Tivoli Application Dependency Discovery Manager

V 7.3

管理员指南

Tivoli Application Dependency Discovery Manager

V 7.3

管理员指南

IBM

注意

在使用本资料及其支持的产品之前，请阅读第 245 页的『声明』中的信息。

版本声明

本修订版适用于 IBM Tivoli Application Dependency Discovery Manager V7.3（产品号 5724-N55）以及所有后续发行版和修订版，直到新版本中另有声明为止。

目录

表

关于本资料

本 PDF 文档版本旨在以可打印格式提供信息中心中的相关主题。

本信息中心中使用的约定

在 IBM® Tivoli Application Dependency Discovery Manager (TADDM) 文档中，使用了特定的约定。 这些约定用于引用与操作系统相关的变量和路径、COLLATION_HOME 目录和 collation.properties 文件位置，整个 TADDM 文档（包括消息）都对此文件进行了引用。

依赖于操作系统的变量和路径

在此信息中心内，UNIX 约定用于指定环境变量和表示目录。

使用 Windows 命令行时，对于环境变量，请将 $*variable* 替换为 %*variable*%，并将目录路径中的每个正斜杠 (/) 替换为反斜杠 (\)。

如果您是在 Windows 系统上使用 Bash Shell，那么可以使用 UNIX 约定。

COLLATION_HOME 目录

TADDM 根目录也称为 COLLATION_HOME 目录。

在 AIX® 或 Linux 之类的操作系统上，TADDM 的缺省安装位置是 /opt/IBM/taddm 目录。 因此，在这种情况下，$COLLATION_HOME 目录是 /opt/IBM/taddm/dist。

在 Windows 操作系统上，TADDM 的缺省安装位置是 c:\IBM\taddm 目录。因此，在这种情况下，%COLLATION_HOME% 目录是 c:\IBM\taddm\dist。

collation.properties 文件的位置

collation.properties 文件包含 TADDM 服务器属性，并包含有关每个属性的注释。此文件位于 $COLLATION_HOME/etc 目录中。

术语和定义

请参阅以下术语及定义列表，以了解 IBM Tivoli Application Dependency Discovery Manager (TADDM) 中的重要概念。

访问集合 (access collection)
用于控制对配置项的访问权和修改配置项的许可权的集合。 仅当启用数据级安全时，才能创建访问集合。

异步发现 (asynchronous discovery)
在 TADDM 中，这是指在目标系统上运行发现脚本，以发现 TADDM 服务器无法直接访问的系统。 由于此发现是手动执行的，并且独立于典型凭证发现，因此将其称为"异步"。

业务应用程序 (business application)
组件的集合，它提供可在内部、外部使用或与其他业务应用程序一起使用的业务功能。

CI 请参阅配置项 *(configuration item)*。

集合 (collection)
在 TADDM 中，这是指一组配置项。

配置项 (configuration item, CI)
这是 IT 基础结构的一个组件，它由配置管理控制，因此受正式更改控制约束。TADDM 数据库中的每个 CI 都有与其关联的持久对象和变更历史记录。 CI 的示例包括操作系统、第 2 层接口和数据库缓冲池大小。

带凭证发现 (credentialed discovery)
这是用于发现有关下列各项的详细信息的 TADDM 传感器扫描：

- 运行时环境中的每个操作系统。 此扫描也称为第 2 层发现，并且需要操作系统凭证。
- 运行时环境中使用的应用程序基础结构、已部署的软件组件、物理服务器、网络设备、虚拟系统和主机数据。 此扫描也称为第 3 层发现，并且需要操作系统凭证和应用程序凭证。

无凭证发现 (credential-less discovery)
用于发现有关运行时环境中活动计算机系统的基本信息的 TADDM 传感器扫描。 此扫描也称为第 1 层发现，并且不需要凭证。

数据管理门户网站 (Data Management Portal)
这是基于 Web 的 TADDM 用户界面，用于查看和处理 TADDM 数据库中的数据。 此用户界面适用于域服务器部署、同步服务器部署以及流式服务器部署中的每个存储服务器。 此用户界面在所有部署中都非常相似，只是在同步服务器部署中具有几项附加功能用于添加域以及对域进行同步。

发现工作程序线程 (discover worker thread)
在 TADDM 中，这是运行传感器的线程。

发现管理控制台 (Discovery Management Console)
这是用于管理发现的 TADDM 客户机用户界面。 此控制台也称为"产品控制台"。 它适用于域服务器部署以及流式服务器部署中的发现服务器。 控制台的功能在这两种部署中相同。

发现服务器 (discovery server)
这是在流式服务器部署中运行传感器但自身没有数据库的 TADDM 服务器。

域 (domain)
在 TADDM 中，这是公司或其他组织的基础结构的逻辑子集。 域可以勾勒出组织、功能或地理界限。

域服务器 (domain server)
这是在流式服务器部署中运行传感器且自身有数据库的 TADDM 服务器。

域服务器部署 (domain server deployment)
这是有一个域服务器的 TADDM 部署。 域服务器部署可以是同步服务器部署的组成部分。

在域服务器部署中，必须将以下 TADDM 服务器属性设置为以下值：

```
com.collation.cmdbmode=domain
```

在上下文中启动 (launch in context)

这是一个概念，即，通过单点登录以及位于适当位置的目标 UI，从一个 Tivoli® 产品 UI 无缝地切换到另一个 Tivoli 产品 UI（在不同的控制台中，或者在同一个控制台或门户网站界面中），以便用户继续执行其任务。

第 1 层发现 (Level 1 discovery)

用于发现有关运行时环境中活动计算机系统的基本信息的 TADDM 传感器扫描。 由于此扫描不需要凭证，因此也称为无凭证发现。 它使用堆栈扫描传感器和 IBM® Tivoli® Monitoring Scope 传感器。 第 1 层发现非常浅。 它只收集每个已发现接口的主机名、操作系统名称、IP 地址、标准域名和介质访问控制 (MAC) 地址。 此外，MAC 地址发现限于 Linux on System z® 和 Windows 系统。 第 1 层发现不会发现子网。 如果任何已发现的 IP 接口不属于在第 2 层或第 3 层发现期间发现的现有子网，那么将根据 `collation.properties` 文件中 `com.collation.IpNetworkAssignmentAgent.defaultNetmask` 属性的值创建新的子网。

第 1 层发现 (Level 2 discovery)

用于发现有关运行时环境中每个操作系统的详细信息的 TADDM 传感器扫描。 此扫描也称为需要凭证的发现，并且需要操作系统凭证。 第 2 层发现收集应用程序名称以及与每个正在运行的应用程序相关联的操作系统名称和端口号。 如果某个应用程序已建立与其他应用程序的 TCP/IP 连接，那么收集此信息作为依赖关系。

第 3 层发现 (Level 3 discovery)

用于发现运行时环境中使用的应用程序基础结构、已部署的软件组件、物理服务器、网络设备、虚拟系统和主机数据的相关详细信息的 TADDM 传感器扫描。 此扫描也称为需要凭证的发现，并且需要操作系统凭证和应用程序凭证。

多承租 (multitenancy)

在 TADDM 中，这是指服务供应商或 IT 供应商使用一个 TADDM 安装来发现多个客户环境。 并且，该服务供应商或 IT 供应商还可以看到来自所有客户环境的数据，但在每个客户环境中，特定于相应客户的数据只能在该客户环境的用户界面中显示或者在该客户环境中的报告中查看。

产品控制台 (Product Console)

请参阅发现管理控制台 *(Discovery Management Console)*。

基于脚本的发现 (script-based discovery)

在 TADDM 中的带凭证发现中，这是指使用传感器为支持异步发现而提供的相同传感器脚本。

SE 请参阅服务器等效 *(server equivalent)*。

服务器等效 (server equivalent, SE)

这是 IT 基础结构的一个代表单元，定义为安装了服务器软件（例如数据库、Web 服务器或应用程序服务器）的计算机系统（配有标准配置、操作系统、网络接口和存储接口）。 服务器等效这一概念还包括那些为了优化服务器功能而提供服务的网络子系统、存储子系统和其他子系统。 服务器等效依赖于操作系统：

操作系统	CI 的大致数目
Windows	500

操作系统	CI 的大致数目
AIX	1000
Linux	1000
HP-UX	500
网络设备	1000

存储服务器 (storage server)

这是一个 TADDM 服务器，用于处理从发现服务器接收的发现数据并将该数据存储在 TADDM 数据库中。 主存储服务器既负责协调发现服务器和所有其他存储服务器，又充当存储服务器。 所有并非主存储服务器的存储服务器都称为辅助存储服务器。

流式服务器部署 (streaming server deployment)

这是有一个主存储服务器和至少一个发现服务器的 TADDM 部署。 此类部署还可以包括一个或多个可选的辅助存储服务器。 主存储服务器和辅助存储服务器共享一个数据库。 发现服务器没有数据库。

在此类部署中，发现数据从多个发现服务器并行流向 TADDM 数据库。

在流式服务器部署中，必须将以下 TADDM 服务器属性设置为下列其中一个值：

- `com.collation.taddm.mode=DiscoveryServer`
- `com.collation.taddm.mode=StorageServer`

对于除主存储服务器以外的所有服务器，还必须设置下列属性（用于指定主存储服务器的主机名和端口号）：

- `com.collation.PrimaryStorageServer.host`
- `com.collation.PrimaryStorageServer.port`

如果设置了 com.collation.taddm.mode property，那么不得设置 com.collation.cmdbmode 属性，或者必须将该属性注释掉。

同步服务器 (synchronization server)

这是用于对来自企业中所有域服务器的发现数据进行同步并且自身有数据库的 TADDM 服务器。 此服务器并不直接发现数据。

同步服务器部署 (synchronization server deployment)

这是一种 TADDM 部署，它包含一个同步服务器部署以及两个或两个以上域服务器部署，并且每个服务器部署都有自身的本地数据库。

在此类部署中，同步服务器在批处理同步过程中复制来自多个域服务器的发现数据，且每次复制一个域。

在同步服务器部署中，必须将以下 TADDM 服务器属性设置为以下值：

`com.collation.cmdbmode=enterprise`

此类部署已过时。 因此，在需要多个服务器的新 TADDM 部署中，请使用流式服务器部署。可以将同步服务器转换为流式服务器部署的主存储服务器。

TADDM 数据库 (TADDM database)

在 TADDM 中，这是用于存储配置数据、依赖关系和变更历史记录的数据库。

除发现服务器和辅助存储服务器以外，每个 TADDM 服务器都有其自身的数据库。 发现服务器没有数据库。 各个存储服务器共享主存储服务器的数据库。

TADDM 服务器 (TADDM server)

这是可以表示下列任何一项的通用术语：

- 域服务器部署中的域服务器
- 同步服务器部署中的同步服务器
- 流式服务器部署中的发现服务器
- 流式服务器部署中的存储服务器（包括主存储服务器）

目标系统 (target system)

在 TADDM 发现过程中，这是指所要发现的系统。

利用率发现 (utilization discovery)

用于发现主机系统的利用率信息的 TADDM 传感器扫描。 利用率发现需要操作系统凭证。

管理

TADDM 概述

IBM Tivoli Application Dependency Discovery Manager (TADDM) 是一款配置管理工具，用于帮助 IT 操作人员确保并改善应用程序环境中的应用程序可用性。TADDM 使用对资产及其应用程序依赖关系的自动化、无代理发现来提供配置项 (CI) 的详细信息，并包含发现库技术以帮助利用来自其他源的数据。

TADDM 为运营工作人员提供了应用程序的自顶向下视图，以便他们可以快速了解其业务关键应用程序的结构、状态、配置和变更历史记录。 发生性能和可用性问题时，此视图能帮助工作人员立即隔离问题，并且更高效地规划应用程序更改而不致引起混乱。TADDM 数据库是一种配置管理数据库，它不需要定制基础结构建模就能进行创建和维护。 TADDM 还提供完整的跨层依赖关系图、拓扑视图、更改跟踪、事件传播以及详细的报告和分析。

TADDM 依赖于对信息的发现，这种发现是使用作为 TADDM 产品一部分而部署的传感器所执行的。 将使用发现进程中产生的数据来创建链接物理和逻辑拓扑的跨层依赖关系图。 此分层目录表示整个运行时环境。

下列步骤是 TADDM 所执行操作的高级摘要：

1. 传感器确定并收集每个应用程序、系统和网络组件的身份、属性和设置。
2. 配置数据、依赖关系和变更历史记录存储在 TADDM 数据库中，拓扑存储在 TADDM 服务器上。 发现 CI 后，会将其存储在以下源的 TADDM 数据库中：
   - 传感器
   - 发现库簿，也称为身份标记语言 (IdML) 簿，由外部管理软件系统生成
   - API
3. 发现的数据以运行时跨层应用程序拓扑的形式显示在 TADDM 用户界面中。 后续发现将更新该拓扑。 此外，TADDM 还将维护基础结构配置和依赖关系的变更历史记录。
4. TADDM 生成存储在 TADDM 数据库中的信息的报告和其他拓扑视图。

TADDM 发现的实体

表 1 列示并描述了 TADDM 在环境中发现的实体。

表 *1.* 带有描述的已发现实体

实体	描述
网络层	以下设备是通过对每个设备使用 MIB2 (RFC 1213) 参数值所发现的： • 路由器 • 交换机 • 负载均衡器 • 防火墙 • 通用 IP 设备
系统层	以下设备是在系统层发现的： • 服务器主机和磁盘 • 主机 IP 接口 • 数据库服务器 • 负载均衡器或集群
应用程序层	以下组件是在应用程序层发现的。 此外，对于每个组件（除通用过程外），还会发现版本信息、配置文件和属性、主机信息和特定于供应商的扩展。 • 定制服务器，基于设计的定制模板 • Java EE 应用程序服务器和配置 • Java EE 与 Java SE 组件和模块 • Web 服务器组件 • Web 模块、配置文件和安装目录 • 通用 JVM 进程 • 数据库
基础结构服务	将发现支持应用程序环境的系统基础结构服务，以及这些服务组件和应用程序组件之间的依赖关系结构。基础结构服务中包含以下组件： • DNS 和 NFS 服务 • LDAP
关系结构	除了发现组件，还会在每层的以下支持级别上发现网络、系统和应用程序层的物理和逻辑连接： • 第 3 层 IP 连接 • 第 2 层连接 • 应用程序组件运行时依赖关系 • 基础结构服务依赖关系

将在以下实体之间发现配置和相互依赖关系：

- 应用程序组件，如 Web 服务器、应用程序服务器和数据库
- 系统组件，如主机、操作系统、负载均衡器和数据库服务器
- 网络组件，如路由器、交换机和防火墙

- 基础结构服务，如 DNS 和 LDAP 服务

注：使用虚拟 IP 地址或多个网络接口控制器可能会使 TADDM 错报结果。 在规划发现时，请考虑网络基础结构。

发现进程概述

发现是一种多层次过程，用于收集有关整个应用程序基础结构的配置信息，从而标识运行时环境中使用的已部署软件组件、物理服务器、网络设备、虚拟系统和主机数据。 发现通过 TADDM 产品中包含的传感器执行。

传感器的任务是发现配置项 (CI)，创建模型对象并将模型对象持久存储到 TADDM 数据库中。传感器所使用的协议特定于这些协议专用于发现的资源。 以下是示例协议：

- Cisco 发现协议 (CDP)
- Java™ 管理扩展 (JMX)
- 安全 Shell (SSH)
- 简单网络管理协议 (SNMP)
- 结构化查询语言 (SQL)

在可能的情况下，会在 TADDM 服务器和目标系统之间使用安全连接。

TADDM 不会运行 IPv6 网络中的发现，但是 IPv4 网络中运行的发现会发现 IPv6 属性。

传感器

TADDM 提供了各种专用传感器，以便在应用程序软件、主机和网络层之间发现典型数据中心内几乎所有的组件。 也可以为唯一组件开发定制传感器。 传感器位于 TADDM 服务器上，用于收集配置属性和依赖关系。

传感器为非侵入性的，也即它们运行在 TADDM 服务器上而非客户机工作站上。 因此，通过使用 TADDM，可以收集与发现相关的信息，而不必对要发现的每个客户机工作站上的代理程序进行本地安装和维护。

由于传感器使用安全网络连接、加密访问凭证和主机本机实用程序，因此比较安全，并且传感器提供的数据采集级别与使用位于客户机工作站上的软件时相同。

传感器具有以下三个可配置的方面：

作用域

发现作用域通常是一个有效的 IP 范围、子网或特定的地址。 它设置了发现的范围。

访问列表

访问列表是凭证（如用户名、密码和简单网络管理协议 (SNMP) 共用字符串）的集合，供传感器在访问应用程序基础结构中的配置项时使用。 必须针对要发现的配置项来配置访问列表。

调度 发现可根据需要运行，根据调度运行，或由外部触发的事件驱动。 调度标识了传感器是根据需要还是根据调度运行。

传感器如何发现配置项：

这些步骤概述了传感器如何在环境中发现配置项 (CI)。

1. 要识别指定作用域中的活动 IP 设备，传感器会在几个端口（如 22、23 和 135）上尝试 TCP 连接以检测响应。 只要有响应，就足以告知传感器该设备存在。
2. 传感器在几个端口（如 22 和 135）上尝试连接到 IP 设备，以确定用于发现主机的技术。
3. 如果打开了使用安全 Shell (SSH) 协议的端口，传感器将尝试使用访问列表中的凭证建立 SSH 连接。 传感器将依次尝试类型为**计算机系统**或 **windows 计算机系统**的访问列表条目，直到某个条目起作用或传感器到达访问列表末尾而仍未成功。
4. 如果打开了 Windows Management Instrumentation (WMI) 端口，将建立与网关计算机系统（如果可以为目标系统找到一个网关计算机系统）之间的 SSH 连接。传感器将依次尝试类型为 **windows 计算机系统**的访问列表条目，直到某个条目起作用或传感器到达访问列表末尾而仍未成功。
5. 如果无法建立会话，将运行 SNMP 传感器。 如果会话已建立，将运行计算机系统传感器。
6. 计算机系统传感器尝试确定所安装操作系统的类型。
7. TADDM 运行特定于操作系统的传感器，以发现操作系统的更多详细信息。
8. 在基于特定条件（如端口号和进程名称）对操作系统进行深度发现期间，TADDM 运行特定于软件的传感器以发现应用程序详细信息。

如何启动应用程序传感器：

此信息描述如何启动应用程序传感器。

GenericServerSensor 运行以下命令：

在 Linux、Solaris、AIX 和 Linux on System z® 操作系统上

- **lsof -nP -i** 用于获取端口信息
- **ps axww** 用于获取命令行信息

在 Windows 操作系统上

- **netstat.exe -nao** 用于获取端口信息
- **wmic process list** 用于获取命令行信息

进程标识 (PID) 用于合并输出。 然后，模板匹配器按照已合并的数据进行操作。 如果在 collation.properties 文件中将记录级别设置为 DEBUG，那么这些命令的输出将位于以下日志中：

- GenericServerSensor.log
- DiscoverManager.log

合并的数据必须与传感器模板中定义的条件相匹配。 可以在 DB2® 传感器的以下样本模板定义中找到启动传感器的模板条件。

请运行以下命令（重定向到文件十分有用），并将 *<username>* 和 *<password>* 替换为有效的用户名以及相关联的密码（例如 ...dist/sdk/bin/api.sh -u administrator -p collation find --depth=5 AppServerTemplate）：

```
...dist/sdk/bin/api.sh -u <username> -p <password> find --depth=5 AppServerTemplate
```

上述命令将生成 XML 输出，即模板定义。 在模板定义中，如果 <order> 元素的值小于 0，那么该模板是传感器模板。 如果 <order> 元素的值大于 0，那么该模板是定制服务器模板。 匹配从 <order> 元素的最小值开始执行，这也是传感器的匹配优先级比定制服务器高的原因。

以下是 DB2 传感器的样本模板定义。 请注意两个 <operand1> 元素，其中一个值为 db2tcpcm，另一个值为 db2agent。 <boolExp> 元素的值指示这两个 <operand1> 值都必须存在还是只能存在其中一个值。 <boolExp> 元素的值为 1 指示逻辑运算符 OR，这表示只能存在其中一个 <operand1> 值。 <boolExp> 元素的值为 0 指示逻辑运算符 AND，这表示两个 <operand1> 值都必须存在。

```
<Template array="18" guid="C1A992327AFF33409C41D5C71046DBB9"
lastModified="1177555771479"
xsi:type="coll:com.collation.platform.model.discovery.template.AppServerTemplate">
   <displayName>DB2</displayName>
   <name>DB2</name>
   <type>DatabaseServer</type>
   <internal>true</internal>
   <filterSet guid="B599AED918F436C99FDA0E8EDA578F02"
   lastModified="1177555771475"
   parent="C1A992327AFF33409C41D5C71046DBB9"
   xsi:type="coll:com.collation.platform.model.discovery.template.FilterSet">
      <displayName>DB2</displayName>
      <filterList array="1"
      guid="BBE4D351653B37E38BFFD2DEBD532EE8"
      lastModified="1177555771476"
      parent="B599AED918F436C99FDA0E8EDA578F02"
      xsi:type="coll:com.collation.platform.model.discovery.template.Filter">
         <displayName>unknown</displayName>
         <operand1>db2tcpcm</operand1>
         <operator>contains</operator>
         <part>Program Name</part>
      </filterList>
      <filterList array="2"
      guid="63816C902B0A317F8C3B24C7A1EEBC17"
      lastModified="1177555771471"
      parent="B599AED918F436C99FDA0E8EDA578F02"
      xsi:type="coll:com.collation.platform.model.discovery.template.Filter">
         <displayName>unknown</displayName>
         <operand1>db2agent</operand1>
         <operator>contains</operator>
         <part>Program Name</part>
      </filterList>
      <boolExp>1</boolExp>
   </filterSet>
   <index>0</index>
   <order>-10</order>
   <enabled>true</enabled>
   <action>1</action>
   <source>0</source>
   <seedClass>com.collation.discover.seed.app.db.db2.Db2Seed</seedClass>
</Template>
```

发现层次

TADDM 提供四个层次的发现：第 1 层发现、第 2 层发现、第 3 层发现和利用率发现。

第 1 层发现

用于发现有关运行时环境中活动计算机系统的基本信息的 TADDM 传感器扫描。 此扫描也称为无凭证发现，因为它不需要凭证。 它使用堆栈扫描传感器和 IBM Tivoli Monitoring 作用域传感器。

第 1 层发现非常浅。 它只收集每个已发现接口的主机名、操作系统名称、IP 地址、标准域名和介质访问控制 (MAC) 地址。 此外，MAC 地址发现限于 Linux on System z 和 Windows 系统。

第 1 层发现不发现子网。 如果任何已发现的 IP 接口不属于在第 2 层或第 3 层发现所发现的现有子网，那么将根据 `collation.properties` 文件中 com.collation.IpNetworkAssignmentAgent.defaultNetmask 属性的值来创建新的子网。

第 2 层发现

用于发现有关运行时环境中每个操作系统的详细信息的 TADDM 传感器扫描。此扫描也称为凭证发现，它需要操作系统凭证。

第 2 层发现收集与每个运行的应用程序相关联的应用程序名称、操作系统名称和端口号。 如果某个应用程序与另一个应用程序建立了 TCP/IP 连接，将收集此信息作为依赖关系。

第 3 层发现

用于发现有关运行时环境中所使用应用程序基础结构、已部署软件组件、物理服务器、网络设备、虚拟系统和主机数据的详细信息的 TADDM 传感器扫描。此扫描也称为凭证发现，它需要操作系统凭证和应用程序凭证。

利用率发现

用于发现有关主机系统的利用率信息的 TADDM 传感器扫描。 利用率发现需要操作系统凭证。

第 2 层和第 3 层发现收集的信息比第 1 层发现更详细。 如果在第 2 层或第 3 层发现期间创建的对象与先前由第 1 层发现创建的对象相匹配，那么由第 1 层发现创建的对象将替换为新创建的对象，这就转而导致更改这些对象的全局唯一标识 (GUID)。 因此，通常，第 1 层数据不应该用于与其他产品的集成。

发现概要文件

要运行发现，必须指定发现概要文件，该文件为发现定义一组选项。 通过发现概要文件，您可以配置个别传感器，管理同一传感器的多项配置，根据条件组选择合适的配置，以及管理要在运行单个发现时应用的不同传感器的配置组。

通过选择相应的发现概要文件，您可以控制发现的深度或发现层次。

缺省情况下，TADDM 提供四个发现概要文件。 其中三个用于您可以选择的三层发现（第 1 层、第 2 层或第 3 层），具体取决于要执行的是无凭证发现还是凭证发现。余下的第 4 个概要文件用于利用率发现。

如果未指定概要文件，缺省情况下将使用第 3 层发现概要文件，但您也可以在发现管理控制台中更改缺省概要文件。

有关发现概要文件的其他信息，请参阅 TADDM Wiki (https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Application%20Dependency%20Discovery%20Manager/page/A%20Flexible%20Approach%20to%20Discovery) 上的《灵活的发现方法》。

启用和禁用传感器

即使概要文件中已经启用了传感器，也可以全局禁用该传感器。您还可以全局启用传感器，并允许概要文件中的设置起作用。

例如，如果某个传感器已全局启用并在概要文件中启用，该传感器将会运行。 如果该传感器已全局启用但在概要文件中禁用，那么为运行发现而选择相应的概要文件时，该传感器不会运行。

要对具有 osgi 目录 ($COLLATION_HOME/osgi/plugins) 的传感器进行全局启用和禁用，必须更改 osgi 目录中的 **AgentConfigurations**。

例如，对于 Db2Sensor，请查找以下目录：

- $COLLATION_HOME/osgi/plugins/com.ibm.cdb.discover.sensor.app.db.db2_*x.x.x*/Db2Sensor.xml
- $COLLATION_HOME/osgi/plugins/com.ibm.cdb.discover.sensor.app.db.db2windows_*x.x.x*/Db2WindowsSensor.xml

其中 *x.x.x* 是传感器插件版本，例如：7.3。

编辑这些 XML 文件时，要启用传感器，请将 enabled 设置为 true。要禁用传感器，请将 enabled 设置为 false。

对于不使用 osgi/plugins 目录的传感器，其配置信息存储在 etc/discover-sensors 目录中的传感器配置 XML 文件中。

异步和基于脚本的发现

在异步和基于脚本的发现中，传感器提供发现脚本，而不运行单独的命令，传感器将针对目标系统运行该发现脚本。

并不是所有的传感器都支持异步和基于脚本的发现。只有提供发现脚本的传感器才支持这些类型的发现。

有关哪些传感器支持异步和基于脚本的发现的信息，请参阅 TADDM *Sensor Reference* 中的支持基于脚本和异步发现的传感器主题。

与非基于脚本的发现的一些区别

异步发现和基于脚本的发现与非基于脚本的凭证发现的区别体现在以下几个重要方面：

- 相比于非基于脚本的第 2 层或第 3 层发现的发现结果，异步或基于脚本的发现的发现结果可能不够完整。 大多数传感器在非基于脚本的发现中所发现的模型对象、属性和关系数量比在异步或基于脚本的发现中所发现的更多。
- 在异步或基于脚本的发现中，应用程序传感器仅针对给定的目标系统启动一次。 但当应用程序侦听多个端口时，将会发现所有的应用程序实例。
  在非基于脚本的发现中，将针对每个应用程序实例启动一个应用程序传感器。

异步发现：

您可以运行异步发现，以发现那些无法直接通过 TADDM 服务器进行访问的系统。 包括位于安全位置的系统（例如，不可通过网络访问的系统）、不运行安全 Shell (SSH) 的系统以及含有敏感信息（无法获取凭证）的系统。

在异步发现中，用户可以在目标系统上运行发现脚本。 发现脚本包含一个主脚本和多个传感器脚本。 在典型发现中运行传感器脚本时，每个传感器脚本都会提供一个发现功能，该功能与传感器具有的功能相类似。

发现脚本的输出是一个包含发现结果的归档文件。 必须将此文件复制到 TADDM 服务器中。在 TADDM 发现期间，TADDM 传感器处理此归档文件中的发现结果（而不是运行命令）。

由于此发现以手动方式执行，并且独立于需要凭证的典型发现，因此将其称为"异步"。

要运行异步发现，需要有异步发现传感器。 有关更多信息，请参阅 TADDM *Sensor Reference*。

有关将传感器配置为运行异步发现的信息，请参阅 第 93 页的『针对异步发现进行配置』。

基于脚本的发现：

在基于脚本的发现中，您可以使用需要凭证的典型发现中的发现脚本。 在此类发现中，将使用异步发现中的同一传感器脚本。

在基于脚本的发现中，传感器不会运行个别命令。相反，传感器脚本在目标系统上运行。 可能不需要特定于应用程序的凭证。

例如，要在典型发现中发现 IBM WebSphere® 应用程序，如果已启用安全性，那么您必须为 WebSphere 应用程序创建一个带有凭证的访问列表条目。 但使用基于脚本的发现时，不需要 WebSphere 访问列表条目。 基于脚本的发现也无需使用特定于应用程序的协议（如 Java 管理扩展 (JMX)），这样可以通过 IBM Tivoli Monitoring 扩展应用程序发现。

有关将传感器配置为运行基于脚本的发现的信息，请参阅 第 96 页的『针对基于脚本的发现进行配置』。

并发发现

您可以同时运行多个发现，这称为并发发现。 例如，由于大型发现可能需要几个小时才能完成，因此您可能希望在大型发现完成前就开始运行较小的发现。您必须先正确配置并发发现，然后才能运行这些发现。

可以使用另一个发现概要文件运行并发发现，而不使用用于启动第一个发现的发现概要文件。

要管理并发发现，请使用发现管理控制台或 `api.sh` 脚本。 有关使用 `api.sh` 脚本的更多信息，请参阅 TADDM *SDK Developer's Guide* 中的 *Command-line interface API* 主题。

可以在同一目标系统上运行并发发现。 如果有两个或多个发现要监视某些相同的 IP 地址，每个发现都将独立运行。

如果在发现运行时发生密码更改，并且已启动并发发现，则该并发发现中的传感器将立即使用新的凭证，前提是这些传感器在密码更改前未启动。

TADDM 不支持具有基于概要文件的访问列表的并发发现。

如果在某个发现运行时对定制服务器模板进行了更改，那么已启动的所有并发发现都将继续使用定制服务器模板的原有版本。 下一个启动的单独非并发发现将使用定制服务器模板的新版本。

确定显示的 FQDN

您可以配置一个首选方法，用于确定每个已发现系统的标准域名 (FQDN)。

对于第 1 层发现，FQDN 是对 IP 地址进行反向查找的结果。 此查找使用操作系统提供的解析器库以及其中提供的任何配置。例如，在操作系统级别，如果主机文件的优先级高于 DNS，那么将优先考虑主机文件内的信息。

对于第 2 层发现，TADDM 使用操作系统提供的解析器库对所有已发现的 IP 地址执行反向查找。 此外，操作系统配置还指定了反向查找获取信息的位置。 如果未配置 DNS 或者 DNS 返回不需要的 FQDN，您可以使用主机文件将其覆盖。

查找已发现的 IP 地址后，会尝试将 FQDN 匹配到计算机系统。 有多种不同方法可获取 FQDN，将按预定义的顺序尝试每种方法，直到找到有效的 FQDN 为止。 您可以修改顺序以使您的首选方法具有更高优先级。 提供了以下方法：

方法 1

TADDM 选择 FQDN 主机部分与已发现系统主机名相匹配的 IP 接口的 FQDN。 如果存在多个匹配项，选择的 FQDN 将取决于 `com.collation.platform.os.FqdnPriorities` 属性中定义的域名优先级。 此属性按优先级顺序列出域名。 要设置域的优先级，请以逗号分隔的一行列表形式输入域名：

```
com.collation.platform.os.FqdnPriorities=domain1.company.com,
domain2.company.com,domain3.company.com
```

域的优先级最高的 FQDN 将作为该域的 FQDN 返回。 此方法使用发现的有关接口 FQDN 和计算机系统名称的信息。

如果未定义优先级，则 TADDM 检查所有 IP 接口。 TADDM 检查的是与给定 IP 接口关联的 FQDN 是否与计算机系统名称相同，或者此 FQDN 的主机名部分是否与计算机系统名称相同。 第一个符合条件的 FQDN 将作为 FQDN 返回。

例如，名为"myname"的计算机系统有两个接口，其 FQDN 如下：

- interface #1 myname.domain1.com
- interface #2 myname.domain2.com

如果未定义 `com.collation.platform.os.FqdnPriorities`，则第一个匹配作为 FQDN 名称返回。两个名称的 FQDN 主机部分都与所发现系统的主机名匹配，但返回的 FQDN 是"myname.domain1.com"。 要设置选择名称的优先级，应

使用属性 com.collation.platform.os.FqdnPriorities。 例如，如果 com.collation.platform.os.FqdnPriorities 条目包含以下信息：

```
com.collation.platform.os.FqdnPriorities=domain2.com,domain1.com
```

在这种情况下，返回的 FQDN 是"myname.domain2.com"，因为此名称的优先级较高。

方法 2

com.collation.platform.os.command.fqdn 属性在用于进行反向查找的 TADDM 服务器上指定外部命令。 以下示例显示如何使用此属性，请在一行上输入属性：

```
com.collation.platform.os.command.fqdn=nslookup $1
| grep Name | awk '{print $2}'
com.collation.platform.os.command.fqdn.AIX=nslookup $1
| grep Name | awk '{print $2}'
com.collation.platform.os.command.fqdn.Linux=nslookup $1
| grep Name | awk '{print $2}'
com.collation.platform.os.command.fqdn.SunOS=nslookup $1
| grep Name | awk '{print $2}'
com.collation.platform.os.command.fqdn.Windows=nslookup $1
```

方法 3

com.collation.platform.os.command.hostOfHostname 属性在用于提供 FQDN 的目标系统上指定外部命令。 以下示例显示如何在 UNIX 上使用此属性，请在一行上输入属性：

```
com.collation.platform.os.command.hostOfHostname=host `hostname`
| awk {'print $1'}
```

方法 4

使用主接口的 FQDN。 主 IP 接口被指定为最低的 IP 值，IP 值按升序排列。

方法 5

使用主接口的 IP 地址。

方法 6

使用计算机系统的名称。

方法 7

设置为会话上下文 IP。

方法 8

将 CS 的 FQDN 设置为会话 IP 的 FQDN。

您可以通过设置 com.collation.platform.os.fqdnSearchOrder 属性来定义这些方法的尝试顺序。 此属性的值是方法编号的逗号分隔列表。 缺省值是 1,2,3,4,5,6,7,8。在这种情况下，TADDM 首先尝试使用方法 1。如果此方法未能返回有效的 FQDN，将尝试方法 2，以此类推，直到获得有效的 FQDN 为止。 有效的 FQDN 是符合 RFC 1035 中指定的规则的标准域名。

此解决方案也适用于使用 SNMP 传感器发现的计算机系统。 您可以定义哪些解决方案具有更高优先级并因此可用于更快找到 FQDN。

在所有情况下，正确配置的 DNS 都是设置主机名的首选方法。 如果不能使用 DNS，请使用主机文件。 DNS 或主机文件的使用是提供 IP 地址名称解析的标准方法。 虽然 TADDM 提供了更优先的方法，但是由于其他任何方法都是 TADDM 特有的，因此这些方法可能导致名称与其他管理系统中的名称不一致。

