
第 5 章 安装 WebSphere Product Center

要安装和配置 WebSphere Product Center，请执行本章列示的任务。

满足先决条件

- 确保完成特定操作环境的所有安装需求
 - 在 WebSphere Application Server 上创建一个用户，该用户具有对 WebSphere Product Center 目录结构的写访问权
 - 假定将 bash 用作缺省 shell 程序脚本。如果正在使用另一个 shell，如 “sh”，则首先执行 bash。必须将 bash 安装在 /bin/bash 目录中
-

准备安装介质

使用 Passport Advantage

如果从 Passport Advantage 获取了 WebSphere Product Center 产品，则从 Passport Advantage 下载压缩的 WebSphere Product Center 文件，然后将安装文件解压缩到期望的安装目录中。

使用安装介质的一般过程

本节描述使用安装介质的一般过程。

调用图形 WebSphere Product Center 安装程序

图形 WebSphere Product Center 安装程序将提供一个向导，该向导允许您进行关于 WebSphere Product Center 的安装的选择。通过从其中一个安装介质安装程序目录中运行可执行文件来调用安装程序。

操作系统	安装程序目录
AIX	WPC_AIX
Linux	WPC_Linux
HP-UX	WPC_HP

AIX 上的示例：

本节描述如何在 AIX 环境中调用 WebSphere Product Center 的安装程序。

如果正在 AIX 计算机上运行 CDE： 如果您正在运行公共桌面环境

(CDE) 并且直接在 AIX 计算机上工作，则可以浏览安装介质并双击 bin 文件 (*.bin) 以调用安装程序。

也可以从命令行浏览安装介质并执行 bin 文件 (.bin)。

如果正在通过 X 仿真软件连接至 AIX 计算机：如果您正在使用 Windows 计算机来通过 X 仿真软件连接至 AIX 计算机，则执行下列操作来调用安装程序：

- 1. 为 X 仿真软件配置环境。
- 2. 执行特定于操作系统的 .bin 文件 (.bin)。以下示例显示如何在 AIX 计算机上执行此操作。

```
./setupAIX.bin
```

图形安装程序将在您用来连接至 AIX 计算机的 Windows 计算机上启动。

执行静默安装

可以执行 WebSphere Product Center 的静默安装，对于这种安装，您在一个文件中提供安装选项，而不是在安装向导的屏幕上提供那些选项。当您必须执行多次完全相同的安装时，这种安装方式对您特别有帮助。

执行静默安装时，可以使用选项文件，也可以不使用它。选项文件存放静默安装使用的安装参数。使用下列其中一个选项在命令行处运行安装程序可执行文件：

- 手工设置安装选项
- 使用选项文件来设置安装选项

手工执行静默安装设置选项

当执行静默安装时，需要下列选项：

安装位置	<div>-P installLocation=<dir></div> <div>注意：当指定 <dir> 名称时，要使用全路径名。</div>
	<div>对特定的操作系统包含应用程序服务器和数据库的组合的选择。可将这些选项包括为手工设置的安装选项的一部分或选项文件的一部分。</div> <div>对于 AIX</div> <div>-W adapterFeatures_aix.selectedFeature=<index></div> <div>索引值</div> <div><ul style="list-style-type: none">• WebSphere/DB2 = 0</div>

应用程序服务器和数据库组合	<ul style="list-style-type: none"> • WebSphere/Oracle= 1 <p>对于 Linux</p> <p>-W adapterFeatures_linux.selectedFeature=<index></p> <p>索引值</p> <ul style="list-style-type: none"> • WebSphere/DB2 = 0 • WebSphere/Oracle= 1 • WebLogic/Oracle= 2 <p>对于 HPUX</p> <ul style="list-style-type: none"> • WebLogic/Oracle <p>由于只有一个应用程序服务器和数据库组合选项，所以只需要包括安装位置选项。</p> <p>./<installer file> -P installLocation=<dir> -silent</p> <p>注意： 有关每个应用程序服务器和数据库组合的版本，请参阅本文档中的软件需求部分。</p>
---------------	--

要使用必需的选项进行静默安装，请使用以下命令行：

```
./<name of installer file> -W adapterFeatures_<operating
system>.selectedFeature=<index> - P installLocation=<dir> -
silent
```

