 `IDTokenN` 參數。

帳號鎖定

認證服務提供一項功能，在認證失敗次數超過某個特定值後將**封鎖**使用者。這項功能預設爲關閉，但可以使用 Access Manager 主控台啓用。

備註 – 只有拋出 [密碼無效] 異常的模組可以利用帳號鎖定功能。

核心認證服務包含用於啓用和自訂此功能的屬性，包括但不限於：

- 會啓用帳號鎖定的**登入失敗封鎖模式**。
- **登入失敗封鎖計數**定義使用者被封鎖前可嘗試認證的次數。此計數僅對單個使用者 ID 有效；只有同一個使用者 ID 失敗次數達到指定的次數後才會被封鎖。
- **登入失敗封鎖間隔**定義使用者被封鎖前，必須完成登入失敗封鎖計數值的時間 (以分鐘計)。
- **發送封鎖通知的電子郵件位址**指定使用者封鎖通知將發送到的電子郵件位址。
- **N 次失敗後警告使用者**指定對使用者顯示警告訊息前，可發生的認證失敗次數。這允許管理員在使用者得到即將封鎖的警告之後設定附加的登入嘗試次數。
- **登入失敗封鎖持續時間**定義封鎖使用者後，再次嘗試認證前必須等待的時間 (以分鐘計)。
- **封鎖屬性名稱**定義使用者設定檔中要針對實體鎖定的 LDAP 屬性設定爲「**非使用中**」。
- **封鎖屬性值**定義**封鎖屬性名稱**中指定的 LDAP 屬性將設定爲：**非使用中**或**使用中**。

有關任何帳號封鎖的電子郵件通知都會發送給管理員。帳號鎖定活動也會被記錄。

備註 – 有關在 Microsoft® Windows 2000 作業系統上使用此功能的特殊說明，請參閱附錄 A，「AMConfig.properties 檔案」中的「簡易郵件傳輸協定 (SMTP)」。

Access Manager 支援兩種帳號鎖定類型：實體鎖定與記憶體鎖定，定義於下列章節中。

實體鎖定

這是 Access Manager 的預設鎖定行爲。藉由變更使用者設定檔中的 LDAP 屬性爲非使用中，啓動鎖定。**封鎖屬性名稱**屬性定義用於封鎖的 LDAP 屬性。

備註 – 以別名爲名稱的使用者是藉由配置 LDAP 設定檔中使用者別名清單屬性 (`amUser.xml` **中的** `iplanet-am-user-alias-list`) ，以對映至現有 LDAP 使用者設定檔的使用者。藉由增加 `iplanet-am-user-alias-list` 至核心認證服務之 [別名搜尋屬性名稱] 欄位，可驗證以別名爲名稱的使用者。也就是說，如果一個別名使用者被封鎖，其對映至的實際 LDAP 設定檔也將被鎖定。這只適用於使用 LDAP 和成員身份之外的認證模組的實體封鎖。

記憶體鎖定

將**登入失敗封鎖持續時間**屬性的值變更爲大於零，可啓用記憶體鎖定。啓用後，使用者帳號會依指定分鐘數鎖定於記憶體中。經過該段時間後，將解除鎖定帳號。以下是使用記憶體鎖定功能時，一些特殊的考量：

- 若重新啓動了 Access Manager，所有鎖定於記憶體中的帳號都會解除鎖定。
- 若使用者的帳號被鎖定在記憶體中，而管理員將帳號鎖定機制變更爲實體鎖定 (藉由將鎖定持續時間設回零)，則使用者帳號將在記憶體中被解除鎖定，鎖定計數也會重設。
- 記憶體封鎖後，當使用 LDAP 與成員身份之外的認證模組時，若使用者嘗試以正確的密碼登入，則將傳回 [**使用者於此範圍中並無設定檔**] 錯誤，而不是傳回 [**使用者不在使用中**] 錯誤。

備註 – 如果在使用者設定檔中設定了失敗的 URL 屬性，則封鎖警告訊息和指出使用者帳號已遭鎖定的訊息都不會顯示，系統會將使用者重新導向至定義的 URL。

認證服務容錯移轉

若主伺服器因爲硬體或軟體問題失敗或伺服器暫時關機，則認證服務容錯移轉會自動將認證請求重新導向至輔助伺服器中。

必須先在可使用認證服務的實例上建立認證環境。如果此 Access Manager 實例無法使用，則可透過認證容錯移轉機制在不同的 Access Manager 實例上建立認證環境。認證內容會依下列順序檢查伺服器的可用性：

1. 認證服務 URL 會傳到 AuthContext API。例如：

   ```
   AuthContext(orgName, url)
   ```

   如果使用此 API，僅使用 URL 參照的伺服器。即使伺服器上可以使用認證服務，也不會發生容錯移轉。
2. 認證環境將檢查定義於 `AMConfig.properties` 檔案的 `com.iplanet.am.server*` 屬性中的伺服器。
3. 如果步驟 2 失敗，則認證環境會從可提供命名服務的伺服器查詢平台清單。在共用一個 Directory Server 實例安裝多個 Access Manager 實例時 (通常是爲容錯移轉)，會自動建立此平台清單。

   例如，如果平台清單包含 `Server1`、`Server2` 及 `Server3` 的 URL，則認證環境會在 `Server1`、`Server2` 及 `Server3` 間循環，直到成功認證至其中一個爲止。

   平台清單有時不是從同一個伺服器取得，其視命名服務的可用性而異。另外，命名服務的容錯移轉可能先發生。多重命名服務 URL 指定於 `com.iplanet.am.naming.url` **特性**中 (在 `AMConfing.properties` 中)。第一個可用的命名服務 URL 會用來辨識伺服器，該伺服器包含將發生容錯移轉的伺服器 (位於其平台伺服器清單中) 的清單。

完全合格的網域名稱對映

完全合格的網域名稱 (Fully Qualified Domain Name, FQDN) 對映可讓認證服務在使用者輸入錯誤的 URL 時採取修正行動 (例如指定部分的主機名稱或 IP 位址以存取受保護的資源)。FQDN 對映是藉由修改 `AMConfig.properties` 檔案中的 `com.sun.identity.server.fqdnMap` 屬性來啓用。指定此特性的格式爲：

`com.sun.identity.server.fqdnMap[`*invalid-name* `]=`*valid-name*

值 *invalid-name* 是使用者可能鍵入的無效 FQDN 主機名稱，*valid-name* 則爲篩選器將重新導向使用者至的實際主機名稱。只要符合規定的要求，可以指定任意數量的對映 (如程式碼範例 1-1 所示)。若未設定此特性，使用者將被傳送到在 `com.iplanet.am.server.host=`*server_name* 特性 (在 `AMConfig.properties` 檔案中) 中配置的預設伺服器名稱。

範例 4–1 AMConfig.properties 中的 FQDN 對映屬性

```
com.sun.identity.server.fqdnMap[isserver]=isserver.mydomain.com
com.sun.identity.server.fqdnMap[isserver.mydomain]=isserver.mydomain.com
com.sun.identity.server.fqdnMap[
            IP address]=isserver.mydomain.com
```

FQDN 對映的可能用法

此特性可用於爲多個主機名稱建立對映，例如在伺服器上的應用程式可由多個主機名稱存取時便可使用此特性。此特性亦可用於配置 Access Manager，使其不對某些 URL 採取修正動作。例如，若使用 IP 位址存取應用程式的使用者不需要重新導向時，可藉由指定對映項目來實作此功能，例如：

`com.sun.identity.server.fqdnMap[`*IP address* `]=`*IP address*。

備註 – 如果定義了一個以上的對映，請確定在無效的 FQDN 名稱中沒有重疊值。如果沒有這麼做，可能會導致應用程式無法存取。

永久性 Cookie

永久性 cookie 將於 Web 瀏覽器關閉後仍持續存在，使用者可以新的瀏覽器階段作業登入而不必重新認證。cookie 的名稱由 `AMConfig.properties` 中 `com.iplanet.am.pcookie.name` 特性所定義；預設值爲 `DProPCookie`。cookie 值是一個 3DES 加密的字串，包含使用者 DN、範圍名稱、認證模組名稱、最長階段作業時間、閒置時間和快取時間。

▼ 若要啓用永久性 Cookie

1. **開啓核心認證模組中的永久性 `Cookie` 模式**。
2. **配置核心認證模組中永久性 `Cookie` 最長時間屬性之時間值**。

3 **將 iPSPCookie 參數 (值為 yes) 附加到使用者介面登入 URL。**

一旦使用者使用此 URL 進行認證，若瀏覽器關閉，其可開啓一個新的瀏覽器視窗並重新導向至主控台，而不需重新認證。這項作業的運作時間爲直到步驟 2 中定義的時間結束爲止。

可以使用認證 SPI 方法開啓永久性 Cookie 模式：

`AMLoginModule.setPersistentCookieOn()`。

「舊有」模式的多重 LDAP 認證模組配置

做爲一種容錯移轉形式，或當 Access Manager 主控台僅提供一個值欄位時可用來配置屬性的多個值，管理員可於一個範圍之下定義多重 LDAP 認證模組配置。儘管這些附加配置不會顯示在主控台中，但它們仍可在找不到用於請求使用者認證的初始搜尋時與主配置配合使用。例如，一個範圍可於兩種不同網域中透過 LDAP 伺服器爲認證定義搜尋，或於一個網域中配置多重使用者命名屬性。就後者而言，在主控台中只有一個文字欄位，如果使用主要搜尋條件找不到使用者，LDAP 模組將會使用次要範圍搜尋。依照下列步驟配置其他的 LDAP 配置。

▼ 若要增加其他的 LDAP 配置

1 **撰寫一個 XML 檔案，其中包含完整屬性集和次要 (或第三) LDAP 認證配置需要的新值。**

檢視 `amAuthLDAP.xml` (位於 `etc/opt/SUNWam/config/xml`) 以參照可用的屬性。但於此步驟中建立的 XML 檔案是以 `amadmin.dtd` 爲基礎，不同於 `amAuthLDAP.xml`。可以爲此檔案定義任何或是所有屬性。程式碼範例 1-2 是子配置檔案的範例，其包含對 LDAP 認證配置可用的所有屬性之值。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
  Copyright (c) 2002 Sun Microsystems, Inc. All rights reserved.
  Use is subject to license terms.
-->
<!DOCTYPE Requests
    PUBLIC "-//iPlanet//Sun ONE Access Manager 6.0 Admin CLI DTD//EN"
    "jar://com/iplanet/am/admin/cli/amAdmin.dtd"
>
<!--
  Before adding subConfiguration load the schema with
GlobalConfiguration defined and replace corresponding
 serviceName and subConfigID in this sample file OR load
 serviceConfigurationRequests.xml before loading this sample
-->
<Requests>
```

```
<realmRequests DN="dc=iplanet,dc=com">
    <AddSubConfiguration subConfigName = "ssc"
        subConfigId = "serverconfig"
        priority = "0" serviceName="iPlanetAMAuthLDAPService">

              <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-server"/>
            <Value>vbrao.red.iplanet.com:389</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-base-dn"/>
            <Value>dc=iplanet,dc=com</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="planet-am-auth-ldap-bind-dn"/>
            <Value>cn=amldapuser,ou=DSAME Users,dc=iplanet,dc=com</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-bind-passwd"/>
            <Value>
                  plain text password</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-user-naming-attribute"/>
            <Value>uid</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-user-search-attributes"/>
            <Value>uid</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-search-scope"/>
            <Value>SUBTREE</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-ssl-enabled"/>
            <Value>false</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-return-user-dn"/>
            <Value>true</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-auth-level"/>
            <Value>0</Value>
        </AttributeValuePair>
        <AttributeValuePair>
```

```
            <Attribute name="iplanet-am-auth-ldap-server-check"/>
            <Value>15</Value>
        </AttributeValuePair>

    </AddSubConfiguration>

</realmRequests>
</Requests>
```

2 **複製純文字密碼做為建立於步驟 1 之 XML 檔案中 iplanet-am-auth-ldap-bind-passwd 的值。**

此屬性的值於程式碼範例中以粗體顯示。

3 **使用 `amadmin` 指令行工具載入 XML 檔案。**

```
./amadmin -u amadmin -w administrator_password -v -t name_of_XML_file.
```

請注意次要 LDAP 配置不會顯示並且不能使用主控台修改。

提示 – 這是多重 LDAP 配置可用的範例。請參閱 `/AccessManager-base/SUNWam/samples/admin/cli/bulk-ops/` 中的 `serviceAddMultipleLDAPConfigurationRequests.xml` 指令行範本。可於 `/AccesManager-base/SUNWam/samples/admin/cli/` 的 `Readme.html` 中取得說明。

階段作業升級

認證服務可讓您根據相同使用者對單一範圍第二次執行的成功認證對有效的階段作業記號進行升級。若具有有效階段作業記號的使用者試圖認證到由目前範圍保護的資源，且第二次認證請求成功，階段作業會根據新認證使用新特性更新。若認證失敗，則將傳回使用者目前的階段作業而不進行更新。 若具有有效階段作業的使用者試圖認證到由不同範圍保護的資源，使用者將收到詢問其是否要認證到新組織的訊息。使用者在此時可以維持目前階段作業，或是嘗試認證到新範圍。成功的認證將導致舊階段作業被銷毀，並建立新的階段作業。

在階段作業升級期間，如果登入頁逾時，將會重新導向到原始的成功 URL。逾時值的決定是基於：

- 爲每個模組設定的頁面逾時值 (預設爲 1 分鐘)
- `AMConfig.properties` 中的 `com.iplanet.am.invalidMaxSessionTime` 特性 (預設值爲 10 分鐘)

- iplanet-am-max-session-time (預設值爲 120 分鐘)

com.iplanet.am.invalidMaxSessionTimeout 和 iplanet-am-max-session-time 的值應大於頁面逾時值，否則階段作業升級期間的有效階段作業資訊將會遺失，而且到前一個成功 URL 的 URL 重新導向將會失敗。

驗證外掛程式介面

管理員可以撰寫適合其範圍的使用者名稱或是密碼驗證邏輯，並外掛至認證服務中。(這項功能只有 LDAP 和成員身份認證模組支援。)認證使用者或變更密碼之前，Access Manager 將呼叫此外掛程式。如果驗證成功，認證將繼續；如果失敗，將拋出認證失敗頁面。外掛程式會延伸 com.iplanet.am.sdk.AMUserPasswordValidation 類別，其爲「服務管理 SDK」的一部分。關於此 SDK 的資訊可於 Access Manager Javadocs 的 com.iplanet.am.sdk 套裝軟體中取得。

▼ 若要撰寫與配置驗證外掛程式

1. **新的外掛程式類別將延伸 com.iplanet.am.sdk. AMUserPasswordValidation 類別，並實作 validateUserID() 與 validatePassword() 方法。如果驗證失敗，應該會?出 AMException。**
2. **編譯外掛程式並將 .class 檔案置於想要的位置中。更新類別路徑，以便在執行階段期間可由 Access Manager 存取。**
3. **以頂層管理員的身份登入 Access Manager 主控台。按一下 [配置] 標籤，然後移至管理服務的屬性。於 [使用者 ID 和密碼驗證外掛程式類別] 欄位中鍵入外掛程式類別的名稱 (包括套裝軟體名稱)。**
4. **登出並登入。**

JAAS 共用狀態

共用狀態提供認證模組間使用者和密碼的共用。爲每個認證模組定義的選項用於：

- 範圍 (或組織)
- 使用者
- 服務
- 角色

在失敗時，模組會提示需要的憑證。在認證失敗後，模組停止執行，或清除登出共用狀態。

啓用 JAAS 共用狀態

若要配置 JAAS 共用狀態：

- 使用 `iplanet-am-auth-shared-state-enabled` 選項。
- 共用狀態選項的用法爲：`iplanet-am-auth-shared-state-enabled=true`
- 此選項預設爲 true。
- 此變數指定於認證鏈接配置的 [選項] 欄中。

失敗時，認證模組會根據 JASS 規格中建議的 `tryFirstPass` 選項行爲提示用戶提供所需的憑證。

JAAS 共用狀態儲存選項

若要配置 JAAS 共用狀態儲存選項：

- 使用 `iplanet-amauth-store-shared-state-enabled` 選項。
- 儲存共用狀態選項的用法爲：`iplanet-am-auth-store-shared-state-enabled=true`
- 此選項預設爲 false。
- 此變數指定於認證鏈接配置的 [選項] 欄中。

在確定、中斷或登出後，將清除共用狀態。

◆ ◆ ◆ 第 5 章

管理策略

本章描述 Sun Java™ System Access Manager 的策略管理功能。Access Manager 的「策略管理」功能使頂層管理員或頂層策略管理員可檢視、建立、刪除和修改用於所有範圍的特定服務的策略。它也爲範圍或子範圍管理員或策略管理員提供一種方式，以檢視、刪除和修改範圍層級的策略。

本章包含下列小節：

簡介

策略定義指定擁有組織受保護資源存取權限的規則。公司擁有需要保護、管理和監視的資源、應用程式和服務。策略透過定義使用者對特定資源行動的時機和方法，控制存取權限以及這些資源的用途。策略定義特定主體的資源。

備註 – 主體可以是個人、企業、角色或群組；或是任何可以具有識別的個體。如需更多資訊，請參閱 Java™ 2 Platform Standard Edition Javadoc (`http://java.sun.com/j2se/1.4.2/docs/api/java/security/Principal.html`)。

單一策略可以定義二進位或非二進位決策。二進位決策爲 *[***是***]*/*[***否***]*、*[***真***]*/*[***假***]* 或 *[***允許***]*/*[***拒絕***]*。非二進位決策代表屬性值。例如，郵件服務可能包含一個 `mailboxQuota` 屬性，並且爲每位使用者設定了最大儲存值。一般來說，策略是配置爲定義主體可以在什麼情況下對哪一個資源進行什麼動作。

策略管理功能

策略管理功能提供建立及管理策略的**策略服務**。策略服務允許管理員定義、修改、取得、取消及刪除權限，以保護 Access Manager 部署內的資源。通常，策略服務包括資料庫、允許建立、管理及評估策略的介面之程式庫、及策略執行程式或**策略代理程式**。依預設，Access Manager 將 Sun Java Enterprise System Directory Server 用於資料儲存，並爲策略評估和策略服務自訂提供 Java 和 C API (如需更多資訊，請參閱「Sun Java System Access Manager 7.1 Developer's Guide」)。它也讓管理員可使用 Access Manager 主控台來管理策略。Access Manager 提供一個啓用策略的服務，即「URL 策略代理程式」服務，它使用可下載的策略代理程式來強制執行策略。

URL 策略代理程式服務

在安裝時，Access Manager 提供的「URL 策略代理程式」服務可定義策略來保護 HTTP URL。此服務可讓管理員透過策略執行程式或**策略代理程式**建立與管理策略。

策略代理程式

策略代理程式是儲存企業資源的伺服器之策略執行點 (PEP)。策略代理程式與安裝在不同的 Web 伺服器上，且於使用者發出對受保護的 Web 伺服器上的網路資源的請求時，做爲一個額外的認證步驟。此認證在執行資源的任何使用者認證請求之外。此代理程式保護 Web 伺服器，並且資源也會受到認證外掛程式的保護。

例如，受遠端安裝的 Access Manager 保護之人力資源 Web 伺服器可能已安裝一個代理程式。此代理程式可以防止沒有適當策略的人員檢視機密薪資資訊或其他敏感資料。策略是由 Access Manager 管理員所定義、儲存在 Access Manager 部署中，且由策略代理程式用於允許或拒絕使用者存取遠端 Web 伺服器的內容。

最新的 Access Manager 策略代理程式可以從 Sun Microsystems 下載中心下載。

有關安裝與管理策略代理程式的更多資訊，請參閱「Sun Java System Access Manager Policy Agent 2.2 User's Guide」。

備註 – 策略是以一般順序進行評估，但在評估時，如果一個動作值評估爲 *deny*，就不會評估後續策略，除非策略配置服務中已啓用 [繼續評估拒絕決定] 屬性。

Access Manager 策略代理程式僅在 Web URL (`http://...` 或 `https://...`) 上執行決策。然而，可使用 Java 和 C 策略評估 API 編寫代理程式，以在其他資源上強制執行策略。

此外，策略配置服務中的 [資源比較程式] 屬性可能也需要從預設配置變更爲：