跟踪发现

从发现启动时到更新变更历史记录和构建拓扑依赖关系时，您可以跟踪发现的各个阶段。 发现的每个阶段都会记录在关联的日志文件中。

发现运行阶段和日志文件

在启动发现后，系统会对每个发现指定一个唯一标识（运行标识）。 *YYYY-MM-DD-hh:mm:ss:SSS* 时间戳记标识发现的运行，例如 20110517225225948。 *YYYY-MM-DD* 部分表示年月日。 *hh:mm:ss.sss* 表示 24 小时制的时间，精确到千分之一秒。在上例中，日期为 2011/05/17，时间为 22:52:25.948。可以使用此标识在 $COLLATION_HOME/log/sensors 目录中为每个传感器分别创建日志文件。时间戳记用于这些日志文件中。

在发现期间，过程流管理器监视发现的状态和传感器事件的状态。 过程流管理器还管理不同服务之间的交接。 过程流活动存储在发现服务器或域服务器上的 $COLLATION_HOME/log/services/ProcessFlowManager.log 文件中。

以下示例显示了过程流管理器监视的各种活动，以及这些信息在日志文件中的存储方式。

Starting discovery:

```
- 2011-05-17 22:53:01,643 ProcessFlowManager [RMI TCP Connection(42)-127.0.0.1] INFO
processflowmgr.ProcessFlowManagerImpl - [ProcessFlowManagerImpl.I.0] startDiscovery()
started discovery with run id 2,011,051,722,525,948
- 2011-05-17 22:53:01,643 ProcessFlowManager [RMI TCP Connection(42)-127.0.0.1] INFO
processflowmgr.ProcessFlowManagerImpl - [ProcessFlowManagerImpl.I.22] startDiscovery()
 setting the discoveryRun's run id to 2,011,051,722,525,948
- 2011-05-17 22:53:01,973 ProcessFlowManager [RMI TCP Connection(42)-127.0.0.1] INFO
processflowmgr.ProcessFlowManagerImpl -
Discovery run, 2011051722525948 started with profile Level 2 Discovery
```

Discovery done:

```
- 2011-05-17 22:56:11,689 ProcessFlowManager [RMI TCP Connection(45)-127.0.0.1] INFO
processflowmgr.ProcessFlowManagerImpl - [ProcessFlowManagerImpl.I.36]
discoveryDone(2,011,051,722,525,948) called by Discovery Manager
```

Discovery event:

```
- 2011-05-17 22:53:49,901 ProcessFlowManager [RMI TCP Connection(45)-127.0.0.1] INFO
processflowmgr.ProcessFlowManagerImpl - [ProcessFlowManagerImpl.I.32]
discoveryProgress(2,011,051,722,525,948, Discovered - The CustomAppServerSensor
(JavaServer 9.156.47.175:36750) sensor discovered the following: CustomAppServerResult,
 JavaServer,9.156.47.175:36750.) called by Discovery Manager
```

拓扑构建器阶段和日志文件

拓扑构建器构建所发现项目之间的关系和依赖性。 拓扑构建器会运行 $COLLATION_HOME/etc/TopologyBuilderConfigurationDefault.xml 文件中列出的一系列代理程序。拓扑代理程序按指定时间间隔运行。 但是，在发现期间或发现完成时发生的事件也会触发拓扑构建器。 每个代理程序都执行特定的任务，例如联合、确定依赖关系、构建依赖关系图表和除去旧信息。 拓扑构建器日志文件存储在域服务器、同步服务器和主存储服务器上的 $COLLATION_HOME/log/services/TopologyBuilder.log 和 $COLLATION_HOME/log/agents/*.log 文件中。

以下示例显示了构建关系时的各个阶段，以及这些信息在日志文件中的存储方式。

Starting builder execution:

```
- 2011-05-17 22:56:11,717 TopologyBuilder [RMI TCP Connection(158)-127.0.0.1]
INFO cdb.TivoliStdMsgLogger
- CTJOT0400I Topology builder is starting.
```

Topobuilder done:

```
- 2011-05-17 23:16:39,429 TopologyBuilder [TopologyBuilderEngineThread$Dependency@0.5]
 INFO engine.TopologyBuilderEngine - Topology agent completed :
 all normally in seconds 30.367
```

Moving to next TopoAgent:

```
- 2011-05-17 23:16:29,774 TopologyBuilder [TopologyBuilderEngineThread$Dependency@0.5]
INFO cdb.TivoliStdMsgLogger - CTJOT0403I Topology builder agent class
com.ibm.cdb.topomgr.topobuilder.agents.ComputerSystemConsolidationAgent is stopping.
- 2011-05-17 23:16:30,078 TopologyBuilder [TopologyBuilderEngineThread$Dependency@0.5]
INFO cdb.TivoliStdMsgLogger - CTJOT0402I Topology builder agent class
com.ibm.cdb.topomgr.topobuilder.agents.ComputerSystemTypeAgent is starting.
```

如果您发现问题，例如拓扑构建器挂起，请在日志文件中检查上次启动的拓扑代理程序来确定问题。 如果 TopologyBuilder.log 文件中没有任何条目，请在 TopologyManager.log 文件中检查上次启动的代理程序的时间戳记之后的条目。 如果您知道造成问题的代理程序，您也可以检查 $COLLATION_HOME/log/agents/agentName.log 文件来进行确定。

其他服务和日志文件

变更管理器处理事件并更新变更历史记录。 此处理与发现阶段无关，它从其他服务（例如拓扑构建器进程和批量装入程序）接收事件。 在您打开拓扑视图时，视图管理器会构建 GUI 高效呈现拓扑所需的结构。服务日志存储在 $COLLATION_HOME/log/services 目录中。每个服务日志都与服务同名，例如，services/ChangeManager.log 文件。

以下示例显示这些信息在服务日志文件中的存储方式。

ChangeManager:

```
2011-05-19 13:22:42,342 ChangeManager [ChgWork-1] INFO changemgr.
ChangeManagerPersisterImpl -
[ChangeManagerPersister.I.3] Got a create or delete event
```

ViewManager:

```
2011-05-19 16:37:22,428 ViewManager [RMI TCP Connection(174)-127.0.0.1]
INFO viewmgr.ViewMetaLoader - [ViewMetaLoader.I.31] getViewMeta()
found view meta definition for view Business Application Topology
```

最近一次成功的凭证高速缓存

TADDM 可以高速缓存最近一次工作的访问凭证。 在下一次发现（第 2 层发现或者基于脚本的发现）中，可以复用这些凭证。

在目标初始发现期间，TADDM 服务器对访问列表进行循环访问并对发现目标验证每一项。 找到有效凭证后，将其存储在高速缓存中并在连续发现同一发现目标期间复用。

高速缓存可以存储以下两个值：

凭证
在发现期间找到发现目标的有效凭证后，将该值存储在高速缓存中。 在下一次发现期间，从高速缓存中读取凭证并检查其是否仍然有效。 如果仍然有效，那么将其用于发现。如果不再有效并且禁用了回退，那么最后一次尝试失败的信息存储在服务器中并且停止发现。 启用回退后，服务器对访问列表进行循环访问并尝试找到新的有效凭证。 要启用回退，请将 com.ibm.cdb.security.auth.cache.fallback.failed 属性设置为 true。

最后一次尝试失败（以及最后一个错误）的信息
在发现期间找不到发现目标的有效凭证后，将该值存储在高速缓存中。 如果禁用了回退，那么将显示最后一次尝试失败的信息并且停止发现。 如果启用了回

退，服务器对访问列表进行循环访问并尝试找到新的有效凭证。要启用回退，请将 com.ibm.cdb.security.auth.cache.fallback.invalid 属性设置为 true。

缺省情况下，启用这两种情况中的回退。 通过适当地设置访问凭证高速缓存属性，您可以定制回退行为和凭证高速缓存。

注：按照 IP 地址、位置标记、凭证类型以及在连接期间使用的协议来高速缓存凭证。除去访问条目后，还将除去所有关联的高速缓存条目。 凭证高速缓存可以受新实用程序 cachemgr 管理。

限制

- 凭证高速缓存不用于第三级别发现。 仅用于第二级别计算机系统发现和基于脚本的传感器。
- 高速缓存不跟踪作用域访问限制更改。 例如，如果发现目标在作用域访问限制内并且已经发现和高速缓存，然后从作用域限制中移出，那么仍使用高速缓存的值。
- 高速缓存的值优先于已简要描述的访问列表。例如，如果使用主要访问列表来运行发现并且存储了有效凭证，那么即使在概要文件中指定了其他凭证，仍然使用高速缓存的值。

您可以使用 cachemgr 实用程序来除去高速缓存的值。如果您经常针对同一发现目标或作用域使用具有不同访问条目的不同概要文件，那么您可以对其禁用高速缓存。 否则，在发现中可能会使用错误的凭证。

拓扑构建过程概述

TADDM 定期运行拓扑构建过程。 在发现或批量装入操作后一直到拓扑构建过程完成前，TADDM 数据库中可能存在未调和的对象，拓扑关系也可能不完整。

无论使用何种类型的 TADDM 部署，此过程都相同。

拓扑构建包含以下操作：

清除 TADDM 数据库

该过程将删除旧实体，除去缺少源或目标的依赖关系，并除去已取代的其他项。

建立配置项之间的依赖关系

该过程在通信的过程之间建立依赖关系，如应用程序和底层数据库之间，发送和接收 WebSphere MQ 队列之间。 它还在应用程序集群的各部分之间或者只是在两个计算机系统之间建立依赖关系。

创建和扩增配置项

该过程使用来自现有配置项和连接的信息来合成新的配置项。 例如，TADDM 可以根据从先前的发现和批量装入操作中得出的信息创建一个名为『ApplicationServerClusters』 的新配置项。

创建拓扑视图的信息

该过程会生成并存储信息以供数据管理门户网站用于更快地显示拓扑视图。

导出数据

该过程会查询 TADDM 数据库，从而将配置项信息导出到外部系统。 例如，作为拓扑代理程序以实现与 Registry Services 的集成。

日志文件和日志记录

《TADDM 故障诊断指南》及其主题描述了 TADDM 日志文件以及如何设置日志记录以进行故障诊断。

保护环境安全

在安全环境中，TADDM 强制执行认证以帮助保护机密信息。

可以使用数据管理门户网站配置用户帐户。 每个用户都必须具备有效的用户帐户，才能使用数据管理门户网站访问有关网络和基础结构组件的已发现信息。

登录发现管理控制台并选择**建立安全 (SSL) 会话**选项时，所有数据在通过网络发送之前都将进行加密（包括用户名和密码）。

在发现进程中，TADDM 服务器将使用安全 Shell (SSH) 协议与所有支持 SSH 的计算机主机和其他设备进行安全通信。

服务器支持基于密钥的 SSH 认证以及基于登录且基于密码的 SSH 认证。 使用基于登录且基于密码的 SSH 认证时，将使用访问列表中定义的用户名和密码来登录要发现的计算机主机。

另请参阅第 82 页的『安全性属性』。

控制用户对配置项的访问权

TADDM 通过使用访问权集合、角色和许可权来控制用户对配置项的访问权。

对配置项的访问控制是通过以下过程建立的：

1. 配置项聚集为访问权集合。
2. 定义聚集许可权集合的角色。
3. 定义用户或用户组，并将角色指定给每个用户或用户组以授予特定许可权（针对特定的访问权集合）。

在 TAADDM 中的安全性上下文中，用户是获得了配置项访问权的人员，用户组由具有相同角色或许可权的若干用户组成。

您可以在数据管理门户网站中创建用户和用户组。用户和用户组对配置项的访问权由指定给每个用户或用户组的角色以及访问权集合所定义。 可随时更改这些指定。

许可权

许可权授予用户执行操作或访问特定配置项的权力。 许可权聚集到角色中，通过将拥有这些许可权的角色指定给用户可授予用户许可权。

TADDM 提供四种许可权，它们分别属于数据级别许可权或方法级别许可权。

数据级别许可权

"读"和"更新"是数据级别许可权。

读　　用户可以查看有关配置项的信息。

更新　　用户可以更改有关配置项的信息。

方法级别许可权

"发现"和"管理"是方法级别许可权。

发现 用户可以从发现管理控制台的"编辑"菜单之类的位置启动发现、创建和更新发现作用域对象或者创建新对象。

没有发现许可权的用户无法登录到发现管理控制台，也无法查看数据管理门户网站中的发现选项卡。

管理 用户可以创建或更新用户、角色和许可权。 用户还可以通过授权管理器来配置授权策略。

启用数据级安全性

对于 AIX、Linux、Linux on System z 和 Windows 操作系统，可以通过编辑 collation.properties 文件来启用数据级安全性。

要启用数据级安全性（以便有选择地授予读访问权和更新许可权），请完成下列步骤：

1. 在 collation.properties 文件中，找到以下行，将属性值从 false 更改为 true：
   ```
   com.collation.security.enabledatalevelsecurity=false
   ```
2. 保存文件。
3. 停止 TADDM 服务器。
4. 重新启动 TADDM 服务器。

注：在流式服务器部署中，必须更新每台存储服务器上的 collation.properties 文件，并重新启动每台存储服务器。

您可以通过创建访问权集合来设置更详细的许可权。如果已启用数据级安全性，就可以使用访问权集合来保护大多数 TADDM 资源的安全。 如果启用了数据级安全，那么用户只能修改其具有更新许可权的访问权集合中所含的 CI。

创建访问权集合时不会显示辅助资源（如物理地理资源，包括 SiteInfo 属性）。

角色

角色是一组可指定给用户的许可权。 指定角色即赋予了特定的访问功能。

将角色指定给用户时，必须为该角色指定一个或多个访问权集合。 这将角色的作用域限为仅适合该用户的那些访问权集合。

例如，Sarah 负责公司的 NT 服务器和工作站，可向她分配由这些系统组成的访问权集合的主管人员角色。 Jim 负责 Linux 系统，可向他指定由这些 Linux 系统组成的访问权集合的主管人员角色。虽然对 Sarah 和 Jim 指定了相同的角色（因为他们执行的操作相同），但他们可访问的资源不同。

注： 如果您使用的是同步服务器，那么必须为每个 TADDM 域创建该角色，然后将域服务器与同步服务器同步。

预定义角色

TADDM 提供以下预定义角色：

操作员
该角色具有读许可权。

主管 该角色具有读、更新和发现许可权。

管理员
该角色具有读、更新、发现和管理许可权。

您可以创建的其他角色

可以创建更多角色来指定其他许可权组合。以下组合可能特别有用：

读取 + 更新
在指定的访问权集合中读取并更新对象的许可权。

读取 + 更新 + 管理
在指定的访问权集合中读取和更新对象，以及创建用户、角色和许可权的许可权。

访问权集合

TADDM 并不逐个管理对配置项的访问权。 而是将配置项聚集成称为访问权集合的组。访问权集合是一组出于安全目的而合在一起进行管理的配置项。

通过创建角色并将角色指定给用户来对每个访问权集合的安全性进行管理。 角色仅应用于在将该角色指定给用户时所指定的访问权集合。 因此，访问权集合用于显示角色的作用域。

在您安装 TADDM 时，会创建名为 DefaultAccessCollection 的访问权集合，它包含所有配置项。 缺省情况下，除非启用了数据级安全性，否则所有用户都对此访问权集合拥有读取和更新许可权。

注：用户无权读取和更新访问权集合，而只能读取和更新各个配置项。 但是，用户对分配给他们的访问权集合的成员访问权集合具有读取和更新许可权。

复位安全策略

如果需要将安全策略（许可权、角色和访问权集合）重置为其缺省状态，可以通过替换两个文件来实现此目的。 但重置安全策略需要删除并重新创建所有用户。

关于此任务

安全策略存储在 $COLLATION_HOME/var/policy 目录中的以下两个文件中，这些文件用于初始化安全策略：

- AuthorizationPolicy.xml
- AuthorizationRoles.xml

初始化安全策略后，这两个文件将重命名并存储在同一目录中。 例如，已重命名以下文件：

- AuthorizationPolicy.backup.xml
- AuthorizationRoles.backup.xml

包含提供的安全策略的缺省文件版本也位于同一目录中。 以下文件是缺省版本：

- DefaultPolicy.xml

- DefaultRoles.xml

过程

要恢复缺省安全策略，请完成下列步骤：

1. 果保存当前策略文件，请重命名这些文件，或者将其移动到其他目录。
2. 删除已创建的所有用户。
3. 删除 $COLLATION_HOME/var/ibmsecauthz 目录。
4. 创建 DefaultPolicy.xml 文件的副本，将其命名为 AuthorizationPolicy.xml。
5. 创建 DefaultRoles.xml 文件的副本，将其命名为 AuthorizationRoles.xml。
6. 重新启动服务器。
7. 根据需要创建用户。

锁定

您可以使用锁定功能在超出配置的失败登录尝试允许次数后锁定单一用户或全部用户，以使他们无法登录 TADDM。 使用锁定功能可以进行更好的认证控制，并有助于阻止密码暴力破解。

如果单一用户超出配置的失败登录尝试次数，那么将触发局部锁定。 其结果是，该用户在配置的时间长度内无法登录 TADDM。

如果触发了全局锁定，那么在配置的时间长度内，任何用户都无法登录 TADDM。 下列两种情况都将触发全局锁定：

- 各个不同用户的活动锁定总数超出配置的全局最大允许锁定数目。
- 非重复用户名的失败登录尝试次数超出配置的限制。

触发锁定时，现有会话不受影响。

通过在 collation.properties 文件中配置属性，可以指定允许的失败登录尝试次数以及锁定保持活动状态的时间长度。有关这些属性的更多信息，请参阅第 79 页的『锁定属性』。

全局锁定时间经过之后，将自动解除所有进行中的局部锁定。

在同步服务器部署中，所有 TADDM 域的安全性由同步服务器控制。 启用域服务器与同步服务器之间的同步时，将清除域服务器上所有在其连接到同步服务器之前已处于活动状态的锁定。

计入总数的失败登录尝试可以是任何类型，例如，使用 CLI API、Java API、工具（脚本）、SOAP、REST、发现管理控制台或数据管理门户网站进行的登录。 锁定功能适用于使用 TADDM API 实现的集成，但不适用于使用单点登录功能的登录或基于数据库的集成（例如 Tivoli Common Reporting）。

TADDM 服务器管理员可以使用 $COLLATION_HOME/bin/lockmgr.sh 脚本来清除局部或全局锁定。可以从下列服务器中运行此脚本：

- 域服务器部署中的域服务器
- 同步服务器部署中的同步服务器
- 流式服务器部署中的主存储服务器

可以在使用下列选项的情况下运行 lockmgr.sh 脚本：

lockmgr.sh -s
显示锁定状态。

lockmgr.sh -g
清除活动的全局锁定。

lockmgr.sh -u *username*
清除特定用户的活动局部锁定。

lockmgr.sh -h
显示 lockmgr.sh 脚本的帮助信息。

加密

加密是将数据变换为无法理解的形式的过程，以实现无法获取原始数据或只能使用解密过程来获取原始数据的效果。

Fix Pack 5 TADDM 使用"*com.collation.security.algo.aes.keylength*"属性来确定"符合 *FIPS* 的 *IBMJCEFIPS*"安全提供程序的算法（AES 128 或 AES 256），以对以下各项进行加密：

- 密码，包括 collation.properties 中的条目和 userdata.xml 文件
- 存储在数据库中的访问列表条目

例如：

此属性定义 AES 的密钥长度 -com.collation.security.algo.aes.keylength=128。

初次安装 TADDM 时，将生成一个加密密钥并使用此新的加密密钥对密码进行加密。加密密钥的缺省位置为 etc/TADDMSec.properties 文件。

更改 TADDM 加密密钥的位置

要更改加密密钥的位置，请在 collation.properties 文件中更改 com.collation.security.key 属性的值。 可以将该属性设置为相对于 $COLLATION_HOME 目录的其他位置。

为了避免数据丢失，请将加密密钥的备份副本存储在其他位置。 原始副本出现问题时也可恢复密钥。

在域服务器部署中更改 TADDM 加密密钥

注：在流式服务器部署和同步服务器部署中，TADDM 不支持在完成安装后更改加密密钥。

要在域服务器部署中更改 TADDM 加密密钥，请使用 bin/changekey.sh 脚本（或者等效的批处理脚本文件）。 此脚本将迁移 collation.properties 和 userdata.xml 文件中的加密条目，并迁移存储在数据库中的访问列表条目。要使用 bin/changekey.sh 脚本，请确保以安装期间定义的非 root 用户身份登录。

成功使用此脚本之后，必须重新启动 TADDM。

脚本的运行格式

```
./changekey.sh $COLLATION_HOME admin_user admin_password
```

示例

```
./changekey.sh /opt/IBM/taddm/dist administrator taddm
```

符合 FIPS

通过将 FIPSMode 属性 **com.collation.security.FIPSMode** 设置为 true，可以将 TADDM 的运行方式配置为使用符合 FIPS 标准的算法进行加密。

请在下列文件中将 **com.collation.security.FIPSMode** 属性设置为 true：

- $COLLATION_HOME/dist/etc/collation.properties
- $COLLATION_HOME/dist/sdk/etc/collation.properties
- 每个连接到符合 FIPS 标准的 TADDM 的 TADDM SDK 安装中的 sdk/etc/collation.properties。

com.collation.security.FIPSMode 属性的缺省值为 false。

在 FIPS 方式下，TADDM 使用以下 FIPS 140-2 核准的密码提供程序：

- IBMJCEFIPS（证书 376）
- IBMJSSEFIPS（证书 409）

有关证书 376 和 409 的更多信息，请参阅美国国家标准技术学会 (NIST) Web 站点：http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2004.htm。

FIPS 方式可用于所有类型的 TADDM 发现，但存在下列例外：

- 第 2 层 SNMP 发现
- 第 2 层 i5/OS 发现
- 第 2 层 ZEnterprise 发现
- 第 2 层 VMware ESXi 发现
- 第 3 层 VMware Virtual Center 发现
- 第 3 层 JBoss 发现
- 第 3 层 Oracle Application Server 发现
- 第 3 层 WebLogic 发现
- 第 3 层 SAP CCMS 和 SLD 发现
- 第 3 层 EMC 发现
- Fix Pack 1 第 3 层 Sybase 发现
- 第 2 层和第 3 层发现，在这些发现中，只有在 Windows TADDM 服务器、Windows 网关和 Windows 发现目标未以符合 FIPS 兼容的方式运行时，才使用 Windows Management Instrumentation (WMI) 或 PowerShell 会话（在 TADDM 7.3.0.2 或更高版本中不支持 PowerShell 会话）来发现 Windows 平台。 要将 Windows 服务器配置为以符合 FIPS 标准的方式运行，请参阅 Windows 文档，例如 http://support.microsoft.com/kb/811833。

在 FIPS 方式下，使用 SSH 的 TADDM 传感器无法连接到仅支持 SSHv1 协议或者仅支持使用过弱密码的 SSHv2 协议的服务器。 TADDM 无法验证目标服务器上的 SSH 实现是否符合 FIPS 标准。 您必须检查环境中使用的 SSH 实现是否符合 FIPS 标准。

在 FIPS 方式下，以安全方式使用 TADDM SDK 和发现管理控制台时，仅支持 IBM Java。

相关概念:

『SP800-131 合规性』
可以将 TADDM 配置为支持美国国家标准技术学会 (NIST) SP800-131a 安全标准。

SP800-131 合规性

可以将 TADDM 配置为支持美国国家标准技术学会 (NIST) SP800-131a 安全标准。

与其他标准（例如 FIPS 140-2 标准）相比，SP800-131a 安全标准需要更长的密钥长度和更健壮的密码术。另外，此标准还要求使用传输层安全性 (TLS) V1.2。 有关更多信息，请参阅http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf。

要启用 SP800-131a 方式，请在下列文件中将 `com.ibm.jsse2.sp800-131` 属性设置为 `strict`：

- `$COLLATION_HOME/dist/etc/collation.properties`
- `$COLLATION_HOME/dist/sdk/etc/collation.properties`
- 每个连接到符合 SP800-131 标准的 TADDM 的 TADDM SDK 安装中的 `sdk/etc/collation.properties`。

缺省情况下，未设置 `com.ibm.jsse2.sp800-131` 属性。

支持 SP800-131a 合规性方式的 TADDM 发现类型与支持 FIPS 方式的 TADDM 发现类型相同。

在 SP800-131 方式下，TADDM 使用最安全的 SSL 协议 (TLS V1.2) 进行加密通信。请确保满足下列要求。

- 通过 Web SSL 端口 (HTTPS) 使用数据管理门户网站时，必须先将 Web 浏览器配置为支持 TLS V1.2 协议。
- 在安全方式下使用 TADDM SDK 和发现管理控制台时，必须在 Java 运行时环境中启用 TLS V1.2 协议。 另外，仅支持 IBM Java。
- 当 SSL 证书不符合 SP800-131a 标准时，需要重新创建该证书。 要了解所需执行的步骤，请参阅第 29 页的『安装定制的 SSL 证书以用于 TADDM』。

相关概念:

第 19 页的『符合 FIPS』
通过将 FIPSMode 属性 **`com.collation.security.FIPSMode`** 设置为 `true`，可以将 TADDM 的运行方式配置为使用符合 FIPS 标准的算法进行加密。

同步服务器部署的安全性

如果您使用同步服务器部署，必须在针对您的环境配置同步服务器时进行安全性更改。

如果使用的是基于 TADDM 文件的注册表并且 TADDM 域已添加到同步服务器，那么必须在同步服务器中重新创建域中已存在的所有用户，包括已授权访问权集合的已指定角色和访问权。 如果使用的是轻量级目录访问协议 (LDAP) 或 WebSphere 联合存储库用户注册表，那么必须在同步服务器中为所有访问 TADDM 的用户添加权限。

向同步服务器添加域时，将授权同步服务器执行新域的认证和授权。

域的登录在同步服务器中处理。此外，安全管理器方法调用由同步服务器处理。

以下列表汇总了配置同步服务器需要了解的其他安全性信息：

- 要使 TADDM 正常运行，数据管理门户网站必须正在同步服务器上运行。 TADDM 域将安全性操作授权给数据管理门户网站执行，并且此授权每 2.5 分钟更新一次。 如果 5 分钟后此授权仍未更新，TADDM 域将不再授权安全性操作，并且将同步服务器视为不存在而继续。 在这种情况下，必须重新启动 TADDM UI 以重新建立与同步服务器的会话。
- 在以下每种情境中，都必须重新启动 TADDM UI 以重新建立与正确的同步服务器的会话：
  - 正在运行 UI 的域将添加到正在运行同步服务器的数据管理门户网站中。
  - 在域连接到数据管理门户网站时 UI 将在该域中打开，但稍后此同步服务器变为不可用，例如在该同步服务器重新启动期间或出现网络问题时。
- TADDM 服务器中存储的角色、许可权和访问权集合将从域同步到同步服务器。 用户到角色的映射将不同步。
- 在将为域所创建的角色从该域同步到同步服务器后，该同步服务器可以使用这些对象。
- 不会将用户同步到同步服务器。
- 中央用户注册表（例如 LDAP 或 WebSphere 联合存储库注册表）是同步服务器认证的首选方法。 使用中央用户注册表，用户密码将存储到一个位置。
- 访问权集合不能跨域。
- 同步操作将从域执行到同步服务器。同步服务器中创建的对象不会传播到域。
- 在域中创建并填充访问权集合，并与同步服务器同步。
- 在域中创建角色，并与同步服务器同步。
- 在同步服务器上授权用户以提供对多个域中的访问权集合的访问权。

流式服务器部署的安全性

如果您使用流式服务器部署，将授权主存储服务器执行认证和授权。

如果使用 TADDM 基于文件的注册表，必须在主存储服务器上创建和授权 TADDM 用户。 如果使用轻量级目录访问协议 (LDAP) 或 WebSphere 联合存储库用户注册表，必须在主存储服务器上授权 TADDM 用户。 TADDM 认证的首选注册表类型是带中央用户注册表的，如 LDAP 注册表或 WebSphere 联合存储库注册表。

发现服务器和辅助存储服务器的登录在主存储服务器中处理。 因此，将根据已配置主存储服务器的用户注册表执行用户认证。 此外，安全管理器功能由主存储服务器处理。

要使 TADDM 正常运行，主存储服务器必须处于运行状态。

如果主存储服务器停止或重新启动，那么必须重新启动 TADDM 用户界面，从而与主存储服务器重新建立会话。

配置 LDAP

您可配置外部 LDAP 服务器进行用户认证。

开始之前

如果要经过 LDAP 用户注册表验证，请配置 LDAP V2 或 V3 注册表。

关于此任务

使用 LDAP 和/或 VMM 时，LDAP 用户和/或组将始终存储在 LDAP/VMM 中，并且不需要在 TADDM 中创建 LDAP 用户和/或组。 TADDM 仅用于向 LDAP 用户和组指定角色。 只有用户/组到角色的映射（称为许可权）才需要在 TADDM 中进行创建和存储。 管理员用户标识是一种特殊的内部 TADDM 用户，无论配置了何种用户注册表，都始终使用基于文件的安全性对其进行处理。 此用户始终可用于最先向 LDAP 用户和组指定角色。

过程

要使用 LDAP 或 VMM 进行用户认证，请完成以下步骤：

1. 通过配置 collation.properties 文件中的相应属性，可将 TADDM 配置为使用 LDAP 注册表。
2. 使用 TADDM 管理员用户标识登录到数据管理门户网站。
3. 完成下列其中一个步骤：
   - 在"用户"窗格中，使用**搜索用户**字段可搜索相应用户的 LDAP 注册表。
   - 在"用户组"窗格中，使用**搜索组**字段可搜索相应用户组的 LDAP 注册表。

   注：搜索结果将列示由 LDAP 注册表搜索返回的用户名称或组名称。 这不是为了创建用户，或将用户从 LDAP 复制到 TADDM。 此列表的目的是用于显示需要为用户创建哪些 TADDM 许可权。
4. 列出用户（或组）之后，向其指定所需的 TADDM 角色。 TADDM 中只存储这些许可权，而不存储 LDAP 用户（或组）。

下一步做什么

要为 LDAP 配置 SSL，请完成以下步骤：

1. 在 `collation.properties` 文件中，找到以下属性，并将属性值从 `false` 更改为 `true`：

   ```
   com.collation.security.auth.ldapUseSSL
   ```
2. 对以下信任库和密钥库属性进行相应配置：

   ```
   com.collation.security.auth.ldapClientKeyStore
   com.collation.security.auth.ldapClientKeyStorePassphrase
   com.collation.security.auth.ldapClientTrustStore
   com.collation.security.auth.ldapClientTrustStorePassphrase
   ```
3. 需要时，可通过配置以下属性来更改 LDAP 服务器用于侦听 SSL 连接的端口：

   ```
   com.collation.security.auth.ldapPortNumber
   ```

配置 WebSphere 联合存储库

如果您为使用 WebSphere 联合存储库的中央用户注册表配置了 Tivoli WebSphere 应用程序，那么您可以在联合存储库注册表中配置 WebSphere 联合存储库。

将 TADDM 服务器配置为使用 WebSphere 联合存储库

WebSphere 联合存储库是 WebSphere 中灵活的元存储库，它支持包括 Microsoft Active Directory 在内的多种用户注册表。

开始之前

如果在环境中使用其他 Tivoli 产品，那么必须将 TADDM 配置为使用 WebSphere 联合存储库，并且在 TADDM 和任何以下产品之间需要单点登录：

- IBM Tivoli Change and Configuration Management Database (CCMDB) 或 IBM SmartCloud Control Desk (SCCD)
- IBM Tivoli Business Service Manager

TADDM 需要标准 WebSphere 分发中不存在的其他服务，因此在为联合存储库配置 TADDM 时，必须使用以下某个 WebSphere 安装：

- 随 CCMDB 或 SCCD 一起安装的 WebSphere Application Server Network Deployment
- WebSphere Application Server 与 IBM Tivoli Business Service Manager 一起安装

要查看产品的受支持版本，请转至第 171 页的『受支持的版本』部分。

您必须先在 WebSphere Application Server Network Deployment 服务器上配置 WebSphere 联合存储库认证服务，然后才能启动此过程。 有关更多信息，请参阅 IBM Tivoli Change and Configuration Management Database (CCMDB) 文档或 IBM SmartCloud Control Desk (SCCD) 文档。

关于此任务

此配置将在使用 WebSphere 轻量级第三方认证 (LTPA) 令牌的 Tivoli 应用程序之间启用单点登录。 例如，如果将 TADDM 配置为使用 CCMDB 或 SCCD 所使用的那些 WebSphere 联合存储库，那么在 IBM Tivoli CCMDB 或 IBM SCCD 与 TADDM 之间，将支持单点登录以实现"在上下文中启动"。

要将 TADDM 自动配置为使用 WebSphere 联合存储库，请安装 TADDM 并在安装期间选择 **WebSphere 联合存储库**作为用户注册表。

在所有 TADDM 服务器类型的所有部署中都支持此配置。

过程

要手动执行配置，请完成下列步骤：

1. 停止 TADDM 服务器。
2. 指定此 TADDM 服务器使用的用户管理模块。以下值有效：

   file 该值用于表示基于文件的用户注册表。 （这是缺省值。）

   ldap 该值用于表示 LDAP 用户注册表。

   vmm 该值用于表示使用 WebSphere Application Server 联合存储库的用户注册表。

   例如，在 `$COLLATION_HOME/etc/collation.properties` 文件中：

```
com.collation.security.usermanagementmodule=vmm
```

3. 在 collation.properties 文件中指定 WebSphere 主机名和端口。 例如：

   ```
   com.collation.security.auth.websphereHost=localhost
   com.collation.security.auth.webspherePort=2809
   ```

   在 collations.properties 文件中指定 WebSphere 端口时，使用以下属性：com.collation.security.auth.webspherePort。 WebSphere 端口应该是 WebSphere 服务器的引导程序端口。对于 WebSphere Application Server 和嵌入式版本的 WebSphere Application Server，缺省端口为 2809。 对于 IBM Tivoli CCMDB 或 IBM SCCD 使用的 WebSphere Application Server Network Deployment，缺省端口为 9809。

4. 在 collation.properties 文件中指定 WebSphere 管理员用户名和密码。 例如：

   ```
   com.collation.security.auth.VMMAdminUsername=administrator
   com.collation.security.auth.VMMAdminPassword=password
   ```

5. 对认证服务配置文件进行以下更改：
   - 对于 Linux、AIX 和 Linux on System z 操作系统，此文件位于以下路径：$COLLATION_HOME/etc/ibmessclientauthncfg.properties。
   - 对于 Windows 操作系统，此文件位于以下路径：%COLLATION_HOME%\etc\ibmessclientauthncfg.properties。

   在 authnServiceURL 属性中，替换安装了 WebSphere 实例的系统的标准域名和 WebSphere 实例的 HTTP 端口。

   ```
   # This is the URL for the Authentication Service
   authnServiceURL=http://localhost:9080/TokenService/services/Trust
   ```

6. 将 WebSphere orb.properties 和 iwsorbutil.jar 文件复制到 TADDM 安装使用的 JRE 中。 例如，在 TADDM Linux 安装中，执行以下操作：
   a. 将 dist/lib/websphere/6.1/orb.properties 复制到 dist/external/ jdk-Linux-i686/jre/lib/。
   b. 将 dist/lib/websphere/6.1/iwsorbutil.jar 复制到 dist/external/ jdk-Linux-i686/jre/lib/ext/。
7. 在 sas.client.props 文件中指定 WebSphere 主机名和端口：
   - 对于 Linux、AIX 和 Linux on System z 操作系统，此文件位于以下路径：$COLLATION_HOME/etc/sas.client.props。
   - 对于 Windows 操作系统，此文件位于以下路径：%COLLATION_HOME%\etc\sas.client.props，例如：

     ```
     com.ibm.CORBA.securityServerHost=host1.austin.ibm.com
     com.ibm.CORBA.securityServerPort=2809
     ```