示例

以下示例显示如何在 AIX 计算机上使用 WebSphere/DB2 组合执行此操作：

```
./setupAIX.bin -W adapterFeatures_aix.selectedFeature=0 - P
installLocation=/opt/IBM/WPC - silent
```

使用选项文件执行静默安装

通过使用以下命令行，可使用选项文件执行静默安装：

```
./<name of installer file> - P -options <options file name > -
silent
```

选项文件必须包含静默安装的必需选项，就如在上一节中讨论的那样。

示例

以下示例显示如何在 AIX 计算机上执行此操作。

```
./setupAIX.bin -P - options /home/WPC/install.txt - silent
```

记录用户选择

安装程序可以将交互式安装期间的用户选择记录到文件中。此文件是在安装完成时生成的。可以将此文件用作其它机器上的静默安装的响应文件（将记录的用户选择应用于这些安装）。

```
./<name of installer file> - options - record <file name>
```

安装 WebSphere Product Center

本节提供有关安装 WebSphere Product Center 的指示信息。您可以选择使用图形安装程序来安装软件产品，也可以从命令行以静默方式进行安装。

使用图形安装程序来安装 WebSphere Product Center

执行下列操作以安装 WebSphere Product Center:

1. 按照“调用图形 WebSphere Product Center 安装程序”中的描述来调用安装程序。
2. 选择要用于安装向导的语言，然后单击“确定”。
3. 在“欢迎”屏幕上，单击**下一步**。
4. 在“IBM 软件许可协议接受”屏幕上，单击“我接受许可协议中的条款”，然后单击**下一步**。
5. 从可用的组合列表中选择应用程序服务器和数据库组合，然后单击**下一步**。
6. 在产品目录屏幕上，执行下列其中一项任务：
 - 在“目录名”字段中输入要在其中安装 WebSphere Product Center 的目录的完整路径，然后单击**下一步**
 - 单击“浏览”以选择目录，然后单击**下一步**
 - 接受缺省路径并单击**下一步**
7. 总结屏幕列示将要安装的功能部件、指定的产品目录以及所需的磁盘空间容量。单击**下一步**。
8. 在安装程序成功完成之后，单击**完成**。

注意：日志详细信息是在“log.txt”文件中提供的，它位于：<install location>/log.txt

设置环境变量

设置 TOP 变量

用途：将 shell 变量 TOP 设置为指向 WebSphere Product Center 安装目录。

编辑以下文件：

```
$HOME/.bash_profile
```

注意：上面用于概要文件的名称可能随配置的不同而不同。建议使用“`.bash_profile`”。

UNIX 上的示例：

如果 WebSphere Product Center 安装在以下目录中：

```
/home/user/john/WPCV50
```

则从命令行：

```
export TOP=/home/user/john/WPCV50
```

源 \$TOP/setup/init_ccd_vars.sh

用途：定义 WebSphere Product Center 初始化脚本的位置。

```
$TOP/setup/init_ccd_vars.sh
```

注意：应该修改缺省 shell 程序初始化脚本 `.bash_profile` 以设置 TOP 和源 `init_ccd_vars.sh`。确保是对 `~/.bash_profile` 进行修改。根据 bash 安装的不同，该文件的位置可能会有所变化。

编辑 `.bash_profile` 以包括以下内容：

```
# WPC init

export TOP=<path to WPC{VERSION} directory>

WPC_INIT_VARS=$TOP/setup/init_ccd_vars.sh

if [ -f $WPC_INIT_VARS ]; then
    . $WPC_INIT_VARS
else
    echo "Error: file '$WPC_INIT_VARS' not found"
```

编辑初始化脚本（init_ccd_vars.sh）

以“init_ccd_vars.sh 文件”为基础以初始化 WebSphere Product Center shell 变量。要使 WebSphere Product Center 能够正确地运行，必须在“init_ccd_vars.sh”文件中设置几个关键变量，本节列示了那些变量。如果未设置这些变量，启动 WebSphere Product Center 时就会发生错误。