```
serviceType=Name_of_LDAPService
|class=com.sun.identity.policy.plugins.SuffixResourceName|wildcard=*
```

```
|delimiter=,|caseSensitive=false
```

或者，也可以提供如 LDAPResourceName 等實作來實作 com.sun.identity.policy.interfaces.ResourceName，並正確配置 [資源比較程式]。

策略代理程式程序

當網路瀏覽器請求一個駐留在受策略代理程式保護的伺服器之 URL 時，保護網路資源的程序即開始。伺服器的已安裝策略代理程式會截取請求，並檢查現有的認證憑證 (階段作業記號)。

如果代理程式截取了請求並驗證了現有階段作業記號，隨後便會執行下列程序。

1. 如果階段作業記號爲有效，允許或拒絕使用者存取。如果記號爲無效，使用者將被重新導向至認證服務，如下列步驟所述。

   假設代理程式截取了一個沒有現存階段作業記號的請求，代理程式將重新導向使用者到登入頁，不論該資源是否已經使用不同的認證方法保護。
2. 一旦正確的認證了使用者的憑證，代理程式會核發一個請求給命名服務，以將使用的 URL 定義爲連接至 Access Manager 的內部服務。
3. 若資源符合在代理程式配置的不予執行清單，則允許存取。
4. [命名服務] 會傳回策略服務、階段作業服務和記錄服務的定址器。
5. 代理程式會傳送請求給 [策略服務]，以取得適用於使用者的策略決策。
6. 基於存取資源的策略決策，決定使用者是否可以存取。如果策略決策建議不同的認證層級或認證機制，代理程式將重新導向請求到認證服務，直到驗證所有準則爲止。

策略類型

使用 Access Manager 配置的策略有兩種：

- 第 89 頁的 「一般策略」
- 第 94 頁的 「參照策略」

一般策略

在 Access Manager 中，定義存取權限的策略是指**一般**策略。一般策略由**規則**、**主體**、**條件**及**回應提供者**組成。

規則

規則包含一個服務類型、一或多個動作，以及一個值。基本上，規則定義策略。

- 服務類型定義受保護資源的類型。

- **動作**爲一項可於資源上執行的作業之名稱；Web 伺服器動作的範例有：POST 或 GET。人力資源服務可允許的一個動作可以變更爲一個住家電話號碼。
- **值**定義動作的權限，例如允許或拒絕。

備註 – 部份服務可接受只定義動作但沒有資源。

主體

主體定義策略將影響的使用者或使用者集合 (例如，擁有特定角色的群組或使用者)。主體的一般原則是，只有當使用者爲策略中至少一個主體的成員時，策略才適用。預設主體爲：

Access Manager 識別主體	此主體暗指您於 [範圍主體] 標籤下建立與管理的識別可做爲主體的一個成員增加。
經認證的使用者	此主體類型表示具有有效 SSOToken 的任何使用者均爲此主體的成員。 所有認證的使用者都將成爲此主體的成員，即使這些使用者已在與定義策略之組織不同的範圍內進行認證。如果資源所有者想要將存取權限授予其他組織的使用者所管理的資源，這個功能很有用。若要限制對特定組織的成員存取保護的資源，請使用組織主體。
Web 服務用戶端	此主體類型表示，如果包含在 SSOToken 中的任何主體之 DN 與此主體的任意所選值相符，則由 SSOToken 識別的 Web 服務用戶端 (WSC) 爲此主體的成員。有效值爲本機 JKS 鍵值儲存區中可信任憑證的 DN (與可信任 WSC 的憑證相對應)。此主體取決於 Liberty Web 服務架構，並且僅應該由 Liberty 服務提供者用來授權 WSC。 確定建立鍵值儲存區後再將此主體加入策略。以下位置可以找到設定鍵值儲存區的資訊： *AccessManager-base* `/SUNWam/samples/saml/xmlsig/keytool.html`

在範圍的 [策略配置服務] 中選取下列的附加主體後，便可使用它們：

Access Manager 角色	此主體類型表示 Access Manager 角色的任何成員均爲此主體的成員。使用舊有模式下的 Access Manager 及基於版本 6.3 的主控台，可以建立 Access Manager 角色。這些角色具有 Access Manager 代管的物件類別。Access Manager 角色僅可透過代管 Access Manager 策略服務存取。
LDAP 群組	此主體類型表示 LDAP 群組的任何成員均爲此主體的成員。

LDAP 角色	此主體類型表示 LDAP 角色的任何成員均爲此主體的成員。LDAP 角色是使用 Directory Server 角色功能的任何角色定義。這些角色具有 Directory Server 角色定義代管的物件類別。可以在策略配置服務中修改 LDAP 角色搜尋篩選器，以縮小範圍和改善效能。
LDAP 使用者	此主體類型表示任何 LDAP 使用者均爲此主體的成員。
組織	此主體類型表示範圍的任何成員均爲此主體的成員。

Access Manager 角色與 LDAP 角色的比較

Access Manager 角色是使用 Access Manager 建立的，這些角色具有 Access Manager 指派的物件類別。LDAP 角色是使用 Directory Server 角色功能的任何角色定義。這些角色具有 Directory Server 角色定義代管的物件類別。所有 Access Manager 角色皆可用來做爲 Directory Server 角色。不過，不是所有的 Directory Server 角色都一定會是 Access Manager 角色。藉由配置第 112 頁的「策略配置服務」，您可從現有目錄取用 LDAP 角色。Access Manager 角色僅可透過代管 Access Manager 策略服務存取。可以在策略配置服務中修改 LDAP 角色搜尋篩選器，以縮小範圍和改善效能。

巢式角色

在策略定義中，巢式角色可以正確評估爲 LDAP 角色。

條件

條件可讓您定義對策略的限制。例如，如果您在爲薪津應用程式定義策略，可以定義僅在特定幾小時限制此動作存取應用程式的條件。或者，如果請求來自給定 IP 位址集或企業內部網路，可能希望定義僅允許此動作存取的條件。

此條件可能還用於在同一網域的不同 URL 中配置不同的策略。例如，`http://org.example.com/hr/*jsp` 僅可以藉由 `org.example.net` 在上午 9 時至下午 5 時之間進行存取。配合使用 IP 條件與時間條件便可達此目的。將規則資源指定爲 `http://org.example.com/hr/*.jsp`，此策略會套用於 `http://org.example.com/hr` 下的所有 JSP (包括子目錄中的 JSP)。

備註 – 參照、規則、資源、主體、條件、動作及值等術語分別對應於 `policy.dtd` 中的元素 *Referral*、*Rule*、*ResourceName*、*Subject*、*Condition*、*Attribute* 及 *Value*。

您可增加的預設條件有：

作用中的階段作業時間

根據使用者階段作業資料設定條件。您可以修改的欄位爲：

最長階段作業時間	指定自階段作業初始開始時可套用策略的最大持續時間。
終止階段作業	選取此欄位時，如果階段作業時間超過 [最大階段作業時間] 欄位中定義所允許的最大時間，則使用者階段作業將被終止。

認證鏈

若使用者成功認證到指定範圍內的認證鏈接，則會套用策略。若未指定範圍，則在任何範圍內的認證鏈接中進行的認證都滿足條件。

認證層級 (大於或等於)

若使用者的認證層級大於或等於條件中設定的認證層級，則會套用策略。此屬性指示指定範圍內認證的信任層級。

認證層級 (小於或等於)

若使用者的認證層級小於或等於條件中設定的認證層級，則會套用策略。此屬性指示指定範圍內認證的信任層級。

認證模組實例

若使用者成功認證到指定範圍內的認證模組，則會套用策略。若未指定範圍，則在任何範圍內認證模組的認證都滿足條件。

目前的階段作業特性

根據使用者的 Access Manager 階段作業中設定的特性值來判定策略是否適用於請求。於策略評估期間，僅當使用者階段作業具有條件中定義的每個特性值時，條件才傳回 true。對於條件中以多重值定義於的特性，若記號具有至少一個條件中爲特性列出的值即可。

IP 位址/DNS 名稱

根據 IP 位址範圍設定條件。您可以定義的欄位爲：

IP 位址自/至	指定 IP 位址的範圍。
DNS 名稱	指定 DNS 名稱。此欄位可以爲完整的主機名稱或以下之一格式的字串： *domainname* **.domainname*

LDAP 篩選條件

當定義的 LDAP 篩選器在策略配置服務中指定的 LDAP 目錄中尋找使用者項目時，就會套用策略。這僅於定義策略所在的範圍內才適用。

範圍認證

若使用者成功認證到指定範圍，則會套用策略。

時間 (星期幾、日期、時間和時區)

根據時間限制來設定條件。這些欄位包括：

日期自/至	指定日期範圍。
時間	指定一天內的時間範圍。
星期幾	指定星期幾的範圍。
時區	指定時區 (標準或自訂)。自訂時區僅可爲 Java 識別的時區 ID (例如，PST)。如果未指定值，則預設值爲 Access Manager JVM 中設定的時區。

回應提供者

回應提供者爲提供策略型回應屬性的外掛程式。回應提供者屬性會和策略決策一起傳送給 PEP。Access Manager 包括一個實作，即 `IDResponseProvider`。此版本的 Access Manager 不支援自訂回應提供者。代理程式 PEP 通常會將這些回應以標頭的形式傳遞給應用程式。應用程式通常使用這些屬性將應用程式頁面個人化，例如入口網站頁面。

策略建議

如果無法根據條件的決定來套用策略，條件可能會產生建議訊息，指出無法將策略套用至請求的原因。這些建議訊息會在策略決策中傳播至 [策略執行點]。[策略執行點] 可以擷取此建議，並嘗試採取適當的行動，例如將使用者重新導向回認證機制，以便進行更高層級認證。採取建議的適當行動後，接著，使用者可能會收到更高層級認證的提示，只要能夠使用策略，使用者可能可以存取資源。

以下類別有更多資訊：

```
com.sun.identity.policy.ConditionDecision.getAdvices()
```

如果條件不符，只有 `AuthLevelCondiiton` 和 `AuthSchemeCondition` 會提供建議。

`AuthLevelCondition` 建議與以下鍵值相關聯：

```
com.sun.identity.policy.plugin.AuthLevelCondition.AUTH_LEVEL_CONDITION_ADVICE
```

AuthSchemeCondition 建議與以下鍵值相關聯：

com.sun.identity.policy.plugin.AuthLevelCondition.AUTH_SCHEME_CONDITION_ADVICE

自訂條件也會產生建議。但是，Access Manager 策略代理程式僅回應認證層級認證和認證方案建議。可以寫入自訂代理程式來瞭解及回應其他建議，而現有 Access Manager 代理程式可以延伸來瞭解及回應其他建議。如需更多資訊，請參閱「Sun Java System Access Manager Policy Agent 2.2 User's Guide」。

參照策略

管理員可能需要將一個範圍的策略定義委託給另一個範圍。(或者，可以將資源的策略決策委託給其他策略產品。)**參照**策略爲策略建立和評估控制此策略委託。它是由一項或多項**規則**及一個或多個**參照**組成。

策略配置服務包含一個稱爲「組織別名參照」的全域屬性。此屬性可讓您在子範圍內建立策略，而無需從頂層或父系範圍內建立參照策略。您只能建立用於保護 HTTP 或 HTTPS 資源的策略，而且這些資源的完全合格主機名稱必須符合範圍的範圍別名/DNS 別名。依預設，此屬性定義爲「否」。

規則

規則定義其策略定義與評估正在被參照的資源。

參照

參照定義策略評估所參照的組織。依預設，有兩種類型的參照：同級範圍及子範圍。此二者分別委託至相同層級上的範圍與子層級上的範圍。如需更多資訊，請參閱第 104 頁的 「建立同級範圍與子範圍的策略」。

備註 – 被參照的範圍可以僅爲那些已參照了該範圍的資源 (或子資源) 定義或評估策略。然而，此限制不會套用至頂層範圍。

策略定義類型文件

建立與配置好策略之後，會將其以 XML 的形式儲存於 Directory Server。在 Directory Server 中，以 XML 編碼的資料會儲存在同一位置。雖然策略是使用 amAdmin.dtd (或主控台) 定義和配置，實際上是做爲基於 policy.dtd 的 XML 儲存在 Directory Server。policy.dtd 包含從 amAdmin.dtd 中擷取的 policy 元素標籤 (不含策略建立標籤)。因此，

當策略服務從 Directory Server 載入策略時，將根據 `policy.dtd` 剖析 XML。只有在使用指令行建立策略時，才會使用 `amAdmin.dtd`。本節將描述 `policy.dtd` 的結構。`policy.dtd` 位於下列位置：

```
AccessManager-base/SUNWam/dtd (Solairs)
AccessManager-base/identity/dtd (Linux)
AccessManager-base/identity/dtd (HP-UX)
AccessManager-base\identity\dtd (Windows)
```

備註 – 本章其他部分僅提供 Solaris 目錄資訊。請注意，Linux、HP-UX 與 Windows 的目錄結構不同。

Policy 元素

Policy 是根元素，其定義策略的權限或**規則**，及套用規則的對象或**主體**。它也定義策略是否爲**參考** (委託的) 策略，及該策略是否有任何限制 (或**條件**)。可能包含下列一或多個子元素：*Rule*、*Condition*、*Subject*、*Referral* 或 *response provider*。必要的 XML 屬性爲 `name`，其指定策略的名稱。`referralPolicy` 屬性辨識策略是否爲參照策略；若未定義，預設值爲一般策略。選用的 XML 屬性包括 *name* 與 *description*。

備註 – 將策略標示爲**參照**時， 策略評估期間將略過主體與條件。相對的，將策略標示爲**一般**時，策略評估期間將略過所有參考。

Rule 元素

Rule 元素定義策略特性並可接受三個子元素：*ServiceName*、*ResourceName* 或 *AttributeValuePair*。可定義爲其建立策略服務類型或應用程式，以及於其中執行的資源和動作。規則可被定義爲不具任何動作；例如，參照策略規則不具任何動作。

備註 – 已定義策略不含已定義 *ResourceName* 元素是可接受的。

ServiceName 元素

ServiceName 元素定義套用策略的服務之名稱。此元素代表服務類型。不包含任何其他元素。此值與服務的 XML 檔案中定義之值完全相同 (以 `sms.dtd` 爲根據)。*ServiceName* 元素的 XML 服務屬性爲服務的名稱 (可接受字串值)。

ResourceName 元素

ResourceName 元素定義據以行動的物件。策略已經特別配置爲保護這個物件。不包含任何其他元素。*ResourceName* 元素的 XML 服務屬性爲物件的名稱。*ResourceName* 的範例有：網路伺服器上的 `http://www.sunone.com:8080/images`，或目錄伺服器上的 `ldap://sunone.com:389/dc=example,dc=com`。一個更特定的資源範例是 `salary://uid=jsmith,ou=people,dc=example,dc=com`，對其執行動作的物件是 John Smith 的薪金資訊。

AttributeValuePair 元素

AttributeValuePair 元素定義動作和動作的值。它被用來做爲 第 97 頁的 「Subject 元素」、第 97 頁的 「Referral 元素」及 第 98 頁的 「Condition 元素」的子元素。其同時包含 *Attribute* 與 *Value* 元素，而且沒有 XML 服務屬性。

Attribute 元素

Attribute 元素定義動作的名稱。一個動作爲在資源上執行的作業或事件。POST 或 GET 爲 Web 伺服器資源上執行的動作，READ 或 SEARCH 爲目錄伺服器上執行的動作。*Attribute* 元素必須與 *Value* 元素配對使用。*Attribute* 元素本身不包含其他任何元素。*Attribute* 元素的 XML 服務屬性爲動作的名稱。

Value 元素

Value 元素定義動作值。Allow/deny 或 yes/no 爲動作值範例。其他動作值可以是布林值、數字或字串。其值在定義於服務的 XML 檔案中 (以 `sms.dtd` 爲根據)。*Value* 元素不包含其他任何元素，而且也不包含 XML 服務屬性。

備註 – 拒絕規則永遠優先於允許規則。例如，如果一個策略是拒絕，另一種是允許，則結果是拒絕 (假如同時滿足這兩種策略條件)。由於拒絕策略可能導致潛在的衝突，因此建議您使用拒絕策略時要非常謹慎。如果使用明確的拒絕規則，透過不同主體 (如角色和/或群組成員身份) 爲給定使用者指定的策略也可能會導致拒絕對資源存取。通常，策略定義程序應該僅使用允許規則。如果未套用其他策略則可能使用預設的拒絕。

Subject 元素

Subject 子元素辨識套用策略的主體集合；此簡介集合是根據群組中的成員、角色的擁有權或個別使用者進行選擇的。它接受 *Subject* 子元素。XML 屬性可定義爲：

name。可定義物件集合的名稱。

description。可定義主體的描述。

includeType。目前不使用。

Subject 元素

Subject 子元素辨識套用策略的主體集合；此集合指出 Subject 元素所定義的集合中較特別的物件。成員可以根據角色、群組成員或只是一些個別使用者。其包含子元素第 96 頁的 「AttributeValuePair 元素」。必要的 XML 屬性爲 `type`，其辨識可從其中取得定義特殊之主體的一般物件集合。其他的 XML 屬性包括 `name`，其定義物件集合的名稱、`includeType`，其定義是否已定義物件集合，並決定策略是否適用於「非」主體成員的使用者。

備註 – 定義多重主體時，至少一項主體必須套用到使用者，才能套用策略。若主體定義的 `includeType` 設爲 false，使用者不可以是策略套用的主體之成員。

Referrals 元素

Referrals 子元素辨識策略參照集合。它接受 *Referral* 子元素。可對其定義的 XML 屬性爲 `name`，其定義物件集合的名稱及 `description` ，其接受描述。

Referral 元素

Referral 子元素辨識特定策略參照。其接受子元素 第 96 頁的 「AttributeValuePair 元素」。對其而言必要的 XML 屬性是 `type`，其辨識可從其中取得定義特殊之參照的一般指定集合。它也可包含定義集合名稱的 `name` 屬性。

Conditions 元素

Conditions 子元素辨識策略限制集合 (時間範圍、認證層級等等)。它必須包含一或多個 *Condition* 子元素。可對其定義的 XML 屬性爲 `name`，其定義物件集合的名稱及 `description` ，其接受描述。

備註 – Conditions 元素爲策略中的選擇性元素。

Condition 元素

Condition 子元素辨識特定策略限制 (時間範圍、認證層級等等)。其接受子元素 第 96 頁的 「AttributeValuePair 元素」。它的必要 XML 屬性爲 `type`，其辨識可從其中取得定義特殊的條件之一般限制集合。它也可包含定義集合名稱的 `name` 屬性。

增加啓用策略的服務

只有當服務模式的 `<Policy>` 元素配置爲 `sms.dtd` 時，才可以爲指定服務的資源定義策略。

依預設，Access Manager 提供 URL 策略代理程式服務 (`iPlanetAMWebAgentService`)。此服務於下列目錄中的 XML 檔案中定義：