   注：对于 WebSphere Application Server 和嵌入式版本的 WebSphere Application Server，缺省端口为 2809。 对于 IBM Tivoli CCMDB 或 IBM SCCD 使用的 WebSphere Application Server Network Deployment，缺省端口为 9809。
8. 在 sas.client.props 文件中指定 WebSphere 管理员用户名和密码。 例如：

   ```
   # RMI/IIOP user identity
   com.ibm.CORBA.loginUserid=administrator
   com.ibm.CORBA.loginPassword=password
   ```

9. 可选： 要在 sas.client.props 文件中对登录密码进行加密，请完成下列步骤：

a. 将 sas.client.props 文件复制回 TADDM 服务器的 $COLLATION_HOME/etc 目录中。

b. 根据已安装 WebSphere 的操作系统来对密码进行加密，如下所示。

- 对于 Linux、AIX 和 Linux on System z 操作系统：

  使用 PropFilePasswordEncoder.sh 命令。

- 对于 Windows 操作系统：

  使用 PropFilePasswordEncoder.bat，例如：

  ```
  C:\WebSphere\profiles\AppSrv01\bin\PropFilePasswordEncoder C:\temp\sas
    .client.props com.ibm.CORBA.loginPassword
  ```

c. 将 sas.client.props 文件复制回 TADDM 服务器的 etc 目录中。

10. 启动 TADDM 服务器。

下一步做什么

安装完成后，您可以使用 TADDM 基于文件的本地存储库中定义的缺省管理员用户来配置其他 TADDM 用户，包括 TADDM 管理员。这些其他 TADDM 用户都已使用 WebSphere 联合存储库进行认证。

Tivoli CCMDB 或 IBM SCCD 有一些安全配置，这些配置允许在 Maximo® 用户和组应用程序中创建和维护组和组成员资格。

如果对 Tivoli CCMDB 或 IBM SCCD 进行了这方面的配置，那么 TADDM 将使用自己的存储库，而不使用 Tivoli CCMDB 或 IBM SCCD 的存储库。 在这种情况下，在 Tivoli CCMDB 或 IBM SCCD/Maximo 以及 TADDM 中都必须创建用户。

可通过 WebSphere 联合存储库将 TADDM 配置为使用外部用户注册表中的用户和组定义。 但是，TADDM 不能使用存储在 Tivoli CCMDB 中的用户和组定义，因为它们不受 WebSphere 联合存储库的支持。

更新认证服务 LTPA 密钥

如果要对 WebSphere 联合存储库使用单点登录，必须使认证服务轻量级第三方认证 (LTPA) 密钥与 WebSphere 联合存储库所使用的相应密钥保持同步。

过程

如果更改了 WebSphere 联合存储库所使用的 LTPA 密钥，请使用以下过程再同步认证服务所使用的密钥：

1. 导出新的 WebSphere LTPA 密钥：
   a. 在 WebSphere 管理控制台中，浏览到**安全管理、应用程序和基础结构** > **认证机制和到期**。
   b. 对于**跨单元单点登录**，请为要包含导出的 LTPA 密钥的文件指定文件名和密码。
2. 在命令提示符下，转至相应 WebSphere 概要文件的 bin 目录。
3. 运行以下 WebSphere **wsadmin** 命令：

   ```
   wsadmin> $AdminTask importESSLTPAKeys {-pathname pathname -password password}
   ```

   其中，*pathname* 和 *password* 是在导出 LTPA 密钥时为文件名和密码指定的值。
4. 重新启动 WebSphere 服务器。

保护认证通道安全

如果将 TADDM 配置为使用 WebSphere 联合存储库，那么可以保护认证客户机和认证服务之间的通信安全。

关于此任务

TADDM 使用支持单点登录的认证服务。 在安装 IBM Tivoli Change and Configuration Management Database (IBM SmartCloud Control Desk (SCCD)) 或 IBM Tivoli Business Service Manager 时安装认证服务。

要查看产品的受支持版本，请转至第 171 页的『受支持的版本』部分。

您可以通过以下两种机制确保认证客户机和认证服务之间的通信安全：

- SSL
- 客户机认证

为 *SSL* 配置认证通道：

您可以使用 WebSphere 签署者证书来配置认证客户机和认证服务器之间的 SSL，以确保通信安全。

过程

要配置认证客户机和认证服务器之间的 SSL，请完成下列步骤：

1. 执行下列其中一项操作：
   a. 如果您使用的是由 Tivoli Integrated Portal 安装的 WebSphere 实例，请浏览到 **SSL 证书和密钥管理** > **管理端点安全性配置** > **Node1** > **密钥库和证书** > **NodeDefaultTrustStore** > **签署者证书**。
   b. 如果您使用的是由 Tivoli Change and Configuration Management Database (CCMDB) 或 IBM SmartCloud Control Desk 安装的 WebSphere 实例，请浏览到 **SSL 证书和密钥管理** > **管理端点安全性配置** > **ctgNode01** > **密钥库和证书** > **NodeDefaultTrustStore** > **签署者证书**。
2. 将 WebSphere 签署者证书导出到文件（例如，将 dummyclientsigner 导出到 `signer1.cert`，并将 dummyserversigner 导出到 `signer2.cert`）。 如果不确定要导出哪些证书，必须导出所有的签署者证书。
3. 将 `.cert` 文件复制到 TADDM 服务器。 如下创建信任库，并导入 WebSphere 签署者证书：

```
$COLLATION_HOME/external/jdk-Linux-i686/jre/bin/keytool \
-genkey -alias truststore -keystore truststore.jks
$COLLATION_HOME/external/jdk-Linux-i686/jre/bin/keytool \
-import -trustcacerts -alias default -file signer1.cert -keystore truststore.jks
$COLLATION_HOME/external/jdk-Linux-i686/jre/bin/keytool \
-import -trustcacerts -alias dummyserversigner -file signer2.cert -keystore truststore.jks
```

4. 在 `$COLLATION_HOME/etc/collation.properties` 条目中包含信任密钥库密码和位置：

```
com.collation.security.auth.ESSClientTrustStore=/opt/IBM/taddm/dist/etc/truststore.jks
com.collation.security.auth.ESSClientTrustPwd=password
```

5. 更新 `ibmessclientauthncfg.properties` 文件中的 Tivoli Authentication Service URL，以使用 https 和端口 9443。 请确保 WebSphere 主机名正确，使用该主机名替代 localhost，并确保非 https 条目已注释掉。

```
# This is the URL for the ESS Authentication Service
#authnServiceURL=http://localhost:9080/TokenService/services/Trust
authnServiceURL=https://localhost:9443/TokenService/services/Trust
```

配置客户机认证：

要配置认证客户机和认证服务器之间的客户机认证，建议您启用 WebSphere 应用程序安全性。

开始之前

在启用 WebSphere 应用程序安全性后，您可以将名为 TrustClientRole 的角色添加到 TADDM 安装期间指定的 WebSphere 管理员用户。 此方法通过限制只有具有 TrustClientRole 的用户才能对认证服务进行认证，从而为认证服务提供了更多安全性。

过程

要将 TrustClientRole 添加到 TADDM 安装期间指定的 WebSphere 管理员，请完成下列步骤：

1. 登录到 WebSphere 管理控制台。
2. 在**安全性**选项卡下，单击**企业应用程序**。 此时将显示"企业应用程序"窗格。
3. 在"企业应用程序"表中，单击"名称"列中的"认证服务"应用程序 (authnsvc_ctges)。 此时将显示"企业应用程序 > authnsvc_ctges"窗格。
4. 在"企业应用程序 > authnsvc_ctges"窗格的"详细属性"列表中，单击**安全角色到用户/组的映射**。 此时将显示"企业应用程序 > authnsvc_ctges > 安全角色到用户/组的映射"窗格。
5. 在"企业应用程序 > authnsvc_ctges > 安全角色到用户/组的映射"窗格上的表中，完成下列步骤：
   - 在表中，选中 TrustClientRole 旁边的复选框。
   - 取消选中**所有人**复选框。
   - 单击**查找用户**或**查找组**。 此时将显示"企业应用程序 > authnsvc_ctges > 安全角色到用户/组的映射 > 查找用户或组"窗格。
   - 在"企业应用程序 > authnsvc_ctges > 安全角色到用户/组的映射 > 查找用户或组"窗格中，完成下列步骤：
     - 使用"限制和搜索"字符串输入框搜索用户或组。 当找到组或用户时，将显示在"可用"列表中。
     - 从"可用"列表中，选择所需的用户或组。
     - 单击**移动**将该用户或组添加到**选定**列表中。
   - 单击**确定**。 此时将显示"企业应用程序 > authnsvc_ctges > 安全角色到用户/组的映射"窗格。
   - 在"企业应用程序 > authnsvc_ctges > 安全角色到用户/组的映射"窗格中，取消选中**每个人**复选框。
   - 单击**确定**。 此时将显示"企业应用程序 > authnsvc_ctges"窗格。
   - 单击**保存**保存配置。 此时将显示"企业应用程序"窗格。
   - 单击**确定**。 此时将显示"企业应用程序 > authnsvc_ctges"窗格。

为 Microsoft Active Directory 进行配置

可使用 Microsoft Active Directory 作为认证方法，用于 TADDM 通过 LDAP 或 WebSphere 联合存储库作为中介进行认证。 如果需要单点登录到 TADDM，应该使用 WebSphere 联合存储库。

关于此任务

您可以使用 Active Directory 注册表中定义的用户，而无需定义新用户，方法是将 TADDM 配置为使用 Active Directory。可以将 TADDM 配置为使用 Active Directory 作为 LDAP 注册表，或将 TADDM 配置为使用 WebSphere 联合存储库，然后配置 Active Directory 的 WebSphere 联合存储库。

在 TADDM 安装期间针对 Active Directory 进行配置时，可将 TADDM 配置为使用 Active Directory 中的任何用户作为 TADDM 管理员。 该管理员能够配置 TADDM 的访问权，以及授予其他用户对 TADDM 对象和服务的访问权。

在所有 TADDM 服务器类型的所有部署中都支持此配置。

过程

要配置 Microsoft Active Directory，请完成下列步骤：

执行下列其中一项操作：

- 要使用 LDAP 配置 Microsoft Active Directory：
  1. 为 LDAP 配置 TADDM。 有关为 LDAP 配置 TADDM 的更多信息，请参阅第 21 页的『配置 LDAP』。
  2. 确保在使用 Active Directory 时将 `collation.properties` 文件中的 **`com.collation.security.auth.ldapFollowReferrals`** 设置为 *true*。
- 要使用 WebSphere 联合存储库配置 Microsoft Active Directory：
  1. 为 WebSphere 联合存储库配置 TADDM。 有关为 WebSphere 联合存储库配置 TADDM 的更多信息，请参阅第 23 页的『将 TADDM 服务器配置为使用 WebSphere 联合存储库』。
  2. 为 Microsoft Active Directory 配置 WebSphere 联合存储库。有关在联合存储库配置中配置受支持实体类型的更多信息，请参阅位于 http://www-01.ibm.com/support/knowledgecenter/SSAW57_6.1.0/com.ibm.websphere.nd.doc/info/ae/ae/twim_entitytypes.html 的 *WebSphere Application Server Information Center* 中名为 **Configuring supported entity types in a federated repository configuration** 的部分。

保护 TADDM Web Service 的安全

您可以将 TADDM 配置为禁用 HTTP 端口，具体方法是将 `collation.properties` 中的 `com.ibm.cdb.secure.tomcat` 属性 (TADDM 7.3.0) 或 `com.ibm.cdb.secure.liberty` 属性（TADDM 7.3.0.1 及更高版本）设置为 `true`。 另外，还可以通过使用 `com.ibm.cdb.http.ssl.protocol` 标志来设置更安全的 SSL 协议。

`com.ibm.cdb.secure.tomcat` 和 `com.ibm.cdb.secure.liberty` 属性的缺省值为 `false`。当 HTTP 端口处于禁用状态时，只能通过 HTTPS 端口访问 TADDM，例如 `https://example.com:9431`。

Limitation： 在流式服务器部署中安装 TADDM 后，并且目录服务器和辅助存储服务器均已启动并且正在运行，那么可以将 com.ibm.cdb.secure.tomcat 或 com.ibm.cdb.secure.liberty 属性设置为 true。 在此类情况下，禁用 HTTP 端口，可以在安全方式下使用 TADDM。 但是，如果要将新的发现服务器或辅助存储服务器添加到部署中，必须临时启用 HTTP 端口，因为 TADDM 安装程序不支持 HTTPS 协议。 要临时禁用安全方式，请完成以下步骤：

1. 将 com.ibm.cdb.secure.tomcat 或 com.ibm.cdb.secure.liberty 属性的值更改为 false。
2. 重新启动 TADDM 服务器。
3. 安装新的发现服务器或辅助存储服务器。
4. 将 com.ibm.cdb.secure.tomcat 或 com.ibm.cdb.secure.liberty 属性的值更改为 true。
5. 重新启动 TADDM 服务器。

com.ibm.cdb.http.ssl.protocol 属性的缺省值为 TLS。 安全值为 TLS、TLSv1.1 和 TLSv1.2。 如果您要使用最安全的协议 TLSv1.1 和 TLSv1.2，那么必须先将 Web 浏览器配置为支持这些协议。

安装定制的 SSL 证书以用于 TADDM

您可以安装您自己的定制 SSL 证书并将其与 TADDM 配合使用。

过程

1. 创建以下密钥库文件的备份副本：
   - $COLLATION_HOME/etc/serverkeys
   - $COLLATION_HOME/etc/jssecacerts.cert
2. 转至 $COLLATION_HOME/etc 目录，打开命令行，通过以下方式输入 keytool 和 TADDM sslpassphrase 参数以及值：
   - Linux 操作系统：

     ```
     keytool=../external/jdk-Linux-x86_64/bin/keytool
     pass=XXXXXXXX30374
     ```

   - Windows 操作系统：

     ```
     set keytool=..\external\jdk-Windows-i386-64\bin\keytool.exe
     set pass=XXXXXXXX30374
     ```

   pass 参数的值是在 collation.properties 文件中指定的 com.collation.sslpassphrase 属性的值。
3. 运行以下命令以从 TADDM 中除去自签名证书和密钥：
   - Linux 操作系统：

     ```
     $keytool -delete -alias collation -noprompt -keystore jssecacerts.cert
     -storepass $pass
     $keytool -delete -alias collation -noprompt -keystore serverkeys -storepass
     $pass
     ```

   - Windows 操作系统：

     ```
     %keytool% -delete -alias collation -noprompt -keystore jssecacerts.cert
     -storepass %pass%
     %keytool% -delete -alias collation -noprompt -keystore serverkeys -storepass
     %pass%
     ```

4. 使用必需的 CN、有效性、算法和其他参数生成 SSL 密钥，然后将其保存到 serverkeys 文件。 例如，您可以运行以下命令：
   - Linux 操作系统：

     ```
     $keytool -genkey -alias collation -keystore serverkeys -validity 3650
     -keyAlg RSA -sigalg SHA256WithRSA -keypass $pass -storepass $pass -dname
     "CN=John Public, OU=Engineering, OU=NA, o=Company, L=Manhattan,
     S=New York, c=US"
     ```

   - Windows 操作系统：

     ```
     %keytool% -genkey -alias collation -keystore serverkeys -validity 3650
     -keyAlg RSA -sigalg SHA256WithRSA -keypass %pass% -storepass %pass% -dname
     "CN=John Public, OU=Engineering, OU=NA, o=Company, L=Manhattan,
     S=New York, c=US"
     ```

5. 创建 serverkeys 文件的另一个备份副本，在此文件中保存生成的 SSL 密钥。
6. 运行以下命令以生成证书签名请求（CSR 文件）：
   - Linux 操作系统：

     ```
     $keytool -certreq -alias collation -storepass $pass -file
     /tmp/certreq.csr -keystore serverkeys
     ```

   - Windows 操作系统：

     ```
     %keytool% -certreq -alias collation -storepass %pass% -file
     C:\temp\certreq.csr -keystore serverkeys
     ```

7. 使用 CSR 文件从官方认证中心获取 SSL 证书。 在 TADDM 服务器上将证书保存为 cert.crt 文件，例如，在 Linux 操作系统的 tmp 目录中，或者在 Windows 操作系统的 C:\temp 目录中。
8. 通过运行以下命令，将生成的 TADDM 证书导入 serverkeys 和 jssecacerts.cert 文件：

   要点：对于 -file 参数，指定在先前步骤中保存 SSL 证书的文件的路径，例如，/tmp/cert.crt（在 Linux 操作系统上）或 C:\temp\cert.crt（在 Windows 操作系统上）。
   - Linux 操作系统：

     ```
     $keytool -import -trustcacerts -alias collation -noprompt -keystore
     serverkeys -storepass $pass -keypass $pass -file /tmp/cert.crt
     $keytool -import -trustcacerts -alias collation -noprompt -keystore
     jssecacerts.cert -storepass $pass -keypass $pass -file /tmp/cert.crt
     ```

   - Windows 操作系统：

     ```
     %keytool% -import -trustcacerts -alias collation -noprompt -keystore
     serverkeys -storepass %pass% -keypass %pass% -file C:\temp\cert.crt
     %keytool% -import -trustcacerts -alias collation -noprompt -keystore
     jssecacerts.cert -storepass %pass% -keypass %pass% -file C:\temp\cert.crt
     ```

9. 重新启动 TADDM 服务器。

下一步做什么

保留在步骤 4 中生成的 serverkeys 文件的备份副本，以及在步骤 7 中保存 SSL 证书的文件的备份副本。如果必须替换或更新证书，那么这些文件是必需的。 要替换或更新证书，请完成以下步骤：

1. 重复第 2 步和第 3 步。
2. 复原 serverkeys 文件。
3. 重复第 8 步和第 9 步。

管理 TADDM 服务器

在为发现配置 TADDM 前，您必须了解如何管理 TADDM 服务器，其中包括许多任务。

检查 TADDM 服务器状态

您可以使用管理员控制台或 **control** 命令来检查 TADDM 服务器的状态。

使用管理员控制台检查状态

要使用管理员控制台检查状态，请打开 Web 浏览器，并输入安装有 TADDM 服务器的系统的 URL 和端口号。 例如以下 URL：

```
http://system.company.com:9430
```

然后会显示管理员控制台，它列示了 TADDM 服务器的组件及其状态。

使用 control 命令检查状态

要使用 **control** 命令检查状态，请完成下列步骤：

1. 以安装过程中定义的非 root 用户身份登录。
2. 从命令提示符转至安装了 TADDM 服务器的目录。
3. 运行以下命令之一：
   - 对于 AIX、Linux 和 Linux on System z 操作系统：

     ```
     $COLLATION_HOME/bin/control status
     ```
   - 对于 Windows 操作系统：

     ```
     %COLLATION_HOME%\bin\control.bat status
     ```

   根据您使用的部署和在各种部署中运行 TADDM 的服务器类型，会显示以下输出：

   同步服务器部署

   同步服务器

   - TADDM 7.3.0：

     ```
     DbInit: Started
     Tomcat: Started
     EcmdbCore: Started

     TADDM: Running
     ```
   - TADDM 7.3.0.1 及更高版本：

     ```
     DbInit: Started
     Liberty: Started
     EcmdbCore: Started

     TADDM: Running
     ```

   域服务器

   - TADDM 7.3.0：

     ```
     Discover: Started
     DbInit: Started
     Tomcat: Started
     Topology: Started
     DiscoverAdmin: Started
     ```

```
Proxy: Started
EventsCore: Started

TADDM: Running
```

- TADDM 7.3.0.1 及更高版本：

```
Discover: Started
DbInit: Started
Liberty: Started
Topology: Started
DiscoverAdmin: Started
Proxy: Started
EventsCore: Started

TADDM: Running
```

流式服务器部署

存储服务器

- TADDM 7.3.0：

```
TADDM: Starting
EtaddmCore: Started
DbInit: Started
Tomcat: Started

TADDM: Running
```

- TADDM 7.3.0.1 及更高版本：

```
TADDM: Starting
EtaddmCore: Started
DbInit: Started
Liberty: Started

TADDM: Running
```

发现服务器

- TADDM 7.3.0：

```
Discover: Started
Tomcat: Started
DiscoverAdmin: Started
ProxyLite: Started
EventsCore: Started

TADDM: Running
```

- TADDM 7.3.0.1 及更高版本：

```
Discover: Started
Liberty: Started
DiscoverAdmin: Started
ProxyLite: Started
EventsCore: Started

TADDM: Running
```

启动 TADDM 服务器

如果在安装时选择了**在引导时启动**选项，那么 TADDM 服务器将在每次系统引导时自动启动。

关于此任务

要点：在启动 TADDM 服务器之前，必须先启动并运行本地或远程数据库服务器。 如果数据库不可用，TADDM 服务器将不能正常初始化或运行。

过程

要手动启动 TADDM 服务器，请完成下列步骤：

1. 以安装过程中定义的非 root 用户身份登录。
2. 打开命令提示符窗口。

   注：在 Windows Server 2008 系统上，如果"用户帐户控制"已开启，请使用管理员特权打开命令提示符窗口。 要执行此操作，可右键单击命令提示符图标，然后单击**以管理员身份运行**。
3. 转至安装了 TADDM 服务器的目录。
4. 使用以下命令之一来运行 start 脚本：
   - 对于 Linux、AIX 和 Linux on System z 操作系统：

     ```
     $COLLATION_HOME/bin/control start
     ```
   - 对于 Windows 操作系统：

     ```
     %COLLATION_HOME%\bin\startServer.bat
     ```

   在 Windows 系统上启动服务器时，可能会看到以下超时错误消息：错误 1053：服务未及时响应启动或控制请求。 出现此错误的原因是 TADDM 服务器的启动时间比允许的启动时间要长。 您可以忽略此消息；启动过程将继续，直到完成为止。

   如果使用 root 用户特权安装了 TADDM 服务器，那么可通过运行以下脚本来手动启动 TADDM 服务器：

   ```
   /etc/init.d/collation start
   ```

停止 TADDM 服务器

您可以手动停止 TADDM 服务器和相关的发现进程。

过程

要手动停止 TADDM 服务器，请完成下列步骤：

1. 以安装过程中定义的非 root 用户身份登录。
2. 打开命令提示符窗口。

   注：在 Windows Server 2008 系统上，如果"用户帐户控制"已开启，请使用管理员特权打开命令提示符窗口。 要执行此操作，可右键单击命令提示符图标，然后单击**以管理员身份运行**。
3. 转至安装了 TADDM 服务器的目录。
4. 使用以下命令之一来运行 stop 脚本：
   - 对于 Linux、AIX 和 Linux on System z 操作系统：

     ```
     $COLLATION_HOME/bin/control stop
     ```
   - 对于 Windows 操作系统：

     ```
     %COLLATION_HOME%\bin\stopServer.bat
     ```

如果使用 root 用户特权安装了 TADDM 服务器，那么可通过运行以下脚本来手动停止 TADDM 服务器：

```
/etc/init.d/collation stop
```

下一步做什么

某些传感器在自己的特殊 Java 虚拟机 (JVM) 中运行。 运行发现时，如果使用控制脚本 (./control stop) 停止 TADDM，那么可能需要手动停止这些额外的 JVM（称为本地锚点）。 如果不停止本地锚点，可能会出现意外行为。 例如，可能降低某些发现的性能。

要验证本地锚点的进程是否不再运行，请输入以下命令：

```
% ps -ef |grep -i anchor
```

此命令识别正在运行的所有本地锚点进程。 输出看上去类似于以下代码示例：

```
 coll   23751  0.0  0.0  6136  428 ?   S  Jun02   0:00 /bin/sh
    local-anchor.sh 8494 <more information here>
```

如果进程正在运行，请通过运行以下命令停止进程：

```
-  % kill -9 23751
```

运行该命令后，通过运行以下命令来验证进程是否已停止：

```
 % ps -ef |grep -i anchor
```

备份数据

定期备份数据，以便可以从系统故障中恢复。

开始之前

备份数据前，请先停止 TADDM 服务器。

过程

要备份 TADDM 服务器的文件，请完成以下任务：

将所有文件保存到安装 TADDM 服务器的目录中。

- 对于 Linux、AIX 和 Linux on System z 操作系统，此目录的缺省路径为 /opt/IBM。
- 对于 Windows 操作系统，此目录的缺省路径为 C:\opt\IBM。

下一步做什么

要备份数据库文件，请使用数据库供应商提供的文档。

恢复数据

在系统发生故障后，您可以恢复配置、数据和数据库文件。 因此，可以从故障发生前的最近一个备份点恢复操作。

过程

要从备份介质恢复数据，请完成下列步骤：

1. 执行下列其中一项操作：
   - 恢复 /opt/IBM 目录并重新启动 TADDM。
   - 恢复 C:\opt\IBM 目录并重新启动 TADDM。
2. 找到数据文件的备份副本。
3. 打开命令提示符窗口。
4. 浏览到安装了 TADDM 服务器的目录。
5. 将数据文件的备份副本复制到安装目录中。
6. 关闭命令提示符窗口。
7. 启动 TADDM 服务器。

下一步做什么

如果数据库受到系统故障的影响，请使用数据库供应商提供的文档恢复数据库文件。

在 TADDM 服务器之间复制发现作用域、概要文件和定制服务器模板

可以使用 **datamover.sh|bat** 命令在 TADDM 服务器之间复制发现作用域、发现概要文件和定制服务器模板。

您可以导出发现作用域、概要文件和定制服务器模板（所有实体）或指定要从服务器导出的实体。 然后您可以在目标服务器上导入相应实体。

限制：要维护数据完整性，必须在相同版本的 TADDM 服务器之间移动数据。

要在 TADDM 服务器之间复制实体，请完成下列步骤：

1. 在源服务器上运行以下命令将所需实体导出到文件：

   ```
   datamover.sh|bat -u user -p password -a action [-t type ] [-f filename]
   ```

   其中：

   user
   TADDM 用户名。

   password
   TADDM 用户密码。

   action
   指定以下某个操作：import、export 或 help。

   可选：*type*
   指定以下某个类型：all、scope、profile 或 template。缺省值为 all。

   可选：*filename*
   指定文件名。 缺省值为 datamover.xml。

   将不会导出缺省发现概要文件，可以导出所有定制服务器模板、用户创建的概要文件和作用域。

运行命令后，将显示已导出的条目的相关信息。 例如，如果输出文件为 exporthost.xml，那么提供以下信息：

```
Exported 6 scopes
Exported 1 profiles
Exported 57 templates
```

2. 将相应文件复制到目标服务器，然后运行 **datamover.sh|bat**，并导入相应实体。

   导入实体时以下规则适用：

   - 如果服务器上存在同名作用域或概要文件，那么将重命名导入的作用域或概要文件。 该文件将重命名为 *name*_TADDM。
   - 如果服务器上存在同名模板，那么导入的模板将合并到现有模板中。

部署发现管理控制台

确认 TADDM 服务器可用后，即可部署发现管理控制台。

过程

要部署发现管理控制台，请完成下列步骤：

1. 为用户提供安装了 TADDM 服务器的系统的 URL（包括端口号）。

   例如，您可以向用户提供与以下 URL 类似的地址：

   ```
   http://system.company.com:9430
   ```

2. 向用户提供其用户名和密码。
3. 指定用户是否应使用安全套接字层 (SSL)。

   在使用 SSL 的情况下，指导用户按照发现管理控制台"安装和启动"页面上的以下指示信息保存 TADDM 服务器的信任密钥库。 有关更多信息，请参阅 TADDM 《安装指南》。

   要点：应该对发现管理控制台和 TADDM 服务器之间的所有通信使用 SSL。

4. 用户需要在用于查看发现管理控制台的系统上安装受支持版本的 Java 运行时环境。有关客户机先决条件的更多信息，请参阅 TADDM 《安装指南》。
5. 让用户参阅 TADDM 《用户指南》，以获取有关如何启动 Discovery Management Console 的信息。

配置 TADDM 通信

要建立 TADDM 通信，必须配置所有必需的服务、连接和防火墙。

TADDM 服务

TADDM 连接可分为三个区域：

公共连接

公共连接涉及在 TADDM 基础结构之外完成的网络连接。 例如，连接到 TADDM 服务器的数据管理门户网站、发现管理控制台或 API 客户机。 公共连接是连接的最高层。

服务器间连接

服务器间连接涉及 TADDM 核心基础结构元素（即发现服务器和存储服务器）之间的网络连接。 服务器间连接是连接的中层。

本地连接

本地连接涉及在某台机器上多个本地服务之间的网络连接。 本地连接是连接的最低层。

您可以在安装阶段为每项服务配置连接，或者在稍后通过更改 collation.properties 配置文件中的配置属性进行配置。

服务缺省接口

要配置服务缺省侦听接口，请更改 collation.properties 文件中的 com.ibm.cdb.global.hostname 属性。

表 2. 服务缺省接口设置

名称	配置属性	缺省接口
全局服务主机	com.ibm.cdb.global.hostname	0.0.0.0

取决于通信类型的侦听接口

要分别为每个连接区域中的各项服务配置侦听接口，请更改 collation.properties 文件中的相应属性。

表 3. 服务缺省接口设置

名称	配置属性	缺省接口
公共连接服务主机	com.ibm.cdb.public.hostname	由 com.ibm.cdb.global.hostname 定义
服务器间连接服务主机	com.ibm.cdb.interserver.hostname	由 com.ibm.cdb.global.hostname 定义
本地连接服务主机	com.ibm.cdb.local.hostname	127.0.0.1

注：如果未指定接口或者接口的值为 0.0.0.0.， 那么必须打开一个本地外部网络接口以与其自身通信。 如果指定了接口，那么必须打开该接口才能与自身通信。

特定服务的侦听接口

您可以在安装阶段为每项服务配置独立的 TCP 端口，或者在稍后通过更改 collation.properties 文件中的各个相应属性进行配置。

服务接口配置

要为每项服务配置特定的侦听接口，请更改 collation.properties 文件中后缀为 host 的相应属性。

TopologyManager 服务的示例：

```
com.ibm.cdb.service.TopologyManager.host=192.168.1.5
```

注：此命名约定不适用于公共或服务器间服务注册表。

服务端口配置

要为每项服务配置特定的侦听端口，请更改 collation.properties 文件中后缀为 port 的相应属性。

如下是 TopologyManager 服务的示例：

```
com.ibm.cdb.service.TopologyManager.port=9550
```

SSL 服务配置

要为每项 SSL 服务配置特定的侦听接口或端口，请更改 collation.properties 文件中中缀为 secure 的相应属性。

如下是 SecureApiServer 服务的示例：

- com.ibm.cdb.service.SecureApiServer.secure.host=192.168.1.5
- com.ibm.cdb.service.SecureApiServer.secure.port=9531

Web 门户网站接口（HTTP 和 HTTPS）配置

要为 Web 门户网站（HTTP 和 HTTPS）配置侦听接口，请更改 collation.properties 文件中的 com.ibm.cdb.service.web.host 属性。

注：与其他服务相比，只需更改一个属性就可以配置 HTTP 和 HTTPS 主机。

数据库连接

要配置特定的数据库连接，请更改 collation.properties 文件中的 com.collation.db.port 和 com.collation.db.server 属性。

例如：

- com.collation.db.port=65432
- com.collation.db.server=9.156.47.156

DNS 连接

如果要使用标准域名 (FQDN) 进行通信，请确保参与通信的主机能够解析来自 DNS 服务的 FQDN。

传感器连接

ping 传感器和端口传感器文档中包含了 ping 传感器和端口传感器用于进行连接的端口配置。 请确保要发现的服务端口处于打开状态。

表 *4. Ping* 传感器和端口传感器缺省端口

端口名称	缺省端口	协议
SSH	22	TCP
Telnet	23	TCP
DNS	53	TCP
WMI	135	TCP
Fix Pack 2 PowerShell	5985, 5986	TCP
LDAP	389	TCP

表 4. *Ping* 传感器和端口传感器缺省端口 (续)

端口名称	缺省端口	协议
SMB	445	TCP
Oracle	1521	TCP
CiscoWorks	1741	TCP

锚点连接

TADDM 可使用以下任一连接类型连接到锚点服务器：*ssh* 或 *direct*。要配置特定的锚点连接类型，请将 `collation.properties` 文件中的 `com.collation.discover.anchor.connectType` 属性更改为 *ssh* 或 *direct*。

要为特定地址配置特定的锚点连接类型，请更改 `collation.properties` 文件中后缀为 IP 地址的 `com.collation.discover.anchor.connectType` 属性。例如：

```
com.collation.discover.anchor.connectType.1.2.3.4=direct
```

另外，可将端口 8497 定义为连接到锚点服务器的缺省端口。 可以使用发现管理控制台来配置此端口。

- 以 *ssh* 方式打开从 TADDM 服务器访问的公共接口上的各个 SSH 通信端口，以及锚点服务器所在机器的回送接口上的锚点连接端口。
- 以 *direct* 方式打开从 TADDM 服务器访问的公共接口上的 SSH 通信端口以及锚点连接端口。

网关连接

通过使用 SSH 连接，TADDM 可以连接到网关服务器。

在网关上，主机 SSH 端口必须处于打开状态，以便在通过 TADDM 服务器访问的公共接口上进行通信。

将服务器主机名解析为标准域名

要确保在服务器之间成功通信，通过使用操作系统的解析器库，主机服务器必须能够将其主机名解析为标准域名 (FQDN)。必须满足以下某个条件：

- 在操作系统主机解析的搜索顺序中，DNS 必须优先搜索本地文件。 要配置此设置，请参阅操作系统文档。
- 在主机文件中，TADDM 服务器的 FQDN 必须优先于短名称。

如果不能满足上述条件，那么您可以将 `collation.properties` 文件中的 `com.collation.serverID` 属性设置为 TADDM 服务器的 IP 或主机名。 此外，请确保将"同步服务器/企业服务器 > 数据管理门户网站 > 域管理 > 域主机名"中的服务器标识设置为相同值。

临时端口

TADDM 通信包括临时端口的使用。这些端口是临时的，且特定于操作系统。每个操作系统都有一个已定义的端口号范围，将会从中随机选择特定端口。TADDM 不定义这些端口。有关该端口范围的信息、必需的配置以及更多详细信息，请参阅您所使用的操作系统的文档。

配置防火墙

要建立 TADDM 通信，必须配置必需的防火墙。 根据所配置的域服务器部署、流式服务器部署或同步服务器部署，此任务的详细信息会有所不同。

各个表显示了防火墙配置信息。每个表都指示了通信方向。 在目标机器上，防火墙上的目标服务端口必须处于打开状态，用于作为出局连接的源以及入局连接的目标。 在源机器上，防火墙上的目标服务端口必须处于打开状态，用于作为出局连接的目标以及入局连接的源。