文件的位置：\$TOP/setup/init_ccd_vars.sh

Set JDK_HOME

用途：定义 Java 运行时安装的位置。此参数必须位于 TOP 参数前面。

UNIX 上的示例：

```
export JDK_HOME=/usr/java131
```

设置公共 Unix 实用程序

用途：将属性设置为 WebSphere Product Center 脚本使用的公共 UNIX 实用程序。

为下列实用程序设置属性：

```
AWK
CP
ECHO
GREP
MKDIR
RM
SED
TOUCH
TR
BC
CAT
```

UNIX 上的示例：

```
export CP=cp
export RM="rm -rf"
export ECHO=echo
export TOUCH=touch
export MKDIR="mkdir -p"
export TR=tr
export GREP=grep
export AWK=awk
export SED=sed
export CAT=cat
```

```
export BC=bc
```

设置配置以便与 WebSphere Application Server 配合使用

用途：配置 WebSphere Product Center 以使用 WebSphere Application Server

注意：本节仅适用于 WebSphere Application Server。

设置下列变量：

WAS_HOME	WebSphere 的安装目录
WAS_NODENAME	WebSphere Application Server 的节点名。通常，这是本地主机名。
WAS_APPSERVERNAME	将要（在 Websphere 中）使用的应用程序服务器的名称。此名称必须是唯一的。
WAS_VHOST	要将 WebSphere Product Center 中间件绑定到的 WebSphere 虚拟主机的名称。仅当将要在同一个 WebSphere 实例上运行多个 WebSphere Product Center 中间件服务器时，才需要此变量。注意：这与 HTTP Server 虚拟主机不同。

为不同的模块设置内存标志

所有不同的 WebSphere Product Center 模块都是使用 Java 虚拟机运行的。用户可以通过更改为此目的提供的内存标志来指定用于这些模块的内存。

例如，以下条目：

```
export SCHEDULER_MEMORY_FLAG='-Xmx1024m -Xms48m'
```

指定 SCHEDULER 模块的初始内存设置（由选项 -Xms 指定）是 48MB，最大内存设置（由选项 -Xmx 指定）是 1024MB。

注意：请参阅 \$TOP/setup/init_ccd_vars.sh 中的注释以了解更多信息。

建议您将下列内存标志设置用于 WebSphere Product Center 服务：

```
export ADMIN_MEMORY_FLAG='-Xmx64m -Xms48m'
export APPSVR_MEMORY_FLAG='-Xmx512m -Xms64m'
export EVENTPROCESSOR_MEMORY_FLAG='-Xmx64m -Xms48m'
export QUEUEMANAGER_MEMORY_FLAG='-Xmx64m -Xms48m'
export SCHEDULER_MEMORY_FLAG='-Xmx1024m -Xms48m'
export WORKFLOWENGINE_MEMORY_FLAG='-Xmx256m -Xms64m'
```

配置运行时属性

下列步骤配置系统使用的运行时属性，这些属性是在文件 `common.properties` 中定义的。

编辑以下文件：

`$STOP/etc/default/common.properties`

设置数据库信息

用途：设置信息，以允许访问数据库。Tomcat 使用此信息来登录到数据库和设置连接池。

设置属性：

`db_userName`

`db_password`

`db_url`

示例：

假定用户名是 “wpcapp”，密码是 “track1”，数据库服务器的主机名是 “jupiter”，Oracle 数据库实例的 SID 是 “pauprd”。

设置下列属性：

`db_userName=wpcapp`

`db_password=track1`

`db_url=jdbc:oracle:thin:@jupiter:1521:pauprd`

注意：引用的端口是侦听器端口。在上面的示例中，端口是 1521。

设置属性：

`db_userName`

`db_password`

`db_url`

示例：

假定用户名是 “wpcapp”，密码是 “track1”，数据库服务器的主机名是 “jupiter”，数据库实例是 “pauprd”。

设置下列属性：

`db_userName=wpcapp`

`db_password=track1`

`db_url=jdbc:db2:pauprd`

设置时区

用途：定义时区属性。

设置属性： `timezone`

值：以小时或秒计设置偏移量

示例：

要设置（GMT-08:00）太平洋时间（美国和加拿大）； **Tijuana**（以小时计）为：

`timezone=-8`

要设置（GMT-08:00）太平洋时间（美国和加拿大）； **Tijuana**（以分钟计）为：

`timezone=-480`

设置 FTP 目录

用途：定义使用 `ftp` 将数据文件传送到的目录（对于导入）。如果不将 `FTP` 用于数据导入，则可以将此属性保留为缺省值。如果不将 `FTP` 用于数据导入，则可以将此属性保留为缺省值。

