

WebSphere Liberty Profile 中的 EAR 部署

聽起來你已經成功使用 Maven 的 `mvn package` 命令建立了一個 EAR（企業檔案）文件，並且建立過程中沒有任何錯誤。然而，當你將這個 EAR 部署到 WebSphere Liberty Profile（WLP）並啟動伺服器時，遇到了「找不到模組」的錯誤。這很令人困惑，因為平順的建立過程通常意味著一切正常，但伺服器卻沒有如預期地識別你的應用程序。讓我們來分析這個問題，並找出可能的解決方案。

理解問題

EAR 文件是一個標準的 Java EE 檔案，通常包含一個或多個模組，如 WAR 文件（用於網頁應用程序）、JAR 文件（用於 EJBs 或庫）和部署描述符（`application.xml`）。當你將 EAR 部署到 WLP 時，伺服器應該能夠檢測到這些模組並啟動應用程序。「找不到模組」的消息表明 WLP 可能無法找到 EAR 中的任何模組，或者不識別它們，從而阻止應用程序啟動。

由於你的 Maven 建立過程成功（「一切順利」），問題可能出在以下三個方面之一：1. **EAR 文件內容**：EAR 可能不包含預期的模組，或者部署描述符可能遺失或不正確。2. **部署過程**：你將 EAR 部署到 WLP 的方式可能不符合伺服器期望的方式來找到和處理它。3. **伺服器配置**：WLP 可能未設置來識別 EAR 中的模組，因為缺少功能或配置錯誤。

讓我們探討這些可能性，並提供可操作的步驟來診斷和解決問題。

可能的原因及解決方案

1. EAR 文件可能為空或打包不正確 即使建立過程成功，EAR 可能不包含任何模組（例如 WAR 或 JAR 文件），或者 `application.xml` 文件，它告訴伺服器包含哪些模組，可能遺失或配置錯誤。

- **為什麼會發生：**在 Maven EAR 專案中，`maven-ear-plugin` 負責組裝 EAR。它根據你的 `pom.xml` 配置或依賴項來包含模組。如果沒有指定模組，或者依賴項（如 WAR）沒有正確定義或解析，EAR 可能為空或缺少正確的 `application.xml`。
- **如何檢查：**
 - 使用工具如 `unzip` 打開你的 EAR 文件，或者在終端運行 `jar tf myApp.ear` 列出其內容。
 - 查找：
 - * 模組文件（例如 `my-web.war`, `my-ejb.jar`）在 EAR 的根目錄。
 - * `META-INF/application.xml` 文件。
 - 在 `application.xml` 中，你應該看到定義模組的條目，例如：

```
<?xml version="1.0" encoding="UTF-8"?>
<application>
    <module>
```

```

<web>
    <web-uri>my-web.war</web-uri>
    <context-root>/myapp</context-root>
</web>
</module>
</application>

```

- **如何修復：**

- 驗證你的 EAR 模組的 `pom.xml`。確保你已經指定了要包含的模組的依賴項，例如：

```

<dependencies>
    <dependency>
        <groupId>com.example</groupId>
        <artifactId>my-web</artifactId>
        <type>war</type>
    </dependency>
</dependencies>

```

- 如果需要，配置 `maven-ear-plugin`：

```

<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-ear-plugin</artifactId>
    <version>3.3.0</version>
    <configuration>
        <modules>
            <webModule>
                <groupId>com.example</groupId>
                <artifactId>my-web</artifactId>
                <contextRoot>/myapp</contextRoot>
            </webModule>
        </modules>
    </configuration>
</plugin>

```

- 使用 `mvn clean package` 重新建立並重新檢查 EAR 內容。

如果 EAR 為空或 `application.xml` 遺失/不正確，這可能是根本原因。修復 Maven 配置應該能解決問題。

2. 部署方法問題 你將 EAR 部署到 WLP 的方式也可能是問題。WLP 支持兩種主要部署方法：`dropins` 目錄和 `server.xml` 中的明確配置。

- **使用 dropins 目錄：**

- 如果你將 EAR 放在 `wlp/usr/servers/<serverName>/dropins/` 目錄中，WLP 應該自動檢測並部署它。
- 然而，對於 EAR 文件，自動部署可能不總是如預期工作，特別是如果需要額外配置（如上下文根）。