```
/etc/opt/SUNWam/config/xml/
```

不過您可以增加其他策略服務到 Access Manager。一旦建立了策略服務，就可以透過 `amadmin` 指令行公用程式將其增加至 Access Manager。

▼ 增加啓用策略的服務

1. **在 XML 檔案中以 `sms.dtd` 為根據開發新的策略服務。Access Manager 提供兩種策略服務 XML 檔案，您會想要使用以下兩種檔案作為新策略服務檔案的基礎：**

   `amWebAgent.xml` - 這是預設 URL 策略代理程式服務的 XML 檔案。它位於 `/etc/opt/SUNWam/config/xml/` 中。

   `SampleWebService.xml` - 這是位於 `AccessManager-base/samples/policy` 中的範例策略服務檔案。

2. **將 XML 檔案儲存到您即將從其中載入新策略服務的目錄。例如：**

   ```
   /config/xml/newPolicyService.xml
   ```

3. **使用 `amadmin` 指令行公用程式載入新的策略服務。例如：**

   ```
   AccessManager-base/SUNWam/bin/amadmin
       --runasdn “uid=amAdmin,ou=People,default_org,
   root_suffix
       --password password
       --schema /config/xml/newPolicyService.xml
   ```

4. **載入新的策略服務後，您可以透過 Access Manager 主控台，或透過 `amadmin` 載入新策略，來定義策略定義的規則。**

建立策略

您可透過策略 API 與 Access Manager 主控台建立、修改和刪除策略，並透過 amadmin 指令行工具建立和刪除策略。您也可以使用 amadmin 公用程式在 XML 中取得和列出策略。本節重點在透過 amadmin 指令行公用程式與透過 Access Manager 主控台建立策略。如需有關策略 API 的更多資訊，請參閱「Sun Java System Access Manager 7.1 Developer's Guide」。

策略通常是以 XML 檔案建立，並透過 amadmin 指令行公用程式增加至 Access Manager，然後透過 Access Manager 主控台管理 (但可透過主控台建立策略)。這是因爲不能直接使用 amadmin 修改策略。若要修改策略，必須先從 Access Manager 刪除策略，然後使用 amadmin 加入修改後的策略。

通常策略是在範圍 (或子範圍) 層級建立，可在範圍的整個樹狀結構中使用。

▼ 使用 amadmin 建立策略

1 **根據 amadmin.dtd 建立策略 XML 檔。此檔案位於下列目錄：**

AccessManager-base/SUNWam/dtd。

下列是策略 XML 檔的範例。此範例包含所有預設的主體及條件值。如需這些值的定義，請參閱第 89 頁的 「策略類型」。

```
<Policy name="bigpolicy" referralPolicy="false" active="true" >
<Rule name="rule1">
<ServiceName name="iPlanetAMWebAgentService" />
<ResourceName name="http://thehost.thedomain.com:80/*.html" />
<AttributeValuePair>
<Attribute name="POST" />
<Value>allow</Value>
</AttributeValuePair>
<AttributeValuePair>
<Attribute name="GET" />
<Value>allow</Value>
</AttributeValuePair>
</Rule>
<Subjects name="subjects" description="desccription">
<Subject name="webservicescleint" type="WebServicesClients" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/><Value>CN=sun-unix,
OU=SUN Java System Access Manager, O=Sun, C=US</Value>
</AttributeValuePair>
</Subject>
<Subject name="amrole" type="IdentityServerRoles" includeType="inclusive">
```

```
<AttributeValuePair><Attribute name="Values"/><Value>
cn=organization admin role,o=realm1,dc=red,dc=iplanet,dc=com</Value>
</AttributeValuePair>
</Subject>
<Subject name="au" type="AuthenticatedUsers" includeType="inclusive">
</Subject>
<Subject name="ldaporganization" type="Organization" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/>
<Value>dc=red,dc=iplanet,dc=com</Value>
</AttributeValuePair>
</Subject>
<Subject name="ldapuser" type="LDAPUsers" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/>
<Value>uid=amAdmin,ou=People,dc=red,dc=iplanet,dc=com</Value>
</AttributeValuePair>
</Subject>
<Subject name="ldaprole" type="LDAPRoles" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/>
<Value>cn=Organization Admin Role,o=realm1,dc=red,dc=iplanet,dc=com</Value>
</AttributeValuePair>
</Subject>
<Subject name="ldapgroup" type="LDAPGroups" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/>
<Value>cn=g1,ou=Groups,dc=red,dc=iplanet,dc=com</Value>
</AttributeValuePair>
</Subject>
<Subject name="amidentitysubject" type="AMIdentitySubject" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/>
<Value>id=amAdmin,ou=user,dc=red,dc=iplanet,dc=com</Value>
</AttributeValuePair>
</Subject>
</Subjects>
<Conditions name="conditions" description="description">
<Condition name="ldapfilter" type="LDAPFilterCondition">
<AttributeValuePair><Attribute name="ldapFilter"/>
<Value>dept=finance</Value>
</AttributeValuePair>
</Condition>
<Condition name="authlevelge-nonrealmqualified" type="AuthLevelCondition">
<AttributeValuePair><Attribute name="AuthLevel"/>
<Value>1</Value>
</AttributeValuePair>
</Condition>
```

```
<Condition name="authlevelle-realmqaulfied" type="LEAuthLevelCondition">
<AttributeValuePair><Attribute name="AuthLevel"/>
<Value>/:2</Value>
</AttributeValuePair>
</Condition>
<Condition name="sessionproperties" type="SessionPropertyCondition">
<AttributeValuePair><Attribute name="valueCaseInsensitive"/>
<Value>true</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="a"/><Value>10</Value>
<Value>20</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="b"/><Value>15</Value>
<Value>25</Value>
</AttributeValuePair>
</Condition>
<Condition name="activesessiontime" type="SessionCondition">
<AttributeValuePair><Attribute name="TerminateSession"/>
<Value>session_condition_false_value</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="MaxSessionTime"/>
<Value>30</Value>
</AttributeValuePair>
</Condition>
<Condition name="authelevelle-nonrealmqualfied"
        type="LEAuthLevelCondition">
<AttributeValuePair><Attribute name="AuthLevel"/>
<Value>2</Value>
</AttributeValuePair>
</Condition>
<Condition name="ipcondition" type="IPCondition">
<AttributeValuePair><Attribute name="DnsName"/>
<Value>*.iplanet.com</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="EndIp"/>
<Value>145.15.15.15</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="StartIp"/>
<Value>120.10.10.10</Value>
</AttributeValuePair>
</Condition>
<Condition name="authchain-realmqualfied"
        type="AuthenticateToServiceCondition">
```

```
<AttributeValuePair><Attribute name="AuthenticateToService"/>
<Value>/:ldapService</Value>
</AttributeValuePair>
</Condition>
<Condition name="auth to realm"
    type="AuthenticateToRealmCondition">
<AttributeValuePair><Attribute name="AuthenticateToRealm"/>
<Value>/</Value>
</AttributeValuePair>
</Condition>
<Condition name="authlevelge-realmqualified"
    type="AuthLevelCondition">
<AttributeValuePair><Attribute name="AuthLevel"/>
<Value>/:2</Value>
</AttributeValuePair>
</Condition>
<Condition name="authchain-nonrealmqualfied"
   type="AuthenticateToServiceCondition">
<AttributeValuePair><Attribute name="AuthenticateToService"/>
<Value>ldapService</Value>
</AttributeValuePair>
</Condition>
<Condition name="timecondition" type="SimpleTimeCondition">
<AttributeValuePair><Attribute name="EndTime"/>
<Value>17:00</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="StartTime"/>
<Value>08:00</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="EndDate"/>
<Value>2006:07:28</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="EnforcementTimeZone"/>
<Value>America/Los_Angeles</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="StartDay"/>
<Value>mon</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="StartDate"/>
<Value>2006:01:02</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="EndDay"/>
<Value>fri</Value>
```

```
</AttributeValuePair>
</Condition>
</Conditions>
<ResponseProviders name="responseproviders"
     description="description">
<ResponseProvider name="idresponseprovidere"
   type="IDRepoResponseProvider">
<AttributeValuePair>
<Attribute name="DynamicAttribute"/>
</AttributeValuePair>
<AttributeValuePair>
<Attribute name="StaticAttribute"/>
<Value>m=10</Value>
<Value>n=30</Value>
</AttributeValuePair>
</ResponseProvider>
</ResponseProviders>
</Policy>
```

2 **策略 XML 檔案開發完成後，您可使用下列指令加以載入：**

```
AccessManager-base/SUNWam/bin/amadmin
--runasdn "uid=amAdmin,ou=People,default_org,
root_suffix"
--password password
--data policy.xml
```

若要同時加入多重策略，請將這些策略放在一個 XML 檔案中，這一點與在每個 XML 檔案中放一個策略相反。如果使用多重 XML 檔案連續快速載入策略，則內部策略索引可能會損毀，而且某些策略可能不參與策略評估。

透過 `amadmin` 建立策略時請確定：當建立認證方案條件時認證模組是以範圍註冊；當建立範圍、LDAP 群組、LDAP 角色和 LDAP 使用者時對應的 LDAP 物件範圍、群組、角色和使用者存在；當建立 `IdentityServerRoles` 主體時 Access Manager 角色存在；當建立子範圍或同級範圍參照時相關範圍存在。

請注意，在 `SubrealmReferral`、`PeerRealmReferral` 的 Value 元素之內容中，`Realm` 主體、`IdentityServerRoles` 主體、`LDAPGroups` 主體、`LDAPRoles` 主體和 `LDAPUsers` 主體必須爲完整的 DN。

▼ 以 Access Manager 主控台建立一般策略

1 **選擇您要為其建立策略的範圍。**

2 **按一下 [策略] 標籤。**

3 按一下 [策略] 清單中的 [新建策略]。

4 增加策略的名稱與說明。

5 若您要策略為使用中，請選取 [使用中] 屬性中的 [是]。

6 此時無需定義一般策略的所有欄位。您可以建立策略，隨後再加入規則、主體、條件和回應等。如需更多資訊，請參閱第 106 頁的「管理策略」。

7 按一下 [確定]。

▼ 以 Access Manager 主控台建立參照策略

1 選擇您要建立策略的範圍。

2 於 [策略] 標籤中按一下 [新增參考]。

3 增加策略的名稱與說明。

4 若您要策略為使用中，請選取 [使用中] 屬性中的 [是]。

5 此時無需定義參照策略的所有欄位。您可以建立策略，隨後再加入規則和參照等。如需更多資訊，請參閱第 106 頁的「管理策略」。

6 按一下 [確定]。

建立同級範圍與子範圍的策略

要爲同級組織或子範圍建立策略，必須先在父系範圍 (或另一個同級範圍) 中建立參照策略。參照策略的規則定義中必須包含正由子範圍管理的資源前綴。在父系範圍 (或另一個同級範圍) 中建立參照策略後，可在子範圍 (或另一個同級範圍) 建立一般策略。

在此範例中，`o=isp` 是父系範圍，`o=example.com` 是子範圍，該子範圍管理 `http://www.example.com` 的資源和子資源。

▼ 建立子範圍的策略

1 **於** o=isp **建立參照策略。如需參照策略的相關資訊，請參閱程序**第 110 頁的 「修改參照策略」。

參照策略必須定義 http://www.example.com 做爲規則中的資源，且必須包含以 example.com 做爲參照中的值之 SubRealmReferral。

2 **瀏覽至子範圍** example.com。

3 **目前，**isp **將資源參考為** example.com**，可以為資源** http://www.example.com **或任何以** http://www.example.com **開頭的資源建立一般策略**。

若要定義由 example.com 管理的其他資源之策略，必須在 o=isp 建立額外的參考策略。

將策略匯出到其他 Access Manager 實例

Access Manager 可讓您使用 amadmin 指令行工具匯出策略。當您想要將許多現有策略移到另一個 Access Manager 實例，或希望檢查您以批次模式對現有實例所做的變更時，這個工具很有用。若要匯出策略，請使用 amadmin 指令行公用程式將指定的策略匯出到檔案。語法爲：

amamdin - u *username* –w *password* –ofilename *output_file.xml* –t *policy_data_file.xml*

您可以在策略名稱中使用萬用字元 (*) 來符合任何字元的字串。

下列是 *policy_data_file.xml* 的範例：

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!--
    Copyright (c) 2005 Sun Microsystems, Inc. All rights reserved
    Use is subject to license terms.
-->

<!DOCTYPE Requests
    PUBLIC "-//iPlanet//Sun Java System Access Manager 6.2 Admin CLI DTD//EN"
    "/opt/SUNWam/dtd/amAdmin.dtd"
>>

<!--  CREATE REQUESTS -->

<!-- to export to file use option -ofilename fileName -->

<Requests>
```

```
<RealmRequests >
<RealmGetPolicies realm="/" >
<AttributeValuePair>
<Attribute name="policyName"/>
<Value>p*</Value>
</AttributeValuePair>
</RealmGetPolicies>
</RealmRequests>

<RealmRequests >
<RealmGetPolicies realm="/" >
<AttributeValuePair>
<Attribute name="policyName"/>
<Value>g10</Value>
<Value>g11</Value>
</AttributeValuePair>
</RealmGetPolicies>

</RealmRequests>
<RealmRequests >
<RealmGetPolicies realm="/realm1" >
<AttributeValuePair>
<Attribute name="policyName"/>
<Value>*</Value>
</AttributeValuePair>
</RealmGetPolicies>
</RealmRequests>

</Requests>
```

策略將匯出至 *Output_file.xml* 檔案。現在可以對檔案中包含的策略定義進行任何變更。您必須修改輸出檔案，使它與 `amadmin` 指令公用程式相容，然後才能將策略匯入到另一個 Access Manager 實例。如需如何匯入策略的說明，包括 `amadmin` 相容策略資料檔的範例，請參閱使用 amadmin 建立策略。

管理策略

建立一般策略或參照策略並加入 Access Manager 後，您即可透過 Access Manager 主控台管理策略，方法是修改規則、主體、條件與參照。

修改一般策略

透過 [策略] 標籤，您可修改用來定義存取權限的一般策略。您可定義和配置數個規則、主體、條件和資源主較程式。此節列出和說明其步驟。

▼ 增加或修改一般策略的規則

1. **若您已建立策略，按一下您要增加規則的策略名稱。若還沒建立，請參閱****第 103 頁的「以 Access Manager 主控台建立一般策略」**。

2. **於 [規則] 功能表下，按一下 [新建]**。

3. **為規則選取下列預設服務類型之一。啓用策略的服務越多，您可以參閱的清單就越大：**

探索服務	定義探索服務查詢的授權動作，並修改 Web 服務用戶端對特定資源的協定呼叫。
Liberty 個人設定檔服務	定義 Liberty 個人設定檔服務查詢的授權動作，並修改 Web 服務用戶端對特定資源的協定呼叫。
URL 策略代理程式	定義 URL 策略代理程式服務的授權動作。可用來定義保護 HTTP 及 HTTPS URL 的策略。這是 Access Manager 策略最常用的使用案例。