设置属性：

`ftp_root_dir`

注意：存储在 `/u01/ftp/COMPANY_CODE/files` 中的文件将显示在文档库的 `ftp` 部分中。如果文件被放到 `/u01/ftp` 中，则它们不会显示。因此，必须将此属性设置为组织的 `ftp` 目录。

例如，`/u01/ftp/wpc/files`。

UNIX 上的示例：

`ftp_root_dir=/u01/ftp/`

注意：`ftp` 目录以正斜杠 “/” 结尾。

WebSphere Product Center 将读写这个 `FTP` 目录。

设置此目录以允许 WebSphere Product Center 用户有读 / 写权限。

设置 Tmp 目录

用途：此设置标识服务器上的临时目录。

设置属性：

tmp_dir

UNIX 上的示例：

tmp_dir=/tmp/

注意：tmp 目录以正斜杠 “/” 结尾。

设置此目录以允许 WebSphere Product Center 用户有读 / 写权限。

测试 WebSphere Product Center 许可文件是否有效

脚本 \$STOP/bin/go/start_local.sh 脚本用来启动所有 WebSphere Product Center 中间件服务（即模块），并且是运行 WebSphere Product Center 所必需的。仅当存在指定 WebSphere Product Center 模块有效的有效许可证文件 \$STOP/etc/default/ccd_license.xml 时，才能成功地启动该模块。如果该许可证文件无效，则将无法启动该模块。错误将记录在每个模块的特定日志文件中，那些日志文件位于 \$STOP/logs/svc 目录中。

注意：许可证文件是与安装文件分开提供的。如果需要许可证文件来操作 WebSphere Product Center，请与 WebSphere Product Center 支持联系。

要测试 WebSphere Product Center 许可证文件的有效性，请运行以下脚本：

\$STOP/bin/test_license.sh

如果 IP 不正确，则许可证验证将报告期望的 IP 以及当前本地 IP。

应用程序服务器配置

本节提供将 WebSphere Product Center 与 WebSphere Application Server 配合运行所需的指示信息。

IBM WebSphere Application Server 配置

为了将 WebSphere Product Center 与 WebSphere Application Server 配合使用，请遵循下列各节中描述的配置指示信息。

- 验证配置设置
- 启动 WebSphere 缺省服务器和管理控制台
- 添加 WebSphere 组并更改 WebSphere 设置
- 运行 WebSphere 脚本
- 启动 WebSphere 应用程序

验证配置设置

除了标准变量以外，验证是否在 \$TOP/setup/init_ccd_vars.sh 中定义了下列配置设置：

名称	值
WAS_HOME	WebSphere 的主目录。 即 /usr/WebSphere/AppServer。
WAS_APPSERVERNAME	将要（在 WebSphere 中）使用的应用程序服务器的名称。此名称必须是唯一的。
WAS_NODENAME	正在其上运行的 WebSphere Application Server 的节点名。通常，这将与系统的主机名相同。
WAS_VHOST	要将 WebSphere Product Center 中间件绑定到的 WebSphere 虚拟主机的名称。仅当将要在同一个 WebSphere 实例上运行多个 WebSphere Product Center 中间件服务器时，才需要此变量。注意：这与 HTTP Server 虚拟主机不同。

启动缺省服务器和管理控制台

要启动 WAS 缺省服务器，请作为 root 用户发出以下命令：

```
${WAS_HOME}/bin/startServer.sh server1
```

添加 WAS 组

在 AIX 上，可以使用 SMIT 管理工具来添加组。

注意：确保 CCD 用户是 WAS 组的一部分。

- 登录到 WebSphere 管理控制台。（通常，此控制台是在 <http://your-server-name:9090/admin> 位置运行的。）
- 单击“服务器 -> 应用程序服务器”
- 单击缺省服务器（server1）
- 单击“过程定义”
- 单击“过程执行”
- 将 UMASK 更改为“002”。
- 将“作为组运行”更改为已创建的 WAS 组。如果 WebSphere