要点：低层客户机也必须提供高级服务。 例如，对于服务器间连接，公共服务也必须处于打开状态。

如果表中指定的方向描述为回送，那么必须在此接口上开启所有通信。 在更改任何缺省端口配置时，请确保已打开了与环境配置相应的端口。

在域服务器部署中配置防火墙

在域服务器部署中必须配置防火墙，从而使特定端口处于打开状态，以便进行通信。

下图显示了域服务器部署中的 TADDM 通信。

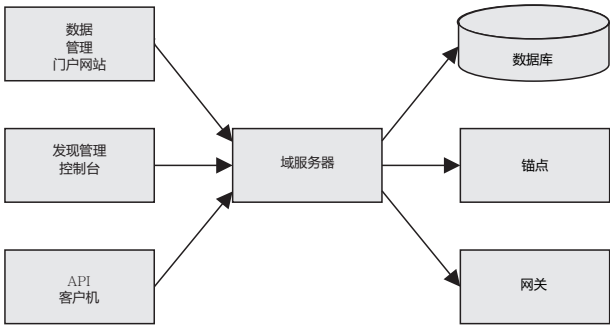

图 *1.* 域服务器部署中的 *TADDM* 通信

连接服务：

对于域服务器部署，您可以配置公共连接服务、服务器间连接服务以及本地连接服务。

公共连接服务

下表显示了域服务器公共连接服务的缺省主机设置。

表 *5.* 域服务器公共连接服务的缺省主机设置

名称	配置属性	缺省接口
公共服务主机	com.ibm.cdb.public.hostname	由 com.ibm.cdb.global.hostname 定义

下表显示了域服务器公共连接服务的缺省端口设置。

表 *6.* 域服务器公共连接服务的缺省端口设置

名称	配置属性	协议	缺省端口
API 服务器端口	com.ibm.cdb.service.ApiServer.port	TCP	9530

表 *6.* 域服务器公共连接服务的缺省端口设置（续）

名称	配置属性	协议	缺省端口
安全 API 服务器端口	com.ibm.cdb.service.SecureApiServer.secure.port	TCP	9531
HTTP 端口（没有 SSL）	com.ibm.cdb.service.web.port	TCP	9430
HTTPS 端口（具有 SSL）	com.ibm.cdb.service.web.secure.port	TCP	9431
GUI 服务器通信端口	com.ibm.cdb.service.ClientProxyServer.port	TCP	9435
GUI 服务器 SSL 通信端口	com.ibm.cdb.service.SecureClientProxyServer.secure.port	TCP	9434
公共服务注册表端口	com.ibm.cdb.service.registry.public.port	TCP	9433

本地连接服务

未显式设置本地服务端口。 为本地服务定义的接口上的所有端口必须处于打开状态。 缺省接口为回送接口。

下表显示了域服务器本地连接服务的缺省主机设置。

表 *7.* 域服务器本地连接服务的缺省主机设置

名称	配置属性	缺省接口
本地服务主机	com.ibm.cdb.local.hostname	127.0.0.1

在域服务器部署中配置通信：

要在域服务器部署中成功建立通信，请配置公共连接服务和本地连接服务。

下表显示可在域服务器部署中进行连接的元素，以及必须打开才能成功进行通信的端口。

数据库服务器和域服务器之间的通信

表 *8.* 数据库服务器和域服务器之间的通信。

元素 A	端口	说明	元素 B	配置属性
数据库服务器	5000	←	域服务器	

发现管理门户网站、API 客户机、Web 门户网站和数据管理门户网站客户机、以及域服务器之间的通信

表 *9.* 发现管理门户网站、*API* 客户机、*Web* 门户网站和数据管理门户网站客户机、以及域服务器之间的通信。

元素 A	端口	说明	元素 B	配置属性
发现管理门户网站	9433	→	域服务器 - 公共服务注册表	com.ibm.cdb.service.registry.public.port
	9435	→	域服务器 - ClientProxyServer	com.ibm.cdb.service.ClientProxyServer.port
	9434	→	域服务器 - SecureClientProxyServer	com.ibm.cdb.service.SecureClientProxyServer.secure.port

表 *9.* 发现管理门户网站、*API* 客户机、*Web* 门户网站和数据管理门户网站客户机、以及域服务器之间的通信。 *(续)*

元素 A	端口	说明	元素 B	配置属性
API 客户机	9433	→	域服务器 - 公共服务注册表	com.ibm.cdb.service.registry.public.port
	9530	→	域服务器 - API 服务器	com.ibm.cdb.service.ApiServer.port
	9531	→	域服务器 - 安全 API 服务器	com.ibm.cdb.service.SecureApiServer.secure.port
Web 门户网站和数据管理门户网站客户机	9430	→	域服务器 - Web	com.ibm.cdb.service.web.port
	9431	→	域服务器 - 安全 Web	com.ibm.cdb.service.web.secure.port

锚点、网关和域服务器之间的通信

表 *10.* 锚点、网关和域服务器之间的通信。

元素 A	端口	说明	元素 B	配置属性
锚点（以 SSH 方式）- SSH	22	←	域服务器（以 SSH 方式）	
锚点（以 direct 方式）- SSH		←	域服务器（以 direct 方式）	
锚点（以 SSH 方式）- SSH 通道转发	8497	↔	域服务器（以 SSH 方式）	
锚点（以 direct 方式）- direct		←	域服务器（以 direct 方式）	
网关 - SSH	22	←	域服务器	

本地通信

表 *11.* 域服务器的本地连接通信配置。

本地通信	说明	配置属性
域服务器 - 本地服务注册表	↔	com.ibm.cdb.local.hostname
域服务器 - 本地服务		
域服务器 - 127.0.0.1		

在流式服务器部署中配置防火墙

在流式服务器部署中必须配置防火墙，从而使特定端口处于打开状态，以便进行通信。

下图显示了流式服务器部署中的 TADDM 通信。

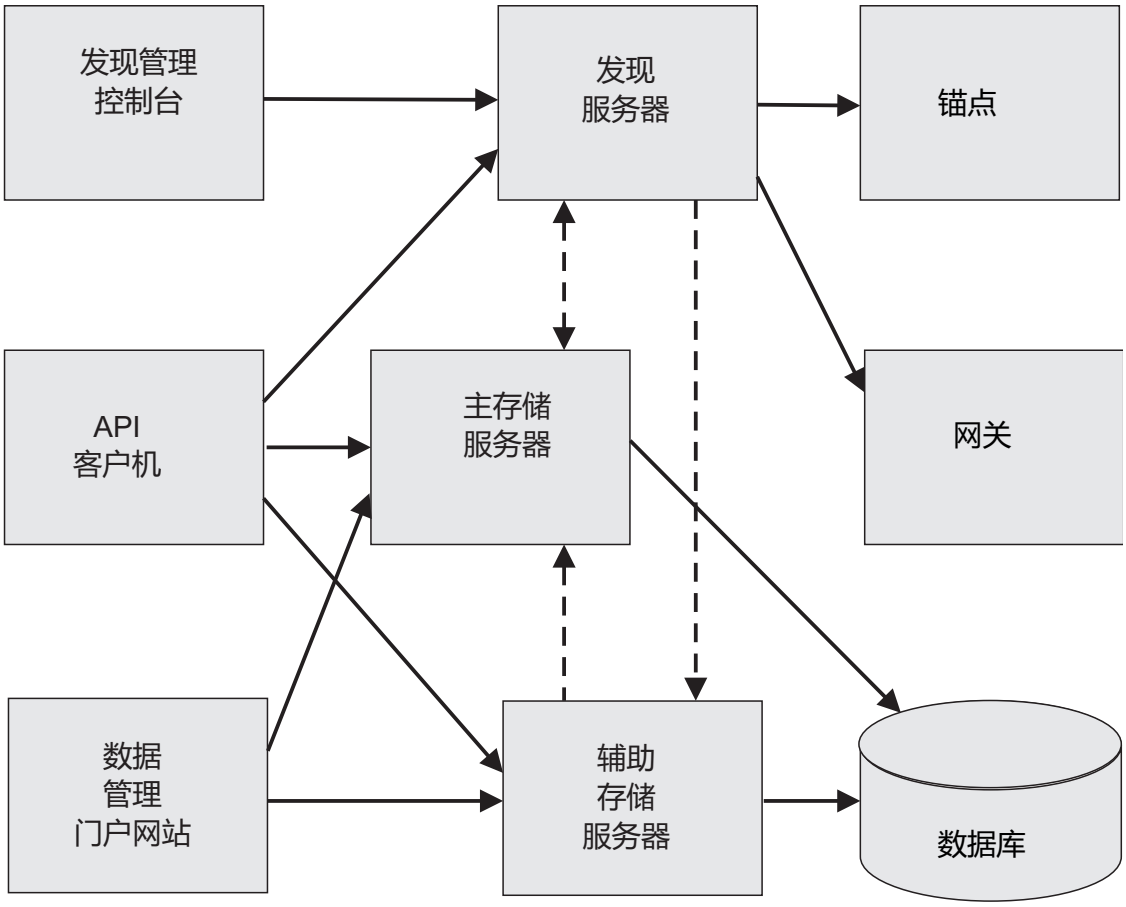

图 2. 流式服务器部署中的 *TADDM* 通信

连接服务：

对于流式服务器部署，您可以配置公共连接服务、服务器间连接服务以及本地连接服务。

要点：本节随后提供的属性的缺省端口仅适用于 `collation.properties` 文件中列示的那些属性。如果 `collation.properties` 文件中未编码某个属性或者将其注释掉，那么该属性将缺省为随机端口。 特别是，请确保在 `collation.properties` 文件中列示 `com.ibm.cdb.service.RegistriesURLProvider.port` 属性，以确保启动成功。

公共连接服务

下表显示了主存储服务器、辅助存储服务器以及发现服务器公共连接服务的缺省主机设置。

表 12. 主存储服务器、辅助存储服务器以及发现服务器公共连接服务的缺省主机设置

名称	配置属性	缺省接口
公共服务主机	com.ibm.cdb.public.hostname	由 com.ibm.cdb.global.hostname 定义

下表显示了主存储服务器、辅助存储服务器以及发现服务器公共连接服务的缺省端口设置。

表 *13.* 主存储服务器、辅助存储服务器以及发现服务器公共连接服务的缺省端口设置

名称	配置属性	协议	缺省端口
API 服务器端口	com.ibm.cdb.service.ApiServer.port	TCP	9530
安全 API 服务器端口	com.ibm.cdb.service.SecureApiServer.secure.port	TCP	9531
HTTP 端口（没有 SSL）	com.ibm.cdb.service.web.port	TCP	9430
HTTPS 端口（具有 SSL）	com.ibm.cdb.service.web.secure.port	TCP	9431
GUI 服务器通信端口	com.ibm.cdb.service.ClientProxyServer.port	TCP	9435
GUI 服务器 SSL 通信端口	com.ibm.cdb.service.SecureClientProxyServer.secure.port	TCP	9434
公共服务注册表端口	com.ibm.cdb.service.registry.public.port	TCP	9433

服务器间连接服务

下表显示了主存储服务器和辅助存储服务器服务器间连接服务的缺省主机设置。

表 *14.* 主存储服务器和辅助存储服务器服务器间连接服务的缺省主机设置

名称	配置属性	缺省接口
服务器间服务主机	com.ibm.cdb.interserver.hostname	由 com.ibm.cdb.global.hostname 定义

下表显示了主存储服务器服务器间连接服务的缺省端口设置。

表 *15.* 主存储服务器服务器间连接服务的缺省端口设置

名称	配置属性	协议	缺省端口
TopologyManager 端口	com.ibm.cdb.service.TopologyManager.port	TCP	9550
SecurityManager 端口	com.ibm.cdb.service.SecurityManager.port	TCP	9540
RegistriesURLProvider 端口	com.ibm.cdb.service.RegistriesURLProvider.port	TCP	9560
服务器间服务注册表端口	com.ibm.cdb.service.registry.interserver.port	TCP	4160

下表显示了辅助存储服务器服务器间连接服务的缺省端口设置。

表 *16.* 辅助存储服务器服务器间连接服务的缺省端口设置

名称	配置属性	协议	缺省端口
TopologyManager 端口	com.ibm.cdb.service.TopologyManager.port	TCP	9550
RegistriesURLProvider 端口	com.ibm.cdb.service.RegistriesURLProvider.port	TCP	9560
服务器间服务注册表端口	com.ibm.cdb.service.registry.interserver.port	TCP	4160

本地连接服务

未显式设置本地服务端口。 为本地服务定义的接口上的所有端口必须处于打开状态。 缺省接口为回送接口。

下表显示了主存储服务器、辅助存储服务器以及发现服务器本地连接服务的缺省主机设置。

表 *17.* 主存储服务器、辅助存储服务器以及发现服务器本地连接服务的缺省主机设置

名称	配置属性	缺省接口
本地连接区域服务主机	com.ibm.cdb.local.hostname	127.0.0.1

在流式服务器部署中配置通信：

要在流式服务器部署中成功建立通信，请配置公共连接服务、服务器间连接服务以及本地连接服务。

下表显示可在流式服务器部署中进行连接的元素，以及必须打开才能成功进行通信的端口。

服务器间通信

表 *18.* 流式服务器部署中的服务器间连接通信配置。

元素 A	端口	说明	元素 B	配置属性	TLS 支持
发现服务器			**主存储服务器**		
	9433	→	主存储服务器	com.ibm.cdb.service.registry.public.port	是
	4160	→	主存储服务器 - 服务器间服务注册表	com.ibm.cdb.service.registry.interserver.port	否
	9560	→	主存储服务器 - Registries URLProvider	com.ibm.cdb.service.Registries URLProvider.port	是
	9540	→	主存储服务器 - SecurityManager	com.ibm.cdb.service.Security Manager.port	是
	9550	→	主存储服务器 - TopologyManager	com.ibm.cdb.service.Topology Manager.port	是
	9430	←	主存储服务器 - web	com.ibm.cdb.service.web.port	否
发现服务器			**辅助存储服务器**		
	4160	→	辅助存储服务器 - 服务器间服务注册表	com.ibm.cdb.service.registry.interserver.port	否
	9560	→	辅助存储服务器 - Registries URLProvider	com.ibm.cdb.service.Registries URLProvider.port	是
	9550	→	辅助存储服务器 - TopologyManager	com.ibm.cdb.service.Topology Manager.port	是

表 *18.* 流式服务器部署中的服务器间连接通信配置。 *(续)*

元素 A	端口	说明	元素 B	配置属性	TLS 支持
辅助存储服务器			**主存储服务器**		
	4160	→	主存储服务器 - 服务器间服务注册表	com.ibm.cdb.service.registry.interserver.port	否
	9560	→	主存储服务器 - Registries URLProvider	com.ibm.cdb.service.RegistriesURLProvider.port	是
	9540	→	主存储服务器 - SecurityManager	com.ibm.cdb.service.SecurityManager.port	是
	9550	→	主存储服务器 - TopologyManager	com.ibm.cdb.service.TopologyManager.port	是
数据库服务器	5000	←	**主存储服务器**		否
数据库服务器	5000	←	**辅助存储服务器**		否

发现管理门户网站、API 客户机、Web 门户网站和数据管理门户网站客户机、以及 TADDM 服务器之间的通信

表 *19.* 发现管理门户网站、*API* 客户机、*Web* 门户网站和数据管理门户网站客户机、以及 *TADDM* 服务器之间的通信。

元素 A	端口	说明	元素 B	配置属性	TLS 支持
发现管理门户网站	9433	→	发现服务器 - 公共服务注册表	com.ibm.cdb.service.registry.public.port	是
	9435	→	发现服务器 - ClientProxyServer	com.ibm.cdb.service.ClientProxyServer.port	否
	9434	→	发现服务器 - SecureClient ProxyServer	com.ibm.cdb.service.SecureClientProxyServer.secure.port	是

表 *19.* 发现管理门户网站、*API* 客户机、*Web* 门户网站和数据管理门户网站客户机、以及 *TADDM* 服务器之间的通信。*(续)*

元素 A	端口	说明	元素 B	配置属性	TLS 支持
API 客户机	9433	→	• 发现服务器 - 公共服务注册表 • 主存储服务器 - 公共服务注册表 • 辅助存储服务器 - 公共服务注册表	com.ibm.cdb.service.registry.public.port	是
	9530	→	• 发现服务器 - API 服务器 • 主存储服务器 - API 服务器 • 辅助存储服务器 - API 服务器	com.ibm.cdb.service.ApiServer.port	否
	9531	→	• 发现服务器 - 安全 API 服务器 • 主存储服务器 - 安全 API 服务器 • 辅助存储服务器 - 安全 API 服务器	com.ibm.cdb.service.SecureApiServer.secure.port	是
Web 门户网站和数据管理门户网站客户机	9430	→	• 发现服务器 - Web • 主存储服务器 - Web • 辅助存储服务器 - Web	com.ibm.cdb.service.web.port	否
	9431	→	• 发现服务器 - 安全 Web • 主存储服务器 - 安全 Web • 辅助存储服务器 - 安全 Web	com.ibm.cdb.service.web.secure.port	是

锚点、网关和发现服务器之间的通信

表 *20.* 锚点、网关和发现服务器之间的通信。

元素 A	端口	说明	元素 B	配置属性
锚点（以 SSH 方式）- SSH	22	←	发现服务器（以 SSH 方式）	
锚点（以 direct 方式）- SSH		←	发现服务器（以 direct 方式）	

表 *20.* 锚点、网关和发现服务器之间的通信。 *(续)*

元素 A	端口	说明	元素 B	配置属性
锚点（以 SSH 方式）- SSH 通道转发	8497	⟷	发现服务器（以 SSH 方式）	
锚点（以 direct 方式）- direct		⟵	发现服务器（以 direct 方式）	
网关 - SSH	22	⟵	发现服务器	

本地通信

表 *21.* 流式服务器部署中的本地连接通信配置。

本地通信	说明	配置属性
发现服务器		
发现服务器 - 本地服务注册表	⟷	com.ibm.cdb.local.hostname
发现服务器 - 本地服务		
发现服务器 - 127.0.0.1		
主存储服务器		
主存储服务器 - 本地服务注册表	⟷	com.ibm.cdb.local.hostname
主存储服务器 - 本地服务		
主存储服务器 - 127.0.0.1		
辅助存储服务器		
辅助存储服务器 - 本地服务注册表	⟷	com.ibm.cdb.local.hostname
辅助存储服务器 - 本地服务		
辅助存储服务器 - 127.0.0.1		

在同步服务器部署中配置防火墙

在同步服务器部署中必须配置防火墙，从而使特定端口处于打开状态，以便进行通信。

下图显示了同步服务器部署中的 TADDM 通信。

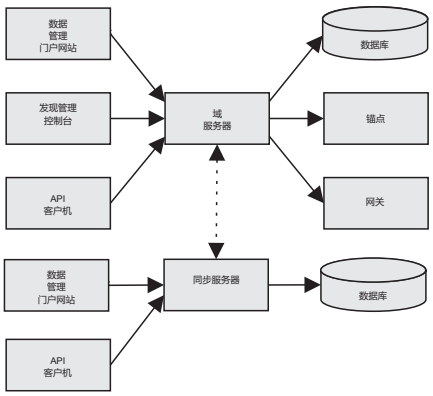

图 *3.* 同步服务器部署中的 *TADDM* 通信

连接服务配置：

对于同步服务器部署，您可以配置公共连接服务、服务器间连接服务以及本地连接服务。

要点：本节随后提供的属性的缺省端口仅适用于 collation.properties 文件中列示的那些属性。如果 collation.properties 文件中未编码某个属性或者将其注释掉，那么该属性将缺省为随机端口。 特别是，请确保在 collation.properties 文件中列示 com.ibm.cdb.service.RegistriesURLProvider.port 属性，以确保启动成功。

公共连接服务

下表显示了域服务器和同步服务器公共连接服务的缺省主机设置。

表 *22.* 域服务器和同步服务器公共连接服务的缺省主机设置

名称	配置属性	缺省接口
公共服务主机	com.ibm.cdb.public.hostname	由 com.ibm.cdb.global.hostname 定义

下表显示了域服务器公共连接服务的缺省端口设置。

表 *23.* 域服务器公共连接服务的缺省主机设置

名称	配置属性	协议	缺省端口
API 服务器端口	com.ibm.cdb.service.ApiServer.port	TCP	9530
安全 API 服务器端口	com.ibm.cdb.service.SecureApiServer.secure.port	TCP	9531
HTTP 端口（没有 SSL）	com.ibm.cdb.service.web.port	TCP	9430
HTTPS 端口（具有 SSL）	com.ibm.cdb.service.web.secure.port	TCP	9431
GUI 服务器通信端口	com.ibm.cdb.service.ClientProxyServer.port	TCP	9435
GUI 服务器 SSL 通信端口	com.ibm.cdb.service.SecureClientProxyServer.secure.port	TCP	9434
公共服务注册表端口	com.ibm.cdb.service.registry.public.port	TCP	9433

下表显示了同步服务器公共连接服务的缺省端口设置。

表 *24.* 同步服务器公共连接服务的缺省端口设置

名称	配置属性	协议	缺省端口
API 服务器端口	com.ibm.cdb.service.ApiServer.port	TCP	9530
安全 API 服务器端口	com.ibm.cdb.service.SecureApiServer.secure.port	TCP	9531
HTTP 端口（没有 SSL）	com.ibm.cdb.service.web.port	TCP	9430
HTTPS 端口（具有 SSL）	com.ibm.cdb.service.web.secure.port	TCP	9431
公共服务注册表端口	com.ibm.cdb.service.registry.public.port	TCP	9433

服务器间连接服务

下表显示了域服务器和同步服务器服务器间连接服务的缺省主机设置。

表 *25.* 域服务器和同步服务器服务器间连接服务的缺省主机设置

名称	配置属性	缺省接口
服务器间服务主机	com.ibm.cdb.interserver.hostname	由 com.ibm.cdb.global.hostname 定义

下表显示了域服务器服务器间连接服务的缺省端口设置。

表 *26.* 域服务器服务器间连接服务的缺省端口设置

名称	配置属性	协议	缺省端口
TopologyManager 端口	com.ibm.cdb.service.TopologyManager.port	TCP	9550
SecurityManager 端口	com.ibm.cdb.service.SecurityManager.port	TCP	9540
RegistriesURLProvider 端口	com.ibm.cdb.service.RegistriesURLProvider.port	TCP	9560
服务器间服务注册表端口	com.ibm.cdb.service.registry.interserver.port	TCP	4160

下表显示了同步服务器服务器间连接服务的缺省端口设置。

表 *27.* 同步服务器服务器间连接服务的缺省端口设置

名称	配置属性	协议	缺省端口
RegistriesURLProvider 端口	com.ibm.cdb.service.RegistriesURLProvider.port	TCP	9560
EnterpriseSecurityManager 端口	com.ibm.cdb.service.EnterpriseSecurityManager.port	TCP	9570
服务器间服务注册表端口	com.ibm.cdb.service.registry.interserver.port	TCP	4160

本地连接服务

未显式设置本地服务端口。 为本地服务定义的接口上的所有端口必须处于打开状态。 缺省接口为回送接口。

下表显示了域服务器和同步服务器本地连接服务的缺省主机设置。

表 *28.* 域服务器和同步服务器本地连接服务的缺省主机设置

名称	配置属性	缺省接口
本地服务主机	com.ibm.cdb.local.hostname	127.0.0.1

在同步服务器部署中配置通信：

要在同步服务器部署中成功建立通信，请配置公共连接服务、服务器间连接服务以及本地连接服务。

下表显示可在同步服务器部署中进行连接的元素，以及必须打开才能成功进行通信的端口。

服务器间通信

表 *29.* 同步服务器部署中的服务器间连接通信配置。

元素 A	端口	说明	元素 B	配置属性
域服务器			**同步服务器**	
	4160	→	同步服务器 - 服务器间服务注册表	com.ibm.cdb.service.registry.interserver.port
	9560	→	同步服务器 - RegistriesURLProvider	com.ibm.cdb.service.RegistriesURLProvider.port
	9570	→	同步服务器 - EnterpriseSecurityManager	com.ibm.cdb.service.EnterpriseSecurityManager.port
域服务器			**同步服务器**	
域服务器 - 服务器间服务注册表	4160	←	同步服务器	com.ibm.cdb.service.registry.interserver.port
域服务器 - Registries URLProvider	9560	←		com.ibm.cdb.service.RegistriesURLProvider.port
域服务器 - 安全管理器	9540	←		com.ibm.cdb.service.SecurityManager.port
域服务器 - 拓扑管理器	9550	←		com.ibm.cdb.service.TopologyManager.port
数据库服务器	5000	←	**域服务器**	
数据库服务器	5000	←	**同步服务器**	

发现管理门户网站、API 客户机、Web 门户网站和数据管理门户网站客户机、域服务器和同步服务器之间的通信

表 *30.* 发现管理门户网站、*API* 客户机、*Web* 门户网站和数据管理门户网站客户机、域服务器和同步服务器之间的通信。

元素 A	端口	说明	元素 B	配置属性
发现管理门户网站	9433	→	域服务器 - 公共服务注册表	com.ibm.cdb.service.registry.public.port
	9435	→	域服务器 - ClientProxyServer	com.ibm.cdb.service.ClientProxyServer.port
	9434	→	域服务器 - SecureClientProxyServer	com.ibm.cdb.service.SecureClientProxyServer.secure.port
API 客户机	9433	→	• 域服务器 - 公共服务注册表 • 同步服务器 - 公共服务注册表	com.ibm.cdb.service.registry.public.port
	9530	→	• 域服务器 - API 服务器 • 同步服务器 - API 服务器	com.ibm.cdb.service.ApiServer.port
	9531	→	• 域服务器 - 安全 API 服务器 • 同步服务器 - 安全 API 服务器	com.ibm.cdb.service.SecureApiServer.secure.port

表 30. 发现管理门户网站、API 客户机、Web 门户网站和数据管理门户网站客户机、域服务器和同步服务器之间的通信。 (续)

元素 A	端口	说明	元素 B	配置属性
Web 门户网站和数据管理门户网站客户机	9430	→	• 域服务器 - Web • 同步服务器 - Web	com.ibm.cdb.service.web.port
	9431	→	• 域服务器 - 安全 Web • 同步服务器 - 安全 Web	com.ibm.cdb.service.web.secure.port

锚点、网关和域服务器之间的通信

表 31. 锚点、网关和域服务器之间的通信。

元素 A	端口	说明	元素 B	配置属性
锚点（以 SSH 方式）- SSH	22	←	域服务器（以 SSH 方式）	
锚点（以 direct 方式）- SSH		←	域服务器（以 direct 方式）	
锚点（以 SSH 方式）- SSH 通道转发	8497	↔	域服务器（以 SSH 方式）	
锚点（以 direct 方式）- direct		←	域服务器（以 direct 方式）	
网关 - SSH	22	←	域服务器	

本地通信

表 32. 同步服务器部署中的本地连接通信配置。

本地通信	说明	配置属性
域服务器		
域服务器 - 本地服务注册表	↔	com.ibm.cdb.local.hostname
域服务器 - 本地服务		
域服务器 - 127.0.0.1		
同步服务器		
同步服务器 - 本地服务注册表	↔	com.ibm.cdb.local.hostname
同步服务器 - 本地服务		
同步服务器 - 127.0.0.1		

TADDM 服务器属性参考

collation.properties 文件包含 TADDM 服务器的属性。 您可以编辑其中的某些属性。

collation.properties 文件位于 $COLLATION_HOME/etc 目录中。 该文件包含每个属性的注释。

如果更新 collation.properties 文件，必须保存该文件并重新启动服务器，才能使更改生效。

有作用域的属性和无作用域的属性

collation.properties 文件包含两种类型的属性：有作用域的属性和无作用域的属性。

有作用域的属性

可向其附加 IP 地址或作用域集名称的属性。 该 IP 地址或作用域集名称使该属性依赖于要发现的主机。 您只能使用不包含空格、撇号 (')、句点 (.) 和正斜杠 (/) 的作用域集名称。

无作用域的属性

无法限制为特定于某一对象的属性。

例如，以下属性是无作用域的属性：

- com.collation.log.filesize
- com.collation.discover.agent.command.lsof.Linux

但如果向 com.collation.discover.agent.command.lsof.Linux 属性附加 IP 地址或作用域集名称，那么该属性也可以变成有作用域的属性，如以下示例所示：

- 附加 IP 地址 129.42.56.212 的示例：

  com.collation.discover.agent.command.lsof.Linux.129.42.56.212=sudo lsof
- 附加名为"scope1"的作用域集的示例：

  com.collation.discover.agent.command.lsof.Linux.scope1=sudo lsof

不得更改的属性

更改 collation.properties 文件中的某些属性可能会造成系统无法运行。

以下属性不得更改：

com.collation.version

标识产品版本。

com.collation.branch

标识代码分支。

com.collation.buildnumber

标识构建编号。 该编号由构建进程设置。

com.collation.oalbuildnumber

标识另一个构建进程的构建编号。

com.collation.SshWeirdReauthErrorList=Permission denied

此属性的值必须为 Permission denied。

需要此属性的原因是 Windows 系统可能会随机拒绝有效的登录尝试。 可以尝试使用先前在发现运行期间起过作用的用户名和密码对。

访问凭证高速缓存属性

这些属性适用于访问凭证高速缓存。

com.ibm.cdb.security.auth.cache.disabled=false

缺省值为 false。

该属性确定是否禁用凭证高速缓存。

该属性是有作用域并且已简要描述的属性。 您可以附加 IP 地址、作用域集名称或概要文件名称。 在发现管理控制台中，您还可以在概要文件配置中对其进行设置。

com.ibm.cdb.security.auth.cache.fallback.failed=true

缺省值为 true。

当高速缓存包含有效凭证时，该属性开启回退，但是在检索时，它未能验证。如果启用了回退并且高速缓存的凭证不再有效，那么高速缓存循环访问所有可用的访问条目类型，直到找到匹配项。

该属性是有作用域并且已简要描述的属性。 您可以附加 IP 地址、作用域集名称或概要文件名称。

以下条目是可在 collation.properties 文件中找到的条目的示例：

```
com.ibm.cdb.security.auth.cache.fallback.failed=false
com.ibm.cdb.security.auth.cache.fallback.failed.10.160.160.11=true
com.ibm.cdb.security.auth.cache.fallback.failed.ScopeA=true
com.ibm.cdb.security.auth.cache.fallback.failed.GroupA=true
com.ibm.cdb.security.auth.cache.fallback.failed.Level_2_Discovery=false
```

在发现管理控制台中，您还可以在**平台属性**选项卡上的概要文件配置中设置该属性。

com.ibm.cdb.security.auth.cache.fallback.invalid=true

缺省值为 true。

当从高速缓存读取的条目包含无效尝试（最后一次访问失败，没有有效凭证）时，该属性开启回退。 如果启用了回退，那么高速缓存循环访问所有可用的访问条目类型，直到找到匹配项。

该属性是一个有作用域并且已简要描述的属性，您可以附加 IP 地址、作用域集名称或概要文件名称。

以下条目是可在 collation.properties 文件中找到的条目的示例：

```
com.ibm.cdb.security.auth.cache.fallback.invalid=false
com.ibm.cdb.security.auth.cache.fallback.invalid.10.160.160.11=true
com.ibm.cdb.security.auth.cache.fallback.invalid.ScopeA=true
com.ibm.cdb.security.auth.cache.fallback.invalid.GroupA=true
com.ibm.cdb.security.auth.cache.fallback.invalid.Level_2_Discovery=false
```

在发现管理控制台中，您还可以在**平台属性**选项卡上的概要文件配置中设置该属性。

Fix Pack 5 **com.ibm.cdb.security.auth.cache.itm.disabled=true**

缺省值为 true。

该属性确定是否为 OSLC 发现禁用凭证高速缓存。

该属性是有作用域并且已简要描述的属性。 您可以附加 IP 地址、作用域集名称或概要文件名称。 在发现管理控制台中，您还可以在概要文件配置中对其进行设置。

相关概念:

第 12 页的『最近一次成功的凭证高速缓存』

TADDM 可以高速缓存最近一次工作的访问凭证。 在下一次发现（第 2 层发现或者基于脚本的发现）中，可以复用这些凭证。

API 端口属性

这些属性适用于 API 端口。

com.ibm.cdb.service.ApiServer.port=9530

缺省值为 9530。 此值必须为整数。

此属性指定 API 服务器侦听非 SSL 请求的端口。 可将该值设置为服务器上的任何可用端口。 将该 API 用于连接的任何客户机都必须为非 SSL 连接指定该端口。

com.ibm.cdb.service.SecureApiServer.secure.port=9531

缺省值为 9531。 此值必须为整数。

此属性指定 API 服务器侦听 SSL 请求的端口。 可将该值设置为服务器上的任何可用端口。 将该 API 用于连接的任何客户机都必须为 SSL 连接指定该端口。

清除代理程序属性

清理代理程序用于除去孤立的别名和配置项，或者修正表中缺少的行。 大部分清除代理程序读取 collation.properties 文件中定义的属性。

AliasesCleanupAgent

此代理程序用于从 ALIASES 表中除去不再与 CI 命名属性匹配的别名。另外，还将除去 PERSOBJ 表中没有任何相应 CI 的别名和行。 此代理程序从 collation.properties 文件中读取以下属性：

Fix Pack 2 **com.ibm.cdb.topomgr.topobuilder.deleteAliasesWithoutMaster**

缺省值为 true。

该属性指定是否从 ALIASES 表中删除不具有对应的主别名的别名。 缺省情况下，启用删除。

com.ibm.cdb.topomgr.topobuilder.max.row.fetch

缺省值为 1000。

该属性配置用于从 ALIASES 表中访存别名的批量大小。

如果将该属性设置为 -1，那么此代理程序不验证别名。

com.ibm.cdb.topomgr.topobuilder.max.row.delete

缺省值为 5000。

该属性配置用于删除别名的批量大小。

如果将该属性设置为 -1，那么此代理程序不除去别名，而只报告损坏的别名。

com.ibm.cdb.topomgr.topobuilder.agents.AliasesCleanupAgent.maxNumberOfMastersToScan

缺省值为 1000。

该属性配置要求在此代理程序的一次运行期间验证别名的 CI 数目。

com.ibm.cdb.topomgr.topobuilder.cleanupOrphanedAliasesAndPersobj

缺省值为 true。 此代理程序运行清除。

此属性用于允许或禁止清除 ALIASES 表中没有任何相应 CI 的别名以及 PERSOBJ 表中没有任何相应 CI 的 GUID。

com.ibm.cdb.topomgr.topobuilder.DelayToRemoveAliases

缺省值为 12（小时）。 此代理程序将除去存在时间超过 12 小时的孤立别名。

此属性用于定义此代理程序在除去没有相应 CI 的别名前等待的时间，以小时计。 这可以保护由于 CI 存储未完成而没有相应 CI 的新别名。

请谨慎地使用此属性。 请勿将其设置为过小的值。

AliasesJnTableCleanupAgent

该代理程序会从 ALIASES_JN 表中除去旧的行。 该表包含对 ALIASES 表进行的更改的历史记录。 该表用于查找数据库中配置项过度合并的可能性。 此代理程序从 collation.properties 文件中读取以下属性：

Fix Pack 2

com.ibm.cdb.topomgr.topobuilder.agents.AliasesJnTableCleanupAgent.maxRow

缺省值为 5000。 建议不要更改缺省值。

该属性指定了代理程序一次删除的最大行数。

com.ibm.cdb.topomgr.topobuilder.agents.AliasesJnTableCleanupAgent.removeOlderThanDays

缺省值为 30（天）。

此属性会除去存在时间超过指定时间的行。 缺省情况下，它会除去超过 30 天的行。

如果将该属性设置为 0 或更低的值，那么将禁用该代理程序。

com.ibm.cdb.topomgr.topobuilder.agents.AliasesJnTableCleanupAgent.timeout

缺省值为 1800（秒）。

该属性指定时间，代理程序在经过指定的时间之后即超时。 如果指定的时间不足以删除所有旧的行，那么代理程序在下次运行时尝试删除这些行。

DependencyCleanupAgent

此代理程序将除去休眠的 Relationship 对象。 此代理程序从 collation.properties 文件中读取以下属性：

com.ibm.cdb.topomgr.topobuilder.agents.DependencyCleanupAgent.timeout

缺省值为 600 秒。 经过此时间后，代理程序将停止除去对象，即使仍有未除去的对象也是如此。

com.ibm.cdb.topomgr.topobuilder.agents.DependencyCleanupAgent.removeOlderThanDays