4. **按 [下一步]**。

5. **輸入規則的名稱與資源名稱**。

   目前，Access Manager 策略代理程式僅支援 `http://` 與 `https://` 資源，而不支援以 IP 位址取代主機名稱。

   協定、主機、連接埠及資源名稱等均支援使用萬用字元。例如：

   ```
   http*://*:*/*.html
   ```

   對 URL 策略代理程式服務而言，若未輸入連接埠埠號，則 `http://` 的預設埠號爲 80、`https://` 的預設埠號爲 443。

6. **為此規則選取動作。依據服務類型，您可選取下列項目：**

   - 查尋 (探索服務)
   - 更新 (探索服務)
   - 修改 (Liberty 個人設定檔服務)
   - 查詢 (Liberty 個人設定檔服務)
   - GET (URL 策略代理程式)
   - POST (URL 策略代理程式)

7. **選取動作值**。

   - 互動同意 — 呼叫 Liberty 互動協定以達成資源同意。此值僅用於 Liberty 個人設定檔服務類型。

- 互動值 — 呼叫 Liberty 互動協定以取得資源上的值。此值僅用於 Liberty 個人設定檔服務類型。
- 允許 — 可讓您存取與規則中定義的資源相符的資源。
- 拒絕 — 不允許您存取與規則中定義的資源相符的資源。

  策略中的拒絕規則總是要優先於允許規則。例如，如果指定的資源有兩種策略，一種是拒絕存取，另一種是允許存取，則結果是拒絕存取 (假如同時滿足這兩種策略條件)。由於拒絕策略可能導致這兩種策略之間產生潛在的衝突，因此建議您使用拒絕策略時要非常謹慎。通常，策略定義程序應該僅使用允許規則，在所有策略均不適於完成此拒絕存取時才使用預設拒絕規則。

  如果使用明確的拒絕規則，即使有一個或多個策略允許存取，透過不同主體如角色和或群組成員身份爲給定使用者指定的策略也可能會導致拒絕對資源存取。例如，如果存在一個適用於員工角色之資源的拒絕策略，還存在另一個適用於管理員角色之相同資源的允許策略，系統將會拒絕指定給使用者 (員工角色和管理員角色) 的策略決策。

  解決此問題的一種方法爲使用條件外掛程式設計策略。在上述情況中，「角色條件」(將拒絕策略套用於認證爲員工角色之使用者，並將允許策略套用至認證爲經理角色之使用者) 協助區分這兩種策略。另一種方法爲使用 `authentication level` 條件，其中管理員角色是在較高認證層級進行認證。

8 **按一下 [完成]。**

▼ 增加或修改一般策略的主體

1 **若您已建立策略，按一下您要增加主體的策略名稱。若您尚未建立策略，請參閱****第 103 頁的 「以 Access Manager 主控台建立一般策略」**。

2 **於 [主體] 清單下，按一下 [新建]。**

3 **選取其中一個預設主體類型。如需主體類型的說明，請參閱****第 90 頁的 「主體」**

4 **按 [下一步]。**

5 **輸入此主體的名稱。**

6 **選取或取消選取 [排除] 欄位。**

  如果未選取此欄位 (預設)，則此策略將套用於屬於此主體成員的身份。如果選取此欄位，則此策略將套用於**不屬於**此主體成員的身份。

  如果策略中存在多重主體，並且至少一個主體表示策略套用於給定身份，則策略將套用於此身份。

7 執行搜尋，以便顯示要加入至此主體的識別。此步驟不適用於 [已認證的使用者] 主體或 [Web 服務用戶端] 主體。

預設 (*) 搜尋式樣將顯示所有合格的項目。

8 選取要為此主體加入的個別身份，或按一下 [全部加入] 以立即加入所有身份。按一下 [新增]，以將識別移至選取的清單。此步驟不適用於認證使用者主體。

9 按一下 [完成]。

10 若要從策略中移除某主體，請選取此主體並按一下 [刪除]。按一下主體名稱可以編輯任何主體定義。

▼ 將條件增加至一般策略

1 若您已建立策略，按一下您要增加規則的策略名稱。若您尚未建立策略，請參閱第 103 頁的 「以 Access Manager 主控台建立一般策略」。

2 於 [條件] 清單下，按一下 [新建]。

3 選取條件類型並按 [下一步]。

4 定義條件類型的欄位。

5 按一下 [完成]。

▼ 將回應提供者增加至一般策略

1 若您已建立策略，按一下您要增加回應提供者的策略名稱。若您尚未建立策略，請參閱第 103 頁的 「以 Access Manager 主控台建立一般策略」。

2 於 [回應提供者] 清單下，按一下 [新建]。

3 輸入回應提供者的名稱。

4 定義下列值：

StaticAttribute — 這些是格式爲屬性值的靜態屬性，定義在儲存於策略中的 `IDResponseProvider` 的實例內。

DynamicAttribute — 此處所選擇的回應屬性首先需要於對應之範圍的「策略配置服務」中定義。定義的屬性名稱應該是那些存在於所配置資料存放區 (IDReposotories) 中的屬性名稱的子集。如需如何定義屬性的詳細資料，請參閱「策略配置」屬性定義。若要選取特定或多個屬

性，請按住 Control 鍵，並按一下滑鼠左鍵。

5 **按一下 [完成]。**

6 **若要從策略中移除回應提供者，請選取主體，然後按一下 [刪除]。按一下名稱可以編輯任何回應提供者定義。**

修改參照策略

您可將範圍的策略定義和決策委派其他使用參照策略的範圍。自訂參照可用以從任何策略目標點取得策略決策。建立參照策略後，可增加或修改關聯的規則、參照和資源提供者。

▼ 增加或修改參照策略的規則

1 **若您已建立策略，按一下您要增加規則的策略名稱。若還沒建立，請參閱第 104 頁的「以 Access Manager 主控台建立參照策略」。**

2 **於 [規則] 功能表下，按一下 [新建]。**

3 **為規則選取下列預設服務類型之一。啓用策略的服務越多，您可以參閱的清單就越大：**

探索服務	定義探索服務查詢的授權動作，並修改 Web 服務用戶端對特定資源的協定呼叫。
Liberty 個人設定檔服務	定義 Liberty 個人設定檔服務查詢的授權動作，並修改 Web 服務用戶端對特定資源的協定呼叫。
URL 策略代理程式	定義 URL 策略代理程式服務的授權動作。可用來定義保護 HTTP 及 HTTPS URL 的策略。這是 Access Manager 策略最常用的使用案例。

4 **按 [下一步]。**

5 **輸入規則的名稱與資源名稱。**

目前，Access Manager 策略代理程式僅支援 `http://` 與 `https://` 資源，而不支援以 IP 位址取代主機名稱。

協定、主機、連接埠及資源名稱等均支援使用萬用字元。例如：

```
http*://*:*/*.html
```

對 URL 策略代理程式服務而言，若未輸入連接埠埠號，則 `http://` 的預設埠號爲 80、`https://` 的預設埠號爲 443。

備註 – 步驟 6 與 7 不適用於參照策略。

6 **按一下 [完成]。**

▼ 增加或修改策略的參照

1 **若您已建立策略，按一下您要增加回應提供者的策略名稱。若您尚未建立策略，請參閱****第 104 頁的 「以 Access Manager 主控台建立參照策略」**。

2 **於 [參照] 清單下，按一下 [新增]。**

3 **定義 [規則] 欄位中的資源。這些欄位包括：**

參照— 顯示目前的參照類型。

名稱— 輸入參照的名稱。

資源名稱— 輸入資源的名稱。

篩選器— 指定將於 [值] 欄位中顯示之範圍名稱的篩選器。依預設，將顯示所有範圍名稱。

值— 選取參照的範圍名稱。

4 **按一下 [完成]。**

若要從策略中移除某個參照，請選取此參照，然後按一下 [刪除]。

可以透過按一下參照名稱旁邊的 [編輯] 連結，編輯任何參照定義。

▼ 將回應提供者增加至參照策略

1 **若您已建立策略，按一下您要增加回應提供者的策略名稱。若您尚未建立策略，請參閱****第 104 頁的 「以 Access Manager 主控台建立參照策略」**。

2 **於 [回應提供者] 清單下，按一下 [新建]。**

3 **輸入回應提供者的名稱。**

4 **定義下列值：**

StaticAttribute	這些是格式爲屬性值的靜態屬性，定義在儲存於策略中的 `IDResponseProvider` 的實例內。
DynamicAttribute	此處所選擇的回應屬性首先需要於對應之範圍的「策略配置服務」中定義。定義的屬性名稱應該是那些存在於所配置資料存放

區 (IDReposotories) 中的屬性名稱的子集。如需如何定義屬性的詳細資料，請參閱「策略配置」屬性定義。若要選取特定或多個屬性，請按住 Control 鍵，並按一下滑鼠左鍵。

5 **按一下 [完成]。**

6 **若要從策略中移除回應提供者，請選取主體，然後按一下 [刪除]。按一下名稱可以編輯任何回應提供者定義。**

策略配置服務

策略配置服務用來爲每個組織透過 Access Manager 主控台配置每個策略相關屬性。您也可定義資源名稱實作和 Directory Server 資料存放區，以和 Access Manager 策略架構一起使用。[策略配置服務] 中指定的 Directory Server 用於 LDAP 使用者、LDAP 群組、LDAP 角色和組織策略主體的成員身份評估。

持續的主體結果時間

若要改善策略評估表現，成員身份評估將快取一段時間 (以策略配置服務中 [主體結果存在時間] 屬性中定義的時間爲基準)。將一直使用這些快取成員身份決策，直到 [主體結果存在時間] 屬性定義之時間結束。在這之後，成員身份評估會用於反映目錄中使用者的目前狀態。

動態屬性

這些爲允許的動態屬性名稱，其顯示於清單中，並可選取以定義策略回應提供者動態屬性。定義的名稱需要與資料儲存庫中定義的屬性名稱相同。

amldapuser 定義

`amldapuser` 是在安裝中建立的使用者，預設由 [策略配置] 服務中指定的 Directory Server 使用。若有必要，範圍的管理員或策略管理員可變更此值。

加入策略配置服務

建立範圍時，會自動設定範圍的 [策略配置] 服務屬性。然而，若有必要您可加以修改。

基於資源的認證

有些組織需要有進階認證方案，使用者可根據特定模組、根據試圖存取的資源進行認證。基於資源的認證是 Access Manager 的一項功能，使用者必須通過用以保護資訊的特定認證模組的認證，而非預設認證模組。此功能僅適用於首次使用者認證。

備註 – 這是與第 83 頁的 「階段作業升級」中描述的基於資源認證不同的功能。該特定功能並不具有任何限制。

限制

基於資源的認證有下列限制：

- 若適用於資源的策略具有多重認證模組，系統將任意選取一個認證模組。
- 層級和方案是唯一可以爲此策略定義的條件。
- 此功能不能跨不同 DNS 網域運作。

▼ 配置基於資源的認證

Access Manager 和策略代理程式都安裝好之後，就可以配置基於資源的認證。要這樣做，必須先將 Access Manager 指向 Gateway servlet。

1 **開啓** `AMAgent.properties`。

`AMAgent.properties` 可以在 `/etc/opt//SUNWam/agents/config/` 中找到 (於 Solaris 環境中)。

2 **註釋下面的行：**

```
#com.sun.am.policy.am.loginURL = http://Access
Manager_server_host.domain_name:port/amserver/UI/Login.
```

3 **增加下列行到檔案中：**

```
com.sun.am.policy.am.loginURL =
http://AccessManager_host.domain_name:port/amserver/gateway
```

備註 – 閘道 servlet 使用策略評估 API 開發，並可用來撰寫自訂機制以完成基於資源的認證。請參閱「Access Manager 開發者指南」中「Sun Java System Access Manager 7.1 Developer's Guide」中的第 3 章「Using the Policy APIs」。

---

4 **重新啓動代理程式。**

◆ ◆ ◆ 第 6 章

管理主體

主體介面在一個範圍內啓用基本識別管理。您建立於主體介面中的識別可用於以「Access Manager 身份識別主體」類型建立之策略的主體定義中。

您可以建立與修改的識別爲：

使用者

使用者代表個人的識別。可於群組中建立與刪除使用者，並可在角色和/或群組中增加或移除。您亦可爲使用者指定服務。

▼ 建立或修改使用者

1 **按一下 [使用者] 標籤**。

2 **按一下 [開啓新檔]**。

3 **輸入下列欄位的資料：**

使用者 ID。此欄位中爲登入 Access Manager 的使用者名稱。此特性可爲非 DN 值。

名字。此欄位中爲使用者的名字。

姓氏。此欄位中爲使用者的姓氏。

全名 — 此欄位中爲使用者的全名。

密碼。— 此欄位中爲 [使用者 ID] 欄位中所指定名稱的密碼。

密碼 (確認) — 確認密碼。

使用者狀態。此選項指出是否允許使用者透過 Access Manager 認證。

4 按一下 [建立]。

5 一旦建立了使用者，您可以按一下使用者名稱來編輯使用者資訊。如需使用者屬性的資訊，請參閱「使用者」屬性。您可執行的其他修改：

- 第 115 頁的 「建立或修改使用者」
- 第 116 頁的 「將使用者增加至角色或群組」
- 第 116 頁的 「將服務增加至識別」

▼ 將使用者增加至角色或群組

1 按一下您要修改的使用者名稱。

2 選取角色或群組。僅會顯示已指定給使用者的角色與群組。

3 從 [可用的] 清單選取角色或群組並按一下 [增加]。

4 一旦角色或群組顯示於 [選取的] 清單中，按一下 [儲存]。

▼ 將服務增加至識別

1 選取您要增加服務的識別。

2 按一下 [服務] 標籤。

3 按一下 [加入]。

4 依據您所選取的識別類型，將顯示下列服務清單：

- 認證配置
- 探索服務
- Liberty 個人設定檔服務
- 階段作業
- 使用者

5 選取您要增加的服務，並按 [下一步]。

6 編輯服務的屬性。如需有關服務的說明，請按一下步驟 4 中的服務名稱。

7 **按一下 [完成]**。

代理程式設定檔

Access Manager 策略代理程式保護 Web 伺服器與 Web 代理伺服器上的內容以防止未授權的侵入。它們根據管理員所配置的策略控制對服務與 Web 資源的存取。

代理程式物件定義策略代理程式設定檔，可讓 Access Manager 儲存認證及其他與保護 Access Manager 資源之特定代理程式有關的設定檔資訊。經由 Access Manager 主控台，管理員可以檢視、建立、修改與刪除代理程式設定檔。

在代理程式物件建立頁面，可以定義代理程式認證至 Access Manager 所需的 UID/密碼。若您具有使用同一 Access Manager 建立的多重 Web 容器，您可以對不同代理程式啓用多重 ID，並由 Access Manager 個別地啓用與停用。您亦可集中管理代理程式的某些喜好設定值，而不需在每台機器上編輯 `AMAgent.properties`。

▼ 建立或修改代理程式

1 **按一下 [代理程式] 標籤**。

2 **按一下 [開啓新檔]**。

3 **輸入下列欄位值：**

名稱。 輸入代理程式的名稱或識別。這是代理程式將用來登入 Access Manager 的名稱。不接受多位元的名稱。

密碼。 輸入代理程式密碼。此密碼必須與 LDAP 認證期間代理程式所使用的密碼不同。

確認密碼。確認密碼。

裝置狀態。輸入代理程式的裝置狀態。若設爲 [使用中]，代理程式將可以認證至 Access Manager，並與其通訊。若設爲 [非使用中]，代理程式將無法認證至 Access Manager。

4 **按一下 [建立]**。

5 **一旦您建立了代理程式，您可以另外編輯下列欄位：**

描述。輸入代理程式的簡要描述。例如，您可以輸入代理程式實例名稱或其保護的應用程式的名稱。

代理程式金鑰值。以一個「金鑰/值」對設定代理程式特性。此特性爲 Access Manager 所使用，以接收有關使用者之憑證指定的代理程式請求。目前，僅有一個特性有效，且將忽略所有其他特性。請使用以下格式：

agentRootURL=`protocol://` *hostname:port/*

此項目必須精確，而且 agentRootURL 大小寫相符。

protocol　代表所使用的通訊協定，可能是 HTTP 或 HTTPS。

hostname　代表代理程式所在電腦的主機名稱。這部電腦也託管代理程式所保護的資源。

port　代表安裝代理程式的連接埠號碼。代理程式在這個連接埠上偵聽內送的流量，並截取存取主機資源的所有請求。

配置 Access Manager 以保護 Cookie 免遭劫持

Cookie 遭受劫持係指侵入者 (駭客，可能使用不受信任的應用程式) 對 Cookie 進行未經授權的存取。若被劫持的 Cookie 是階段作業 Cookie，視系統配置方式而定，Cookie 劫持可能增加對受保護 Web 資源的未經授權存取威脅。

Sun 文件提供一份技術說明，標題爲「Precautions Against Session-Cookie Hijacking in an Access Management Deployment」，其中說明了要對抗與階段作業 Cookie 劫持有關的特定安全威脅可以採取的預防措施。請參閱下列文件：

「Technical Note: Precautions Against Cookie Hijacking in an Access Manager Deployment」

篩選的角色

篩選的角色是使用 LDAP 篩選器建立的動態角色。建立角色時，所有使用者都會透過篩選器的篩選並指定給此角色。篩選器會在項目中尋找任何屬性值對 (例如，ca=user*)，並自動將包含該屬性的使用者指定給角色。

▼ 建立篩選角色

1. **於 [瀏覽] 窗格中，移至將建立角色的組織。**
2. **按一下 [開啓新檔]。**
3. **輸入篩選角色的名稱。**
4. **輸入搜尋條件的資訊。**
   例如，

```
(&(uid=user1)(|(inetuserstatus=active)(!(inetuserstatus=*))))
```

若篩選器依預設爲空白，將建立下列角色：

```
(objectclass = inetorgperson)
```

5 按一下 [建立] 以根據篩選條件初始搜尋。由篩選條件定義的識別將自動地指定給角色。

6 一旦建立了篩選角色，按一下角色名稱以檢視屬於角色的使用者。您亦可按一下 [服務] 標籤來增加服務至角色。

角色

角色的成員是角色的 LDAP 項目。角色自己的條件已定義爲含屬性的 LDAP 項目，由項目的辨別名稱 (Distinguished Name, DN) 屬性所辨識。一旦建立了角色，您可以手動增加服務與使用者。