缺省服务器不是作为 root 用户运行的，则 WebSphere 用户也应该位于 WAS 组中。

- 更改对 WebSphere 目录的许可权，以使 WAS 组具有写许可权。

```
# chgrp -R wasgroup ${WAS_HOME}
```

```
# chmod -R g+w ${WAS_HOME}
```

运行 WAS 脚本

在完成上述 WebSphere Product Center 安装步骤之后，按列示的顺序运行下列脚本：

1. \$TOP/bin/websphere/create_appsvr.sh
2. \$TOP/bin/websphere/create_vhost.sh
3. \$TOP/bin/websphere/install_war.sh

在同一 WebSphere 服务器上运行 WebSphere Product Center 的多个实例

要使用一台 WebSphere 服务器来运行 WebSphere Product Center 的多个实例，必须执行下列任务：

- 要求 WebSphere 管理服务服务器作为 root 用户运行。
- 如上所述正常地安装每个实例，但先不要运行 create_appsvr.sh 或 install_war.sh
- 确保每个实例都在 init_ccd_vars.sh 中定义唯一的 WAS_APPSERVERNAME 和 WAS_VHOST，并在 common.properties 中定义唯一的应用程序端口
- 在每个实例下，按照以下顺序运行下列脚本：

1. \$TOP/bin/websphere/create_vhost.sh
2. \$TOP/bin/websphere/create_appsvr.sh
3. \$TOP/bin/websphere/install_war.sh

- 为每个实例创建提供程序符号链接（如果 install_war.sh 未能创建符号链接的话）。

BEA WebLogic 配置

本节描述为 Websphere Product Center 环境正确配置 WebLogic 的必需任务。

- 设置缺省主目录
- 使用 \$WL_HOME/common/bin/dmwiz.sh 向导创建新的域
- 更新 config.xml 文件
- 复制或链接 WebLogic 域应用程序目录
- 验证公用文件目录

对于本文档，本节中的 WebLogic 设置指示信息按如下所示配置：

- 域名 “wpc_domain”
- 侦听端口 “7507”
- 包含在目录 “/wpc/envs/wl_username/wl_domain” 中的域文件
- Servername 设置为 “wpc_server”
- WebLogic 用户名为 “wl_username”
- WebLogic 密码为 “wl_password”

设置缺省主目录

要设置 WebLogic 的缺省主目录，在 \$TOP/setup/init_ccd.vars.sh 中配置以下参数：

- WPC_WL_DOMAIN_DIR (*home of the WebLogic WPC domain*)
- WPC_WL_DOMAIN (*domain name for the WPC domain*)
- WPC_WL_SERVER (*server name for the WPC domain*)
- WPC_WL_PW (*password to start the WebLogic server*)
- WPC_WL_USER (*WebLogic server user name*)

示例：

```
export WPC_WL_DOMAIN_DIR="/wpc/envs/wl_username/wl_domain/wpc_domain"
export WPC_WL_DOMAIN=wpc_domain
export WPC_WL_SERVER=wpc_server
export WPC_WL_PW=wl_password
export WPC_WL_USER=wl_username
```

创建新的 WebLogic 域

要创建新的 WebLogic 域，使用以下目录中的 dmwiz.sh 向导：

\$WL_HOME/common/bin/dmwiz.sh

1. 运行 \$WL_HOME/common/bin/dmwiz.sh
2. 选择域类型 (3) WLS 域
3. 选择服务器类型 (1) 单个服务器（独立的服务器）
4. 选择域名 (1) 修改当前选择
5. 配置单个服务器
6. 修改服务器名 wpc_server
7. 修改服务器侦听端口 7507
8. 选择创建域的位置 > 修改当前选择 /wpc/envs/wl_username/wl_domain
9. 输入用户名 wl_username
10. 输入密码 wl_password

11. 输入验证密码 wl_password

12. 按 Enter 键以创建域并遵循提示退出向导。

完成 dmwiz 向导时，将会显示以下消息：**域配置向导成功创建了 "wpc_domain"**

既然在 “/wpc/envs/wl_username/wl_domain/wpc_domain” 中创建了 WebLogic 域，就可以通过编辑 “/wpc/envs/wl_username/wl_domain/wpc_domain/config.xml” 来配置它了