缺省值为 90 天。 存在时间超过指定值的 Relationship 对象将被视为休眠对象。

ObjectsWithoutAliasesCleanupAgent

此代理程序用于除去在 ALIASES 表中没有别名的 CI。此代理程序从 collation.properties 文件中读取以下属性：

com.ibm.cdb.topomgr.topobuilder.agents.ObjectsWithoutAliasesCleanupAgent.maxToRemove

缺省值为 1000

该属性用于此限制代理程序在一次运行期间除去的 CI 数目。 如果将该属性设置为 -1，那么此代理程序将退出而不执行任何清理，并且将显示 ObjectsWithoutAliasesCleanupAgent 已禁用消息。

PersobjCleanupAgent

此代理程序用于修正 PERSOBJ 表中所有缺少的行。 它不使用 collation.properties 文件中的任何配置。 此代理程序将显示有关所修正的行数的摘要，如以下示例所示：

```
2012-08-22 18:12:21,500 TopologyBuilder [TopologyBuilderEngineThread$Cleanup@4.0]
INFO agents.PersobjCleanupAgent - Fixed 10 rows in PERSOBJ table
```

StorageExtentCleanupAgent

此代理程序将除去休眠的 StorageExtent 对象。 此代理程序从 collation.properties 文件中读取以下属性：

com.ibm.cdb.topomgr.topobuilder.agents.StorageExtentCleanupAgent.timeout
缺省值为 1800 秒。 经过此时间后，代理程序将停止除去对象，即使仍有未除去的对象也是如此。

com.ibm.cdb.topomgr.topobuilder.agents.StorageExtentCleanupAgent.removeOlderThanDays
缺省值为 1 天。存在时间比其父 ComputerSystems 多 1 天的 StorageExtent 对象将被视为休眠对象。

VlanInterfaceCleanupAgent

此代理程序将除去休眠的 VlanInterface 对象。 此代理程序从 collation.properties 文件中读取以下属性：

com.ibm.cdb.topomgr.topobuilder.agents.VlanInterfaceCleanupAgent.timeout
缺省值为 1800 秒。 经过此时间后，代理程序将停止除去对象，即使仍有未除去的对象也是如此。

com.ibm.cdb.topomgr.topobuilder.agents.VlanInterfaceCleanupAgent.removeOlderThanDays
缺省值为 1 天。存在时间比其父 Vlans 多 1 天的 VlanInterface 对象将被视为休眠对象。

可能需要提升特权的命令

这些属性指定 TADDM 所使用的操作系统命令，这些命令可能需要提升的特权（例如 root 用户或超级用户）才能在目标系统上运行。

通常，sudo 在 UNIX 和 Linux 系统上使用，用于提供特权升级。 可以使用以下备选方法来代替 sudo：

- 在目标可执行程序上启用 setuid 访问权
- 向与目标可执行程序关联的组添加发现服务帐户
- 为发现服务帐户使用 root 用户（不推荐）

对于每个属性，可全局配置 sudo，这意味着在每个操作系统目标上通过 sudo 运行命令，或限制到特定 IP 地址或作用域集。

要点：在每个需要升级特权的目标系统上，必须使用 NOPASSWD 选项配置 sudo。 否则，发现将挂起，直到 sudo 超时。

com.collation.discover.agent.command.hastatus.Linux=sudo /opt/VRTSvcs/bin/hastatus
com.collation.discover.agent.command.haclus.Linux=sudo /opt/VRTSvcs/bin/haclus
com.collation.discover.agent.command.hasys.Linux=sudo /opt/VRTSvcs/bin/hasys
com.collation.discover.agent.command.hares.Linux=sudo /opt/VRTSvcs/bin/hares
com.collation.discover.agent.command.hagrp.Linux=sudo /opt/VRTSvcs/bin/hagrp
com.collation.discover.agent.command.hatype.Linux=sudo /opt/VRTSvcs/bin/hatype
com.collation.discover.agent.command.hauser.Linux=sudo /opt/VRTSvcs/bin/hauser

- 这些属性是发现 Veritas Cluster 组件所必需的。
- 要执行这些命令而不使用 sudo，TADDM 服务帐户必须是目标上的 Veritas 管理组成员。

com.collation.discover.agent.command.vxdisk=vxdisk
com.collation.discover.agent.command.vxdg=vxdg
com.collation.discover.agent.command.vxprint=vxprint
com.collation.discover.agent.command.vxlicrep=vxlicrep
com.collation.discover.agent.command.vxupgrade=vxupgrade

- 这些属性将发现 Veritas 标准存储信息和其他 Veritas 特定信息（如磁盘组、Veritas 卷、plex 和子磁盘）。

com.collation.platform.os.command.ps.SunOS=/usr/ucb/ps axww
com.collation.platform.os.command.psEnv.SunOS=/usr/ucb/ps axwweee
com.collation.platform.os.command.psParent.SunOS=ps -elf -o ruser,pid,ppid,comm
com.collation.platform.os.command.psUsers.SunOS=/usr/ucb/ps auxw

- 这些属性是发现有关 Solaris 系统的进程信息所必需的。

  您可以通过向属性名附加 SunOS 版本号来指定特定 Solaris 版本。 例如，以下属性特定于 Solaris 10：

  ```
  com.collation.platform.os.command.ps.SunOS5.10=sudo /usr/ucb/ps axww
  ```

com.collation.platform.os.command.ps.Linux=ps axww
com.collation.platform.os.command.psEnv.Linux=ps axwweee
com.collation.platform.os.command.psParent.Linux=ps -ax -o ruser,pid,ppid,comm
com.collation.platform.os.command.psUsers.Linux=ps auxw

- 这些属性是发现有关 Linux 系统的进程信息所必需的。

com.collation.platform.os.command.ps.AIX=ps axww
com.collation.platform.os.command.psEnv.AIX=ps axwweee
com.collation.platform.os.command.psParent.AIX=ps -elf -o ruser,pid,ppid,comm
com.collation.platform.os.command.psUsers.AIX=ps auxw

- 这些属性是发现有关 AIX 系统的进程信息所必需的。

com.collation.platform.os.command.ps.HP-UX=sh UNIX95= ps -elfx -o pid,tty,state,time,args
com.collation.platform.os.command.psEnv.HP-UX=ps -eflx
com.collation.platform.os.command.psParent.HP-UX=sh UNIX95= ps -elfx -o ruser,pid,ppid,comm
com.collation.platform.os.command.psUsers.HP-UX=ps -efx

- 这些属性是发现有关 HP-UX 系统的进程信息所必需的。

com.collation.discover.agent.command.lsof.Vmnix=lsof
com.collation.discover.agent.command.lsof.Linux=lsof
com.collation.discover.agent.command.lsof.SunOS.1.2.3.4=sudo lsof
com.collation.discover.agent.command.lsof.Linux.1.2.3.4=sudo lsof
com.collation.discover.agent.command.lsof.HP-UX=lsof
com.collation.discover.agent.command.lsof.AIX=lsof

- 这些属性是发现进程或端口信息所必需的。

  您可以通过向属性名附加 SunOS 版本号来指定特定 Solaris 版本。 例如，以下属性特定于 Solaris 10：

  ```
  com.collation.discover.agent.command.lsof.SunOS5.10=sudo /usr/local/bin/lsof
  ```

com.collation.discover.agent.command.dmidecode.Linux=dmidecode
com.collation.discover.agent.command.dmidecode.Linux.1.2.3.4=sudo dmidecode

- 这些属性是发现 Linux 系统上的制造商、型号和序列号所必需的。

com.collation.discover.agent.command.vmcp.Linux=
该属性可用于发现在 z/VM® 操作系统上运行的目标 Linux 虚拟系统上的访客用户标识。

com.collation.discover.agent.command.cat.SunOS=cat
com.collation.discover.agent.command.cat.SunOS.1.2.3.4=sudo cat

- 该属性是发现 Solaris 系统上检查点防火墙的配置信息所必需的。

com.collation.discover.agent.command.interfacesettings.SunOS=sudo ndd
com.collation.discover.agent.command.interfacesettings.Linux=sudo mii-tool
com.collation.discover.agent.command.interfacesettings.SunOS.1.2.3.4=sudo ndd
com.collation.discover.agent.command.interfacesettings.Linux.1.2.3.5=sudo mii-tool
com.collation.discover.agent.command.interfacesettings.HP-UX=lanadmin
com.collation.discover.agent.command.interfacesettings.AIX=netstat

- 这些属性是发现高级网络接口信息（例如，接口速度）所必需的。

com.collation.discover.agent.command.adb.HP-UX=adb
com.collation.discover.agent.command.adb.HP-UX.1.2.3.4=sudo adb

- 该属性是发现有关 HP 系统的处理器信息所必需的。

com.collation.discover.agent.command.kmadmin.HP-UX=kmadmin
com.collation.discover.agent.command.kmadmin.HP-UX.1.2.3.4=sudo /usr/sbin/kmadmin

- 该属性是发现 HP 系统上的内核模块所必需的。

com.collation.platform.os.command.partitionTableListing.SunOS=prtvtoc

- 该属性是发现有关 Solaris 系统的分区表信息所必需的。

com.collation.platform.os.command.lvm.lvdisplay.1.2.3.4=sudo lvdisplay -c
com.collation.platform.os.command.lvm.vgdisplay.1.2.3.4=sudo vgdisplay -c
com.collation.platform.os.command.lvm.pvdisplay.1.2.3.4=sudo pvdisplay -c

- 这些属性是发现存储卷信息所必需的。

com.collation.platform.os.command.lputil.SunOS.1.2.3.4=sudo /usr/sbin/lpfc/lputil

- 该属性是发现 Solaris 系统上的 Emulex 光纤通道 HBA 信息所必需的。

com.collation.platform.os.command.crontabEntriesCommand.SunOS=crontab -l
com.collation.platform.os.command.crontabEntriesCommand.Linux=crontab -l -u
com.collation.platform.os.command.crontabEntriesCommand.AIX=crontab -l
com.collation.platform.os.command.crontabEntriesCommand.HP-UX=crontab -l

- 这些属性是发现 **crontab** 条目所必需的。可以通过在属性结尾添加 IP 地址或作用域集名称来将这些属性指定为有作用域属性。 以下示例使用了附加的 IP 地址：

  ```
  com.collation.platform.os.command.crontabEntriesCommand.AIX.1.2.3.4=crontab -l
  ```

com.collation.platform.os.command.filesystems.Linux=df -kTP
com.collation.platform.os.command.filesystems.SunOS=df -k | grep -v 'No such file or directory' | grep -v 'Input/output error' | awk '{print $1, $2, $4, $6}'
com.collation.platform.os.command.filesystems.AIX=df -k | grep -v 'No such file or directory' | grep -v 'Input/output error' | awk '{print $1, $2, $3, $7}
com.collation.platform.os.command.filesystems.HP-UX=df -kP | grep -v 'No such file or directory' | grep -v 'Input/output error' | grep -v Filesystem

- 这些属性是发现文件系统所必需的。

com.collation.platform.os.command.fileinfo.ls=sudo ls
com.collation.platform.os.command.fileinfo.ls.1.2.3.4=sudo ls
com.collation.platform.os.command.fileinfo.cksum=sudo cksum
com.collation.platform.os.command.fileinfo.cksum.1.2.3.4=sudo cksum
com.collation.platform.os.command.fileinfo.dd=sudo dd
com.collation.platform.os.command.fileinfo.dd.1.2.3.4=sudo dd

- 这些属性是特权文件捕获所必需的。
- 特权文件捕获在以下情境中使用：发现服务帐号不具备发现所需的应用程序配置文件的读访问权。

com.collation.discover.agent.WebSphereVersionAgent.versionscript=sudo
如果发现用户没有对目标 WebSphere Application Server 系统的访问权，那么可以启用该属性以访问 WebSphere `versionInfo.sh` 文件。

com.collation.platform.os.command.fileinfo.OnlyDirectoryRecursive
此标志更改发现配置文件的方式。缺省值为 `False`。

如果将此标志更改为 `True`，那么此机制不会使用 find 命令以递归方式发现目录内容。

将此标志设置为 False 时，此机制会使用 find 命令以递归方式发现文件，而不指定文件的确切位置。

上下文菜单服务和数据集成服务属性

这些属性适用于上下文菜单服务 (CMS) 和数据集成服务 (DIS)。

com.ibm.cdb.DisCmsIntegration.enabled=true

缺省值为 true。

此属性指定是否启用 CMSDISAgent 拓扑构建器代理程序，以定期更新在上下文菜单服务和数据集成服务数据库中注册的 TADDM 数据。

com.ibm.cdb.DisCmsIntegration.dbUser=*user*

此属性指定上下文菜单服务和数据集成服务数据库的数据库用户标识。

com.ibm.cdb.DisCmsIntegration.dbPassword=*password*

此属性指定上下文菜单服务和数据集成服务数据库用户的密码。

com.ibm.cdb.DisCmsIntegration.dbUrl=*url*

此属性指定上下文菜单服务和数据集成服务数据库的数据库 URL。

com.ibm.cdb.DisCmsIntegration.dbDriver=*driver*

此属性指定上下文菜单服务和数据集成服务的数据库驱动程序。

com.ibm.cdb.DisCmsIntegration.changehistory.days_previous=30

缺省值为 30。

此属性指定变更历史记录显示在上下文菜单服务和数据集成服务的变更报告中的天数。

数据库属性

这些属性适用于 TADDM 数据库。

com.collation.db.password=password

此属性为数据库用户指定数据库密码，该密码存储在 TADDM 服务器上。

com.collation.db.archive.password=password

此属性为数据库归档用户指定数据库密码，该密码存储在 TADDM 服务器上。

com.ibm.cdb.db.max.retries

此属性指定尝试建立数据库连接时重试的次数。

com.ibm.cdb.db.timeout

此属性指定两次重试之间的休眠时间（以毫秒计）。

com.ibm.cdb.db.connection.ssl.enable=false

此属性针对数据库用户指定是否在 SSL 方式下建立数据库连接。

缺省值为 false。

com.ibm.cdb.db.connection.ssl.truststore.file=filename

此属性针对数据库用户指定用于与数据库建立 SSL 连接的信任库文件。 此信任库文件必须位于 $COLLATION_HOME/etc/ 目录中。

com.ibm.cdb.db.connection.ssl.truststore.password=password

此属性针对数据库用户指定用于与数据库建立 SSL 连接的信任库密码。

com.ibm.cdb.db.archive.connection.ssl.enable=false

此属性针对归档数据库用户指定是否在 SSL 方式下建立数据库连接。

缺省值为 false。

com.ibm.cdb.db.archive.connection.ssl.truststore.file=filename
此属性针对归档数据库用户指定用于与数据库建立 SSL 连接的信任库文件。 此信任库文件必须位于 $COLLATION_HOME/etc/ 目录中。

com.ibm.cdb.db.archive.connection.ssl.truststore.password=password
此属性针对归档数据库用户指定用于与数据库建立 SSL 连接的信任库密码。

要对 collation.properties 文件中的数据库密码进行加密，请完成下列步骤：

1. 编辑数据库用户和/或使用明文对用户密码进行归档。
2. 停止 TADDM 服务器。
3. 运行 encryptprops.sh 文件或 encryptprops.bat 文件（位于 $COLLATION_HOME/bin 目录中）。此脚本将对密码进行加密。
4. 重新启动 TADDM 服务器。

发现属性

这些属性普遍适用于发现。 影响特定传感器的 TADDM 服务器属性在各种传感器的 TADDM *Sensor Reference* 中有说明。

Fix Pack 4 **com.discover.anchor.maxChannelNumber**
此属性指定 TADDM 服务器与锚点之间的 SSH 会话中同时打开的最大通道数。如果打开的通道数过大，那么这种锚点上的发现可能会挂起，并且这种作用域中包含的传感器可能会超时。在这种情况下，请使用此属性来控制打开的通道数。

缺省值为 50。

Fix Pack 4 **com.collation.platform.os.copyToLocal.preferScpCommand**
此属性指定是否使用外部命令 **scp** 将文件从远程主机（通常是发现目标）复制到 TADDM 服务器。外部命令 **scp** 是在 com.collation.platform.os.scp.command 属性中定义的。要支持使用外部命令 **scp**，请将此属性设置为 true。

此属性的缺省值为 false。

注：此属性仅适用于使用基于密钥的登录建立的 SSH 会话（请参阅第 102 页的『使用安全外壳协议 (SSH) 来配置发现』）。对于使用用户名和密码进行认证的情况，将使用内部命令 **scp**，而无论 com.collation.platform.os.copyToLocal.preferScpCommand 属性值如何。

此属性具有作用域。您可以将 IP 地址或作用域集的名称附加到此属性。例如：

```
com.collation.platform.os.copyToLocal.preferScpCommand.12.234.255.4=true
```

com.collation.platform.os.scp.command
此属性指定指向操作系统 **scp** 命令的路径。当内部 SSH 客户机未能在 TADDM 服务器与远程主机（通常是发现目标）之间发送文件时，可以使用此属性。您也可以使用备用命令，但它必须与 **scp** 命令具有相同的语法。

示例值：/usr/local/bin/scp。

Fix Pack 3 **com.collation.platform.session.ssh.winAuth**

此属性指定在使用 SSH 会话时是否尝试使用 Windows 凭证进行登录。 缺省值为 true。

如果在尝试使用 Windows 凭证登录到非 Windows 服务器的发现期间存在风险，那么可以将值设置为 false。 这可以避免锁定 Windows Active Directory 帐户。

Fix Pack 3 **com.collation.platform.os.ignoreL2InterfaceDescription**

此属性指定计算机系统签名计算期间想要忽略的已发现的 L2Interfaces 的描述。例如，如果不想要使用 Microsoft Load Balancer Interface 来计算计算机系统的签名，请指定以下值：

```
com.collation.platform.os.ignoreL2InterfaceDescription=Microsoft Load
Balancer Interface
```

此属性的值被作为正则表达式。 这意味着您可以添加多个接口描述，并且无需使用任何分隔符，例如，逗号。

Fix Pack 3 **com.ibm.cdb.topomgr.topobuilder.agents.ConnectionDependencyAgent2.dependencyPlaceholders**

如果设置为 true，那么此属性针对未发现的依赖关系创建占位符"应用程序服务器"。

注：缺省情况下，不会将此属性包含在 collation.properties 文件中。 您必须在此手动添加。

在第一次将值置为 true 时，必须重新启动 TADDM 以针对 LogicalConnection 和 SSoftwareServer 类启用扩展属性。 这些扩展属性是此功能正常运行所必需的。

有关占位符的更多信息，请参阅第 114 页的『针对占位符发现进行配置』。

com.collation.platform.session.EncodingOverRide

此属性指定在发现会话期间使用的编码类型。当目标服务器所使用的编码与 TADDM 服务器中的编码不同时，这尤其有用。

此属性的值是编码的名称，例如 UTF-8。缺省情况下，此属性未包括在 collation.properties 文件中，您必须将其添加到该文件中。

您还可以将作用域或 IP 地址添加到此属性。例如：

```
com.collation.platform.session.EncodingOverRide.37.53.105.24=UTF-8
```

com.collation.discover.anchor.forceDeployment=true

缺省值为 true。

此属性指定是否要在发现启动期间部署已发现的作用域的锚点。

如果将该值设置为 false，那么仅当满足以下任一条件时才部署锚点：

- 如果作用域中的任何 IP 地址都无法通过 ping 进行访问
- 如果任何已发现的 IP 地址上的端口 22 都无法访问

如果链式锚点存在，那么此条件适用于链中的所有锚点。 如果链中的某个锚点通过某个条件进行限制，那么该锚点之前的所有锚点都必须满足此条件，才能部署所有锚点。

com.collation.discover.anchor.lazyDeployment=false

缺省值为 false。

此属性指定复制传感器所需文件的条件，值为 false 表示在部署锚点时复制，值为 true 表示在即将启动需要这些文件的传感器时复制。

例如，IBM WebSphere 传感器在 dist/lib/websphere 目录中具有依赖关系。该目录的大小为 130 MB。 如果该属性的值为 false，那么依赖关系数据将在部署锚点时复制到目标主机。 如果值为 true，那么将在锚点上即将运行 WebSphere 传感器时复制该数据。 如果不通过该锚点运行任何 WebSphere 传感器，130 MB 的数据将不会发送到远程主机。

com.collation.discover.DefaultAgentTimeout=600000

该值为 600000（以毫秒为单位），即 10 分钟。

此属性指定了传感器的超时数（以毫秒为单位）。 不应更改此缺省超时数。 您可以改为指定个别传感器的超时数。

要覆盖特定传感器的超时，请将下行添加到collation.properties 文件中：

```
com.collation.discover.agent.sensorNameSensor.timeout=
  timeInMilliseconds
```

例如：

```
com.collation.discover.agent.OracleSensor.timeout=1800000
```

com.collation.IpNetworkAssignmentAgent.defaultNetmask=*ip_start-ip_end/netmask*[, ...]

该属性定义在第 1 层发现期间发现的 IP 地址如何分配给生成的子网。 第 1 层发现不会发现子网。 相反，它将生成 IpNetwork 对象来包含不与在第 2 层或第 3 层发现期间发现的现有子网相关联的任何接口。 该配置属性定义要创建的 IpNetwork 对象以及每个子网要包含的节点数量。 （它还应用于在第 2 层或第 3 层发现期间发现的但由于某种原因无法分配到发现的子网的任何接口。）

该属性的值是包括一个或多个以逗号分隔的条目的单个行。 每个条目描述一个使用 IPv4 点分十进制格式的 IP 地址范围，同时将子网掩码指定为 8 到 31 之间的整数。 然后，指定范围中发现的接口将放入不超过子网掩码所指定大小的已创建子网中。

例如，以下值定义两个子网地址范围以及不同的子网掩码：

```
9.0.0.0-9.127.255.255/23, 9.128.0.0-9.255.255.255/24
```

指定的地址范围可重叠。 如果发现的 IP 地址与多个定义的范围相匹配，那么会在属性值中列出子网时，将该地址分配到第一个匹配的子网中。

创建或更改此配置属性并重新启动 TADDM 服务器之后，任何后续的第 1 层发现将使用定义的子网。 要在 TADDM 数据库中重新分配现有的 IpInterface 对象，请转至 $COLLATION_HOME/bin 目录并运行以下某个命令：

- **adjustL1Networks.sh**（Linux 和 UNIX 系统）
- **adjustL1Networks.bat**（Windows 系统）

如果未正确指定该值，那么仅当运行命令行实用程序 **adjustL1Networks.sh**（Linux 和 UNIX 系统）或 **adjustL1Networks.bat**（Windows 系统）时才会显示相应消息。 否则，消息

位于 $COLLATION_HOME/log/services 目录的 TopologyBuilder.log 文件中，以及 $COLLATION_HOME/log/agents 目录的 IpNetworkAssignmentAgent.log 文件中。

此脚本会将第 1 层发现期间发现的所有 IpInterface 对象重新分配到相应的子网，如配置属性中所述。 然后会从数据库中删除不包含任何接口的任何生成的 IpNetwork 对象。 脚本完成后，TADDM 接口可能会由于对象修改而显示有关组件更改的多个通知。 可通过刷新窗口来清除这些通知。

注：使用此命令之前，请确保 TADDM 服务器正在运行并且当前未在执行发现或批量装入操作。 同步服务器上不支持此脚本。

com.collation.number.persist.discovery.run=30

缺省值为 30。

指定在数据管理门户网站和发现管理控制台中的发现历史记录中保存信息的发现数。

要在流式服务器部署中更改缺省值，请在主存储服务器上输入新值。

com.collation.platform.os.hostappdescriptorfiles.dir="*path*"

为部署了计算机系统（主机）的组件应用程序描述符文件的目录指定其标准路径。如果要使用应用程序描述符向业务应用程序添加计算机系统，该属性是必需的。 可以将该属性的作用域限定为特定主机名或 IP 地址，从而为每个主机指定不同的位置。 以下示例显示如何指定主机应用程序描述符路径：

- Linux 和 UNIX 系统：/home/taddm/hostappdescriptors
- Windows 系统：c://taddm//hostappdescriptors

com.collation.platform.session.GatewayForceSsh

指定是否强制网关独立于锚点操作。 有效值为 *true* 和 *false*。如果网关和锚点位于同一个系统上，请将该值设置为 true 以解决 Cygwin 问题。 该值设置为 true 时，在网关和锚点之间会使用 SSH 会话而非本地会话来传输通信。

com.collation.rediscoveryEnabled=false

缺省值为 false。

该属性适用于对已发现的配置项进行重新发现。 数据管理门户网站中提供了重新发现功能。

限制： 重新发现不能使用定制概要文件中的凭证，它使用全局列表中的凭证。

注：

要在域服务器部署中启用重新发现，请在域服务器上将该值设置为 true。

要在流式服务器部署中启用重新发现，请在发现服务器和存储服务器上将该值设置为 true。

流式服务器部署中的重新发现

在流式服务器部署中使用重新发现时，配置项可由不同的发现服务器发现，但只有最后发现该配置项的发现服务器才能对该配置项进行重新发现。 由于存在多台发现服务器，因此每台发现服务器都会覆盖配置项的重新发现信息。

在发现服务器上启用重新发现时，将为每个已发现的对象创建有关重新发现的其他信息。

在存储服务器上启用重新发现时，会将有关重新发现的其他信息存储在每个已发现的对象中。

如果在发现服务器上启用了重新发现，但在存储服务器在禁用了重新发现，那么将无法在 TADDM 数据库中获取有关重新发现的信息。 此外，还必须确保将相同的凭证用于发现服务器和存储服务器。

com.ibm.cdb.discover.sensor.sys.utilization.workingdir=/tmp/taddm

缺省值为 /tmp/taddm。

此属性指定要在目标系统上运行的 IBM Tivoli Utilization 传感器脚本的根路径。 如果未指定该值，那么将使用 com.ibm.cdb.taddm.script.path 属性定义的路径。

com.ibm.cdb.locationTag

指定 TADDM 服务器上创建的各个配置项 (CI) 的位置标记属性。 位置标记属性（标识 CI 的位置）用于支持静态位置标记。 指定此标记之前，必须将 com.ibm.cdb.locationTaggingEnabled 值设置为 true。

com.ibm.cdb.locationTaggingEnabled=false

缺省值为 false。

指定是否启用了位置标记功能。 将此属性的值设置为 true，以执行以下操作：

- 指定 TADDM 服务器上创建的各个配置项 (CI) 的位置标记属性（静态位置标记）。 请参阅 com.ibm.cdb.locationTag 属性以获取详细信息。
- 使用命令行界面 (CLI) 为单一发现期间创建的配置项 (CI) 指定动态位置标记。动态位置标记将覆盖已存在的位置标记（静态位置标记）。
- 为使用批量装入程序装入数据时创建的配置项 (CI) 指定动态位置标记。
- 在运行 BIRT 报告时指定位置标记值，以过滤数据并且仅报告与该指定位置相关的信息。
- 为拓扑构建过程中创建的配置项创建位置标记值。

com.ibm.cdb.taddm.host

指定 TADDM 服务器主机别名。 如果未指定该值，那么将使用系统主机名。如果 TADDM 服务器无法解析系统主机名或者系统主机名解析为 localhost，那么必须以手动方式指定此属性。

com.ibm.cdb.taddm.script.path=/tmp/taddm

缺省值为 /tmp/taddm。

此属性指定要在目标系统上运行的传感器脚本的根路径。 在此位置中，将使用 host_alias/discovery_number/sensor_name 格式创建一个子目录树。host_alias 名称从 com.ibm.cdb.taddm.host 属性中检索。如果未指定此属性，那么将使用系统主机名。 要区分同一发现服务器上的并发发现，必须对 discovery_number 目录指定一个数字。此发现脚本和发现结果将使用此目录结构进行存储。

com.collation.discover.agent.signature.ignore.*1.2.3.4*=true

此属性用于在签名计算期间跳过 IP 地址。

在某些配置中，计算机系统的签名可能不是唯一的，从而导致与 TADDM 数据库中的现有条目进行协调时发生问题。例如，当您使用的虚拟机的虚拟网卡具

有有效的硬件地址和 IP 地址时，可能会发生这种情况。 在此类情况下，必须排除签名计算并使用其他命名规则，例如 ProductModel、Manufacturer 或 SerialNumber。

对于每个要忽略的 IP 地址，请添加 com.collation.discover.agent.signature.ignore.*1.2.3.4*=true 属性，其中 *1.2.3.4* 是要忽略的 IP 地址。

如果要忽略多个 IP 地址，您可以创建发现作用域。 请向 collation.properties 文件添加 com.collation.discover.agent.signature.ignore.*blacklist*=true 属性，其中 *blacklist* 是包含所有要忽略的 IP 地址的发现作用域。

高级发现属性：

高级发现属性指定用于存储工作项的缓冲区容量、特定发现元素的重新启动次数或者将统计信息输出到日志时的时间值。除非必须周密地调整发现进程，否则请勿更改这些属性。

com.ibm.cdb.discover.buffers.workitem.capacity=64

缺省值为 64。 但是，此值始终是 com.collation.discover.dwcount 值（缺省情况下为 32）的两倍。

此属性指定用于存储发现工作项的缓冲区容量。 此属性用于限制发现进程的内存需求，从而避免发生 OutOfMemory 错误。 对于每一次发现，都将启动新的传感器。

请勿将此值设置为小于 com.collation.discover.dwcount 中指定的发现工作程序数，否则将会导致某些工作程序保持空闲状态。

com.ibm.cdb.discover.buffers.workitem.maxresets=10

缺省值为 10。

此属性指定发生意外故障（例如，负责执行发现的 TADDM JVM 发生故障）时传感器可以重新启动的次数。

另外，发现进程的元素的重新启动次数还受 com.ibm.cdb.discover.runrestartlimit（用于指定发现重新启动次数）限制。

com.ibm.cdb.discover.buffers.seed.capacity=100

缺省值为 100。

此属性指定用于存储种子工作项的缓冲区容量。 此属性用于限制发现进程的内存需求，从而避免发生 OutOfMemory 错误。

com.ibm.cdb.discover.buffers.result.capacity=100

缺省值为 100。

此属性指定用于存储结果工作项的缓冲区容量。 此属性用于限制发现进程的内存需求，从而避免发生 OutOfMemory 错误。 对于每个结果工作项，可以启动新传感器。

请将此值设置为与 com.ibm.cdb.discover.buffers.discovered.capacity 大小相同。

com.ibm.cdb.discover.buffers.result.maxresets=10

缺省值为 10。

此属性指定发生意外故障（例如，负责执行发现的 TADDM JVM 发生故障）时发现进程可以为结果工作项启动新传感器的次数。

另外，发现进程的元素的重新启动次数还受 com.ibm.cdb.discover.runrestartlimit（用于指定发现重新启动次数）限制。

com.ibm.cdb.discover.buffers.discovered.capacity=100

缺省值为 100。

此属性指定用于存储已发现的工作项的缓冲区容量。 每个已发现的工作项都表示一个存储在数据库中的发现结果。

请勿将此值指定为小于 com.collation.discover.observer.topopumpcount 中指定的数据库写线程数。

com.ibm.cdb.discover.buffers.statistics.interval.seconds=60

缺省值为 60。 请指定以秒计的值。

此属性指定将发现缓冲区统计信息保存到日志时的时间值。 日志为 /log/services/DiscoveryState.log。

com.ibm.cdb.discover.buffers.timeout.interval.seconds=600

缺省值为 600，即 10 分钟。请指定以秒计的值。

此属性指定检查工作项是否超时的时间值。

com.ibm.cdb.discover.runcontroller.statistics.interval.seconds=60

缺省值为 60。 请指定以秒计的值。

此属性指定将发现运行统计信息保存到日志时的时间值。 日志为 /log/services/DiscoveryRunController.log。

com.ibm.cdb.discover.runrestartlimit=11

缺省值为 11。

此属性指定发生故障后未初始化的发现可以重新启动的次数。 如果尚未针对发现作用域的所有元素启动发现进程，那么发现处于未初始化状态。

com.collation.discovery.oracle.tablelimit=1000

缺省值为 1000。 该属性只支持正值。

该属性控制 Oracle 传感器发现的表数量。

并发发现属性：

这些属性适用于并发发现。

com.collation.discover.concurrent.discovery=true

缺省值为 true。

该属性用于启用并发发现。

com.collation.discover.max.concurrent.discoveries=10

缺省值为 10。

该属性定义并发发现的最大数量。

异步发现属性：

这些属性适用于异步发现。

com.ibm.cdb.discover.asd.AsyncDiscoveryResultsDirectory=var/asdd

缺省值为 var/asdd，它相对于 com.collation.home 目录。

该属性定义 TADDM 服务器上包含异步发现结果的归档文件的根目录位置。位置可以是相对路径或绝对路径。 相对路径相对于 com.collation.home 目录。

com.ibm.cdb.discover.asd.ProcessUnreachableIPs=false

缺省值为 false。

该属性用于启用在异步发现中使用的不可达 IP 地址的处理。 要启用这些地址的处理，将值设置为 true。

com.ibm.cdb.tarpath=tar

缺省值为 tar。

在异步发现中，此属性指定 *TADDM* 服务器上 **tar** 命令的路径。

在诸如 AIX 或 Linux 之类的操作系统上，通常不需要该属性，因为 **tar** 命令已安装且可用。 但是，要在运行 Windows 操作系统的 TADDM 服务器上生成异步发现脚本包或处理发现归档文件，您必须安装第三方 tar 程序并指定该程序的完整路径名。

以下示例显示如何在 AIX 操作系统的 TADDM 服务器上指定 **tar** 命令的路径：

```
com.ibm.cdb.tarpath=tar
```

com.ibm.cdb.targettarpath=tar

缺省值为 tar。

该属性指定异步发现中目标系统上 **tar** 命令的路径。

在诸如 AIX 或 Linux 之类的目标操作系统上，通常不需要该属性，因为 **tar** 命令已安装且可用。 但由于文件名的长度限制，要在 Solaris 操作系统上生成发现归档文件，您必须使用 gtar 归档实用程序，并且必须指定该实用程序的路径。

以下示例显示了如何根据操作系统指定目标系统上 **tar** 命令的路径：

对于 AIX

```
com.ibm.cdb.targettarpath.AIX=tar
```

对于 Solaris

```
com.ibm.cdb.targettarpath.SunOS=/usr/sfw/bin/gtar
```

基于脚本的发现属性：

这些属性适用于基于脚本的发现。

Fix Pack 4 **com.ibm.cdb.discover.enableOutputFileSplittingProcess=true**

缺省值为 true。

该属性指定在基于脚本的发现期间创建的主要输出文件是否拆分为更小的文件。缺省情况下，此文件将会进行拆分。此设置可防止在输出文件很大时出现性能问题。另请参阅 com.ibm.cdb.discover.numberOfLinesForOutputFileSplittingProcess 属性。