▼ 建立或修改角色

1 按一下 [角色] 標籤。

2 於 [角色] 清單中按一下 [新建]。

3 輸入角色的名稱。

4 按一下 [建立]。

▼ 增加使用者至角色或群組

1 按一下您要增加使用者的角色或群組名稱。

2 按一下 [使用者] 標籤。

3 從 [可用的] 清單選取您要增加的使用者並按一下 [增加]。

4 一旦使用者顯示於 [選取的] 清單中，按一下 [儲存]。

群組

群組代表具有共同功能、特性或興趣的使用者集合。通常，此群組並無與之相關聯的權限。群組可存在於兩個層級；於組織內及於其他受管理群組內。

▼ 建立或修改群組

1. **按一下 [群組] 標籤**。
2. **按一下 [群組] 清單上的 [新建]**。
3. **輸入群組的名稱**。
4. **按一下 [建立]**。
   一旦您建立了群組，您可以按一下群組名稱與 [使用者] 標籤，將使用者增加至群組。

第 2 部分

目錄管理和預設服務

這是「Sun Java System Access Manager 7.1 管理指南」的第二部分。「目錄管理」一章中描述以「舊有」模式部署 Access Manager 時，管理「目錄」物件的方法。其他章節描述配置與使用某些 Access Manager 預設服務的方法。本部分包含以下章節：

◆ ◆ ◆ 第 7 章

7 目錄管理

只有在舊有模式下安裝 Access Manager 時，才會顯示 [目錄管理] 標籤。此目錄管理功能爲啓用 Sun Java System Directory Server 的 Access Manager 部署提供了識別管理解決方案。

如需「舊有模式」安裝選項的更多資訊，請參閱「Sun Java Enterprise System 5 Installation Guide for UNIX」

管理目錄物件

[目錄管理] 標籤包含檢視與管理 Directory Server 物件所需的所有元件。本節說明物件類型及有關如何配置它們的詳細資訊。使用 Access Manager 主控台或指令行介面可以定義、修改或刪除使用者、角色、群組、組織、子組織及容器物件。主控台有具權限程度不同的預設管理員，用以建立與管理目錄物件。(可基於角色建立其他管理員。)當與 Access Manager 一起安裝時，可於 Directory Server 內定義管理員。您可管理的 Directory Server 物件有：

組織

組織代表企業用來管理其部門與資源的階層式結構之頂層。安裝時，Access Manager 會動態建立頂層組織 (安裝期間定義) 以管理 Access Manager 企業配置。安裝後可以建立其他組織以管理個別企業。所有建立的組織均位於頂層組織之下。

▼ 建立組織

1 **按一下 [目錄管理] 標籤。**

2 **在 [組織] 清單中，按一下 [新建]。**

3 **輸入欄位的值。僅 [名稱] 是必需的。這些欄位包括：**

名稱 輸入組織名稱的值。

網域名稱 輸入組織的完整領域名稱系統 (DNS) 名稱 (如果有)。

組織狀態 選擇 [**使用中**] 或 [**非使用中**] 狀態。預設值爲 [**使用中**]。在組織存在期間，可以透過選取 [特性] 圖示隨時變更該狀態。如果選擇**非作用中**，系統會在使用者登入組織時停用使用者存取。

組織別名 此欄位定義組織的別名，可讓您使用這些別名經由 URL 登入進行認證。例如，如果您有一個名爲 `exampleorg` 的組織，並將 `123` 與 `abc` 定義爲別名，則您可使用以下任一個 URL 登入該組織：

```
http://machine.example.com/amserver/UI/Login?org=exampleorg

http://machine.example.com/amserver/UI/Login?org=abc

http://machine.example.com/amserver/UI/Login?org=123
```

組織別名在整個組織中必須是唯一的。您可以使用 [唯一屬性清單] 強制唯一性。

DNS 別名名稱 允許爲組織的 DNS 名稱加入別名。此屬性僅接受「實際的」網域別名 (不允許使用隨機字串)。例如，如果您有一個名爲 `example.com` 的 DNS，並將 `example1.com` 與 `example2.com` 定義爲組織 `exampleorg` 的別名，則您可使用以下任一個 URL 登入該組織：

```
http://machine.example.com/amserver/UI/

Login?org=exampleorg

http://machine.example1.com/amserver/

UI/Login?org=exampleorg

http://machine.example2.com/amserver/

UI/Login?org=exampleorg
```

唯一的屬性清單 允許您在組織中加入使用者的唯一屬性名稱清單。例如，如果您加入了指定電子郵件位址的唯一屬性名稱，則無法建立兩個具有相同

電子郵件位址的使用者。此欄位還可以接受以逗號分隔的清單。清單中的任一屬性名稱均定義唯一性。例如，如果欄位包含屬性名稱清單：

```
PreferredDomain, AssociatedDomain
```

而且爲特定使用者將 `PreferredDomain` 定義爲 `http://www.example.com`，則對該 URL 而言，此以逗號分隔的整個清單定義是唯一的。將命名屬性 ou 增加至 [唯一的屬性清單] 將不會對預設群組、使用者容器強制執行唯一性。(ou=Groups,ou=People)。

此一唯一性同時針對所有子組織強制執行。

備註 – 在 [範圍] 模式中無法設定唯一的屬性。在 [舊有] 模式中，也無法在基於 7.0 或 7.1 的主控台中設定它們。若要建立唯一屬性清單，必須登入基於 6.3 的主控台。如需更多資訊，請參閱第 19 頁的「舊有模式 6.3 主控台」。

---

4 **按一下 [確定]。**

新組織會顯示於 [組織] 清單中。若要編輯您建立組織時定義的任一特性，請按一下您要編輯的組織之名稱、變更其特性，然後按一下 [儲存]。

▼ 刪除組織

1 **勾選將要刪除的組織之名稱旁的核取方塊。**

2 **按一下 [刪除]。**

備註 – 執行刪除時不會顯示警告訊息。組織中的所有項目將被刪除，且無法執行還原。

將組織加入到策略

Access Manager 物件會透過策略的主體定義加入策略。當建立或修改策略時，可以將組織、角色、群組及使用者定義爲主體。一旦定義了主體，策略即會套用於物件。如需更多資訊，請參閱第 106 頁的「管理策略」。

容器

當由於物件類別與屬性差異而無法使用組織項目時，會使用**容器**項目。請切記，Access Manager 容器項目與 Access Manager 組織項目不一定等於 LDAP 物件類別 `organizationalUnit` 與 `organization`。它們是抽象的識別項目。理想情況下，將使用組織項目而不是容器項目。

備註 – 容器的顯示是選擇性的。若要檢視容器，您必須在 [配置] > [主控台特性] 下選取 [管理] 服務的 [顯示容器]。

▼ 要建立容器

1. **選取組織或容器的位置連結，新容器將會建立於其中。**
2. **按一下 [容器] 標籤。**
3. **按一下 [容器] 清單中的 [新建]。**
4. **輸入將要建立的容器之名稱。**
5. **按一下 [確定]。**

▼ 要刪除容器

1. **按一下 [容器] 標籤。**
2. **選取要刪除的容器名稱旁邊的核取方塊。**
3. **按一下 [刪除]。**

備註 – 刪除一個容器將會同時刪除該容器中存在的所有物件。包含所有物件和子容器。

群組容器

群組容器用於管理群組。它僅可包含群組與其他群組容器。群組容器「群組」會動態指定爲所有受管理群組的父系項目。如果需要，可以加入附加群組容器。

備註 – 群組容器的顯示是選擇性的。若要檢視群組容器，您必須從 [配置] > [主控台特性] 的 [認證] 服務中選取 [啓用群組容器]。

▼ 建立群組容器

1. **選取包含新群組容器的組織或群組容器的位置連結。**
2. **選取 [群組容器] 標籤。**
3. **按一下 [群組容器] 清單中的 [新增]。**
4. **在 [名稱] 欄位中輸入值，然後按一下 [確定]。新的群組容器會顯示於 [群組容器] 清單中。**

▼ 刪除群組容器

1. **導覽至包含要刪除的群組容器之組織。**
2. **選擇 [群組容器] 標籤。**
3. **選取要刪除的群組容器旁邊的核取方塊。**
4. **按一下 [刪除]。**

群組

群組代表包含一般功能、特性或興趣的使用者集合。通常，此群組並無與之相關聯的權限。群組可以兩個層級存在；於組織內及於其他受管理群組內。存在於其他群組中的群組稱爲**子群組**。子群組是「實際上」存在於父系群組中的子節點。

Access Manager 還支援 **巢式群組**，巢式群組是單一群組中包含現有群組的「陳述」。巢式群組與子群組不同，它可存在於 DIT 中任何之處。它們可讓您爲大量使用者快速設置存取權限。

您可建立的群組有兩種：靜態群組與動態群組。您只能以手動方式將使用者加入靜態群組；動態群組則透過篩選器控制使用者的加入。巢式群組與子群組皆可加入這兩種類型的群組。

靜態群組

靜態群組是根據您指定之管理的群組類型所建立的。群組成員是使用 groupOfNames 或 groupOfUniqueNames 物件類別增加到群組項目。

備註 – 依預設，受管理群組類型爲動態。您可在管理服務配置中變更該預設。

動態群組

動態群組是透過使用 LDAP 篩選器所建立。所有項目都會透過器篩選並動態指定給群組。篩選器可尋找項目中的任一屬性，並傳回包含該屬性的項目。例如，如果要根據建立編號建立群組，可以使用篩選器傳回包含建立編號屬性的所有使用者的清單。

備註 – 應使用 Directory Server 將 Access Manager 配置爲可使用 referential integrity 外掛程式。啓用後的參考完整性外掛程式會在刪除或重新命名工作完成後，立即對指定的屬性執行完整性更新。這可確保在整個資料庫中維持相關項目之間的關係。資料庫索引可增強中的搜尋效能。如需有關啓用此外掛程式的更多資訊，請參閱「Sun Java Access Manager 6 Migration Guide」。

▼ 建立靜態群組

1. **瀏覽將於其中建立新群組的組織、組或群組容器。**
2. **按一下 [群組] 清單的 [新建靜態]。**
3. **在 [名稱] 欄位中輸入群組的名稱。按 [下一步]。**
4. **選取 [使用者可以訂閱該群組] 屬性以允許使用者自行訂閱群組。**
5. **按一下 [確定]。**
   建立群組之後，您便可以選取群組的名稱並按一下 [一般] 標籤，來編輯 [使用者可以訂閱至此群組] 屬性。

▼ 加入或移除靜態群組成員

1. **在 [群組] 清單中選取將對其加入成員的群組。**
2. **在 [選取動作] 功能表中選擇要執行的動作。您可以執行的動作如下所示：**
   - 新建使用者　此動作會建立新的使用者並在儲存該使用者資訊時將其加入群組。
   - 加入使用者　此動作將現有使用者加入群組。選取此動作時，您會建立指定所要加入的使用者之搜尋條件。用於建構條件的欄位會使用 ANY 或 ALL 運算子。

ALL 會傳回所有指定欄位的使用者。ANY 會傳回任一指定欄位的使用者。如果保留某欄位空白，則該欄位將符合該特定屬性的所有可能項目。

建構了此搜尋條件後，即按一下 [下一步]。從傳回的使用者清單中，選取您要加入的使用者，然後按一下 [完成]。

加入群組　此動作將巢式群組加入目前群組。選擇此動作時，您建立了搜尋條件，包括搜尋範圍、群組名稱 (接受「*」萬用字元)，並且您可以指定使用者是否可以自行訂閱群組。輸入資訊後，即按一下 [下一步]。從傳回的群組清單中，選取您要加入的群組，然後按一下 [完成]。

移除成員　此動作將從群組中移除成員 (包括使用者與群組)，但不會刪除它們。選取您要移除的成員，然後從 [選取動作] 功能表中選取 [移除成員]。

刪除成員　此動作將永久刪除您選取的成員。選取您要刪除的成員，然後選擇 [刪除成員]。

▼ 建立動態群組

1 **瀏覽將於其中建立新群組的組織或群組。**

2 **按一下 [群組] 標籤。**

3 **按一下 [新建動態]。**

4 **在 [名稱] 欄位中輸入群組的名稱。**

5 **建構 LDAP 搜尋篩選器。**

依預設，Access Manager 顯示基本搜尋篩選器介面。用於建構篩選器的 [基本] 欄位使用 ANY 或 ALL 運算子。ALL 會傳回所有指定欄位的使用者。ANY 會傳回任一指定欄位的使用者。如果保留某欄位空白，則該欄位將符合該特定屬性的所有可能項目。

6 **按一下 [確定] 後，符合搜尋條件的所有使用者會自動加入群組。**

▼ 若要加入或移除動態群組的成員

1 **在 [群組] 清單中，按一下要對其加入成員的群組之名稱。**

2 **在 [選取動作] 功能表中選擇要執行的動作。您可以執行的動作如下所示：**

加入群組　此動作將巢式群組加入目前群組。選擇此動作時，您建立了搜尋條件，包括搜尋範圍、群組名稱 (接受「*」萬用字元)，並且您可以指定使用者是否可以自行訂閱群組。輸入資訊後，即按一下 [下一步]。從傳回的群組清單中，選取您要加入的群組，然後按一下 [完成]。

移除成員　此動作將從群組中移除成員 (包括群組)，但不刪除它們。選取您要移除的成員，然後選擇 [移除成員]。

刪除成員　此動作將永久刪除您選取的成員。選取您要刪除的成員，然後選擇 [刪除成員]。

將群組加入到策略

Access Manager 物件會透過策略的主體定義加入策略。當建立或修改策略時，可以將組織、角色、群組及使用者定義爲策略主體頁面中的主體。一旦定義了主體，策略即會套用於物件。如需更多資訊，請參閱第 106 頁的 「管理策略」。

使用者容器

使用者容器是預設的 LDAP 組織單元，爲在組織中建立使用者時，所有使用者的指定位置。可以在組織層級和使用者容器層級找到使用者容器 (作爲子使用者容器)。它們僅可包含其他使用者容器與使用者。如果需要，可以將附加使用者容器加入組織。

備註 – 使用者容器的顯示是選擇性的。若要檢視使用者容器，必須在 [管理服務] 中選取 [啓用使用者容器]。

▼ 建立使用者容器

1. **導覽至要在其中建立新使用者容器的組織或使用者容器。**
2. **按一下 [使用者容器] 清單中的 [新建]。**
3. **輸入要建立的使用者容器名稱。**
4. **按一下 [確定]。**

▼ 刪除使用者容器

1. **導覽至包含要刪除的使用者容器之組織或使用者容器。**
2. **選取要刪除的使用者容器名稱旁邊的核取方塊。**
3. **按一下 [刪除]。**

備註 – 刪除一個使用者容器將會同時刪除該使用者容器中存在的所有物件。包含所有使用者和子使用者容器。

使用者

使用者代表個別使用者的識別。透過 Access Manager 識別管理模組，您可以在組織、容器以及群組中建立和刪除使用者；在角色和/或群組中加入或移除使用者。您亦可對使用者指定服務。

備註 – 如果子組織內的使用者是使用與 `amadmin` 相同的使用者 ID 建立的，`amadmin` 的登入會失敗。若發生此問題，管理員應透過 Directory Server 主控台變更使用者的 ID。如此可使管理員登入到預設組織中。此外，認證服務中的 [啓動使用者搜尋] 可以設爲使用者容器，以確保登入時傳回獨特的比對結果。

▼ 建立使用者

1. **導覽至要在其中建立使用者的組織、容器或使用者容器。**
2. **按一下 [使用者] 標籤。**
3. **按一下使用者清單上的 [新建]。**
4. **輸入下列值的資料：**

   使用者 ID　此欄位採用其將登入 Access Manager 的使用者名稱。此特性可爲非 DN 值。

   名字　此欄位中爲使用者的名字。[目前登入] 欄位中的 [名字] 值和 [姓氏] 值可識別使用者。這並非必須填寫的值。

   姓氏　此欄位中爲使用者的姓氏。[名字] 的值與 [姓氏] 的值會識別使用者身份。

   全名　此欄位中爲使用者的全名。

   密碼　此欄位中爲 [使用者 ID] 欄位中所指定名稱的密碼。

   密碼 (確認)　確認密碼。

   使用者狀態　此選項指出是否允許使用者透過 Access Manager 認證。只有作用中的使用者才可以認證。預設值爲 [**使用中**]。

5. **按一下 [確定]。**

▼ 若要編輯使用者設定檔

當尚未被指定管理員角色的使用者進行 Access Manager 認證時，預設的檢視爲使用者自己的 [使用者設定檔]。 此外，具適當權限的管理員可以編輯使用者設定檔。在此檢視中，使用者可以修改其個人設定檔的特定屬性值。[使用者設定檔] 檢視中顯示的屬性可以延伸。如需加入物件與識別的自訂屬性相關之更多資訊，請參閱「Access Manager 開發者指南」。

1 選取要編輯其設定檔的使用者。依預設，會顯示 [一般] 檢視。

2 編輯下列欄位：

名字	此欄位中爲使用者的名字。
姓氏	此欄位中爲使用者的姓氏。
全名	此欄位中爲使用者的全名。
密碼	按一下 [編輯] 連結以加入並確認使用者密碼。
電子郵件位址	此欄位中爲使用者的電子郵件位址。
雇員編號	此欄位中爲使用者的員工號碼。
電話號碼	此欄位中爲使用者的電話號碼。
家庭住址	此欄位中爲使用者的家庭住址。
使用者狀態	此選項指出是否允許使用者透過 Access Manager 認證。只有使用中的使用者才可以透過 Access Manager 進行認證。預設值爲使用中。可以從下拉式功能表中選取以下任一選項：。 ■ [使用中] — 使用者可以透過 Access Manager 進行認證。 ■ [非使用中] — 使用者無法透過 Access Manager 進行認證，但使用者設定檔仍儲存在該目錄。