配置 config.xml

1. 除去下面各部分

- SSL 部分
- Application DefaultWebApp 部分
- 证书应用程序部分

2. 添加下面的 “应用程序” 和 “启动类” 部分：

- 应用程序（在域 xml 参数中添加）

```
<Application Deployed="true" Name="suppliers"
Path="{ACTUAL_PATH_TO_PUBLIC_HTML}" TwoPhase="false">
<WebAppComponent Name="suppliers" Targets="wpc_server"
URI="suppliers"/>
</Application>

<Application Deployed="true" Name="wpc" Path="/applications"
StagedTargets="" TwoPhase="false">
<WebAppComponent Name="wpc" Targets="wpc_server"
URI="wpc.war"/>
</Application>
```

- 启动类（在域 xml 参数中添加）

```
<StartupClass ClassName="austin.appsvr.current.admin.RemoteWeblogic"
FailureIsFatal="true" Name="RemoteWeblogic" Targets="wpc_server"/>
```

3. 更改 DefaultWebApp

将 WebServer 字段中的 DefaultWebApp 从 "DefaultWebApp" 更改为 "wpc"

4. WebLogic 记录 — 如果需要的话，更改日志参数以将 weblogic.log 文件重定向至另一个位置。

配置之后的样本 config.xml 文件

<!--如果域是活动的，则请不要编辑 config.xml 文件。当域活动时，对该文件作出的任何更改都将不会对域的配置产生任何影响并且很有可能会丢失。如果域不活动，则可使用 XML 编辑器编辑此文件。如果进行编辑，则请参阅 BEA Server Configuration Reference 文档，可以从 http://e-docs.bea.com/wls/docs61/config_xml/index.html 中获取该文档。总之，我们建议通过管理控制台更改配置文件-->

```
<Domain Name="wpc_domain">

  <Server
    Name="wpc_server"
    NativeIOEnabled="true"
    ListenAddress=""
    ListenPort="7507" >

    <WebServer
      DefaultWebApp="wpc"
      LogFileName="access.log"
      LoggingEnabled="true"
      Name="wpc_server"
    />

    <Log
      FileName="weblogic.log"
    />
  </Server>

  <Application Deployed="true" Name="suppliers"
    Path="/wpc/envs/wl_username/phtml" TwoPhase="false">
    <WebAppComponent Name="suppliers" Targets="wpc_server" URI="suppliers"/>
  </Application>
  <Application Deployed="true" Name="wpc" Path="/.applications"
    StagedTargets="" TwoPhase="false">
    <WebAppComponent Name="wpc" Targets="wpc_server" URI="wpc.war"/>
  </Application>

  <Security
    Name="wpc_domain"
    GuestDisabled="false"
  />

  <Log
    FileName="wl-domain.log"
  />

  <StartupClass ClassName="austin.appsvr.current.admin.RemoteWeblogic"
    FailureIsFatal="true" Name="RemoteWeblogic" Targets="wpc_server"/>

</Domain>
```

WebLogic 域 applications 目录

将 WebLogic 域 applications 目录
(\$TOP/etc/default/weblogic700/config/wpc_domain/applications/ 目录)
复制到新的域目录中。

要创建指向 applications 目录的符号链接，删除应用程序目录，替换为指向 WebSphere Product Center 代码库的符号链接。

例如（使用上一节中的示例 —

```
(/wpc/envs/wl_username/wl_domain/wpc_domain/applications) :

wl_username@app02:~/wl_domain/wpc_domain$ mv applications applications.bk/

wl_username@app02:~/wl_domain/wpc_domain$ ln -
s /wpc/envs/wl_username/code/wpc42010009/etc/default/weblogic700/config/wpc_domain/applications/.

wl_username@app02:~/wl_domain/wpc_domain/applications$ ls -l

total 28896

-rw-r--r-- 1 wl_username envs 29553587 Dec 31 13:36 wpc.war

wl_username@app02:~/wl_domain/wpc_domain/applications$
```

验证公用文件目录

验证定义为“suppliers”应用程序的目录是否与 WebSphere Product Center 用于 public_html 的目录相同。例如，创建从 \$TOP/public_html 到 ~/phtml 的符号链接

```
wl_username@app02:~/code/wpc42010009$ mv public_html/ public_html.bk

wl_username@app02:~/code/wpc42010009$ ln -s ~/phtml/ public_html

wl_username@app02:~/code/wpc42010009$
```