Fix Pack 4

com.ibm.cdb.discover.numberOfLinesForOutputFileSplittingProcess=10000

缺省值为 10000。

仅当 com.ibm.cdb.discover.enableOutputFileSplittingProcess 属性设置为 true 时，才启用该属性。
该属性指定通过拆分主要输出文件所创建的较小输出文件中允许的大致行数。确切的行数由文件格式确定。在指定的行数之后，仅当到达有意义的数据集末尾以确保整个文件格式正确时，才拆分该文件。这意味着，当该值设置为 10000 时，较小的文件可具有，例如 10200 行。

com.ibm.cdb.taddm.asd.prefix=sh

缺省值为 sh。

此属性指定要添加到发现期间运行的脚本的前缀，例如 *prefix* script.sh。此属性是一个有作用域的属性，您可以追加作用域集的 IP 地址或名称。

com.ibm.cdb.discover.DeleteScriptDiscoveryOutputs=true

缺省值为 true。

此属性指定是否要删除在基于脚本的发现期间复制到 TADDM 服务器以供传感器处理的脚本输出。该输出可能有助于故障诊断，但在缺省情况下，将在发现完成后删除该输出。 如果将此属性的值设置为 false，那么就不会删除脚本输出。

com.ibm.cdb.discover.DeleteRemoteBeforeScriptsRun=false

缺省值为 false。

此属性指定 TADDM 在尝试运行新发现之前，是否从远程目录中除去上一次发现所留下的任何输出。

com.ibm.cdb.discover.PreferScriptDiscovery=false

缺省值为 false。

此属性用于启用基于脚本的发现，且它只对支持基于脚本的发现的传感器有影响。 将该值设置为 true 就会启用基于脚本的发现。

com.ibm.cdb.discover.smallFileSizeLimit=1048576

缺省值为 1048576 (1024*1024 - 1 MB)。

此属性定义将要触发使用校验和的复制操作文件大小限制（以字节为单位）。对于大小小于此限制的文件，将在不计算校验和的情况下进行复制。 对于大小等于或大于此限制的文件，仅当它们不存在于目标目录中且其校验和与本地（源）文件不匹配时，才会进行复制。

可以使用下列值来禁用限制：

- 0 - 复制操作始终使用校验和。
- -1 - 复制操作始终避免使用校验和。

使用 *IBM Tivoli Monitoring*（老方法）的发现属性：

这些属性适用于使用 IBM Tivoli Monitoring（老方法）的发现。

旧的集成方法

该部分说明的是不推荐使用的 TADDM 与 IBM Tivoli Monitoring 集成方法. 从 TADDM V7.3.0 开始，建议使用 OSLC 自动化将 TADDM 与 IBM Tivoli Monitoring 6.3 集成。 不推荐使用 IBM Tivoli Monitoring Scope 传感器进行集成，在后续发行版中将移除这个老方法。 有关用于通过 OSLC 自动化来配置发现过程的属性的更多信息，请参阅 第 172 页的『通过 OSLC 自动化将 TADDM 与 IBM Tivoli Monitoring 相集成』和 第 72 页的『对使用 OSLC 自动化会话的发现应用的属性』。

影响 TADDM 如何发现 Tivoli Monitoring 端点的属性

TADDM 第 2 层和第 3 层发现通常需要有域服务器（在域服务器或同步服务器部署中）或发现服务器（在流式服务器部署中）才能使用以下方法之一直接连接到目标系统：

- 用于基于 UNIX 的目标系统的安全 Shell (SSH)
- 用于 Windows 系统的 Windows Management Instrumentation (WMI)

要使用这些方法，域服务器或发现服务器必须知道用户凭证（帐户和密码）。

使用 IBM Tivoli Monitoring 的发现允许 TADDM 发现没有用户凭证可用的目标系统的第 2 层信息（以及部分第 3 层信息）。 传感器通过 Tivoli Monitoring 基础结构运行，仅需要 Tivoli Enterprise Portal Server 的凭证。 缺省情况下，在配置并运行 IBM Tivoli Monitoring Scope 传感器之后，未来的第 2 层发现使用 Tivoli Monitoring 进行发现。 因为您可能不希望您的环境中出现这一缺省行为，所以 TADDM 提供了以下服务器属性来控制使用 Tivoli Monitoring 还是使用直接连接（SSH 或 WMI）进行发现。 可以在全局层面设置这些属性，也可以针对特定作用域或发现概要文件设置。

com.ibm.cdb.session.allow.ITM=true

缺省值为 true，表示 TADDM 可以使用 IBM Tivoli Monitoring 来发现 Tivoli Monitoring 端点。

此属性指定 TADDM 是否可以使用 IBM Tivoli Monitoring 来发现 Tivoli Monitoring 端点。

要直接连接到 Tivoli Monitoring 端点，请将该值设置为 false。

您还可以使用此属性指定定制发现作用域，如以下示例所示：

com.ibm.cdb.session.allow.ITM.*ip_address*=false

以下示例指定 TADDM 使用发现作用域 10.20.30.40，而且即使受 Tivoli Monitoring 监视也应直接连接到端点：

```
com.ibm.cdb.session.allow.ITM.10.20.30.40=false
```

com.ibm.cdb.session.prefer.ITM=true

缺省值为 true，表示 TADDM 使用 IBM Tivoli Monitoring 来发现 Tivoli Monitoring 端点。

此属性指定如果端点允许使用 IBM Tivoli Monitoring 的发现，TADDM 是否使用 IBM Tivoli Monitoring 作为首选方法来发现 Tivoli Monitoring 端点。 如果 TADDM 使用 IBM Tivoli Monitoring 进行发现但发现不成功，则 TADDM 会直接连接端点。 同样，如果使用 IBM Tivoli Monitoring 的发现

不是首选方法，但直接连接端点不成功，则 TADDM 会尝试使用 IBM Tivoli Monitoring 连接端点，当然前提是端点允许使用 IBM Tivoli Monitoring 的发现。

您还可以使用此属性指定定制发现作用域，如以下示例所示：

com.ibm.cdb.session.prefer.ITM.*ip_address*=false

以下示例指定 TADDM 使用发现作用域 10.20.30.40，而且直接连接 Tivoli Monitoring 端点：

```
com.ibm.cdb.session.prefer.ITM.10.20.30.40=false
```

com.ibm.cdb.session.prefer.ITM.Level_3_Discovery=false

缺省值为 false，表示如果您使用第 3 层发现概要文件，TADDM 将直接连接 Tivoli Monitoring 端点，但对于所有其他发现层次，TADDM 将根据以下属性的值，使用 IBM Tivoli Monitoring 发现 Tivoli Monitoring 端点：

- **com.ibm.cdb.session.allow.ITM**
- **com.ibm.cdb.session.prefer.ITM**

此属性指定如果您使用第 3 层发现概要文件，TADDM 是否使用 IBM Tivoli Monitoring 来发现 Tivoli Monitoring 端点。

如果将该值设置为 true，则 TADDM 可以使用 IBM Tivoli Monitoring 来发现来自第 3 层发现概要文件的 Tivoli Monitoring 端点。

用于调整 TADDM 服务器和门户网站服务器之间连接的属性

对于 IBM Tivoli Monitoring 第 2 层发现，如果 TADDM 服务器和 Tivoli Enterprise Portal Server 之间的连接有延迟，TADDM 会使用以下 TADDM 服务器属性来调整连接恢复行为：

com.collation.discover.agent.ITM.CmdWrapperSelectionPattern=

此属性指定在通过 IBM Tivoli Monitoring 环境运行发现时，必须由脚本回绕的命令。

com.collation.platform.session.ITMSessionConnectionCooldownPeriod=60000

此属性指定在检测到故障后，重新初始化 Tivoli Enterprise Portal Server 连接之前应等待的时间间隔，单位为毫秒。

com.collation.platform.session.ITMSessionConnectionRetryLimit=5

此属性指定如果初始连接失败，在报告错误前应尝试访问连接的次数。

com.collation.platform.session.ITMSessionNumProgressChecks=600

此属性指定在连接失败前，应检查连接进度的次数。

com.collation.platform.session.ITMSessionProgressCheckInterval=1000

此属性指定在每次连接进度检查之间的时间间隔，单位为毫秒。

对使用 *OSLC* 自动化会话的发现应用的属性：

这些属性应用于使用 OSLC 自动化会话的发现。

与通过 OSLC 进行集成相关的属性

com.ibm.cdb.topobuilder.integration.oslc.automationprovider

此属性指定未在 Jazz SM Registry Services 中注册的 OSLC Execute Automation Service Provider 的直接 URL 地址。

以下示例说明用于 ITM 的 OSLC Execute Automation Service Provider 的 URL 地址：

```
com.ibm.cdb.topobuilder.integration.oslc.automationprovider=
http://<AUTOMATION_PROVIDER_INSTALLATION_HOST>:15210/itmautomationprovider
```

以下示例说明如何指定多个 OSLC Execute Automation Service Provider 的 URL 地址：

```
com.ibm.cdb.topobuilder.integration.oslc.automationprovider.1=
http://9.1.1.1:15210/itmautomationprovider
com.ibm.cdb.topobuilder.integration.oslc.automationprovider.2=
http://9.2.2.2:15210/itmautomationprovider
```

com.ibm.cdb.topobuilder.integration.oslc.automation.scope.alwaysrefresh =false

缺省值为 false。

此属性是一个全局属性，用于指定 OSLCAutomationAgent 是否在每次运行期间重建作用域集。 重建作用域集要求连接到 Jazz SM Registry Services 和/或 OSLC Execute Automation Service Provider。

如果此属性设置为 true，那么代理程序将重建作用域集，即使 OSLC Execute Automation Service Provider 所提供的自动化计划自代理程序上次运行以来未曾变更也是如此。

com.ibm.cdb.topobuilder.integration.oslc.frsurl

此属性指定通过 OSLC 与其他产品进行集成时使用的 Jazz SM Registry Services (FRS) IP 地址。 Jazz SM Registry Services 地址必须采用以下格式：

```
protocol://ip_or_hostname:port
```

此属性还由 OSLCAgent 使用。

com.ibm.cdb.topobuilder.integration.oslc.automation.frsurl

此属性以 Jazz SM Registry Services (FRS) 注册集合的完整路径形式指定 IP 地址。当 Jazz SM Registry Services 使用除缺省的 /oslc 以外的其他服务路径时，可能会使用此属性。

与使用自动化会话的发现相关的属性

com.ibm.cdb.session.oslcautomation.pluginId=com.ibm.cdb.session.oslcautomation_1.0.0

缺省值为 com.ibm.cdb.session.oslcautomation_1.0.0。

此属性指定 OSLC 自动化会话插件的 OSGi 捆绑软件标识。

com.ibm.cdb.session.itm.endpointClass=com.collation.platform.session.oslcautomation.OSLCAutomationEndpoint

缺省值为 com.collation.platform.session.oslcautomation.OSLCAutomationEndpoint。

此属性指定要使用的端点类。

com.ibm.cdb.session.allow.OSLCAutomation=true

缺省值为 true。

此属性具有作用域，用于指定 TADDM 在发现期间能否使用 OSLC 自动化会话。

用法示例：

```
com.ibm.cdb.session.allow.OSLCAutomation=true
com.ibm.cdb.session.allow.OSLCAutomation.9.100.1.0=true
com.ibm.cdb.session.allow.OSLCAutomation.scope_set2=true
```

com.ibm.cdb.session.prefer.OSLCAutomation=true

缺省值为 true。

此属性具有作用域，用于指定 OSLC 自动化会话是否为用于发现的首选会话。此属性的值优先于任何其他首选值，例如标准 ITM 会话。

用法示例：

```
com.ibm.cdb.session.prefer.OSLCAutomation=true
com.ibm.cdb.session.prefer.OSLCAutomation.9.100.100.200=true
com.ibm.cdb.session.prefer.OSLCAutomation.scope_name1=true
```

com.ibm.cdb.session.oslcautomation.timeout.httpconnect=60000

缺省值为 60000（60 秒）。此值以毫秒为单位。

此属性是一个全局属性，用于指定 OSLC Execute Automation Service Provider 连接超时值。

com.ibm.cdb.session.oslcautomation.timeout.httpread=240000

缺省值为 240000（4 分钟）。此值以毫秒为单位。

此属性是一个全局属性，用于指定从 OSLC Execute Automation Service Provider 中读取数据时的超时值。

com.ibm.cdb.session.oslcautomation.request.async.maxretries=60

缺省值为 60。

此属性是一个全局属性，用于指定针对异步生成的 AutomationResult 发出的连续请求的最大数目。

com.ibm.cdb.session.oslcautomation.request.async.delay=10000

缺省值为 10000（10 秒）。此值以毫秒为单位。

此属性是一个全局属性，用于指定针对异步生成的 AutomationResult 发出的连续请求之间的延迟时间。

注： Fix Pack 4 为了防止连接服务器的 SSH 会话由于超时问题失败，请尝试为以下属性配置最优值：

com.collation.mindterm.Ssh2Preferences= hello-timeout=30; alive = 25; compression= 9

com.collation.discover.agent.app.packagedapp.mysap.SLDServerPortList = 51200

该属性允许更改 SLD 端口而且指定的端口将会添加到传感器配置中。

com.ibm.cdb.security.auth.cache.itm.disabled=true

缺省值为 true。

该属性确定是否为 OSLC 发现禁用凭证高速缓存。

该属性是有作用域并且已简要描述的属性。 您可以附加 IP 地址、作用域集名称或概要文件名称。 在发现管理控制台中，您还可以在概要文件配置中对其进行设置。

DNS 查找定制属性

这些属性适用于 DNS 查找定制。

com.collation.platform.os.disableDNSLookups=false

缺省值为 false。

有效值为 true 或 false。 如果将此属性更改为 true，那么将对 TADDM 服务器禁用 DNS 查找。

com.collation.platform.os.disableRemoteHostDNSLookups=false

缺省值为 false。

有效值为 true 或 false。 如果将此属性更改为 true，发现的远程主机上将禁用名称查找（仅 DNS）。 此属性强制在 TADDM 服务器上进行所有名称查找。

com.collation.platform.os.command.fqdn=nslookup $1 | grep Name | awk '{print $2}' 缺省值为 nslookup $1 | grep Name | awk '{print $2}'。

该命令用于查找标准域名 (fqdn)。 在大多数情况下，无需此属性，因为缺省的标准域名 (FQDN) 算法在大多数生产环境中都可以使用。 如果不需要此属性，必须将其注释掉。 但是，在标准域名派生自主机名的环境中，可以启用此属性。 例如，如果将主机名配置为 DNS 中的别名，那么请启用此属性。

如果使用此属性，请确保 DNS 可用且配置正确。 否则，**nslookup** 命令可能失败，或响应时间缓慢。

如果启用，此属性将仅在 TADDM 服务器上使用。 目前，仅支持 AIX 和 Linux 操作系统。 Windows TADDM 服务器上不支持此属性。

GUI 属性

这些属性仅适用于 TADDM GUI。

Fix Pack 3 **com.ibm.cdb.gui.supportedJRE.warning=true**

此属性指定当启动"发现管理控制台"时，是否显示警告消息 CTJTG0034E。 此消息将会警告您 Java 运行时环境的版本不受支持。 如果要将 TADDM 和不受支持版本的 Java 运行时环境配合使用，且您不希望显示此消息，请将此属性设置为 false。

此属性的缺省值为 true。

GUI JVM 内存属性：

这些属性适用于 GUI JVM 内存。

com.collation.gui.initial.heap.size=128m

缺省值为 128m。TADDM 用户界面的初始堆大小。

com.collation.gui.max.heap.size=512m

缺省值为 512m。TADDM 用户界面的最大堆大小。

这些属性适用于小型 TADDM 域。 出于缩放大小的目的，将使用 TADDM 服务器的以下类别（基于等效服务器）：

- 小型：最多 1000 台等效服务器
- 中型：1000 到 2500 台等效服务器
- 大型：2500 到 5000 台等效服务器

增加中型和大型环境的这些值将提高某些 GUI 操作的性能。 如果操作时 TADDM 的可用内存不足，那么某些视图将无法正常完成。

对于中型环境：

com.collation.gui.initial.heap.size=256m

缺省值为 256m。

com.collation.gui.max.heap.size=768m

缺省值为 768m。

对于大型环境：

com.collation.gui.initial.heap.size=512m

缺省值为 512m。

com.collation.gui.max.heap.size=1024m

缺省值为 1024m。

***GUI* 端口属性：**

这些属性适用于 GUI 端口。

com.collation.tomcatshutdownport=9436（仅适用于 TADDM 7.3.0）

缺省值为 9436。

该端口用于 Tomcat 关闭命令。

com.ibm.cdb.service.web.port=9430

缺省值为 9430。

该 HTTP 端口不与 SSL 一起使用。

com.ibm.cdb.service.web.secure.port=9431

缺省值为 9431。

该 HTTPS 端口与 SSL 一起使用。

com.ibm.cdb.service.ClientProxyServer.port=9435

缺省值为 9435。

不与 SSL 一起使用的 RMI 数据端口。

com.ibm.cdb.service.SecureClientProxyServer.secure.port=9434

缺省值为 9434。

与 SSL 一起使用的 RMI 数据端口。

com.ibm.cdb.service.registry.public.port=9433

缺省值为 9433。

公共服务注册表端口。

LDAP 属性

这些属性适用于 LDAP。

可将外部 LDAP 服务器用于用户认证。外部 LDAP 服务器同时支持匿名认证和基于密码的认证。

LDAP 服务器主机名、端口号、基本专有名称、绑定专有名称和密码（基于密码的认证需要）可在 collation.properties 文件中配置。 也可以配置可用于搜索以匹配用户标识（UID）的特定命名属性。

在同步和域服务器部署中建议使用 LDAP 配置。 在企业环境中，将域服务器和同步服务器配置为使用相同的用户注册表。 登录到与同步服务器相连接的域服务器时，将在同步服务器上处理此登录。 如果在同步服务器和域服务器之间出现网络连接问题，但域服务器配置为使用与同步服务器相同的用户注册表，那么您无需重新配置就可以成功登录域服务器。

com.collation.security.auth.ldapAuthenticationEnabled=true

缺省值为 true。

此属性用于启用 LDAP 认证。

com.collation.security.auth.ldapBaseDN=ou=People,dc=ibm,dc=com

缺省值为 ou=People,dc=ibm,dc=com。

此属性定义 LDAP 基本专有名称 (DN)。 LDAP 基本专有名称是所有 LDAP 搜索的起点。

com.collation.security.auth.ldapBaseGroupDN

在 collation.properties 文件中，缺省情况下会注释掉此属性。

此属性定义用于搜索组的 LDAP 根分支，它可以不同于所有 LDAP 查询的根分支。 要指定多个 LDAP 根目录分支来搜索组，请使用";"字符将分支名称隔开。

如果没有为此属性指定值，那么缺省值是 com.collation.security.auth.ldapBaseDN 属性的值。

com.collation.security.auth.ldapBindDN=uid=ruser,dc=ibm,dc=com

缺省值为 uid=ruser,dc=ibm,dc=com。

如果使用简单认证，此属性将定义用于认证到 LDAP 的用户标识。

要点：

- 如果未提供 com.collation.security.ldapBindDN 的值或者已注释掉此属性，那么将尝试匿名连接到 LDAP。 以下示例显示了如何使用井号 (#) 注释掉此属性：

  ```
  #com.collation.security.auth.ldapBindDN=uid=ruser,
  dc=ibm,dc=com
  ```

- 如果为 com.collation.security.auth.ldapBindDN 指定了值，那么将使用简单认证，
- 并且还必须指定 com.collation.security.auth.ldapBindPassword 的值。

com.collation.security.auth.ldapBindPassword=ruser

缺省值为 ruser。

如果使用简单认证，此属性将定义用于认证到 LDAP 的用户密码。

com.collation.security.auth.ldapClientKeyStore=*ks_path*

此属性定义 TADDM 服务器上包含证书的密钥库的位置。 密钥库必须包含用于向 LDAP 服务器认证 TADDM 服务器的客户机证书。

com.collation.security.auth.ldapClientKeyStorePassphrase=*ks_passphrase*

可选：此属性定义密钥库的密码。

com.collation.security.auth.ldapClientTrustStore=*ts_path*

此属性定义 TADDM 服务器上包含证书的信任库的位置。 信任库必须包含 LDAP 服务器证书。

com.collation.security.auth.ldapClientTrustStorePassphrase=*ts_passphrase*

可选：此属性定义信任库的密码。

com.collation.security.auth.ldapGroupMemberAttribute=member

缺省值为 member。

此属性定义用于包含 LDAP 中的组成员的属性名称。

com.collation.security.auth.ldapGroupNamingAttribute=cn

缺省值为 cn。

此属性定义用于命名 LDAP 中的用户组的属性名称。

com.collation.security.auth.ldapGroupObjectClass=groupofnames

缺省值为 groupofnames。

此属性定义用于在 LDAP 中表示用户组的类。

com.collation.security.auth.ldapHostName=ldap.ibm.com

缺省值为 ldap.ibm.com。

此属性定义 LDAP 服务器的主机名。

com.collation.security.auth.ldapPortNumber=389

缺省值为 389。

此属性定义 LDAP 服务器的端口。

com.collation.security.auth.ldapUIDNamingAttribute=uid

缺省值为 uid。

此属性定义用于命名 LDAP 中的用户的属性名称。

com.collation.security.auth.ldapUserObjectClass=person

缺省值为 person。

此属性定义用于在 LDAP 中表示用户的类名。

com.collation.security.auth.ldapUseSSL=false

缺省值为 false。

此属性用于通过 SSL 连接启用对 LDAP 用户注册表的认证。

com.collation.security.usermanagementmodule=ldap

缺省值为 ldap。

此属性定义 TADDM 服务器使用的用户管理模块。 有效值为：

- file，表示基于文件的用户注册表。 缺省值为 true。
- ldap，表示 LDAP 用户注册表
- vmm，表示使用 WebSphere Application Server 联合存储库的用户注册表

锁定属性

这些属性适用于锁定。

com.collation.security.lockout.treshold=3

缺省值为 3。

此属性指定触发特定用户的本地锁定之前该用户的失败登录尝试次数。

com.collation.security.lockout.timeout=30

缺省值为 30。

此属性指定在触发了本地锁定时，触发了本地锁定的用户无法登录 TADDM 的时间（以分钟计）。

com.collation.security.lockout.globalthreshold=100

缺省值为 100。

此属性指定同时触发了全局锁定的单一用户锁定数。

com.collation.security.lockout.globaltimeout=30

缺省值为 30。

此属性指定在触发了全局锁定时，所有用户无法登录 TADDM 的时间（以秒为单位）。

com.collation.security.lockout.failedloginthreshold=1000

缺省值为 1000。

此属性指定触发了全局锁定的非重复用户的失败登录尝试总数。

日志记录属性

这些属性适用于日志记录。

com.collation.log.filesize=20MB

缺省值为 20MB。

日志文件的最大大小。 如果文件达到这个大小限制，将创建一个新的日志文件。 当前日志文件将以 *.N* 文件扩展名保存。 *N* 为数字 1 到 **com.collation.log.filecount** 属性中设置的值。可以使用 **com.collation.log.filecount** 属性设置这些文件循环之前可创建和保留的文件数。

可直接输入字节数，或者分别使用 KB 和 MB 指定千字节数或兆字节数。

以下示例是有效的日志文件大小值：

- 1000000
- 512 KB
- 10 MB

com.collation.log.filecount=5

缺省值为 5。

维护的日志文件数。

com.collation.log.level.vm.*vmName*=INFO

缺省值为 INFO。

设置每个虚拟系统的日志级别。

vmName 是与 TADDM 服务名称关联的 Java 虚拟系统。 以下列表提供了其他有效选项：

- Topology
- DiscoverAdmin
- EventsCore
- Proxy
- Discover
- EcmdbCore
- StorageService
- DiscoveryService

以下列表提供了其他有效选项：

- FATAL
- ERROR
- WARNING
- INFO
- DEBUG（设置 DEBUG 选项会降低系统性能。）
- TRACE（设置 TRACE 选项会导致记录密码。）

性能属性

这些属性适用于 TADDM 性能。

com.collation.discover.dwcount=32

缺省值为 32。 此值必须为整数。

此属性将影响发现速率。 发现工作程序线程是运行传感器的线程。 此属性指定可以同时运行的发现工作程序线程数，并且仅适用于流式服务器部署中的发现服务器或域服务器部署中的域服务器。

对于使用 IBM Tivoli Monitoring（使用 IBM Tivoli Monitoring Scope 传感器的老方法）的发现，必须将值设置为 16。对于所有其他发现类型，有效的值范围为 32 - 160。

com.collation.discover.observer.topopumpcount=16

缺省值为 16。 此值必须为整数。

此属性影响将发现结果存储在 TADDM 数据库中的速率。 它指定创建的用于与 TADDM 数据库进行通信的写程序线程的数目。

对于流式服务器部署中的发现服务器，此属性控制发现服务器用于向存储服务器池发送发现结果的线程数。

对于流式服务器部署中的存储服务器，此属性控制用于接收来自发现服务器的发现结果的线程数。

对于域服务器部署中的域服务器，此属性控制用于接收来自发现工作程序线程的发现结果的线程数。

于是，线程将使用连接池中的数据库连接与 TADDM 数据库进行通信（例如，以存储结果和检索数据）。 如果不存在更多使用池的 JDBC 连接，那么线程将创建不使用池的连接。

com.ibm.cdb.discover.observer.topopump.threshold=0.7

com.ibm.cdb.discover.observer.topopump.threshold.*topo_agent_grp_name*=0.7

缺省值为 0.7。 值必须是浮点常数。

此属性指定在拓扑代理程序处于运行状态时可以启动的数据库写程序线程的比例。 您可以分别为特定代理程序组指定阈值，也可以一次为所有这些组指定阈值。 如果未为代理程序组定义该值，那么将使用一般的阈值。 该值允许在拓扑代理程序处于运行状态时限制在 TADDM 数据库中存储发现结果的线程数。

com.ibm.cdb.typesServiceRefreshInterval=120

缺省值为 120。 最小值为 30，最大值为 1800。

此属性指定创建定制查询、显示变更历史记录或显示组件比较信息时更新组件类型的刷新时间间隔（以秒为单位）。

com.ibm.cdb.ea.metaRefreshFrequency=20

缺省值为 20。 此值必须为整数。

此属性指定更新有关已定义的扩展属性（例如存储服务器中的扩展属性）的信息时采用的刷新时间间隔（以秒为单位）。

安全 Shell (SSH) 属性

这些属性适用于安全 Shell (SSH)。

Fix Pack 1 **com.ibm.cdb.platform.SshVersionSessionSkipList**

该属性指定没有为其建立会话的 SSH 服务器的版本。对于这种服务器，会话传感器将会正常完成，而不会发生任何故障。

该属性的值是逗号分隔列表，例如 Cisco,Data ONTAP,SSH-2.0-OpenSSH_5.9 PKIX FIPS,OpenSSH_OA。

com.collation.SshLogInput=false

缺省值为 false。

有效值为 true 或 false。 如果将该值设置为 true，将记录 SSH 输入。

com.collation.SshPort=22

缺省值为 22。 此值必须为整数。

该属性指示服务器用于所有 SSH 连接的端口。

com.collation.SshSessionCommandTimeout=120000

缺省值为 120000。 此值必须为整数。

该值指示允许 SSH 命令运行的时间（毫秒）。 如果从代理程序使用该属性，那么该属性的值必须小于 **AgentRunnerTimeout** 属性的值才有效。

com.collation.SshWeirdReauthErrorList=Permission denied

该属性允许重试之前在发现运行期间已经使用过的用户名和密码对。需要该属性的原因是 Windows 系统会随机拒绝有效登录尝试。 该属性需要具有 Permission denied 设置。 请不要更改该属性。

com.collation.WmiInstallProviderTimeout=240000

缺省值为 240000。 此值必须为整数。

该值指示允许等待初始 WMI InstallProvider 脚本运行的时间（毫秒）。

com.collation.SshSessionReuseSuppressList

某些 SSH 服务器版本不支持复用 TADDM 所实现的连接。 必须将不支持复用的 SSH 服务器版本添加到此属性，这样 TADDM 才能成功发现运行这些 SSH 服务器版本的目标。

该属性的值是逗号分隔列表。仅指定 SSH 服务器版本的开头部分就已足够，例如 SSH-2.0-BoKS_SSH_6。

您可以在会话传感器日志文件中找到 SSH 服务器版本。

安全性属性

这些属性适用于安全性。

Fix Pack 3 **com.ibm.cdb.secure.server=false**

缺省值为 false。

此属性指定公共和外部 RMI 注册程序提供的所有 TADDM 服务是否安全。 如果设置为 true，那么会将所有不安全的公共服务（ClientProxyServer 和 API 服务器）移至内部 RMI 注册程序。 另外，将在外部服务上实施 SSL 协议，例如，RegistriesURLProvider、SecurityManager 和 TopologyManager。

如果将此属性设置为 true，那么还要将 com.collation.security.enablesslforconsole 和 com.collation.security.enforceSSL 属性设置为 true。

此属性可能影响与其他通过不安全连接来连接到 TADDM 的产品进行的集成。

如果修改此属性的缺省值，那么在以下位置进行设置：

- $COLLATION_HOME/dist/etc/collation.properties
- $COLLATION_HOME/dist/sdk/etc/collation.properties
- 每个 TADDM SDK 安装的 sdk/etc/collation.properties。

Fix Pack 5 如果服务器以安全方式运行 (com.ibm.cdb.secure.server = true)，将使用 SSL 协议保护以下端口：

- com.ibm.cdb.service.registry.public.port (Default Value:9433)

如果服务器以安全方式运行 (com.ibm.cdb.secure.server = true)，那么必须在启动数据管理控制台时选中"建立安全 (SSL) 会话"复选框。

Fix Pack 1 **com.ibm.cdb.secure.liberty=false**

缺省值为 false。

有效值为 true 或 false。 要禁用不安全的 HTTP 端口，请将此标志设置为 true。

com.collation.security.privatetruststore=true

缺省值为 true。

有效值为 true 或 false。 在启用 SSL 时，值必须为 true。

com.collation.security.enablesslforconsole=true

缺省值为 true。

有效值为 true 或 false。

com.collation.security.enabledatalevelsecurity=false

缺省值为 false。

有效值为 true 或 false。 要按用户或用户组限制对 TADDM 对象集合的访问权，请将该值设置为 true。

com.collation.security.enforceSSL=false

缺省值为 false。

有效值为 true 或 false。 要禁用不安全的连接并强制使用 SSL 连接，请将该标志设置为 true。

com.collation.security.usermanagementmodule=file

缺省值为 file。

该属性有三个选项：

- file，表示 TADDM 基于文件的用户注册表
- ldap，表示 LDAP 用户注册表
- vmm，表示使用 WebSphere Application Server 联合存储库的用户注册表

com.collation.security.auth.sessionTimeout=240

缺省值为 240。 此值必须为整数。

com.collation.security.auth.searchResultLimit=100

缺省值为 100。 此值必须为整数。

如果有多个用户，请使用该属性。

要点：如果 LDAP 或 WebSphere 联合存储库中的用户超过 100 个，请增加该值，以支持预期的用户数。 例如，com.collation.security.auth.searchResultLimit=150

com.collation.security.auth.websphereHost=localhost

缺省值为 localhost。

输入托管 WebSphere Application Server 的联合存储库功能的系统的标准域名。

com.collation.security.auth.websperePort=2809

缺省值为 2809。

它必须是整数值。 该值指示 WebSphere 系统端口。

com.ibm.cdb.service.SecurityManager.port=9540

对于同步服务器以外的服务器：

缺省值为 9540。

指定安全管理器使用的防火墙端口。

对于同步服务器：
未设置缺省值。
域通过使用 **com.collation.EnterpriseSecurityManager.port** 参数中指定的端口与同步服务器进行通信。 此属性的缺省值为 19433。

com.collation.cdm.analytics.authorizedRole=

可以至允许特定角色使用"分析"窗格。 在缺省情况下，未在 collation.properties 文件中定义此属性，并且"分析"窗格可供所有人使用。属性值必须是允许访问窗格的角色的名称。

可以只允许指定的角色访问"分析"窗格的下列区域：

- Fix Pack 2 分组模式
- 库存摘要
- 应用程序摘要
- 服务摘要
- 系统库存
- 软件服务器目录
- BIRT 报告

com.collation.security.discoverOutsideScope=true

缺省值为 true。

有效值为 true 或 false。 要禁止发现不在作用域内的元素，请将此标志设置为 false。

com.ibm.cdb.secure.tomcat=false（仅适用于 TADDM 7.3.0）

缺省值为 false。

有效值为 true 或 false。 要禁用不安全的 HTTP 端口，请将此标志设置为 true。

com.ibm.cdb.http.ssl.protocol=TLS

缺省值为 TLS。

此属性修改 Web SSL 端口（HTTPS 端口）所使用的 SSL 协议，缺省情况下为 9431。 您可以使用 com.ibm.cdb.service.web.secure.port 属性来设置端口。

要获取受支持的值的列表，请参阅 IBM Java 7 文档 (http://www-01.ibm.com/support/knowledgecenter/SSYKE2_7.0.0/com.ibm.java.security.component.70.doc/security-component/jsse2Docs/protocols.html)。 如果您要使用最安全的协议（例如 TLS V1.1 或 TLS V1.2），那么必须先将 Web 浏览器配置为支持这些协议。 另外，过强的协议可能会影响与其他通过 Web SSL 端口连接到 TADDM 的产品进行的集成。

Fix Pack 5 当客户端使用 com.ibm.cdb.http.ssl.protocol=TLSv1.2 和 JAVA7 时，需要更新以下设置：

```
<JAVA_HOME>/jre/lib/security/java.security
jdk.tls.disabledAlgorithms=SSLv2, SSLv3, TLSv1, TLSv1.1
```

而且还应该在浏览器中禁用 TLSv1 和 TLSv1.1。

com.ibm.cdb.ssl.protocol=TLS

缺省情况下，不会将此属性添加到 collation.properties 文件。 如果未添加，那么缺省值为 TLS。 要修改，请使用新值手动将此属性添加到 collation.properties 文件。

此属性修改以下端口使用的 SSL 协议：

- API 服务器侦听 SSL 请求的端口，缺省情况下为 9531。 您可以使用 com.ibm.cdb.service.SecureApiServer.secure.port 属性来设置端口。
- 与 SSL 一起使用的 RMI 数据端口，缺省情况下为 9434。 您可以使用 com.ibm.cdb.service.SecureClientProxyServer.secure.port 属性来设置端口。

要获取受支持的值的列表，请参阅 IBM Java 7 文档 (http://www-01.ibm.com/support/knowledgecenter/SSYKE2_7.0.0/com.ibm.java.security.component.70.doc/security-component/jsse2Docs/protocols.html)。 如果您要使用最安全的协议（例如 TLS V1.1 或 TLS V1.2），那么必须先将 Web 浏览器配置为支持这些协议。 另外，过强的协议可能会影响与其他通过列出的端口连接到 TADDM 的产品进行的集成。

com.ibm.cdb.http.ssl.ciphers=

将为 LibertyServer 设置密码，而且仅借助给定的密码进行通信。否则，它将选择可能为弱算法的缺省密码。

com.ibm.cdb.rmi.ssl.protocol=

com.ibm.cdb.rmi.ssl.protocol 属性有助于对采用 com.ibm.cdb.ssl.protocol 创建的 SSL 连接启用特定协议。

com.ibm.cdb.rmi.ssl.protocol 必须包含在 com.ibm.cdb.ssl.protocol 受支持的协议列表中。

com.ibm.cdb.rmi.ssl.ciphers=

借助此属性，您可以为 RMI 数据端口和 API 服务器侦听的端口设置密码算法。

临时目录属性

这些属性适于使用临时目录。

临时目录由 TADDM 用于在某些情况下存储临时文件。 例如，锚点日志文件、发现脚本、发现结果以及运行发现时某些传感器所需的信息都可以存储在临时目录中。TADDM 使用三个临时目录：ANCHOR_DIR、ASD_TEMP_DIR 和 TADDM_TEMP_ROOT。

com.ibm.cdb.taddm.anchor.root=.