- **使用 server.xml：**

- 對於 EAR 文件，通常最好在 `wlp/usr/servers/<serverName>/server.xml` 中明確配置應用程序：

```
<server>
    <featureManager>
        <feature>servlet-3.1</feature> <!-- 確保啟用所需功能 -->
    </featureManager>
    <application id="myApp" name="myApp" type="ear" location="${server.config.dir}/apps/myApp.ear"/>
</server>
```

- 將 EAR 放在 `wlp/usr/servers/<serverName>/apps/`（或相應調整 `location` 路徑）。

- **如何檢查：**

- 確認你放置 EAR 的位置以及啟動伺服器的方式（例如 `./bin/server run <serverName>`）。
- 檢查伺服器日誌（例如 `wlp/usr/servers/<serverName>/logs/console.log` 或 `messages.log`）以獲取部署消息。

- **如何修復：**

- 試著在 `server.xml` 中配置 EAR，如上所示，而不是依賴 `dropins`。
- 更改後重啟伺服器：`./bin/server stop <serverName>` 後跟 `./bin/server start <serverName>`。

如果 EAR 沒有正確註冊到伺服器，這可能解釋了錯誤。

3. 缺少伺服器功能 WLP 是一個輕量級伺服器，只加載你在 `server.xml` 中啟用的功能。如果你的 EAR 包含需要特定功能（例如 servlet、EJBs）的模組，而這些功能沒有啟用，WLP 可能無法識別或加載模組。

- **為什麼會發生：**例如，WAR 文件需要 `servlet-3.1` 功能（或更高版本），而 EJB 模組需要 `ejbLite-3.2`。沒有這些功能，伺服器可能無法處理模組。

- **如何檢查：**

- 查看你的 `server.xml` 並檢查 `<featureManager>` 部分。
 - * `<feature>servlet-3.1</feature>` 用於網頁模組。
 - * `<feature>ejbLite-3.2</feature>` 用於 EJB 模組。
- 審查伺服器日誌以獲取有關缺少功能的消息（例如「所需功能未安裝」）。

- **如何修復：**

- 根據應用程序的需求在 `server.xml` 中添加必要的功能：

```
<featureManager>
    <feature>servlet-3.1</feature>
    <!-- 根據需要添加其他功能 -->
</featureManager>
```

- 重啟伺服器以應用更改。

如果缺少功能，啟用它們應該允許 WLP 識別模組。

診斷步驟

為了確定問題，請按照以下步驟進行：

1. 檢查 EAR 文件：

- 運行 `jar tf myApp.ear` 或解壓縮它。
- 確保它包含你的模組（例如 `my-web.war`）和有效的 `META-INF/application.xml`。

2. 檢查 Maven 建立：

- 審查 EAR 模組的 `pom.xml` 以確認依賴項和 `maven-ear-plugin` 配置。
- 再次運行 `mvn clean package` 並檢查建立輸出以獲取有關包含模組的消息（例如「添加模組 `my-web.war`」）。

3. 驗證部署：

- 確認 EAR 的位置（例如 `dropins` 或 `apps`）。
- 如果使用 `dropins`，試著將其移動到 `apps` 並將其添加到 `server.xml`。

4. 檢查伺服器日誌：

- 啟動伺服器並檢查 `console.log` 或 `messages.log` 以獲取超出「找不到模組」的詳細錯誤消息。
- 查找線索，例如「應用程序不包含任何模組」或功能相關錯誤。

5. 使用簡單的 EAR 進行測試：

- 使用已知工作的 `pom.xml` 創建一個包含單個 WAR 文件的基本 EAR，並將其部署到 WLP。如果它工作，問題可能與原始 EAR 的打包有關。
-

最有可能的原因

考慮到你的建立過程順利但 WLP 報告「找不到模組」，最可能的原因是 **EAR 文件不包含任何模組** 或 **application.xml 遺失/不正確**。這可能發生在你的 `pom.xml` 沒有正確指定要包含的模組，即使 Maven 沒有標記錯誤。或者，如果 EAR 正確，問題可能與部署（例如未在 `server.xml` 中配置）或缺少伺服器功能有關。

最終建議

- 從檢查 EAR 的內容 (`jar tf myApp.ear`) 和你的 `pom.xml` 配置開始。
- 如果 EAR 為空或缺少 `application.xml`，修復 Maven 設置並重新建立。
- 如果 EAR 看起來正確，在 `server.xml` 中配置它，確保啟用所需功能，並重新部署。
- 檢查伺服器日誌以獲取更多上下文。

按照這些步驟，你應該能夠確定問題是否在建立、部署或伺服器配置中，並成功在 WLP 上運行你的應用程序。如果需要幫助特定配置，請告訴我！