WebLogic 问题

WebSphere Product Center 消失不见了

如果 applications 目录的位置不正确，则 WebLogic 会自动除去 WebSphere Product Center。解决方案是在 config.xml 文件中重新添加 WebSphere Product Center 并将缺省 Web 应用程序重新设置为 WebSphere Product Center。

嵌入式 LDAP 锁定错误

启动会产生以下错误 -

```
(...)ldap/ldapfiles/EmbeddedLDAP.lok, No locks available,errno=37
```

这种锁定问题的解决方案是将 wpc_server 目录移至本地磁盘。这可从 NFS wpc_domain 建立符号链接。

设置数据库模式

注意：在创建数据库模式之前，必须设置 common.properties 文件。（请参阅本章中的“配置运行时属性”一节。）

验证数据库用户

验证 common.properties 中引用的数据库用户在数据库中是否存在并且

是否具有正确的特权。

执行以下 shell 脚本以测试 WebSphere Product Center 能否与数据库通信：

```
$STOP/bin/test_java_db.sh
```

此脚本尝试对 `common.properties` 进行语法分析并登录到数据库。

- 如果此脚本生成了错误，则请修改 `common.properties` 文件以设置正确的数据库用户，或者与数据库管理员一起验证是否正确地创建了数据库用户。

创建数据库模式

在正确地设置 DB 用户之后，就可以创建 WebSphere Product Center 数据库模式了。

执行 shell 脚本：

```
$STOP/src/db/schema/create/create_schema.sh
```

此脚本首先显示它正在尝试连接的数据库。然后，系统提示用户进行确认，即输入 “y” 继续，或者输入 “n” 停止。

请输入 “y” 以继续。

注意：在创建数据库模式之后，不需要再次运行 `create_schema.sh`。数据库模式只需要创建一次。

设置要测试的公司

本节设置一间用来测试 WebSphere Product Center 安装的公司。必须创建一间公司才能登录到 WebSphere Product Center。要创建新公司，请参考“创建新公司”一节。

创建要测试的公司

要登录到 WebSphere Product Center，必须创建一间公司。执行下列其中一项（或执行这两项）以创建一间公司：

创建 Acme 公司

要创建名为 “acme” 的演示公司，请执行以下 shell 脚本：

```
$STOP/src/db/schema/acme/create_acme.sh --code=acme
```

创建的 Acme 公司有 4 个用户：

Admin、bwilson、Jwilson 和 vjackson

为所有用户都提供了 “trinitron” 密码

创建空公司

要创建名为 “demo” 并且没有任何预定义数据的空公司（比创建 acme 演示公司要快），请执行以下 shell 脚本：

```
$TOP/src/db/schema/cmp/create_cmp.sh --code=demo
```

这就创建了 demo 公司且具有一个用户：

Admin

Admin 的密码是 “trinitron”

注意密码区分大小写。在创建后，Admin 用户具有全部特权，并且应该由管理员使用。

测试 WebSphere Product Center 安装

启动 WebSphere Product Center

要启动 WebSphere Product Center，请执行脚本 start_local.sh，此脚本启动运行 WebSphere Product Center 所需的所有服务。

```
$TOP/bin/go/start_local.sh
```

通过执行以下脚本，验证 WebSphere Product Center 是否已启动并正在运行：

```
$TOP/bin/go/rmi_status.sh
```

验证下列服务是否已启动：

- admin
- appsvr
- eventprocessor
- queuemanager
- scheduler
- workflow

start_local.sh 的正常输出的一个示例：

```
#!/start_local.sh  
killing services on localhost  
killing service 'appsvr_LORAX'  
killing service 'admin_LORAX'  
killing service 'ajp12_LORAX'
```

```
killing service 'ajp13_LORAX'
killing service 'eventprocessor'
killing service 'queuemanager'
killing service 'scheduler'
killing service 'rmi'
starting rmiregistry on port 17507
starting service 'admin_LORAX'
starting service 'appsvr_LORAX'
starting service 'eventprocessor'
starting service 'queuemanager'
starting service 'scheduler'
```

通过执行以下脚本，验证应用程序是否已启动并正在运行：

```
/usr/trigo/pink_tulip/austin/bin/go/rmi_status.sh
```