缺省值为 .\。

此条目指定部署了锚点服务器的 ANCHOR_DIR 目录的位置。 此属性具有作用域，您可以对此属性追加 IP 地址、作用域名称或操作系统。 例如，com.ibm.cdb.taddm.anchor.root.SunOS=。

对于 Windows 系统，将使用以下属性名和缺省值：

com.ibm.cdb.taddm.anchor.root.Windows=%windir%\\temp\\

此属性值使用在目标主机上解析的变量。 Linux、AIX 和 SunOS 变量必须以美元符 ($) 为前缀。 Windows 变量必须括在百分号 (%) 内。 例如，com.ibm.cdb.taddm.anchor.root=$TMP/taddmdirs/anchor 和 com.ibm.cdb.taddm.anchor.root.Windows=%TEMP%\\taddmdirs\\anchor。

如果解析后的属性值是相对目录路径，那么它将具有以下前缀：

- %windir%\\temp\\ - 对于 Windows
- 主目录 - 对于 AIX、Linux 和 SunOS 系统

此路径以 taddmversion/anchor 目录为后缀。例如，/home/taddmusr/taddm7.2.1/anchor 和 c:\Windows\Temp\taddm7.2.1\anchor。

com.ibm.cdb.taddm.asd.temp

此属性指定 ASD_TEMP_DIR 目录的位置，该目录存储发现脚本和发现结果。该属性是有作用域的属性，可以通过将 IP 地址或操作系统附加到该属性来定制该属性。

将在指定的位置创建 taddm*version*/asd/ 目录。 例如，/tmp/taddm7.2.1/asd/。如果指定新位置，所有用户都必须拥有该新位置的所有访问权。

com.ibm.cdb.taddm.file.temp=.

缺省值为 .\

此条目指定 TADDM_TEMP_ROOT 的位置，该目录由各种传感器用于存储运行发现所需的临时数据。 存储临时数据的示例传感器是 DB2® 和 WebLogic 传感器。

TADDM_TEMP_ROOT 目录在主目录的 taddm*version*/temp/ 中创建。例如，/home/taddmusr/taddm7.2.1/temp/。

拓扑构建器属性

这些属性适用于拓扑构建器。

com.collation.topobuilder.RuntimeGcUnknownServerRetentionSpan=5

缺省值为 5。

此属性指定未知进程应保留的时间（以天为单位）。 最大值为 14。 未知进程决定何时需要定制服务器模板，然而不做定期清除，未知进程的数量会随着时间的推移而增加。 这可能会导致拓扑性能问题。 此处理并未除去 zOS 地址空间项。

com.collation.topobuilder.RuntimeGcThreadCount=

缺省值为 4。

此属性用于对 RuntimeGC 代理程序添加并行性，这有助于提高性能。

com.collation.topobuilder.agent.DerivedAppToAppDependencyAgent.ServiceDependency.enabled

缺省值为 false。

当业务应用程序的成员处于服务依赖关系中时，此属性指定拓扑代理程序 DerivedAppToAppDependency 是否创建业务应用程序之间的依赖关系。

要允许此代理程序创建此依赖关系，请将属性设置为 true。

拓扑管理器属性

这些属性适用于拓扑管理器。

com.ibm.JdoQuery.FetchBatchSize=500

缺省值为 500。

批处理大小是可配置的属性并且对应于 **kodo.FetchBatchSize** 属性。 该属性代表在查询运行的结果集中滚动时可访存的行数。

com.ibm.cdb.service.TopologyManager.port=9550

缺省值为 9550。

指定拓扑管理器使用的防火墙端口。

视图管理器属性

这些属性适用于视图管理器。

Fix Pack 2 **com.ibm.taddm.hideNetworkConnectionUnusedColumns.enabled**

缺省值为 `false`。

此属性指定是否在"数据管理门户网站"中显示**网络连接**选项卡的以下列：

- **流**
- **包**
- **八位元**
- **第一次查看**
- **最后一次查看**

要隐藏这些列，请将此属性设置为 `true`。

com.collation.view.maxnodes=500

缺省值为 `500`。 此值必须为整数。

此属性指定在数据管理门户网站的拓扑图形中可查看的最大节点数。如果将此属性设置为较高的值，可以查看更大的拓扑。但是，可能会增大内存需求。

验证数据完整性

您可以通过运行 **verify-data** 命令来验证 TADDM 数据库中的配置项的数据完整性。您可以验证关系、继承映射、重复项和过度合并。

开始之前

请勿运行启用了修复选项的发现、批量装入或同步。 数据完整性工具需要分析大量数据，完成此过程可能需要一些时间，在启用了修复选项的情况下尤其如此。 TADDM 服务器必须已启动并处于运行状态，但是请确保未执行任何任务。

关于此任务

数据完整性验证工具报告并修复 TADDM 数据库中的配置项数据完整性问题。可执行文件脚本位于 `$COLLATION_HOME/bin` 目录中。 此工具在 `verify-data.log` 文件中报告并记录验证结果。 您随时可以停止此工具以及再次运行此工具。

验证关系

关系验证操作将查询并验证所有模型表和交叉表中的外键。

关于此任务

如果启用了修复选项，那么关系验证操作将在数据库中不存在父对象时除去子对象，并清除定义为未包含的关系的无效外键值。 另外，还可能删除大量的低级别配置项。 但是，如果这些项没有父对象，那么可以安全地将其除去。

过程

要验证关系，请运行下列其中一个命令：

- **verify-data.sh -v ro [-a repair]**
- **verify-data.bat -v ro [-a repair]**

验证继承映射

继承映射验证操作将查询所有映射了配置项类的表，并验证是否所有的表对于每一行都包含一个条目。

关于此任务

如果启用了修复选项，那么将重新创建这些记录。

过程

要验证继承映射，请运行下列其中一个命令：

- **verify-data.sh -v io [-a repair]**
- **verify-data.bat -v io [-a repair]**

验证重复项

重复项验证操作根据数据库中的命名规则字段值来搜索重复的配置项。

关于此任务

在启用了修复选项的情况下，重复的对象将合并。 完成合并后，持久对象将保留在数据库中，而瞬时对象将被删除。

合并由多个线程以并行方式执行。 缺省线程数为 5。您可以在 collation.properties 文件中更改线程数，方法是将 **com.ibm.cdb.topomgr.dataverification.generator.ThreadCount** 标志设置为适当的数字，如以下示例所示：

- **com.ibm.cdb.topomgr.dataverification.generator.ThreadCount=10**

在更改线程数之后，必须重新启动 TADDM 服务器。

合并对象期间可能会发生一些错误。 错误原因将记录在日志文件中。

- ERROR_INVALID_DURABLE_GUID
- ERROR_INVALID_TRANSIENT_GUID

错误原因是 ALIASES 表中缺少别名或者对象无效。 您必须等待清除代理程序删除无效记录。

过程

要验证重复项，请运行下列其中一个选项：

- **verify-data.sh -v dup [-a repair]**
- **verify-data.bat -v dup [-a repair]**

验证过度合并

过度合并验证操作使用 ALIASES_JN 表中收集的数据来查找并报告存在大量主别名更改的 GUID。

关于此任务

ALIASES_JN 表包含对 ALIASES 表进行的更改的历史记录。 过度合并是指多个对象将其父代更改为同一个模型对象时的情况。 这时候，子对象围绕一定数目的父对象进行

集群。 在安装 TADDDM 7.2.1 FP3 以前发生的过度合并不会被找到，这是因为，在 ALIASES_JN 表中没有所需的数据。 验证操作没有修复选项，这是因为它可能会错误地找到并报告正面结果。

缺省情况下，对"ComputerSystem"、"AppServer"和"操作系统"类，以及继承自这些类的所有其他类启用详细信息跟踪。 如果要对其他类启用跟踪，可以在 collation.properties 文件中编辑以下属性：

```
com.ibm.tivoli.namereconciliation.service.overmergeClasses
```

以下是一个属性示例，指定该属性可查找"ComputerSystem"、"AppServer"和"操作系统"类：

```
com.ibm.tivoli.namereconciliation.service.overmergeClasses=
ComputerSystem,AppServer,OperatingSystem
```

下列用于运行命令的操作的含义：

- "s1s2s1 - Verification"是指查找以循环方式更改其命名属性值的 CI。 例如，如果计算机系统的签名为 A，然后改为 B，之后又改回 A，该计算机系统会被检测到。
- "s1s2s3 - Verification"是指查找对给定的命名属性更改了若干次的 CI。
- "m1m2m1 - Verification"是指查找 GUID 对其主 GUID 更改了多次的 CI。 例如，别名 A 的主 GUID 为 B，然后指定为 C，之后又再次指定为 B，该别名会被检测到。
- "m1m2m3 - Verification"是指查找 GUID 对其主 GUID 更改了几次的 CI。
- "WinCSLinCSWinCS - Verification"是指查找对其类型更改了几次的 CI。 例如，计算机系统最初存储为 WindowsComputerSystem，然后更新为 LinuxUnitaryComputerSystem，之后又再次更新为 WindowsComputerSystem，该计算机系统会被检测到。

过程

要验证过度合并，请运行下列其中一个命令：

- **verify-data.sh -v om [-a <action>] [-p <class>] [-from <time stamp>] [-to <time stamp>]**
- **verify-data.bat -v om [-a <action>] [-p <class>] [-from <time stamp>] [-to <time stamp>]**

  其中：

  - **<action>**：s1s2s1、s1s2s3、m1m2m1、m1m2m3 和 WinCSLinCSWinCS
  - **<class>**：TADDM 模型中的任何类，例如，ComputerSystem。
  - **<time stamp>**：YYYY-MM-DD HH24:MI:SI 格式的时间戳记。

示例

```
verify-data.sh -v om -a s1s2s1 m1m2m1 WinCSLinCSWinCS
-p ComputerSystem -from 2012-11-13 14:50:00 -to 2012-11-14 14:50:01
```

该命令会查找在 2012-11-13 14:50:00 和 2012-11-14 14:50:01 期间创建的、对"ComputerSystem "类以及继承自它的所有类的类型 s1s2s1、m1m2m1 和 WinCSLinCSWinCS 过度合并的情况。

解决过度合并问题：

当多个对象将其父代更改为同一个模型对象时，就会发生过度合并。 这时候，子对象围绕一定数目的父对象进行集群。

过程

1. 运行过度合并验证。
2. 检查报告的配置项。 验证操作可能会不正确地将这些配置项报告为过度合并。
3. 更正环境中可能是过度合并原因的配置。 配置问题可能包括签名、序列号、VMID 和其他 CI 命名属性相同。
4. 从 TADDM 数据库中除去过度合并的对象。
5. 对删除的对象运行发现，并验证结果。
6. 在解决所有过度合并问题后，除去 ALIASES_JN 表中的所有记录。

管理凭证高速缓存 - cachemgr 实用程序

您可以使用 **cachemgr.sh** 或 **cachemgr.bat** 命令来列出和删除凭证高速缓存的内容。

命令语法

cachemgr -h | **-u** *user* **-p** *password* (**-l**|**-r**) *valid*|*invalid*|*all* [[**-s** *IP*|*scope*|*scope group*|*range*|*subnet*] [**-a** *addressSpace*] [**-n** *accessCredentialName*] [**-c** *type*] [**-d** *yyyy/mm/dd*] [**-k** *key*] [**-t** *locationTag*]]

参数

-a, --addressSpace *addressSpace*
地址空间名称。

-c, --class *type*
执行访问条目的特定类名称描述的所选访问条目的类型。

-d, --date *yyyy/mm/dd*
用于选择在指定时间前未修改的条目的日期阈值。 格式为 yyyy/mm/dd。

-h, --help
显示帮助。

-k, --key *key*
所选高速缓存条目的密钥。

-l, --list *valid*|*invalid*|*all*
受以下参数控制的列表操作：

- *valid* - 只列出高速缓存中的有效认证尝试。
- *invalid* - 只列出高速缓存中的无效认证尝试。
- *all* - 同时列出高速缓存中的有效和无效认证尝试。

-n, --name *accessCredentialName*
访问凭证的名称，与访问列表中的相同。

-p, --password *password*
登录 TADDM 服务器的用户的密码。

-r, --remove *valid|invalid|all*
受以下参数控制的除去操作：

- *valid* - 只除去高速缓存中的有效认证尝试。
- *invalid* - 只除去高速缓存中的无效认证尝试。
- *all* - 同时除去高速缓存中的有效和无效认证尝试。

-s, --scope *IP|scope|scope group|range|subnet*
访问条目的作用域。 受以下参数控制：

- *IP*
- *scope*
- *scope group*
- *range*
- *subnet*

-t, --locationTag *locationTag*
所选访问条目的位置标记。

-u, --username *username*
登录 TADDM 服务器的用户。

示例

- 以下命令列出了作用域 "ScopeSet" 中的计算机的所有无效认证尝试：

```
cachemgr.sh -u user -p password -l invalid -s ScopeSet
```

此命令生成以下输出：

```
以下条目符合所提供的条件：
CachedAuthEntry
 GUID：3B954CE4CFBF346C8DF538F09F1F7FFD
 密钥串：SSH!9.128.109.144!!com.collation.platform.security.auth.HostAuth!null!
 上次修改时间：2013 年 9 月 5 日，星期四 11:00:38
 授权：无效。 错误消息：CTJTP1190E 服务器未完成
授权过程。
CachedAuthEntry
 GUID：ACC2F35A66D3379BAC13FC606C5A08A3
 密钥串： SSH!9.128.109.145!!com.collation.platform.security.auth.HostAuth!null!
 上次修改时间：2013 年 9 月 5 日，星期四 11:00:38
 授权：无效。 错误消息：CTJTP1190E 服务器未完成
授权过程
```

- 以下命令删除 IP 范围 9.123.149.10 - 9.123.149.12 内的无效认证尝试以及访问条目 com.collation.platform.security.auth.HostAuth：

```
cachemgr.sh -u user -p password -r invalid -s 9.123.149.10-9.123.149.12
-c com.collation.platform.security.auth.HostAuth
```

此命令生成以下输出：

```
AuthEntries removed from cache successfully (2).
```

Cachemgr 实用程序返回码

如果写入 cron 脚本或调用 cachemgr 实用程序的一些其他脚本，那么以下返回码指示程序的退出方式。

0 已成功完成程序。

1 所提供的命令行参数无效。 该参数本身或与该参数一起提供的数据不正确。请更正命令，然后重试。

2 日期命令行参数未使用期望的格式。

3 所提供的作用域定义未解析为任何 IP 地址或所提供的访问条目无效。

4 发生了错误，但是该错误未知。 请转至日志目录，然后打开 cachemgr.log 以查找更多信息。

5 所提供的用户没有足够权限（发现）来执行操作。

6 数据库中没有符合所提供的条件的条目。

为发现做好准备

要优化 TADDM 在发现期间从您的环境收集的信息，必须完成配置任务，使您的环境为发现做好准备。

关于此任务

具体配置任务取决于您的环境中需要支持的发现类型和层次。

下一步做什么

除了针对发现配置您的环境之外，还必须相应地配置 TADDM 传感器。 有关如何执行此操作的信息，请参阅 *Sensor Reference*。

有关如何运行发现的信息（包括定义作用域和设置调度），请参阅 TADDM 《用户指南》。

配置用户登录标识

TADDM 需要交互式用户才能成功地运行发现。 您必须配置用户登录标识。

在非交互方式下，交互式用户登录标识将用于所有发现会话，包括服务器到网关的会话以及网关到目标的会话。 用户必须是交互式用户才能运行命令。 但是，命令以非交互方式运行，这意味着用户运行命令，然后等待结果。

在 /etc/passwd 中，请按以下方式设置用户：

taddmusr:x:100:100::/export/home/*taddmusr*:/bin/sh

其中，*taddmusr* 是 TADDM 用户名。

针对备选发现方法进行配置

您可能希望使用备选发现方法，例如异步发现、基于脚本的发现或使用 IBM Tivoli Monitoring 的发现。

注意：

1. 仅当目标计算机系统正在运行 AIX、FreeBSD、HP NonStop、Linux（仅在 x86 系统上）、Solaris 或 Windows 操作系统时，才支持基于异步的发现和基于脚本的发现。
2. 如果目标计算机系统运行的是 Solaris 操作系统，那么在使用了 SunSSH 1.0 时，基于脚本的发现可能不会工作。

针对异步发现进行配置

要运行异步发现，必须先配置该发现。

关于此任务

要针对异步发现进行配置，您必须生成发现脚本包，将该包复制到目标系统，并在目标系统上运行该脚本。 发现脚本的输出是一个包含发现结果的归档文件。 然后您必须将此归档文件移动到 TADDM 服务器上。

注：如果已将发现配置为以异步方式运行，然后升级了 TADDM，那么必须再次生成发现脚本程序包，因为传感器插件标识可能会更改。

过程

1. 要生成发现脚本包，请从 $COLLATION_HOME/bin 目录输入下列某个命令：
   - **常规方法**

     makeASDScriptPackage *OUTPUT_DIR UNAME* [*IPADDRESS*] [*PACKING_METHOD*]

     OUTPUT_DIR

     脚本包的目录路径。

     UNAME

     要运行脚本的目标系统的操作系统。 有效值为 AIX、Linux、SunOS、FreeBSD、Windows 或 NONSTOP_KERNEL。

     ***IPADDRESS*（可选）**

     要运行脚本的目标系统的 IP 地址。

     用于异步发现的脚本使用 collation.properties 文件中定义的 TADDM 服务器属性中的信息，其中有些属性可能是有作用域的。

     有作用域的属性

     可向其附加 IP 地址或作用域集名称的属性。 该 IP 地址或作用域集名称使该属性依赖于要发现的主机。 您只能使用不包含空格、撇号 (')、句点 (.) 和正斜杠 (/) 的作用域集名称。

     如果您定制了任何 TADDM 服务器属性，使其具有作用域，则应该在 **makeASDScriptPackage** 命令中包括 *IPADDRESS* 选项。

     ***PACKING_METHOD*（可选）**

     用于打包文件的方法。 有效值为 tar 或 zip。

     如果没有指定方法，将由操作系统确定方法。 例如，对于 Linux 之类的操作系统，将使用 tar 方法。

     缺省情况下，将在系统路径中搜索归档实用程序。 如有必要，请将 com.ibm.cdb.tarpath 属性添加至 collation.properties 文件，并为归档实用程序指定备用路径。

     在 Solaris 操作系统上，由于文件名的长度限制，您必须使用 gtar 归档实用程序，并且必须指定该实用程序的路径。

     以下示例显示如何在 AIX 操作系统的 TADDM 服务器上指定 **tar** 命令的路径：

     ```
     com.ibm.cdb.tarpath=tar
     ```

     以下示例显示了如何根据操作系统指定目标系统上 **tar** 命令的路径：

对于 AIX

```
com.ibm.cdb.targettarpath.AIX=tar
```

对于 Solaris

```
com.ibm.cdb.targettarpath.SunOS=/usr/sfw/bin/gtar
```

例如，要生成 AIX 操作系统的发现脚本包，请输入以下命令：

```
./makeASDScriptPackage /tmp AIX
```

此命令将在 tmp 目录中创建以下 AIX 脚本包：/tmp/taddm_AIX.tar。

- **扩展方法**

```
makeASDScriptPackage --outputDir OUTPUT_DIR --uname UNAME
[--ipAddress IP_ADDRESS] [--packingMethod PACKING_METHOD] [--sensors SENSOR]
```

--outputDir *OUTPUT_DIR*

请参阅常规方法中 *OUTPUT_DIR* 参数的描述。

--uname *UNAME*

请参阅常规方法中 *UNAME* 参数的描述。

[--ipAddress *IP_ADDRESS*]（可选）

请参阅常规方法中 *IPADDRESS* 参数的描述。

[--packingMethod *PACKING_METHOD*]（可选）

请参阅常规方法中 *PACKING_METHOD* 参数的描述。

[--sensors *SENSOR*]（可选）

要加入软件包的传感器的名称。 下表包含此命令中必须使用的传感器名称。

表 *33.* **makeASDScriptPackage** 命令中使用的传感器名称。

传感器	命令中使用的名称
Apache 传感器	apacheserver
Citrix XenServer 传感器	xenserver
FreeBSD 计算机系统传感器	computersystem
通用服务器传感器	genericserver
HP NonStop 计算机系统传感器	computersystem
IBM AIX 计算机系统传感器	computersystem
IBM DB2 传感器	db2
IBM Lotus® Domino® 服务器传感器	dominoserverinitial
IBM Tivoli Utilization 传感器	utilization
IBM WebSphere MQ Server 传感器	mqserver
IBM WebSphere 传感器	webspherescript
JBoss Application Server 7 传感器	jboss7
KVM 传感器	kvm
Linux 计算机系统传感器	computersystem
Microsoft Exchange 传感器	exchange
Microsoft IIS Web 服务器传感器	iisserver
Oracle 传感器	oracle
Solaris 计算机系统传感器	computersystem

表 33. **makeASDScriptPackage** 命令中使用的传感器名称。 (续)

传感器	命令中使用的名称
WebLogic SSH 传感器	weblogiclaunchersensor
Windows 计算机系统传感器	computersystem

缺省情况下，异步发现传感器会添加到每个软件包中。 所有操作系统传感器都有 computersystem 名称。 它们的区别在于 --uname 参数。例如，如果指定下列参数：

```
[...] --uname Linux --sensors computersystem
```

Linux 计算机系统传感器将添加到软件包中。

例如，要生成 AIX 操作系统的发现脚本包，请输入以下命令：

```
./makeASDScriptPackage --outputDir /tmp --uname AIX --sensors computersystem
```

此命令将在 tmp 目录中创建以下 AIX 脚本包：/tmp/taddm_AIX.tar。

2. 将脚本包从 *OUTPUT_DIR* 复制到目标系统，然后解压缩该脚本包。
3. 作为 UNIX 系统上的 root 用户或者 Windows 系统上的管理员，授予所有脚本文件的执行特权。如果以非 root 用户或非管理员用户身份运行发现脚本，某些传感器脚本可能无法成功完成发现，或者传感器发现的数据可能有限。
4. 运行 **scriptsRunner.sh** 脚本（对于 UNIX 目标）或 **scriptsRunner.bat**（对于 Windows 目标）。
5. 将产生的归档文件（例如，/tmp/taddm${*version*}/asd/taddmasd-${*hostname*}-${*execution timestamp*}.tar）移动到 TADDM 服务器上 collation.properties 文件中的 com.ibm.cdb.discover.asd.AsyncDiscoveryResultsDirectory 属性所定义的位置。
6. 在 collation.properties 文件中，将 com.ibm.cdb.discover.asd.ProcessUnreachableIPs 属性的值设置为 true。
7. 确保您的发现概要文件上已启用异步发现传感器 (ASDSensor)。 缺省情况下，第 2 层和第 3 层发现概要文件中启用了该传感器。
8. 使用目标系统的 IP 地址创建作用域。

下一步做什么

运行发现。 无需 root 用户权限即可运行此发现。

在发现期间，如果 ping、端口或会话传感器无法访问目标系统，该目标系统将确定为不可达。 如果将 com.ibm.cdb.discover.asd.ProcessUnreachableIPs 属性的值设置为 true，那么将运行异步发现传感器来处理目标系统的发现归档文件。 仅当发现作用域中的 IP 地址与生成归档文件的系统的 IP 地址相匹配时，才会处理该归档文件。 将根据归档文件的内容来调度传感器处理其脚本输出。 归档文件在处理后，将重命名为 *tarfilename*.tar_DONE，以防止再次对其进行处理。

发现归档文件仅处理一次。 如果在处理归档文件时未启用传感器以处理其脚本输出，那么即使启用传感器后再次运行发现也并不会处理之前处理过的归档文件，除非完成下列步骤：

1. 将该归档文件重命名为其原始名称。 例如，从文件名中除去 _DONE。

2. $COLLATION_HOME/var/asdd 目录中的 .processed 文件包含已处理的归档文件的列表。 从 .processed 文件中除去该归档文件的名称。

可在单个发现运行时处理来自不同系统的多个归档文件，但在单个发现运行期间，针对每个目标系统仅处理一个归档文件。 如果某个目标系统有多个归档文件，那么仅处理带有最新时间戳记的归档文件。

要在单个发现运行中发现来自不同系统的多个归档文件，请将每个归档文件复制到 com.ibm.cdb.discover.asd.AsyncDiscoveryResultsDirectory 属性所定义的位置。 在发现作用域中包含每个目标系统的 IP 地址。

由于发现脚本使用 **tar** 命令来创建发现归档文件，因此如果您要使用运行 Windows 操作系统的 TADDM 服务器，必须安装 TADDM 的第三方 tar 程序，用于从该归档文件中解压缩文件。 tar 程序的位置由 com.ibm.cdb.tarpath 属性定义。

限制：tar 程序必须支持长文件路径。 GNU Tar 1.13 不受支持，因为它可能会截断长文件名。

针对基于脚本的发现进行配置

要运行基于脚本的发现，必须先配置该发现。

关于此任务

相对于常规传感器，以基于脚本的方式运行的传感器更直观，这意味着传感器使用的所有命令均在一个可查看的脚本中。 有关支持基于脚本的方式的传感器的列表，以及适用于某些传感器的限制，请参阅 *Sensor Reference* 中的 *Sensors that support script-based and asynchronous discovery* 主题。

过程

使用以下某种方式配置传感器：

- **启用支持基于脚本的发现的所有传感器**

  要全局启用支持基于脚本的发现的所有传感器，请打开 collation.properties 文件并将 com.ibm.cdb.discover.PreferScriptDiscovery 属性的值设置为 true。

- **在特定发现概要文件中启用支持基于脚本的发现的所有传感器**

  对于特定发现概要文件，要启用支持基于脚本的发现的所有传感器，请完成以下步骤：

  1. 在"发现管理门户网站"中，选择想要启用基于脚本的方式的发现概要文件。
  2. 在**平台属性**选项卡中，将 com.ibm.cdb.discover.PreferScriptDiscovery 属性的值设置为 true。

- **在发现概要文件中启用支持基于脚本的发现的传感器**

  在发现概要文件中，要启用支持基于脚本的发现的特定传感器，请在相应的发现概要文件中更新该传感器的配置。 请完成下列步骤：

1. 在"发现管理门户网站"中，转至包含想要启用的传感器的发现概要文件。
2. 在**传感器配置**选项卡中，选择传感器并单击**新建**。
3. 在"创建配置"窗口中，指定配置的名称，然后选择**执行基于脚本的发现**选项。
4. 单击**确定**保存配置。

下一步做什么

配置 TADDM 以选择非缺省用户用于发现

缺省情况下，仅将脚本请求的用户用于发现。 如果在使用缺省用户运行发现时遇到问题并且拥有具备所有必需的许可权的其他用户，那么可以配置 TADDM 以选择此用户用于发现。

注：请谨慎地使用以下配置。 如果将无所有必需的许可权的用户用于发现，那么发现可能失败，或者可能无法发现某些目标。

在可以针对每个传感器找到一个软件包的 COLLATION_HOME/osgi/plugins 目录的 plugin.xml 文件中，编辑 script 节点定义，例如，在 IBM WebSphere MQ Server 传感器的 plugin.xml 片段中：

```
<scriptset>
   <ostype>AIX</ostype>
   <mainScript name="sensorCommon.sh" />
   <script name="script.sh" authClassName="com.collation.platform.security.
auth.MQServerAuth" authMode="preferred" hostAuthFallback="true"/>
</scriptset>
```

可以定义下列属性：

authMode

定义对于 authClassName 所指定的类型，TADDM 如何对访问列表中相应类型的条目进行访问。 提供了以下值：

- single - 仅使用脚本所请求使用的用户。 这是缺省值。
- preferred - 首先使用脚本的首选用户，但如果此用户不可用或者失败，那么使用所定义类型的其余访问列表条目。
- regular - 按照指定访问列表条目的顺序使用这些条目，而不检查首选用户。

hostAuthFallback

定义在为特定 authClassName 和/或首选用户与目标建立连接发生问题的情况下，TADDM 是否回退到用于连接该目标的一般用户所建立的会话。 提供了以下值：

- false - 缺省值。
- true。

使用 IBM Tivoli Monitoring（老方法）配置发现

TADDM 可以使用 IBM Tivoli Monitoring 6.2.1 或更高版本的基础结构来执行第 1 层、第 2 层和某些第 3 层发现。

旧的集成方法

该部分说明的是不推荐使用的 TADDM 与 IBM Tivoli Monitoring 集成方法. 从 TADDM V7.3.0 开始，建议使用 OSLC 自动化将 TADDM 与 IBM Tivoli

Monitoring 6.3 集成。 不推荐使用 IBM Tivoli Monitoring Scope 传感器进行集成，在后续发行版中将移除这个老方法。 有关使用 OSLC 自动化来配置发现的更多信息，请参阅 第 99 页的『进行配置以通过 OSLC 自动化会话执行发现』。

如果使用的是 IBM Tivoli Monitoring 6.2.1-TIV-ITM-FP0001、6.2.2-TIV-ITM-FP0002 或更高级别，您可以通过 Tivoli Enterprise Portal 服务器发现 Tivoli Monitoring 端点。 这些修订包用于解决在 TADDM 发现期间提高 Tivoli Monitoring 性能的 APAR IZ63983。 使用较早发行版或级别的 IBM Tivoli Monitoring 通过 Tivoli Enterprise Portal 服务器执行 TADDM 发现，可能会导致处理器和网络负载过量（尤其是在 Tivoli Monitoring 组件上）。

注：仅当 Tivoli Enterprise Portal Server 数据库位于 Microsoft SQL Server 和 DB2 上时，才能使用 IBM Tivoli Monitoring 执行发现。 使用 Apache Derby 数据库作为 Tivoli Enterprise Portal Server 数据库时，无法以此方式执行发现。

特定于使用 Tivoli Monitoring 的发现的 TADDM 服务器属性

要了解特定于使用 IBM Tivoli Monitoring 的发现的 TADDM 服务器属性，包括影响 TADDM 发现 Tivoli Monitoring 端点的方式的属性，请参阅第 70 页的『使用 IBM Tivoli Monitoring（老方法）的发现属性』。

在发现概要文件中，您可以配置影响 TADDM 发现 Tivoli Monitoring 端点的方式的 TADDM 服务器属性。 为此，请根据您使用的是定制概要文件还是缺省概要文件，完成下列步骤：

配置定制概要文件的属性

1. 启动发现管理控制台。
2. 打开**发现概要文件**。
3. 单击要配置的发现概要文件。
4. 单击**平台属性**选项卡。
5. 更改要更新的属性的值，然后为此属性选择**包含**复选框。
6. 保存更改。

配置缺省概要文件的属性

在 `$COLLATION_HOME/etc/collation.properties` 文件中，添加（或编辑）相应属性，如以下示例所示，其中 *discovery_profile* 表示概要文件名称：

```
com.ibm.cdb.session.allow.ITM.discovery_profile=true
```

例如，以下属性指定 TADDM 使用发现概要文件"Utilization Discovery"，并使用 IBM Tivoli Monitoring 来发现 Tivoli Monitoring 端点：

```
com.ibm.cdb.session.allow.ITM.Utilization_Discovery=true
```

注： 在 `collation.properties` 文件中，您必须将概要文件名称中"Utilization"和"Discovery"之间的空格字符替换为下划线字符。

可能需要配置的其他 TADDM 服务器属性

以下配置提示描述了您可能需要配置的其他 TADDM 服务器属性：

- 以下特定于 Windows 系统的属性的值必须设置为 true（这是缺省值）才能启用在发现中使用 IBM Tivoli Monitoring 发现 Windows 目标系统的功能。 如果值设置为 false，TADDM 就无法建立 IBM Tivoli Monitoring 到 Windows 目标系统的会话。

  ```
  com.collation.AllowPrivateGateways=true
  ```

- 在发现期间，Tivoli Enterprise Portal 服务器上可能会出现较高的处理器使用率。 要最大限度减小使用率，可以限制发现期间运行的发现工作程序线程的数量。 请在 TADDM 服务器上设置以下服务器属性：

  ```
  com.collation.discover.dwcount=16
  ```

- 在大型 IBM Tivoli Monitoring 环境中，IBM Tivoli Monitoring 作用域传感器可能会在完成前超时。要允许更长的处理时间，请设置以下服务器属性：

  ```
  com.collation.platform.session.ITMSessionNumProgressChecks=3600
  com.collation.discover.agent.ITMScopeSensor.timeout=3600000
  ```

进行配置以通过 OSLC 自动化会话执行发现

TADDM 可以使用 OSLC 来运行第 2 层和某些第 3 层发现。

开始之前

要对 OSLC Execute Automation Service Provider 所提供的作用域集配置发现，必须满足下列要求：

- 必须至少有一个已安装且正常运行的 OSLC Execute Automation Service Provider。
- TADDM 必须已连接到 OSLC Execute Automation Service Provider。

过程

要通过 OSLC 自动化会话来运行发现，请完成下列步骤：

1. 将所要集成的产品的访问凭证添加到访问列表中。 为此，请创建类型为"集成">"OSLC 自动化"的新访问列表条目。 如果要将 TADDM 与 ITM 集成，请提供 ITM TEPS 凭证。在发现期间，将使用 OSLC 自动化访问列表条目和 ITM 访问列表条目类型来确保与早期版本兼容。
2. 检查发现作用域。 作用域集由 OSLCAutomationAgent 定期创建。 新的作用域集将列示在**作用域集**选项卡中。 如果要将 TADDM 与 ITM 集成，那么将针对每个 ITM TEMS 创建一个作用域集。 您可以使用以下命令来手动运行 OSLCAutomationAgent：

   ```
   /taddm/dist/support/bin/runtopobuild.sh -a OSLCAutomationAgent
   ```

3. 配置允许使用 OSLC 自动化会话的发现属性。 可以在 collation.properties 文件或新的定制发现概要文件中设置这些属性。

   - collation.properties 文件：

     ```
     com.ibm.cdb.session.prefer.OSLCAutomation=true
     com.ibm.cdb.session.allow.OSLCAutomation=true
     ```

     具有作用域的属性的示例：

     ```
     com.ibm.cdb.session.prefer.OSLCAutomation.9.222.222.124=false
     com.ibm.cdb.session.prefer.OSLCAutomation.Level_3_Discovery=false
     ```

   - 定制发现概要文件。 在发现管理控制台中，创建一个新的发现概要文件，并按以下方式配置**平台属性**选项卡：

```
com.ibm.cdb.session.allow.OSLCAutomation=true
com.ibm.cdb.session.prefer.OSLCAutomation=true
```

4. 通过选择下列任意一种方法对 OSLCAutomationAgent 所创建的作用域运行定期发现：
   - 缺省的 L2 或 L3 概要文件（如果 collation.properties 文件配置为支持 OSLC 自动化会话）。
   - 正确配置了**平台属性**选项卡的新发现概要文件。

相关参考:

第 72 页的『对使用 OSLC 自动化会话的发现应用的属性』
这些属性应用于使用 OSLC 自动化会话的发现。

第 185 页的『OSLCAutomationAgent 的命令行界面』
OSLCAutomationAgent 用于从 OSLC Execute Automation Service Provider 中收集数据。 您可以使用命令手动运行该代理程序，并可以使用命令来刷新或更新该代理程序所创建的作用域集。

配置发现的层次

您必须配置发现的层次。

针对第 1 层发现进行配置

第 1 层发现（无凭证发现）需要一些最低配置，该发现会扫描 TCP/IP 堆栈以收集有关活动计算机系统的基本信息。

关于此任务

对于第 1 层发现，您必须在环境中配置希望 TADDM 服务器发现的网络设备。

过程

为此，请完成下列步骤：

1. 根据 SNMP 版本，记录以下信息以供 TADDM 服务器使用：
   - 对于 SNMP V1 和 V2，记录 SNMP MIB2 GET COMMUNITY 字符串。
   - 对于 SNMP V3，记录 SNMP 用户名和密码。
2. 指定对 MIB2 系统、IP、接口和扩展接口的许可权。