備註 – 將使用者狀態變更爲非作用中僅會影響透過 Access Manager 進行認證的動作。Directory Server 使用 *nsAccountLock* 屬性來決定使用者帳號狀態。針對 Access Manager 認證停用的使用者帳號仍可執行毋須 Access Manager 便可執行的作業。若要使目錄中的使用者帳號成爲非使用中 (不僅僅只針對 Access Manager 認證)，請將 *nsAccountLock* 的值設爲 false。若您網站中經授權的管理員定期會將使用者停用，可考慮將 *nsAccountLock* 屬性加入 Access Manager [使用者設定檔] 頁面。如需詳細資訊，請參閱「Sun Java System Access Manager 7.1 Developer's Guide」。

帳號過期日期	如果存在該屬性，則當目前日期和時間超過指定的帳號過期日期時，認證服務將不允許登入。此屬性的格式爲 *mm/dd/yyyy hh:mm*。
使用者認證配置	此屬性設定使用者的認證鏈。
使用者別名清單	此欄位定義可以套用於使用者的別名清單。若要使用此屬性中配置的任何別名，必須將 `iplanet-am-user-alias-list` 屬性加入 LDAP 服務的 [使用者項目搜尋屬性] 欄位，來修改 LDAP 服務。
語言環境個人喜好	此欄位指定使用者的語言環境。
成功的 URL	此屬性指定使用者認證成功後將重新導向至的 URL。
失敗的 URL	此屬性指定使用者認證失敗後將重新導向至的 URL。
密碼重設選項	這是用來選取忘記密碼頁面中問題之選項，目的在取得忘記的密碼。
使用者探索資源提供	設定使用者的 [使用者探索] 服務的資源提供。
MSISDN 編號	定義在使用 MSISDN 認證時使用者的 MSISDN 編號。

▼ 將使用者增加至角色與群組

1. **按一下 [使用者] 標籤。**
2. **按一下您要修改的使用者名稱。**
3. **選取 [角色] 或 [群組] 標籤。**
4. **選取您希望在其中加入使用者的角色或群組，然後按一下 [新增]。**
5. **按 [儲存]。**

備註 – 若要從 [角色] 或 [群組] 移除使用者，請選取角色或群組，然後按一下 [移除]，再按一下 [儲存]。

將使用者加入到策略

Access Manager 物件會透過策略的主體定義加入策略。當建立或修改策略時，可以將組織、角色、群組及使用者定義爲策略主體頁面中的主體。一旦定義了主體，策略即會套用於物件。如需更多資訊，請參閱第 106 頁的 「管理策略」。

角色

角色爲類似**群組**概念的一種 Directory Server 項目機制。群組具有成員；角色也具有成員。角色的成員爲擁有該角色的 LDAP 項目。角色自己的條件已定義爲含屬性的 LDAP 項目，爲項目的識別名稱 (DN) 屬性所辨識。Directory Server 具有數種不同類型的角色，但 Access Manager 只能管理它們的其中之一：受管理角色。

備註 – 其他 Directory Server 角色類型仍可於目錄部署中使用，但無法被 Access Manager 主控台管理。其他 Directory Server 類型則可用於策略的主題定義。如需策略主體相關的更多資訊，請參閱第 99 頁的 「建立策略」。

使用者可擁有一種或多種角色。例如，可以建立具有階段作業服務屬性和密碼重設服務屬性的承包人角色。新承包人雇員加入公司時，管理員可將該角色指定給他們，而不是在承包人項目中設定各自的屬性。若承包人在工程部門工作，且需要適用於工程員工的服務與存取權，那麼管理員可將承包人指派爲工程角色與承包人角色。

Access Manager 使用角色以套用存取控制指令。首次安裝時，Access Manager 會配置定義管理員權限的存取控制指令 (ACI)。系統會接著在角色 (如組織管理角色和組織 Help Desk 管理角色) 中指定這些 ACI， 將這些角色指定給使用者時，會定義使用者的存取權限。

只有在 [管理服務] 中啓用了 [在使用者設定檔頁面上顯示角色] 屬性，使用者才可檢視指定給他們的角色。

備註 – 應使用 Directory Server 將 Access Manager 配置爲可使用 referential integrity 外掛程式。啓用後的參考完整性外掛程式會在刪除或重新命名工作完成後，立即對指定的屬性執行完整性更新。這可確保在整個資料庫中維持相關項目之間的關係。資料庫索引可增強中的搜尋效能。

角色分兩種類型：

- 靜態 — 若要建立靜態角色，在建立角色階段毋須加入使用者即可。角色建立完成後，便可對其加入特定使用者。這樣可讓您在將使用者加入指定角色時，可進行更多的控制。
- 動態 – 動態角色的建立是透過 LDAP 篩選器的使用完成。建立角色時，所有使用者都會透過篩選器的篩選並指定給角色。篩選器會在項目中尋找任何屬性值對 (例如，`ca=user*`)，並自動將包含該屬性的使用者指定給角色。

▼ 建立靜態角色

1 **移至將建立角色的組織。**

2 **按一下 [角色] 標籤。**

配置組織時會建立一組預設角色，它們會顯示於 [角色] 清單中。預設角色爲：

容器說明桌面管理員。容器說明桌面管理員角色對組織單元中的所有項目均具有讀取權限，但是僅對此容器單元中使用者項目的 `userPassword` 屬性具有寫入權限。

組織說明桌面管理員。組織說明桌面管理員對組織中所有項目皆有讀取權限，對 `userPassword` 屬性則有寫入權限。

備註 – 建立子組織時，請記住在子組織中建立管理角色，而不是在父系組織中建立。

容器管理員。容器管理員角色對 LDAP 組織單元中的所有項目均具有讀取寫入權限。在 Access Manager 中，LDAP 組織單元通常稱爲容器。

組織策略管理員。組織策略管理員具有對所有策略的讀取寫入權限，可以建立、指定、修改和刪除此組織內的所有策略。

使用者管理員。依預設，新建組織中的任何使用者項目均爲該組織的使用者容器的成員。使用者管理員對組織中的所有使用者項目均具有讀取寫入存取權限。請記住，此角色對包含角色與群組 DN 的屬性「並不」具有讀取寫入權限，因此，它們不能修改角色或群組的屬性，也不能從中移除使用者。

備註 – 可以透過 Access Manager 配置其他容器，使其具有使用者項目、群組項目甚至是其他容器。若要將管理員角色套用於配置組織後建立的容器，將會使用預設的容器管理員角色或容器說明桌面管理員。

群組管理員。建立群組時建立的群組管理員對特定群組的所有成員均具有讀取寫入權限，可以建立新的使用者、將使用者指定給其管理的群組以及刪除其建立的使用者。

建立群組時將自動產生群組管理員角色，其具有管理群組的必要權限。不會自動將此角色指定給群組成員。此角色必須由群組建立者或任何具有群組管理員角色存取權限的人員指定。

頂層管理員。頂層管理員對頂層組織中的所有項目均具有讀取寫入權限。換句話說，此頂層管理員角色具有 Access Manager 應用程式中每個配置主體所擁有的權限。

組織管理員。組織管理員對組織中的所有項目均具有讀取寫入權限。建立群組時將自動產生組織管理員角色，其具有管理組織的必要權限。

3 **按一下 [新建靜態] 按鈕。**

4 **輸入角色的名稱。**

5 **輸入角色的描述。**

6 **從 [類型] 功能表選擇角色類型。**

角色可以爲「管理」角色或「服務」角色。主控台使用角色類型決定在 Access Manager 主控台中啓動使用者的位置。管理角色會通知主控台，該角色的擁有者具有管理權限；服務角色會通知主控台，該擁有者爲一般使用者。

7 **從 [存取權限] 功能表，選擇預設的權限集以套用至該角色。具有這些權限，便可以存取組織中的項目。顯示的預設許可權未依特定順序排列。這些權限為：**

沒有權限	對角色不設定權限。
組織管理員	組織管理員對配置組織中的所有項目均具有讀取寫入權限。
組織說明桌面管理員	組織說明桌面管理員具有對已配置組織中所有項目的讀取權限，以及對 `userPassword` 屬性的寫入權限。
組織策略管理員	組織策略管理員對組織中的所有策略均具有讀取寫入權限。組織策略管理員無法建立同級組織的參考策略。 通常，「無權限 ACI」會指定給「服務」角色，而爲「管理」角色指定任一預設 ACI。

▼ 將使用者加入到靜態角色

1 **按一下您要增加使用者的角色名稱。**

2 **在 [成員] 清單中，從 [選取動作] 功能表選取 [加入使用者]。**

3 **輸入搜尋條件的資訊。可以選擇基於一個或多個顯示的欄位搜尋使用者。這些欄位包括：**

符合	可讓您對篩選選取您要包含的欄位。`ALL` 會傳回所有指定欄位的使用者。`ANY` 會傳回任一指定欄位的使用者。
名字	依據其名字搜尋使用者。
使用者 ID	依據使用者 ID 搜尋使用者。
姓氏	依據其姓氏搜尋使用者。
全名	依據其全名搜尋使用者。
使用者狀態	依據使用者的狀態 (作用中或非作用中) 搜尋使用者。

4 **按一下 [下一步] 以開始搜尋。會顯示搜尋的結果。**

5 **透過選取使用者名稱旁邊的核取方塊，從傳回的名稱中選擇使用者。**

6 **按一下 [完成]。**
使用者即會指定給角色。

▼ 若要建立動態角色

1 **移至將建立角色的組織。**

2 **按一下 [角色] 標籤。**
配置組織時會建立一組預設角色，它們會顯示於 [角色] 清單中。預設角色爲：

容器說明桌面管理員。容器說明桌面管理員角色對組織單元中的所有項目均具有讀取權限，但是僅對此容器單元中使用者項目的 `userPassword` 屬性具有寫入權限。

組織說明桌面管理員。組織說明桌面管理員對組織中所有項目皆有讀取權限，對 `userPassword` 屬性則有寫入權限。

備註 – 建立子組織時，請記住在子組織中建立管理角色，而不是在父系組織中建立。

容器管理員。容器管理員角色對 LDAP 組織單元中的所有項目均具有讀取寫入權限。在 Access Manager 中，LDAP 組織單元通常稱爲容器。

組織策略管理員。組織策略管理員具有對所有策略的讀取寫入權限，可以建立、指定、修改和刪除此組織內的所有策略。

使用者管理員依預設，新建組織中的任何使用者項目均爲該組織的使用者容器的成員。[使用者管理員] 對組織中的所有使用者項目均具有讀取寫入存取權限。請記住，此角色對包含角色與群組 DN 的屬性「並不」具有讀取寫入權限，因此，它們不能修改角色或群組的屬性，也不能從中移除使用者。

備註 – 可以透過 Access Manager 配置其他容器，使其具有使用者項目、群組項目甚至是其他容器。若要將管理員角色套用於配置組織後建立的容器，將會使用預設的容器管理員角色或容器說明桌面管理員。

群組管理員。群組建立時建立的群組管理員對特定群組的所有成員均具有讀取寫入權限，可以建立新的使用者、將使用者指定給其管理的群組以及刪除其建立的使用者。

建立群組時將自動產生群組管理員角色，其具有管理群組的必要權限。不會自動將此角色指定給群組成員。此角色必須由群組建立者或任何具有群組管理員角色存取權限的人員指定。

頂層管理員。頂層管理員對頂層組織中的所有項目均具有讀取寫入權限。換句話說，此頂層管理員角色具有 Access Manager 應用程式中每個配置主體所擁有的權限。

組織管理員。組織管理員對組織中的所有項目均具有讀取寫入權限。建立群組時將自動產生組織管理員角色，其具有管理組織的必要權限。

3 **按一下 [新建動態] 按鈕**。

4 **輸入角色的名稱**。

5 **輸入角色的描述**。

6 **從 [類型] 功能表選擇角色類型**。

角色可以爲「管理」角色或「服務」角色。主控台使用角色類型決定在 Access Manager 主控台中啓動使用者的位置。管理角色會通知主控台，該角色的擁有者具有管理權限；服務角色會通知主控台，該擁有者爲一般使用者。

7 **從 [存取權限] 功能表，選擇預設的權限集以套用至該角色。具有這些權限，便可以存取組織中的項目。顯示的預設許可權未依特定順序排列。這些權限為：**

沒有權限	對角色不設定權限。
組織管理員	組織管理員對配置組織中的所有項目均具有讀取寫入權限。
組織說明桌面管理員	組織說明桌面管理員具有對已配置組織中所有項目的讀取權限，以及對 `userPassword` 屬性的寫入權限。
組織策略管理員	組織策略管理員對組織中的所有策略均具有讀取寫入權限。組織策略管理員無法建立同級組織的參考策略。 通常，「無權限 ACI」會指定給「服務」角色，而爲「管理」角色指定任一預設 ACI。

8 **輸入搜尋條件的資訊。這些欄位包括：**

符合	允許您在希望篩選所包含的任何欄位中納入運算子。`ALL` 會傳回所有指定欄位的使用者。`ANY` 會傳回任一指定欄位的使用者。
名字	依據其名字搜尋使用者。
使用者 ID	依據使用者 ID 搜尋使用者。
姓氏	依據其姓氏搜尋使用者。
全名	依據其全名搜尋使用者。
使用者狀態	依據使用者的狀態 (作用中或非作用中) 搜尋使用者。

9 **按一下 [確定] 以根據篩選條件開始搜尋。由篩選條件定義的使用者將自動地指定給角色**。

▼ 從角色移除使用者

1. **導覽至包含要修改之角色的組織。**
   從 [識別管理] 模組的 [檢視] 功能表中選取 [組織]，然後選取 [角色] 標籤。
2. **選取要修改的角色。**
3. **從 [檢視] 功能表選擇 [使用者]。**
4. **選取要移除的每個使用者旁邊的核取方塊。**
5. **按一下 [選取動作] 功能表中的 [移除] 使用者。**
   使用者即會從角色中移除。

將角色加入策略

Access Manager 物件會透過策略的主體定義加入策略。當建立或修改策略時，可以將組織、角色、群組及使用者定義爲策略主體頁面中的主體。一旦定義了主體，策略即會套用於物件。如需更多資訊，請參閱第 106 頁的 「管理策略」。

◆ ◆ ◆ 第 8 章

8 目前階段作業

本章描述 Access Manager 之階段作業管理功能。[階段作業管理] 模組爲檢視使用者階段作業資訊和管理使用者階段作業提供了解決方案。它追蹤各個階段作業時間並允許管理員終止階段作業。系統管理員應忽視 [平台伺服器] 清單中所列的 [負載平衡器] 伺服器。

目前階段作業介面

[目前階段作業] 模組介面允許具有適當權限的管理員，檢視目前登入至 Access Manager 的任何使用者之階段作業資訊。

階段作業管理

階段作業管理框架顯示目前受管理的 Access Manager 名稱。

階段作業資訊

[階段作業資訊] 視窗顯示目前登入至 Access Manager 的所有使用者，並且顯示每位使用者的階段作業時間。這些顯示欄位包括：

使用者 ID。顯示目前登入使用者的使用者 ID。

剩餘時間。顯示必須重新認證之前，使用者此階段作業所具有的剩餘時間 (以分鐘計算)。

最長階段作業時間。顯示使用者在階段作業過期而必須重新認證以重新取得存取權限之前可以登入的最長時間 (以分鐘計算)。

閒置時間。顯示使用者已閒置的時間 (以分鐘計算)。

最長閒置時間。顯示在必須重新認證之前，使用者可以閒置的最長時間 (以分鐘計算)。

時間限制由管理員在階段作業管理服務中定義。

在 [使用者 ID] 欄位中輸入字串，然後按一下 [篩選]，可以顯示某個特定的使用者階段作業或使用者階段作業的特定範圍。允許使用萬用字元。

按一下 [重新整理] 按鈕，將更新使用者階段作業顯示。

終止階段作業

具有適當權限的管理員可以隨時終止使用者階段作業。

▼ 若要終止階段作業

1. **選取您要終止的使用者階段作業**。
2. **按一下 [終止]**。

◆ ◆ ◆ 第 9 章

9 密碼重設服務

Access Manager 提供「密碼重設」服務，可讓使用者重設他們用於存取 Access Manager 所保護的特定服務或應用程式的密碼。「密碼重設」服務屬性由頂層管理員定義，可控制驗證憑證 (以機密提問的形式)、控制新建或現有密碼通知的機制以及設定驗證不正確之使用者的鎖定持續時間。

本章包含下列小節：

註冊密碼重設服務

使用者所屬範圍不需要註冊密碼重設服務。如果使用者所屬組織中不存在密碼重設服務，它將繼承在 [服務配置] 中爲此服務定義的值。

▼ 為不同範圍中的使用者註冊密碼重設

1. **瀏覽至您將為使用者註冊密碼的範圍。**
2. **按一下範圍名稱，然後按一下 [服務] 標籤。**
   若尚未加入範圍，按一下 [新增] 按鈕。
3. **選取 [密碼重設]，然後按 [下一步]。**
   將會顯示[密碼重設] 服務屬性。有關屬性定義的描述，請參閱線上說明。
4. **按一下 [完成]。**

配置密碼重設服務

註冊密碼重設服務後，該服務必須由擁有管理員權限的使用者配置。

▼ 若要配置服務

1. **選取要註冊 [密碼重設] 服務的範圍。**
2. **按一下 [服務] 標籤。**
3. **按一下服務清單中的 [密碼重設]。**
4. **會顯示密碼重設屬性，可讓您定義 [密碼重設] 服務的需求。確保已啓用密碼重設服務 (預設為啓用)。至少必須定義以下屬性：**
   - 使用者驗證
     - 機密提問
     - 連結 DN
     - 連結密碼

     [連結 DN] 屬性必須包含擁有重設密碼權限的使用者 (例如說明桌面管理員)。由於 Directory Server 有所限制，因此當連結 DN 爲 `cn=Directory Manager` 時，[密碼重設] 便不起作用。

     其餘屬性均爲選擇性的。如需服務屬性的描述，請參閱線上說明。

     備註 – Access Manager 會自動安裝密碼重設 Web 應用程式，以便產生隨機密碼。但是，您可以寫入自己的外掛程式類別，以產生和通知密碼。請參閱位於以下位置的 Readme.html 檔案，以取得這些外掛程式類別的範例。