验证下列服务是否已启动：

```
admin_LORAX
appsvr_LORAX
eventprocessor
queuemanager
scheduler
```

注意：此过程应该会耗费大约 30-40 秒，这取决于处理器的速度。

检查 RMI 状态

通过执行以下脚本，验证 WebSphere Product Center 是否已启动并正在运行：

```
$STOP/bin/go/rmi_status.sh
```

此脚本与集群中的所有机器上的 RMI 守护进程联系并获取每台机器上的本地服务的列表。将返回一个比较长的名称列表。

以下是 rmi_status.sh 的一个示例，它显示在名为“LORAX”的机器上所有服务都已启动：

```
#!/rmi_status.sh
++ [success] rmistatus (Mon Aug 26 17:29:47 PDT 2003)
rmi://lorax:17507/CMP1/appsvr/appsvr_LORAX
rmi://lorax:17507/CMP1/admin/admin_LORAX
rmi://lorax:17507/CMP1/eventprocessor/eventprocessor_LORAX
rmi://lorax:17507/CMP1/scheduler/scheduler_LORAX
rmi://lorax:17507/CMP1/queuemanager/queuemanager_LORAX
rmi://lorax:17507/CMP1/workflow/workflow_LORAX
```

测试数据库连接

要使用 Java 来测试数据库连接以验证驱动程序，请运行以下脚本。

```
$STOP/bin/test_java_db.sh
```

登录到 WebSphere Product Center

打开 Internet Explorer Web 浏览器并输入 Web 服务器的 URL 和端口。

`http://www.acme.com:7507`

注意：在两层配置中，在 WebSphere Product Center 安装期间已将 Web 服务器端口设置为 7507。如果使用了另一个端口，则更改 `server.xml` 文件中的端口引用，或者（对于三层配置）更改 Apache 配置文件中的端口引用。

通过使用 Acme 公司演示，按如下方式登录：

用户名：bwilson
密码：trinitron
公司代码：acme

如果对 WebSphere Product Center 执行的登录打开了 WebSphere Product Center 主页，则表示安装已成功。请从该应用程序中注销并转到下一个步骤。

此时，如果所有连接的运行都正常，则打开 Internet Explorer Web 浏览器并输入 WebSphere Product Center 服务器的 URL 和端口。

`http://www.acme.com:7507`

如果已使用 `create_acme.sh` 来创建 Acme 演示公司，则按如下方式登录：

用户名：bwilson
密码：trinitron
公司：acme

如果已使用 `create_cmp.sh` 来创建公司，则按如下方式登录：

用户名：Admin
密码：trinitron
公司：acme

如果对 WebSphere Product Center 执行的登录打开了 WebSphere Product Center 主页，则表示安装已成功。

停止 WebSphere Product Center

要在本地机器上停止 WebSphere Product Center，请执行以下脚本：

`$STOP/bin/go/stop_local.sh`

该脚本停止 WebSphere Product Center 启动的所有服务。以下是 `stop_local.sh` 的示例输出，它显示已停止了所有的服务：

```
#!/stop_local.sh
stopping services on localhost
++ [success] stop service 'appsvr_LORAX' (Mon Aug 26 17:55:46 PDT
2002)
Websphere will stop in 5 seconds
++ [success] stop service 'admin_LORAX' (Mon Aug 26 17:55:47 PDT
2002)
admin will stop in 5 seconds
++ [success] stop service 'eventprocessor' (Mon Aug 26 17:55:47 PDT
2002)
event processor stopped
++ [success] stop service 'scheduler' (Mon Aug 26 17:55:47 PDT 2002)
scheduler will stop in 5 seconds
++ [success] stop service 'queuemanager' (Mon Aug 26 17:55:48 PDT
2002)
queue manager stopped
killing service 'rmi'
```

创建新公司

既然 WebSphere 已经安装并测试完毕，请创建一间新公司以便与测试或生产环境配合使用。如果您愿意的话，可以创建多间公司。

例如，要创建名为 “test” 并且没有任何预定义数据的空公司，请执行以下 shell 脚本：

```
$STOP/src/db/schema/cmp/create_cmp.sh --code=test
```

在创建后，该公司具有单个缺省管理员用户：

用户名：Admin
密码：trinitron

注意：Admin 用户自动地被创建为具有全部特权。此用户帐户将由管理员使用。