     PasswordGenerator:

     ```
     AccessManager-base/SUNWam/samples/console/PasswordGenerator
     ```

     NotifyPassword:

     ```
     AccessManager-base/SUNWam/samples/console/NotifyPassword
     ```

     ---
5. **如果使用者要定義其唯一的個人提問，則選取 [啓用個人提問] 屬性。定義屬性後，按一下 [儲存]。**

▼ 本土化機密提問

如果您正在執行 Access Manager 本土化版本，並想以特定於您的語言環境的字元集來顯示機密提問，請執行下列動作：

1. **在 [密碼重設] 服務中，於 [機密提問] 屬性下的 [目前的值] 清單中，增加機密提問關鍵字。例如，**`favorite-color`。

2. **將這個關鍵字與您想顯示此關鍵字值的問題增加到** `amPasswordReset.properties` **檔案。例如：**

   ```
   favorite-color=What is your favorite color?
   ```

3. **將相同的關鍵字與本土化的問題增加至位於** `/opt/SUNWam/locale` **中的** `AMPasswordReset_`*locale*`.properties`。**當使用者嘗試變更他們的密碼時，就會顯示本土化的問題。**

密碼重設鎖定

密碼重設服務包含鎖定功能，此功能限制使用者正確回答其機密提問前可以嘗試的次數。鎖定功能透過密碼重設服務屬性來配置。如需服務屬性的描述，請參閱線上說明。密碼重設支援兩種類型的鎖定，記憶體鎖定和實體鎖定。

記憶體鎖定

鎖定是暫時的，只有當 [密碼重設失敗鎖定持續時間] 屬性的值大於 0，且 [啓用密碼重設失敗鎖定] 屬性已啓用時時才有效用。該鎖定將防止使用者透過密碼重設 Web 應用程式重設密碼。此鎖定會持續 [密碼重設失敗鎖定持續時間] 中指定的時間，或直到伺服器重新啓動。如需服務屬性的描述，請參閱線上說明。

實體鎖定

該鎖定爲一種比較永久的鎖定。當 [密碼重設失敗鎖定計數] 屬性的值設爲 0，且 [啓用密碼重設失敗鎖定] 屬性已啓用時，若使用者回答機密提問答案錯誤，其使用者帳號狀態會變更爲非作用中。如需服務屬性的描述，請參閱線上說明。

一般使用者的密碼重設

以下小節描述使用者使用密碼重設服務的情況。

自訂密碼重設

啓用了密碼重設服務且管理員定義了屬性後，使用者即可登入 Access Manager 主控台，以便自訂其機密提問。

▼ 若要自訂密碼重設

1. **在使用者名稱和密碼成功通過認證後，使用者登入主控台。**
2. **在 [使用者設定檔] 頁面中，使用者選取密碼重設選項。系統會顯示 [可用提問回答] 畫面。**
3. **系統會為使用者顯示管理員為服務定義的提問，如：**
   - 您的寵物叫什麼名字？
     - 您最喜愛哪個電視節目？
     - 您母親的婚前姓是什麼？
     - 您最喜歡的飯店是哪家？
4. **使用者可以選取機密提問，最多不超過管理員為範圍定義的最大問題數 (最大問題數在 [密碼重設服務] 中定義)。然後，使用者提供對所選問題的回答。這些問題與回答為重設使用者密碼的依據 (請參閱下一小節)。如果管理員選取了 [啓用個人提問] 屬性，系統會提供文字欄位，讓使用者輸入特有的機密提問及其回答。**
5. **使用者按一下 [儲存]。**

重設遺忘密碼

如果使用者遺忘密碼，可使用密碼重設網路應用程式隨機產生新密碼，並通知使用者此新密碼。遺忘密碼的典型情形如下：

▼ 重設遺忘密碼

1. **使用者從管理員為他們提供的 URL 登入到密碼重設網路應用程式。例如：**

   `http://hostname:port/ampassword` (適用於預設範圍)

   或

   `http://hostname: port/`*deploy_uri*`/UI/PWResetUserValidation?realm=realmname`，其中 realmname 是範圍的名稱。

   備註 – 若父系範圍的 [密碼重設] 服務沒有啓用，但其子範圍的啓用了，使用者必須使用以下語法存取服務：

   `http://hostname: port/`*deploy_uri*`/UI/PWResetUserValidation?realm=realmname`

   ---

2. **使用者輸入使用者 ID。**

3 **系統向使用者顯示在密碼重設服務中定義且在自訂期間被使用者選取的個人提問。如果使用者先前未登入 [使用者設定檔] 頁面且未自訂個人提問，則不會產生密碼。**

使用者正確回答提問後，系統會產生新密碼並使用電子郵件將其傳送給該使用者。無論使用者是否正確回答了提問，系統均會將嘗試通知傳送給該使用者。爲了接收新密碼和嘗試通知，使用者必須在 [使用者設定檔] 頁面中輸入自己的電子郵件位址。

密碼策略

密碼策略是一組規則，用來規範密碼在指定目錄中的使用方式。密碼策略通常透過 Directory Server 主控台定義在 Directory Server 之中。透過強制以下作業，安全密碼策略可以將密碼被容易猜出的風險降到最低：

- 使用者必須依據排程變更密碼。
- 使用者必須提供比較特殊的密碼。
- 數次輸入錯誤密碼後，系統可能會鎖定帳號。

Directory Server 提供在樹的任一節點設定密碼策略的多種方法，而且存在多種設定策略的方法。如需詳細資訊，請參閱

「Directory Server Enterprise Edition 6.0 管理指南」中的「Directory Server 密碼策略」。

備註 – 在 Directory Server 中，密碼策略包含屬性 `passwordExp`，用於定義使用者密碼是否會在指定的秒數後過期。如果管理員將 `passwordExp` 屬性設定爲 `on`，則會設定使用者密碼的過期以及設定 Access Manager 管理帳號 (例如 `amldap`、`dsame` 及 `puser`) 的過期。當 Access Manager 管理員的帳號密碼過期，而一般使用者已登入，則該使用者會看見密碼變更螢幕。但是，Access Manager 不會指定密碼變更螢幕所屬的使用者。在此情況下，此螢幕是專供管理員使用，一般使用者無法變更密碼。

若要解決此問題，管理員必須登入 Directory Server 並變更 `amldap`、`dsame` 及 `puser` 的密碼，或將 `passwordExpirationTime` 屬性改爲未來的某個時間。

◆ ◆ ◆ 第 10 章

10 記錄服務

Sun Java™ System Access Manager 提供記錄服務，以記錄如使用者作業、流量模式和授權違規等資訊。此外，除錯檔案可幫助管理員排解安裝的疑難。

記錄檔

記錄檔記錄其監視的每個服務的許多事件。管理員應定期查看這些檔案。記錄檔的預設目錄是 `/var/opt/SUNWam/logs` (SPARC 系統)、`/var/opt/sun/identity` (Linux 系統)、`/var/opt/sun/identity` (HP-UX) 及 *jes-install-dir*`\identity` (Windows)。藉由使用 Access Manager 主控台，可在 [記錄服務] 中配置記錄檔目錄。

如需預設記錄檔類型、記錄何種資訊以及記錄檔格式之詳細清單的資訊，請參閱「Sun Java System Access Manager 技術摘要」之「Sun Java System Access Manager 7.1 Technical Overview」中的「Logging Overview」。

有關 [記錄服務] 的屬性定義，請按一下 Access Manager 主控台中的 [說明] 按鈕以查閱線上說明。

Access Manager 服務記錄

服務記錄檔有兩種類型：存取及錯誤。「存取」記錄檔包含嘗試登入與成功登入的記錄。「錯誤」記錄檔記錄 Access Manager 服務中的錯誤。平面記錄檔附加的副檔名爲 `.error` 或 `.access`。資料庫欄位的名稱是以 `_ERROR` 或 `_ACCESS` 結束 (Oracle 資料庫)，或以 `_error` 或 `_access` 結束 (MySQL 資料庫)。例如，平面檔案記錄主控台事件名爲 `amConsole.access`，而記錄同一事件的資料庫欄位名爲 `AMCONSOLE_ACCESS`。以下各節說明記錄服務所記錄的記錄檔。

階段作業記錄檔

記錄服務記錄以下階段作業服務事件：

- 登入
- 登出
- 階段作業閒置逾時
- 階段作業最長逾時
- 登入失敗
- 階段作業重新啓動
- 階段作業銷毀

階段作業記錄檔的前綴是 amSSO。

主控台記錄檔

Access Manager 主控台記錄檔記錄識別相關物件、策略和服務的建立、刪除與修改，其中包括組織、組織單元、使用者、角色、策略和群組。它也記錄使用者屬性的修改，包括密碼以及增加或移除角色和群組中的使用者。此外，主控台記錄檔也寫入委託和資料存放區作業。主控台記錄檔的前綴是 amConsole。

認證記錄檔

認證元件記錄使用者的登入和登出。認證記錄檔的前綴是 amAuthentication。

聯合記錄檔

「聯合」元件記錄聯合相關事件，例如 (但不限於) 建立「認證網域」和建立「寄存提供者」。聯合記錄檔的前綴是 amFederation。

策略記錄檔

策略元件記錄策略相關事件，包括 (但不限於) 策略管理 (策略的建立、刪除和修改) 和策略評估。策略記錄檔的前綴是 amPolicy。

代理程式記錄檔

策略代理程式記錄檔負責記錄有關允許或拒絕使用者存取之記錄資源的異常。代理程式記錄檔的前綴是 amAgent。amAgent 記錄檔只存在於代理程式伺服器中。在 Access Manager 伺服器上於「認證記錄檔」中記錄代理程式事件。如需有關此功能的更多資訊，請參閱有疑問的策略代理程式的相關文件。

SAML 記錄檔

SAML 元件記錄 SAML 相關事件，包括 (但不限於) 指定和工件的建立或移除、回應和請求的詳細資訊以及 SOAP 錯誤。階段作業記錄檔的前綴是 amSAML。

amadmin 記錄檔

指令行記錄檔記錄使用指令行工具的作業中發生的事件錯誤。包括 (但不限於) 載入服務模式、建立策略和刪除使用者。指令行記錄檔的前綴是 amAdmin。amadmin.access 及 amadmin.error 記錄檔位於主記錄目錄的子目錄中。依預設，amadmin 指令行工具記錄檔位於 /var/opt/SUNWam/logs 中。

記錄功能

[記錄服務] 有數個特定功能，可加以啓用以執行額外的功能。包括啓用「安全記錄」、「指令行記錄」和「遠端記錄」。

安全記錄

此選擇性的功能可將額外安全性加入記錄功能中。啓用安全記錄後，可以偵測對安全記錄檔進行的未授權變更或竄改。使用此功能不需用特殊編碼。「安全記錄」是藉由使用由系統管理員配置的預先註冊憑證來達成。會爲每個記錄檔記錄產生和儲存此「清單分析和憑證 (MAC)」。會定期插入特定「簽名」記錄檔記錄，表示寫入該點之記錄內容的簽名。兩種記錄的組合可確保記錄沒有被竄改。有兩個方法可以啓用安全記錄：透過 Java Security Server (JSS) 提供者，以及透過 Java Cryptography Extension (JCE) 提供者。

▼ 透過 JSS 提供者啓用「安全記錄」

1. **以 Logger 名稱建立憑證，然後將其安裝在執行 Access Manager 的部署容器中。**

   如需 Application Server 的說明，請參閱「Sun Java System Application Server Enterprise Edition 8.2 Administration Guide」中的「Working with Certificates and SSL」。

   如需 Web Server 的說明，請參閱「Sun Java System Web Server 7.0 Administrator's Guide」中的「Managing Certificates」。

2. **在 Access Manager 主控台中，開啓 [記錄服務] 配置中的 [安全記錄]，並儲存此變更。管理員亦可修改 [記錄服務] 中其他屬性的預設值。**

   若記錄目錄預設值 (/var/opt/SUMWam/logs) 有所變更，請確定將其權限設爲 0700。若此目錄不存在，記錄服務會建立此目錄，但它會建立權限設爲 0755 的目錄。

此外，若您指定和預設不同的目錄，必須將 Web 容器的 server.policy 檔案中的以下參數更改爲新的目錄：

```
permission java.io.FilePermission “/var/opt/SUNWam/logs/*”,”delete,write”
```

3 **在包含憑證資料庫密碼的** *AccessManager-base*/SUNWam/config **目錄中建立檔案，並將之命名為** .wtpass。

備註 – 其檔名和路徑可在 AMConfig.properties 檔中配置。如需更多資訊，請參閱「*Access Manager Administration Reference*」中 AMConfig.properties 檔案參照一章裡的「Certificate Database」。

請確定部署容器使用者爲因安全性理由對此檔案有讀取權限的管理員。

---

4 **重新啓動伺服器**。

應清除安全記錄目錄，因爲當啓動安全記錄時，部份易引起誤解的驗證錯誤可能會被寫入 /var/opt/SUNWam/debug/amLog 檔案。

若要偵測安全記錄檔中有無未授權的變更或竄改，請查看驗證程序寫入 /var/opt/SUNWam/debug/amLog 的錯誤訊息。若要手動檢查竄改，請執行 VerifyArchive 公用程式。如需更多資訊，請參閱「*Access Manager Administration Reference*」中的 VerifyArchive 指令行一章。

▼ 透過 JCE 提供者啓用安全記錄

1 **使用 Java** keytool **指令建立稱為 Logger 的憑證，並將它安裝在 JKS 金鑰庫中。例如：**

```
JAVA-HOME/jre/lib/security/Logger.jks
```

如需 Application Server 的說明，請參閱「Sun Java System Application Server Enterprise Edition 8.2 Administration Guide」中的「Working with Certificates and SSL」。

如需 Web Server 的說明，請參閱「Sun Java System Web Server 7.0 Administrator's Guide」中的「Managing Certificates」。

2 **在 Access Manager 主控台中，開啓 [記錄服務] 配置中的 [安全記錄]，並儲存此變更。管理員亦可修改 [記錄服務] 中其他屬性的預設值。**

若記錄目錄預設值 (/var/opt/SUMWam/logs) 有所變更，請確定將其權限設爲 0700。若此目錄不存在，記錄服務會建立此目錄，但它會建立權限設爲 0755 的目錄。

此外，若您指定和預設不同的目錄，必須將 Web 容器的 server.policy 檔案中的以下參數更改爲新的目錄：

```
permission java.io.FilePermission “/var/opt/SUNWam/logs/*”,”delete,write”
```

3 **在包含 JKS 金鑰庫密碼的** *AccessManager-base*/SUNWam/config **目錄中建立檔案，並將之命名為** .wtpass。

備註 – 其檔名和路徑可在 AMConfig.properties 檔中配置。如需更多資訊，請參閱「*Access Manager Administration Reference*」中 AMConfig.properties 檔案參照一章裡的「Certificate Database」。

請確定部署容器使用者爲因安全性理由對此檔案有讀取權限的管理員。

4 在位於 *AccessManager-base*/config/xml 目錄的 amLogging.xml 中，編輯下列項目：

```
sun-am-logging-secure-log-helper

                <AttributeSchema name="iplanet-am-logging-secure-log-helper"
                    type="single"
                    syntax="string"
                    i18nKey="">
                    <DefaultValues>
                        <Value>com.sun.identity.log.secure.impl.SecureLogHelperJCEImpl</Value>
                    </DefaultValues>
                </AttributeSchema>

 sun-am-logging-secure-certificate-store

                <AttributeSchema name="iplanet-am-logging-secure-certificate-store"
                    type="single"
                    syntax="string"
                    i18nKey="">
                    <DefaultValues>
                        <Value>/dir-to-signing-cert-store/Logger.jks</Value>
                    </DefaultValues>
                </AttributeSchema>
```

5 刪除現有的服務模式 iPlanetAMLoggingService。例如：

```
./amadmin -u amadmin -w netscape -r iPlanetAMLoggingService
```

6 使用 amadmin 指令行工具將編輯好的 amLoging.xml 載入到 Access Manager。例如：

```
./amadmin -u amadmin -w netscape -s /etc/opt/SUNWam/config/xml/amLogging.xml
```

7 重新啓動伺服器。

若要偵測安全記錄中有無未授權的變更或竄改， 請查看驗證程序寫入 /var/opt/SUNWam/debug/amLog 的錯誤訊息。若要手動檢查竄改，請執行 VerifyArchive 公用程式。如需更多資訊，請參閱「*Access Manager Administration Reference*」中的 VerifyArchive 指令行一章。

指令行記錄

amadmin 指令行工具可建立、修改和刪除 Directory Server 中的識別物件 (例如組織、使用者、角色)。此工具也可載入、建立和註冊服務範本。[記錄服務] 可啓用 -t 選項來記錄這些動作。若啓用 (ACTIVE) AMConfig.properties 中的 com.iplanet.am.logstatus 特性，則會建立記錄檔記錄。(依預設會啓用此特性。)指令行記錄檔的前綴是 amAdmin。如需更多資訊，請參閱「*Access Manager Administration Reference*」中的「The amadmin Command Line Tool」。

記錄特性

AMConfig.properties 檔中有一些特性會影響記錄的輸出：

com.iplanet.am.logstatus=ACTIVE	此特性可啓用或停用記錄。預設爲 ACTIVE。
iplanet-am-logging.*service*.level= *level*	*service* 爲服務的正常記錄檔檔名。例如，若要指定 amSAML.access 的記錄等級，請使用特性 iplanet-am-logging.amSAML.access.level。*level* 爲 java.util.logging.Level 值的其中一個，表示記錄於記錄檔中之詳細資訊的等級。等級可爲 OFF、SEVERE、WARNING、INFO、CONFIG、FINE、FINER、FINEST 及 ALL。大多數服務所記錄的詳細資訊不會高於 INFO 記錄等級。

遠端記錄

Access Manager 支援遠端記錄。從而允許用戶端應用程式 (使用安裝有 Access Manager SDK 的主機) 在部署於遠端機器上的 Access Manager 實例中建立記錄檔記錄。遠端記錄可在以下情況下被啓動：

1. 當 Access Manager 實例的 [記錄服務] 中的記錄 URL 指向遠端實例，且二者之間配置爲信任關係，記錄將寫入遠端 Access Manager 實例。
2. 當 Access Manager SDK 是針對遠端 Access Manager 實例而安裝，且在 SDK 伺服器上執行的用戶端 (或簡單 Java 類別) 使用記錄 API，記錄將寫入遠端 Access Manager 機器。
3. 當記錄 API 是由 Access Manager 代理程式所使用。

▼ 使用 Web 容器啓用遠端記錄

1 登入 Application Server 或 Web Server 的管理主控台，並增加下列 JVM 選項：

- `java.util.logging.manager=com.sun.identity.log.LogManager`
- `java.util.logging.config.file=/` *AccessManager-base* `/SUNwam/lib/LogConfig.properties`

  如需有關 Application Server 管理主控台的更多資訊，請參閱「Sun Java System Application Server Enterprise Edition 8.2 Administration Guide」。

  如需 Web Server 管理主控台的更多資訊，請參閱「Sun Java System Web Server 7.0 Administrator's Guide」。
- 若使用的 Java™ 2 Platform, Standard Edition 爲 1.4 或更高版本，在指令行中執行以下指令將完成此步驟：

  `java -cp /`*AccessManager-base* `/SUNWam/lib/am_logging.jar:/` *AccessManager-base* `/SUNWam/lib/xercesImpl.jar:/` *AccessManager-base* `/SUNWam/lib/xmlParserAPIs.jar:/` *AccessManager-base* `/SUNWam/lib/jaas.jar:/` *AccessManager-base* `/SUNWam/lib/xmlParserAPIs.jar:/` *AccessManager-base* `/SUNWam/lib/servlet.jar:/` *AccessManager-base* `/SUNWam/locale:/` *AccessManager-base*`/SUNWam/lib/am_services.jar:/` *AccessManager-base*`/SUNWam/lib/am_sdk.jar:/` *AccessManager-base*`/SUNWam/lib/jss311.jar:/` *AccessManager-base*`/SUNWam/lib:.`

  `-Djava.util.logging.manager=com.sun.identity.log.LogManager`

  `-Djava.util.logging.config.file=/` *AccessManager-base* `/SUNwam/lib/LogConfig.properties`
- 若使用的 Java 2 Platform, Standard Edition 版本低於 1.4，在指令行中執行以下指令將完成此步驟：

  `java -Xbootclasspath/a:/` *AccessManager-base* `/SUNWam/lib/jdk_logging.jar -cp /`*AccessManager-base* `/SUNWam/lib/am_logging.jar:/` *AccessManager-base* `/SUNWam/lib/xercesImpl.jar:/` *AccessManager-base* `/SUNWam/lib/xmlParserAPIs.jar:/` *AccessManager-base* `/SUNWam/lib/jaas.jar:/` *AccessManager-base* `/SUNWam/lib/xmlParserAPIs.jar:/` *AccessManager-base* `/SUNWam/lib/servlet.jar:/` *AccessManager-base* `/SUNWam/locale:/` *AccessManager-base*`/SUNWam/lib/am_services.jar:/` *AccessManager-base*`/SUNWam/lib/am_sdk.jar:/` *AccessManager-base*`/SUNWam/lib/jss311.jar:/` *AccessManager-base*`/SUNWam/lib:.`

  `-Djava.util.logging.manager=com.sun.identity.log.LogManager`

  `-Djava.util.logging.config.file=/` *AccessManager-base* `/SUNwam/lib/LogConfig.properties`

2 請確定位於 *AccessManager-base*/SUNWam/lib 的 LogConfig.properties 中有配置以下參數：

- `iplanet-am-logging-remote-handler=com.sun.identity`。

  `log.handlers.RemoteHandler`

- `iplanet-am-logging-remote-formatter=com.sun`。

  `identity.log.handlers.RemoteFormatter`

- `iplanet-am-logging-remote-buffer-size=1`

  遠端記錄支援緩衝，以記錄檔記錄的數目爲基準。此值定義了記錄緩衝區大小，以記錄的數目爲單位。一旦緩衝區空間已滿，所有在緩衝區中的記錄都會被清空至伺服器。

- `iplanet-am-logging-buffer-time-in-seconds=3600`

  此值定義要呼叫緩衝區清除器執行緒的逾時期間。

- `iplanet-am-logging-time-buffering-status=OFF`

  此值定義是否要啓用記錄緩衝 (和緩衝區清除器執行緒)。依預設此功能爲關閉。

  如果啓用基於計時器的緩衝 (`iplanet-am-logging-time-buffering-status=ON`)，則當記錄檔的記錄數量達到 `iplanet-am-logging-remote-buffer-size` 中所指定的值，或當計時器逾時 (逾時在 `iplanet-am-logging-buffer-time-in-seconds` 中指定) 時，就會沖寫記錄檔的緩衝區 (轉移到提供記錄服務的 AM 伺服器)。如果計時器在達到緩衝區大小之前逾時，則會傳送位於緩衝區內的記錄。如果停用遠端記錄的基於計時器緩衝，則緩衝區大小會決定沖寫緩衝區的時機。例如，如果緩衝區大小爲 10，而應用程式只傳送 7 條記錄，則不會沖寫緩衝區，也不會向記錄檔寫入記錄。如果應用程式終止，則會沖寫緩衝區中的記錄。

備註 – 每當記錄檔是空白時，安全記錄可能會顯示 [驗證失敗]。這是因爲當已建立檔案的數目與歸檔檔案大小相等時，安全記錄會從此歸檔並重新開始。於大部分的實例中，您可忽略此錯誤。一旦記錄的數目與歸檔檔案大小相等時，將不會顯示錯誤。

---

3 如果搭配用戶端 SDK 使用程式，則需要適當地設定 AMConfig.properties 檔案中的下列特性：

- com.iplanet.am.naming.url
- com.sun.identityagents.app.username
- com.iplanet.am.service.password
- com.iplanet.am.server.protocol
- com.iplanet.am.server.host
- com.iplanet.am.server.port

請參閱 `/opt/SUNWam/war` 目錄中的用戶端 SDK 範例 `README.clientsdk`。它詳述 `AMConfig.properties` 及 make 檔案是如何針對 `/opt/SUNWam/war/clientsdk-samples` 目錄產生的。範例的 makefile 的編譯及執行項目會依序使用這些檔案。

錯誤和存取記錄檔

存在兩種 Access Manager 記錄檔類型：存取記錄檔和錯誤記錄檔。

存取記錄檔記錄有關 Access Manager 部署的一般稽核資訊。記錄檔可能包含一個事件的單一記錄，例如一次成功的認證。記錄檔可能包含相同事件的多個記錄。例如，當管理員使用主控台變更屬性值時，「記錄服務」在一個記錄中記錄變更嘗試。「記錄服務」同時也會在第二個記錄中記錄執行結果。

錯誤記錄檔記錄發生於應用程式中的錯誤。當錯誤記錄檔中記錄了作業錯誤時，作業嘗試會記錄於存取記錄檔中。

平面記錄檔會附加副檔名 `.error` 或 `.access`。資料庫的表格名稱則以 `_ERROR` 或 `_ACCESS` 結尾。例如，記錄主控台事件的平面檔案命名爲 `amConsole.access`，記錄相同事件的資料庫表格命名爲 `AMCONSOLE_ACCESS` 或 `amConsole_access`。

下表提供對每個 Access Manager 元件所產生之記錄檔的簡要描述。

表 10–1 Access Manager 元件記錄檔

元件	記錄檔名稱前綴	記錄的資訊
階段作業	`amSSO`	階段作業管理屬性值，如登入時間、登出時間、逾時限制。
管理主控台	`amConsole`	經由管理主控台執行的使用者動作，如識別相關之物件、範圍，及策略的建立、刪除與修改。
認證	`amAuthentication`	使用者登入和登出。
識別聯合	`amFederation`	聯合相關事件，如「認證網域」的建立和「寄存提供者」的建立。聯合記錄檔的前綴是 `amFederation`。
授權 (策略)	`amPolicy`	策略相關事件，如策略建立、刪除或修改以及策略評估。
策略代理程式	`amAgent`	有關爲使用者存取或拒絕使用者存取之資源的異常。`amAgent` 記錄檔位於安裝策略代理程式的伺服器上。在 Access Manager 主機上於「認證記錄檔」中記錄代理程式事件。
SAML	`amSAML`	SAML 相關事件，如指定和工件的建立或移除、回應和請求的詳細資訊以及 SOAP 錯誤。
指令行	`amAdmin`	使用 amadmin 指令行工具進行作業的過程中發生的事件錯誤。若指定平面檔案記錄，則 amAdmin 記錄檔會位於主記錄目錄 (預設爲 `/var/opt/SUNWam/logs`) 下的 `amadmincli` 子目錄中。例如：載入服務模式、建立策略和刪除使用者。

如需 Access Manager 記錄檔的清單及說明，請參閱「*Access Manager Administration Reference*」中的「Access Manager Log File Reference」。

除錯檔案

除錯檔案並非記錄服務的功能。它們是使用獨立於記錄 API 的其他 API 寫入。除錯檔案儲存在 `/var/opt/SUNWam/debug`。此位置 (以及除錯資訊的等級) 可在 `AMConfig.properties` 檔案中配置，此檔案位於 *AccessManager-base*`/SUNWam/lib/` 目錄中。如需關於除錯特性的更多資訊，請參閱「*Access Manager Administration Reference*」中的 AMConfig.properties 檔案參照一章。

除錯等級

除錯檔案可記錄的資訊分爲幾個等級。除錯等級是以 `AMConfig.properties` 的 `com.iplanet.services.debug.level` 特性設定。

1. Off— 不記錄除錯資訊。
2. Error— 此等級用於生產。生產時，除錯檔案中應無錯誤。
3. Warning— 目前並不建議使用此等級。
4. Message— 此等級利用程式碼追蹤對可能的問題發出警示。大多數 Access Manager 模組使用此等級傳送除錯訊息。

備註 – [Warning] 與 [Message] 等級不可用於生產中。這樣會嚴重降低效能並產生大量的除錯訊息。

除錯輸出檔案

除非模組寫入除錯檔案，否則不會建立除錯檔案。因此，在預設**錯誤**模式下不會產生除錯檔案。登入時若除錯等級設爲**訊息**，則建立的除錯檔案包括：

- amAuth
- amAuthConfig
- amAuthContextLocal
- amAuthLDAP
- amCallback
- amClientDetection
- amConsole
- amFileLookup
- amJSS
- amLog

- amLoginModule
- amLoginViewBean
- amNaming
- amProfile
- amSDK
- amSSOProvider
- amSessionEncodeURL
- amThreadManager

最常使用的檔案是 `amSDK`、`amProfile` 和所有適用於認證的檔案。所擷取的資訊包括日期、時間和訊息類型 ([錯誤]、[警告]、[訊息])。

使用除錯檔案

依預設，除錯等級設爲**錯誤**。當管理員要進行下列作業時，除錯檔案十分有用：

- 寫入自訂認證模組。
- 使用 Access Manager SDK 寫入自訂應用程式。`amProfile` 和 `amSDK` 除錯檔案會擷取此資訊。
- 使用主控台或 SDK 對存取權限進行疑難排解。`amProfile` 與 `amSDK` 除錯檔案也會擷取此資訊。
- 疑難排解 SSL。
- 疑難排解 LDAP 認證模組。`amAuthLDAP` 除錯檔案會擷取此資訊。

應將我們以後可能會收到的疑難排解指南與除錯檔案配合使用。例如，當 SSL 失敗時，某些人可能會開啓除錯訊息並尋找 `amJSS` 除錯檔案中的任何特定憑證錯誤。

◆ ◆ ◆ 第 11 章

11 通知服務

Sun Java System Access Manager 7.1 通知服務可讓您將階段作業通知傳送到遠端 Web 容器。需要啓用這個服務以供遠離 Access Manager 伺服器本身執行的 SDK 應用程式使用。本章說明如何啓用遠端 Web 容器，以接收通知。包含以下小節：

簡介

「通知服務」可讓您將階段作業通知傳送到正在遠端執行 Access Manager SDK 的 Web 容器。通知僅適用於「階段作業」、「策略」及「命名服務」。此外，遠端應用程式必須在 Web 容器內執行。通知的目的在於：

- 同步化各個服務的用戶端快取。
- 在用戶端上啓用更即時的更新。(沒有通知時，改用輪詢。)
- 不需要變更用戶端應用程式即可支援通知。

請注意，唯有在 Web 容器上安裝遠端 SDK 後，才可接收通知。

啓用通知服務

下列是配置遠端 SSO SDK 以接收階段作業通知所需的步驟。

▼ 接收階段作業通知

1. **在電腦 1 上安裝 Access Manager。**
2. **在電腦 2 上安裝 Sun Java System Web Server。**

3. **在安裝 Web Server 的同一電腦上安裝 SUNWamsdk。**
   如需遠端安裝 Access Manager SDK 的說明，請參閱「Sun Java Enterprise System 5 安裝指南」。

4. **請確保下列與安裝 SDK 所在電腦有關的事項為真。**

   a. **請確定已為安裝 SDK 的伺服器上的 /*remote_SDK_server*/SUNWam/lib 及 /*remote_SDK_server*/SUNWam/locale 目錄設定正確的存取權限。**
      這些目錄包含位於遠端伺服器上的檔案及 jar。

   b. **請確定針對 Web Server 的 server.policy 檔案的「授權」區段設定下列權限。**
      server.policy 位於 Web Server 安裝的 config 目錄中。必要時，可以複製及貼上這些權限：

      ```
      permission java.security.SecurityPermission
      "putProviderProperty.Mozilla-JSS"

      permission java.security.SecurityPermission "insertProvider.Mozilla-JSS";
      ```

   c. **請確定在 server.xml 中設定正確的類別路徑。**
      server.xml 也位於 Web Server 安裝的 config 目錄中。通常的類別路徑是：

      ```
      <JAVA javahome="/export/home/ws61/bin/https/jdk"
      serverclasspath="/export/home/ws61/bin/https/jar/webserv-rt.jar:${java.home}/lib/tools.
          bin/https/jar/nova.jar"
      classpathsuffix="::/IS_CLASSPATH_BEGIN_DELIM:                //usr/share/lib/xalan.jar:
                      //lib:/export/SUNWam/locale:                //usr/share/lib/mps/jss3.ja
      envclasspathignored="true" debug="false"
      debugoptions="-Xdebug -Xrunjdwp:transport=dt_socket,
      server=y,suspend=n"
      javacoptions="-g"
      dynamicreloadinterval="2">
      ```

5. **使用安裝在遠端 SDK 伺服器上的 SSO 範例來進行配置。**

   a. **移至 /*remote_SDK_server* /SUNWam/samples/sso 目錄。**

   b. **執行 gmake。**

   c. **將產生的類別檔案從 /*remote_SDK_server*/SUNWam/samples/sso 複製到 /*remote_SDK_server*/SUNWam/lib/。**

6. **將 am.encryption.pwd 的加密值從隨 Access Manager 安裝的 AMConfig.properties 檔案複製到安裝 SDK 的遠端伺服器上的 AMConfig.properties 檔案。**
   am.encryption.pwd 的值用於加密及解密密碼。

7. **以 amadmin 的身份登入 Access Manager。**

   `http://`*AcceessManager-HostName* `:3000/amconsole`

8. **在瀏覽器位置欄位中輸入 `http://`*remote_SDK_host*`:58080/servlet/SSOTokenSampleServlet` 並驗證 SSOToken 來執行 servlet。**

   SSOTokenSampleServlet 用於驗證階段作業記號和增加偵聽程式。執行 servlet 將列印下列訊息：

   ```
   SSOToken host name: 192.18.149.33 SSOToken Principal name:
   uid=amAdmin,ou=People,dc=red,dc=iplanet,dc=com Authentication type used: LDAP
   IPAddress of the host: 192.18.149.33 The token id is
   AQIC5wM2LY4SfcyURnObg7vEgdkb+32T43+RZN30Req/BGE= Property: Company is - Sun
   Microsystems Property: Country is - USA SSO Token Validation test Succeeded
   ```

9. **在安裝用戶端 SDK 所在電腦的 AMConfig.properties 中設定特性 com.iplanet.am.notification.url=：**

   ```
   com.iplanet.am.notification.url=http://clientSDK_host.domain:port
   /servlet
       com.iplanet.services.comm.client.PLLNotificationServlet
   ```

10. **重新啓動 Web Server。**

11. **以 amadmin 的身份登入 Access Manager。**

    `http://`*AcceessManager-HostName* `:3000/amconsole`

12. **再次在瀏覽器位置欄位中輸入 `http://`*remote_SDK_host*`:58080/servlet/SSOTokenSampleServlet` 並驗證 SSOToken 來執行 servlet。**

    執行遠端 SDK 的機器接收到通知時，它會在階段作業狀態變更時呼叫對應的偵聽程式。請注意，唯有在 Web 容器上安裝遠端 SDK，才可接收通知。

▼ 在僅限入口網站安裝中啓用通知服務

本節描述在預設會以**輪詢**模式執行的僅限入口網站安裝中，使用 WebLogic 8.1 啓用通知的步驟。對於也包含 amserver 元件的入口網站實例，則不需要這些程序。amserver 元件會自動配置，以執行通知。

1. **在 WebLogic 中註冊 PLLNotificationServlet。**

   WebLogic 8.1 要求部署 Web 應用程式。此外，servlet URL 必須有效，以便從瀏覽器存取時，可以傳回下列訊息：

   ```
   Webtop 2.5 Platform Low Level notification servlet
   ```

2. **將註冊的 URL 輸入到 `AMConfig.properties`，如下所示：**

   ```
   com.iplanet.am.notifaction.url=http://
   weblogic_instance-host.domain:port/notification/PLLNotificationServlet
   ```

3. **在 AMConfig.properties 中禁用輪詢。這會自動啓用通知：**

   ```
   com.iplanet.am.session.client.polling.enable=false
   ```

4. **重新啓動 WebLogic 並測試配置。**

   如果您已將除錯模式設爲 `message`，您應該會在觸發後，看到抵達入口網站的階段作業通知。例如，從 Access Manager 主控台終止使用者這樣的類似動作將引發通知事件。

索引