针对第 2 层发现进行配置

除了第 1 层发现的需求之外，第 2 层发现还需要进行配置以支持发现详细的主机配置信息。

开始之前

如果目标系统是由 IBM Tivoli Monitoring Scope 传感器发现的 IBM Tivoli Monitoring 端点，那么第 2 层发现就不需要这些目标系统的凭证。 有关更多信息，请参阅以下资源：

- 第 186 页的『将 TADDM 与 IBM Tivoli Monitoring（老方法）相集成』
- 第 97 页的『使用 IBM Tivoli Monitoring（老方法）配置发现』
- TADDM *Sensor Reference*，它提供了关于 IBM Tivoli Monitoring Scope 传感器的信息

关于此任务

在您希望 TADDM 发现的目标操作系统（计算机系统）上，您必须至少配置以下软件：

安全 Shell (SSH)

可以使用 OpenSSH 或操作系统附带的，供应商提供的 SSH 版本。 有关 Windows 操作系统的更多信息，请参阅 第 113 页的『Windows Management Instrumentation (WMI) 依赖关系』。

SUNWscpu（仅适用于 Solaris 环境）

要提供有关进程的完整信息，请安装 SUNWscpu (Source Compatibility) 软件包。

列出打开的文件 (lsof)

要提供有关依赖关系的完整信息，请根据位于 https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Application%20Dependency%20Discovery%20Manager/page/TADDM%20lsof%20requirements 的 TADDM Wiki 中的 *lsof requirements* 的需求，在所有目标计算机系统上安装 LiSt Open Files (lsof) 程序。

创建服务帐户：

必须在所有使用基于 SSH 密钥和基于密码连接发现的计算机系统上创建服务帐户。这是用于发现您网络中的计算机系统（服务器）的主要方法。

关于此任务

要简化发现设置，请在每个要发现的目标计算机系统上创建相同的服务帐户。服务帐户必须允许访问 TADDM 需要发现的目标计算机系统上的所有资源。 服务帐户必须具有对其在每个目标计算机系统上的主目录的写访问特权。 此目录要求大约 20 MB 可用空间。 在发现期间，可将脚本和临时结果文件存储在此目录中。 运行发现后，将删除这些文件。

可以使用具有非 root 特权的服务帐户。 但是，在发现期间使用的一些操作系统命令可能需要提升的特权（例如 root 用户或超级用户）才能在目标计算机系统上运行。

过程

完成以下某个过程可在的目标计算机系统上创建服务帐户：

1. 对于 Linux、Solaris、AIX 和 Linux for System z 操作系统，假设服务帐户名称为 coll，并使用以下命令创建服务帐户：

   ```
   # mkdir -p /export/home/coll
   # useradd -d /export/home/coll -s /bin/sh \
      -c "Service Account" -m coll
   # chown -R coll /export/home/coll
   ```

2. 对于 Windows 计算机系统，请创建一个属于本地管理员组成员的服务帐户。 此帐户可以是本地帐户，也可以是域帐户。 由于 TADDM 依赖于 WMI 进行发现，因此该帐户必须能够访问本地计算机上的所有 WMI 对象。 必须在 Windows Gateway 和所有目标 Windows 计算机系统上创建服务帐户。

   注：该服务帐户必须具有对 \WINDOWS\system32 或 \WINDOWS\system64 目录及其子目录的读/写访问权。 在 Windows Server 2008 系统上，缺省情况下，新用户不具有必需的访问权，因此您必须明确将其授予服务帐户。

使用安全外壳协议 *(SSH)* 来配置发现：

TADDM 服务器可连接到 OpenSSH（V1 或 V2）或供应商提供的 SSH（随操作系统提供）。

TADDM 服务器支持以下认证方法：

- 基于 SSH2 密钥的登录（RSA 或 DSA 密钥）以及基于 SSH1 密钥的登录（仅 RSA）
- 使用 SSH2 的用户名和密码，以及使用 SSH1 的用户名和密码

虽然可以使用任何认证方法，但基于 SSH2 密钥的登录是首选方法。 服务器按照前面列出的顺序自动尝试每种方法，并使用第一个成功起作用的方法。 然后 TADDM 服务器在整个发现运行过程中对该主机使用相同的方法。

注：对于基于 SSH2 密钥的登录，TADDM 服务器会尝试仅用一个密钥（RSA 或 DSA）登录，具体取决于在 TADDM 服务器上找到的密钥。 如果两种密钥都存在，则仅使用 RSA。

为基于密钥登录 *TADDM* 服务器创建密钥对：

可以使用安全 Shell 协议 (SSH) 创建公用/专用密钥对，用于基于密钥登录 TADDM 服务器。

关于此任务

根据所使用的 SSH 版本，基于 SSH 密钥的登录使用表 34 中显示的密钥：

表 *34. SSH* 密钥

SSH 版本/算法	专用密钥	公用密钥
Openssh/SSH2/RSA	$HOME/.ssh/id_rsa	$HOME/.ssh/id_rsa.pub
Openssh/SSH2/DSA	$HOME/.ssh/id_dsa	$HOME/.ssh/id_dsa.pub
Openssh/SSH1/RSA	$HOME/.ssh/identity	$HOME/.ssh/identity.pub
商用/SSH2/RSA	$HOME/.ssh2/id_dss_1024_a	$HOME/.ssh2/id_dss_1024_a.pub

您还可以使用 OpenSSH V2 生成公用/专用密钥对。要使用除 OpenSSH 外的 SSH 程序或其他版本的 OpenSSH 生成公用/专用密钥对，请参阅 SSH 文档。

过程

要使用 OpenSSH V2 生成公用/专用密钥对，请完成下列步骤：

1. 以 TADDM 服务器的所有者身份登录。
2. 要生成 SSH 密钥，请输入以下命令：

   ```
   $ ssh-keygen -t rsa
   ```

   接受命令缺省值。 TADDM 支持使用或不使用口令的密钥对。
3. 在每个想要允许基于密钥登录的目标计算机系统上，在服务帐户的 $*HOME*/.ssh/authorized_keys 文件中插入 id_rsa.pub 文件的内容。 某些 SSH2 实现在 $HOME/.ssh 之外的目录中生成密钥。如果您的 SSH 实施在其他目录中或使用其他名称生

成密钥，那么请根据算法，将专用密钥文件复制、链接或移动到 *$HOME*/.ssh/id_rsa 或 *$HOME*/.ssh/id_dsa 目录中。

添加计算机系统服务帐户的访问列表条目：

要配置使用 Secure Shell (SSH) 进行密码认证，必须添加您在目标系统上创建的计算机系统服务帐户的访问列表条目。

要添加计算机系统服务帐户的访问列表条目，请完成下列步骤：

1. 在 TADDM 启动页面上，确保"管理员控制台"中的所有服务均已启动。
2. 启动发现管理控制台。
3. 选中**建立安全 (SSL) 会话**复选框，以使您使用 SSL 安全选项。 通过该选项，所有数据（包括访问列表用户名和密码）在发现管理控制台和 TADDM 服务器之间传输之前都将进行加密。
4. 添加服务帐户的计算机系统访问列表条目，然后指定登录名和密码。

配置 *System p* 和 *System i*：

通过管理控制台来发现基于 IBM Power5 技术的系统（System p 或 System i®）及其逻辑分区。 TADDM 支持两种类型的管理控制台：硬件管理控制台 (HMC) 和集成虚拟化管理器 (IVM)。

TADDM 使用 SSH 发现管理控制台。 发现作用域必须包括管理控制台的 IP 地址，"访问列表"必须包含指定了正确凭证（用户名和密码）的类型为"计算机系统"的条目。

除这些用户凭证之外，必须使用以下最低许可权在管理控制台上定义发现用户：

- 硬件管理控制台 (HMC)
  - 对于 HMC 管理控制台，需要基于 **hmcoperator** 角色的用户。 例如，根据 hmcoperator 角色创建名为 *taddmViewOnly* 的新角色。 此外，必须为新角色指定下列命令行任务：

    受管系统
    使用 **lshwres** 和 **lssyscfg** 命令时需要

    逻辑分区
    使用 **lshwres**、**lssyscfg** 和 **viosvrcmd** 命令时需要。

    HMC 配置
    使用 **lshmc** 命令时需要。
- 集成式虚拟化管理器 (IVM)。

  对于 IVM 管理控制台，需要具有 **View Only** 角色的用户。

针对第 3 层发现进行配置

除了第 2 层发现的需求之外，第 3 层发现还需要进行配置以支持发现应用程序配置和主机数据。

为发现配置 *Web* 和应用程序服务器：

必须在环境中配置希望 TADDM 服务器发现的 Web 服务器和应用程序服务器。

本部分提供配置 Web 服务器和应用程序服务器的步骤。

Microsoft IIS 服务器无需配置。 对访问权没有特别的要求。 在主机上建立的用户帐户已足够。

对于 Apache Web 服务器，主机系统的 TADDM 服务帐户必须对 Apache 配置文件（如 httpd.conf 文件）具有读许可权。

对于 Oracle iPlanet Web 服务器，主机系统的 TADDM 服务帐户必须对 iPlanet 配置文件具有读许可权。

对于 Lotus Domino 服务器，请确保您满足 TADDM *Sensor Reference* 的 *IBM Lotus Domino server sensor* 主题中的先决条件。

配置 *Oracle Application Server*：

对 Oracle Application Server 的发现使用该 Oracle Application Server 附带的 JAR 文件。 这些 JAR 文件未包含在 TADDM 服务器安装中。

关于此任务

$COLLATION_HOME/etc/collation.properties 文件中有一个属性用于指向 Oracle Application Server 的现有安装。 以下文本位于 $COLLATION_HOME/etc/collation.properties 文件中：

```
# Location of the root directory for Oracle Application Server on
the Tivoli Application Dependency Discovery Manager
server
# 1. An example is /home/oracle/product/10.1.3/OracleAS_1
# 2. A relative directory is relative to com.collation.home
# 3. This directory (and its subdirectories) must be accessible
     for the user under which the server runs, usually the collation user.
# 4. Ignore if you do not intend to discover an Oracle Application server.
```

要指向 Oracle Application Server 的现有安装，请编辑 $COLLATION_HOME /etc/collation.properties 文件中的以下行：

```
com.collation.oracleapp.root.dir=lib/oracleapp
```

在 Oracle Application Server 安装中，包含所需的 JAR 文件的目录属于具有许可权 rwx------ 的 oracle 用户。 这意味着，除了所有者（通常是 Oracle 应用程序）之外，任何人都不能访问这些目录。 如果 TADDM 服务器是使用 oracle 用户运行的，那么可以访问这些目录。 但是，如果不是这样，那么必须将以下目录的目录许可权更改为 711，以便所有用户都可访问这些目录：

- *OracleAppServerHome*
- *OracleAppServerHome*/j2ee
- *OracleAppServerHome*/j2ee/home
- *OracleAppServerHome*/opmn
- *OracleAppServerHome*/opmn/lib，其中 *OracleAppServerHome* 的示例为 /home/oracle/product/10.1.3/OracleAS_1

为了发现 Oracle Application Server，必须将 $COLLATION_HOME/etc/collation.properties 文件中的 com.collation.platform.os.ignoreLoopbackProcesses 属性设置为 true：

```
com.collation.platform.os.ignoreLoopbackProcesses=true
```

过程

要配置访问列表，请完成下列步骤：

1. 从发现管理控制台，创建包含 Oracle Application Server 的发现作用域集，或使用包含 Oracle Application Server 的现有作用域。
2. 要创建访问列表，请单击**访问列表**图标。
3. 在"访问列表"窗口中，单击**添加**。
4. 在"访问详细信息"窗口的**组件类型**字段中，单击**应用程序服务器**。
5. 在**供应商**字段中，单击 **Oracle Application Server**。
6. 输入 Oracle Application Server 的凭证。

配置 *Microsoft Exchange Server*：

必须配置希望 TADDM 服务器发现的 Microsoft Exchange Server。

关于此任务

要发现 Microsoft Exchange Server，Microsoft Exchange Management 服务必须运行在目标 Windows 系统上。 TADDM 服务帐户的 Windows 服务标识必须在 Microsoft Exchange Server 所运行的 Windows 系统上创建。 Windows 服务标识必须具有对以下 WMI 名称空间的完全许可权（即执行方法、完全写入、部分写入、提供程序写入、启用帐户、远程启用、读安全性和编辑安全性）：

- Root\CIMV2
- Root\CIMV2\Applications\Exchange
- Root\MicrosoftExchangeV2

如果 TADDM 服务帐户的 Windows 服务标识具有发现 Microsoft Exchange Server 所需的足够许可权，那么传感器将使用 Windows 服务标识，而不需要单独的 Microsoft Exchange Server 访问列表条目。

如果 TADDM 服务帐户的 Windows 服务标识没有发现 Microsoft Exchange Server 所需的足够许可权，那么必须创建单独的 Microsoft Exchange Server 访问列表。

过程

要配置访问列表，请完成下列步骤：

1. 从发现管理控制台，创建包含 Microsoft Exchange Server 的发现作用域集，或使用包含 Microsoft Exchange Server 的现有作用域。
2. 要创建访问列表，请单击**访问列表**图标。
3. 在"访问列表"窗口中，单击**添加**。
4. 在"访问详细信息"窗口的**组件类型**字段中，单击**消息传递服务器**。
5. 在**供应商**字段中，单击 **Microsoft Exchange Server**。
6. 输入 Microsoft Exchange Server 的凭证。

配置 *VMware* 服务器：

正确配置后，TADDM 发现进程将返回有关 VMware 服务器的信息。

关于此任务

要配置 VMware 服务器进行发现，请为 VMware ESX 控制台中的非 root TADDM 服务帐户设置只读许可权。 另一种方法是为发现使用 root 用户。有关 VMware 服务器的更多信息，您可以在位于 https://communities.vmware.com/welcome 的 VMware 社区中搜索主题。

针对发现设置数据库：

要支持数据库发现，必须为 TADDM 服务器创建 DB2、Oracle 或 Sybase 数据库用户。 TADDM 服务器使用这些数据库用户来收集有关在远程主机上运行的数据库的信息。

创建 *DB2* 用户：

要更完整地发现远程计算机主机上的 DB2 实例，请创建 DB2 用户。

过程

要创建 DB2 用户，请完成下列步骤：

1. 创建对以下各项具有访问权的用户：
   - DB2 数据库 TADDM 服务器
   - DB2 数据库 TADDM 服务器中需要发现的所有实例
2. 将此 DB2 用户配置为对托管 DB2 数据库服务器的系统具有 SSH 访问权。
3. 在 TADDM 服务器访问列表中，完成下列步骤来为 DB2 用户添加用户名和密码：
   a. 在发现管理控制台工具栏中，单击**发现** > **访问列表**。 此时将显示"访问列表"窗格。
   b. 单击**添加**。 此时将显示"访问详细信息"窗口。
   c. 在"访问详细信息"窗口中，完成以下信息：
      1) 在**组件类型**列表中，选择**数据库**。
      2) 在**供应商**列表中，选择 **DB2**。
      3) 输入 DB2 用户的名称、用户名和密码。
   d. 单击**确定**保存信息。 此时将显示具有新信息的"访问列表"窗格。

创建 *Microsoft SQL Server* 用户：

要更完整地发现远程计算机主机上的 Microsoft SQL Server 实例，请创建 Microsoft SQL Server 用户。

过程

要创建 Microsoft SQL Server 用户，请完成下列步骤：

1. 创建具有 `db_datareader` 角色特权和 `VIEW_ANY_DEFINITION` 许可权的 Microsoft SQL Server 用户。此操作可能需要由 Microsoft SQL Server 管理员完成。
2. 在发现管理控制台中，完成下列步骤以在 TADDM 服务器访问列表中添加 Microsoft SQL Server 用户的用户名和密码：
   a. 在工具栏中，单击**发现** > **访问列表**。 此时将显示"访问列表"窗格。
   b. 单击**添加**。 此时将显示"访问详细信息"窗口。

c. 在"访问详细信息"窗口中，请输入以下信息：
   1) 在**组件类型**列表中，选择**数据库**。
   2) 在**供应商**列表中，选择 **Microsoft SQL Server**。
   3) 输入**名称**、**用户名**和**密码**。
d. 单击**确定**保存信息。 此时将显示具有新信息的"访问列表"窗格。

创建 *Oracle* 用户：

要更完整地发现远程计算机主机上的 Oracle 实例，请创建 Oracle 用户。

过程

要创建 Oracle 用户，请完成下列步骤：

1. 创建具有 SELECT_CATALOG_ROLE 特权的 Oracle 用户。此操作可能需要由 Oracle 管理员完成。

   例如，使用以下命令创建 IBM Oracle 用户：

   ```
   create user collation identified by collpassword;
   grant connect, select_catalog_role to collation;
   ```

2. 在发现管理控制台中，完成下列步骤以在 TADDM 服务器访问列表中添加 Oracle 用户的用户名和密码：
   a. 在工具栏中，单击**发现** > **访问列表**。 此时将显示"访问列表"窗格。
   b. 单击**添加**。 此时将显示"访问详细信息"窗口。
   c. 在"访问详细信息"窗口中，完成以下信息：
      1) 在**组件类型**列表中，选择**数据库**。
      2) 在**供应商**列表中，选择 **Oracle**。
      3) 输入计算机的名称、用户名和密码。
   d. 单击**确定**保存信息。 此时将显示具有新信息的"访问列表"窗格。

创建 *Sybase* 用户：

要完全发现远程计算机主机上的 Sybase ASE，请创建一个指定了相应角色的 Sybase 用户。

过程

要创建 Sybase 用户，请完成下列步骤：

1. 使用以下命令创建作为 sa-role 成员的 Sybase 用户。

   ```
   sp_role "grant",sa_role,IBM
   ```

   确保 Sybase IQ 用户是 DBA 的成员。如果 Sybase IQ 用户不是 DBA 成员，那么找不到特定于 Sybase IQ 数据库的信息。

2. 在发现管理控制台中，完成下列步骤以在 TADDM 服务器访问列表中添加 Sybase 用户的用户名和密码：
   a. 要创建访问列表，请单击**访问列表**图标。
   b. 在"访问列表"窗口中，单击**添加**。
   c. 在"访问详细信息"窗口的**组件类型**字段中，单击**数据库**。

d. 在**供应商**字段中，单击**数据库**。

e. 输入凭证（用户名和密码）以建立到 Sybase 服务器的 Java 数据库连接 (JDBC)。

针对 Windows 系统的发现进行配置

为了发现 Windows 计算机系统、TADDM 支持基于网关的发现、基于 SSH 的发现、异步发现和基于脚本的发现。

有关异步发现的详细信息，请参阅第 93 页的『针对异步发现进行配置』。 有关基于脚本的发现的详细信息，请参阅第 96 页的『针对基于脚本的发现进行配置』。

基于网关的发现需要可通过 SSH 访问的专用 Windows 计算机系统来充当网关。 所有发现请求都通过该网关。 网关使用 Windows Management Instrumentation (WMI) 发现目标 Windows 计算机系统。

Fix Pack 2 如果使用 TADDM 7.3.0.2 或更高版本而不是 WMI，那么还可以使用 PowerShell 会话来发现目标 Windows 计算机系统。您可以将 TADDM 配置为仅允许通过 PowerShell 会话进行通信。有关详细信息，请参阅《TADDM 用户指南》中的『进行配置以通过防火墙而不使用锚点来发现』。

基于 SSH 的发现无需专用的网关计算机系统。 发现将通过直接 SSH 连接来连接到目标 Windows 操作系统。

通常，基于网关的发现优先于基于 SSH 的发现，因为配置网关和 WMI（或 PowerShell）比配置 SSH 更容易。缺省情况下，WMI 在 TADDM 支持的所有 Windows 目标系统上都可用。仅针对运行 Windows Server 2008 和更高版本的目标才支持 PowerShell。 必须在网关和目标系统上安装 PowerShell V2 或更高版本。 与需要 SSH 服务器的网关计算机不同，Windows 目标没有特别的软件需求。 但使用 SSH 的发现可能更快，因为发现流中不涉及网关，并且无需部署 WMI 提供程序。

执行直接发现要求每个 Windows 目标系统上都有 SSH 服务器。 另外，对于使用 SSH 执行的直接发现，必须在每个 Windows 目标系统上安装 Microsoft .NET Framework V2 或 V3。缺省情况下，在 Windows Server 2000 上未安装 .NET Framework。

注： Fix Pack 2 如果使用 TADDM 7.3.0.2 或更高版本，那么也可以安装 .NET Framework V4 或 V4.5。

这两种类型的发现都使用 TADDM Windows 发现程序 TaddmTool.exe 文件来执行发现。 对于使用网关的发现，将在发现初始化期间向网关部署 TaddmTool 程序。 对于使用 SSH 的发现，将向每个 Windows 目标计算机系统部署 TaddmTool 程序。 TaddmTool 程序是一个 .NET 应用程序。

缺省情况下，TADDM 配置为仅使用基于网关的发现。此配置由以下两个 TADDM 服务器属性控制，在针对 Windows 计算机系统传感器的 TADDM *Sensor Reference* 中对其作了说明：

- `com.collation.AllowPrivateGateways=true`
- `com.collation.PreferWindowsSshOverGateway=false`

缺省情况下，TADDM 配置为使用 WMI 会话。 要了解何时使用以及如何启用 PowerShell 会话，请参阅第 113 页的『PowerShell 会话』。

无论使用 Windows 网关和 WMI，还是使用通过 SSH 的直接连接，检索到的信息都相同。 以下列表标识了基于网关的发现和基于 SSH 的发现的先决条件：

使用 WMI 的基于网关发现的先决条件

1. 需要专用的 Windows Server 计算机系统来充当网关。 对于网关服务器的操作系统需求与 TADDM 服务器的 Windows 操作系统需求相同。 有关受支持的 Windows 操作系统的详细信息，请参阅《TADDM 安装指南》中的『TADDM 服务器软件需求』主题。
2. 网关必须与要发现的 Windows 计算机位于同一防火墙区域内。
3. 必须在网关计算机系统上安装受支持的 SSH 服务器版本。
4. 该网关使用远程 WMI 发现每个 Windows 目标。 此外，将自动在发现初始化期间向每个 Windows 目标计算机系统部署一个 WMI 提供程序。 该 WMI 提供程序用于发现核心 WMI 中未包含的数据。 在要发现的 Windows 目标计算机系统上启用 WMI。 缺省情况下，大多数 Windows 2000 和更高版本的系统上都启用了 WMI。

Fix Pack 2 **使用 PowerShell 的基于网关发现的先决条件**

1. 需要专用的 Windows Server 计算机系统来充当网关。 对于网关服务器的操作系统需求与 TADDM 服务器的 Windows 操作系统需求相同。 有关受支持的 Windows 操作系统的详细信息，请参阅《TADDM 安装指南》中的『TADDM 服务器软件需求』主题。
2. 必须在网关和目标系统上安装 PowerShell V2 或更高版本。 仅支持运行 Windows Server 2008 或更高版本的目标。
3. 必须运行以下命令来配置网关：

   ```
   Set-Item WSMan:\localhost\Client\TrustedHosts * -Force
   ```

   此命令可设置 **trustedHosts** 列表。 缺省情况下，此列表存在但为空，因此必须先设置此列表，然后才能打开远程会话。 通过使用 `-Force` 参数，PowerShell 无需针对每个步骤向您发出提示即可执行此命令。
4. 必须运行以下命令来配置目标系统：

   ```
   Enable-PSRemoting -Force
   ```

   此命令可启动 WinRM 服务、将其设置为随系统自动启动，并创建一条防火墙规则以允许传入连接。 通过使用 `-Force` 参数，PowerShell 无需针对每个步骤向您发出提示即可执行这些操作。

基于 SSH 的发现的先决条件

1. 必须在每个 Windows 目标系统上安装受支持的 SSH 服务器版本。
2. 必须在每个 Windows Server 目标系统上安装 Microsoft .NET Framework V2 或 V3。

   注： Fix Pack 2 如果使用 TADDM 7.3.0.2 或更高版本，那么也可以安装 .NET Framework V4 或 V4.5。

另请参阅 TADDM *Sensor Reference* 中的配置非管理员 *Windows* 发现主题。

配置 Bitvise WinSSHD

您可以使用 Bitvise WinSSHD 来提供对 Windows 系统的 SSH 访问。

开始之前

对于基于网关的发现，Bitvise WinSSHD 必须安装在网关系统上。 对于直接 SSH 发现，Bitvise WinSSHD 必须安装在每个 Windows 系统上。

有关受支持的 Bitvise WinSSHD 版本的更多信息，请参阅《TADDM 安装指南》中的『Windows 网关』主题。

Bitvise WinSSHD 可从 http://www.bitvise.com/ 获得。

关于此任务

下列步骤描述如何配置 Bitvise WinSSHD 5.22。 根据您拥有的 Bitvise WinSSHD 发行版，具体步骤可能有所不同。

过程

1. 要限制对 TADDM 服务器的 SSH 主机访问，请完成下列步骤：
   a. 在"WinSSHD 控制面板"上，单击**打开简易设置**。
   b. 在**服务器设置**选项卡上，对于**打开 Windows 防火墙**字段，选择**采用高级 WinSSHD 设置中的设定**。
   c. 单击**保存更改**。
   d. 在"WinSSHD 控制面板"上，单击**编辑高级设置**。 此时将显示"高级 WinSSDH 设置"窗口。
   e. 单击**设置 > 会话**。
   f. 将以下项目的值设置为 `0`：
      - IP blocking - 窗口持续时间
      - IP blocking - 锁定时间
   g. 单击**确定**。
   h. 在"WinSSHD 控制面板"上，单击**编辑高级设置**。 此时将显示"高级 WinSSDH 设置"窗口。
   i. 单击**设置 > 访问控制**。
   j. 在右窗格中，单击 **IP 规则**。
   k. 单击**添加**。
   l. 键入 TADDM 服务器的 IP 地址。
   m. 在**有效位数**字段中，键入 `32`。
   n. 在**描述**列表中，键入 `TADDM server`。
   o. 确保选择**允许连接**复选框。
   p. 单击**确定**。
   q. 从列表中除去 `0.0.0.0/0` 条目。
2. 要创建和配置虚拟组与用户，请完成下列步骤：
   a. 在"WinSSHD 控制面板"上，单击**编辑高级设置**。 此时将显示"高级 WinSSDH 设置"窗口。

b. 单击**设置** > **虚拟组**。
c. 要添加新组，单击**添加**。
d. 在**组和 Windows 帐户名称**字段中，键入名称。
e. 单击**确定**。
f. 单击**设置** > **虚拟帐户**。
g. 要添加新帐户，单击**添加**。
h. 在**虚拟帐户名称**字段中，键入名称。
i. 使用虚拟帐户密码的链接设置密码。
j. 从下拉列表选择上一步中创建的虚拟组，并确保选择**使用组缺省 Windows 帐户**复选框。
k. 单击**确定**。

3. 在"WinSSHD 控制面板"中，单击**启动 WinSSHD**。

下一步做什么

如果要发现多个 Windows 服务器，可能会遇到以下消息：

```
A Working gateway cannot be found
```

有关其他可能有帮助的配置的更多信息，请参阅《TADDM 故障诊断指南》中的『网关问题』主题。

配置 Cygwin SSH 守护程序

您可以使用 Cygwin SSH 守护程序 (sshd) 来提供对 Windows 系统的 SSH 访问。

关于此任务

对于基于网关的发现，Cygwin SSH 守护程序必须安装在网关系统上；对于直接 SSH 发现，该守护程序必须安装在每个 Windows 系统上。

有关受支持的 Cygwin SSH 守护程序版本的更多信息，请参阅《TADDM 安装指南》中的『Windows 网关』主题。

要点：要使用 Cygwin SSH 成功完成发现，必须满足以下需求：

- 在 Windows Server 2012 x64 和 Windows Server 2008 x64 上的 Cygwin 64 位版本上支持锚点和网关。
- 发现用户和启动服务的用户必须相同。 发现用户必须是管理员组的成员。

Cygwin 可从 http://www.cygwin.com/ 获得。

过程

要配置 Cygwin SSH 守护程序：

1. 启动 cygwin bash shell。
2. 从系统信息中，使用 **cygwin mkpasswd** 实用程序来创建初始 /etc/passwd。 也可以使用 **mkgroup** 实用程序来创建初始 /etc/ 组。有关更多详细信息，请参阅 *Cygwin User's Guide*。

   例如，以下命令用于从您系统上的本地帐户设置密码文件 passwd：

```
mkpasswd -l > /etc/passwd
```

3. 运行 ssh-host-config 程序安装。
4. 配置 SSH。 对于所有问题回答是。
5. 运行以下命令来启动 SSH 服务器：

```
net start sshd
```

下一步做什么

Cygwin (sshd) 服务必须使用管理域用户帐户来访问网关服务器。 某些传感器（例如 Microsoft Exchange 传感器）需要此用户帐户。 请完成下列步骤：

1. 运行以下命令来配置域用户帐户：

```
mkpasswd -u [domain_user] -d [domain] >> /etc/passwd
mkgroup -d [domain] >> /etc/group
```

2. 启动 services.msc 程序。 请检查创建的 Cygwin (sshd) 服务的登录属性。 验证该服务是否设置为由管理域用户帐户运行。
3. Cygwin (sshd) 配置和日志文件必须由 Cygwin (sshd) 服务用于访问网关的同一个域用户帐户拥有。 请运行以下命令：

```
$ chown [domain_user] /var/log/sshd.log
$ chown -R [domain_user] /var/empty
$ chown [domain_user] /etc/ssh*
```

4. 域用户帐户必须对网关服务器具有以下许可权：
   - 调整进程的内存配额
   - 创建令牌对象
   - 作为服务登录
   - 替换进程级别令牌

如果要发现多个 Windows 服务器，可能会遇到以下消息：

```
A Working gateway cannot be found
```

有关其他可能有帮助的配置的更多信息，请参阅《TADDM 故障诊断指南》中的『网关问题』主题。

配置 Remotely Anywhere

您可以使用 Remotely Anywhere 来提供对 Windows 系统的 SSH 访问。

关于此任务

有关受支持的 Remotely Anywhere 版本的更多信息，请参阅《TADDM 安装指南》中的『Windows 网关』主题。

对于基于网关的发现，Remotely Anywhere 必须安装在网关系统上。

对于直接 SSH 发现，Remotely Anywhere 必须安装在每个 Windows 系统上。

可以在 Remotely Anywhere 中使用缺省配置值。 有关更多信息，请转至 http://remotelyanywhere.com/。

配置 Tectia SSH 服务器

可以使用 Tectia SSH 服务器来提供对 Windows 系统的 SSH 访问。

关于此任务

有关受支持的 Tectia SSH Server 版本的更多信息，请参阅《TADDM 安装指南》中的『Windows 网关』主题。

对于基于网关的发现，Tectia SSH Server 必须安装在网关系统上。

对于直接 SSH 发现，Tectia SSH Server 必须安装在每个 Windows 系统上。

在 Tectia SSH Server 中，可以使用缺省配置值。 有关更多信息，请转至 http://www.ssh.com。

Windows Management Instrumentation (WMI) 依赖关系

TADDM 依靠 Windows Management Instrumentation (WMI) 发现 Windows 计算机系统。 可将 TADDM 配置为在 WMI 遇到问题时重新启动 WMI 服务。 如果重新启动 WMI 服务，那么重新启动前正在运行的所有依赖 WMI 的服务也将重新启动。

以下 TADDM 服务器属性控制 WMI 的重新启动。

注：WMI 重新启动的缺省值为 `false`。将以下属性的值设置为 `true` 可能会提供更可靠的 Windows 发现，但您还必须考虑临时停止和重新启动 WMI 服务的潜在负面影响。

- `com.collation.RestartWmiOnAutoDeploy=false`
- `com.collation.RestartWmiOnAutoDeploy.1.2.3.4=false`
- `com.collation.RestartWmiOnFailure=false`
- `com.collation.RestartWmiOnFailure.1.2.3.4=false`

有关 Windows 计算机系统传感器所使用的 TADDM 服务器属性的更多信息，请参阅 TADDM *Sensor Reference* 的 Windows 计算机系统传感器部分中的 *Configuring the collation.properties file* 主题。

PowerShell 会话

Fix Pack 2

要发现 Windows 计算机系统，可以使用 WMI 或 PowerShell 会话。 相比于 WMI 会话，PowerShell 会话 TADDM 发送的访问目标系统的请求数更少，这样减少了记录的事件数。 PowerShell 会话只能配合基于脚本的传感器使用。 如果要开始使用 PowerShell 会话，必须启用 PowerShell 会话，因为缺省情况下禁用 PowerShell 会话。

可以同时使用两种会话。 如果要运行常规发现和基于脚本的发现，那么不能禁用 WMI 会话，因为如果没有 WMI 会话，常规发现将失败。但是，可以排列使用 PowerShell 会话的优先顺序。

要点：如果只运行常规发现，那么不支持 PowerShell 会话。

可以使用以下属性来控制 PowerShell 会话的使用和优先顺序：

- `com.collation.PowerShellAccessEnabled=false`
- `com.collation.WmiAccessEnabled=true`
- `com.collation.PreferPowerShellOverWMI=true`

- com.collation.PowerShellPorts=5985,5986
- com.ibm.cdb.session.ps.useSSL=false
- com.ibm.cdb.session.ps.allowDNS=true
- com.ibm.cdb.session.ps.fallbackToIP=true
- com.collation.PowerShellTimeoutFudge=10000
- Fix Pack 3 com.ibm.cdb.session.ps.urlPrefix=wsman

要启用 PowerShell 会话，请将 com.collation.PowerShellAccessEnabled 属性设置为 true。 缺省情况下，PowerShell 会话优先于 WMI 会话。

要了解有关这些属性的更多信息，请参阅 TADDM 传感器参考资料中的配置 *collation.properties* 文件条目（针对 Windows 计算机系统传感器）。

注：在非常特定的情况下，当您配置了防火墙以便仅允许通过 PowerShell 会话进行通信时，必须打开 PowerShell 端口并配置 Ping 传感器属性。有关详细信息，请参阅《TADDM 用户指南》中的『进行配置以通过防火墙而不使用锚点来发现』。

示例方案

根据发现 Windows 目标系统的方式，可以按以下方式配置上述属性。

- 仅使用支持基于脚本的发现的传感器。 在此类情况下，可以将 com.collation.PowerShellAccessEnabled 属性设置为 true 来启用 PowerShell 会话，将 com.collation.WmiAccessEnabled 属性设置为 false 来禁用 WMI 会话。但是，当 PowerShell 不可用时，会话和发现将失败。
- 使用支持基于脚本的发现和常规发现的传感器。 在此类情况下，请勿禁用 WMI 会话，因为禁用 WMI 会话会导致常规发现失败。 通过将 com.collation.PowerShellAccessEnabled 属性设置为 true 来启用 PowerShell 会话。 如果要尽可能建立 PowerShell 会话，请勿更改 com.collation.PreferPowerShellOverWMI 属性的缺省值。 在此类情况下，TADDM 会创建能够同时使用 PowerShell 功能和 WMI 功能的混合会话。 仅当 PowerShell 会话不能执行常规传感器所请求的任务时才会使用 WMI 会话。

针对占位符发现进行配置

Fix Pack 3

您可以配置 TADDM 以针对基础结构中未发现的依赖关系创建占位符。

占位符是属于基础结构但是不使用缺省设置在 TADDM 中显示的对象。 不显示的原因可能是未发现连接的一端，没有任何传感器支持此类型的对象，或者未针对其创建定制服务器模板。

占位符是 SSoftwareServer 类。 它们具有 hierarchyDomain 和 hierarchyType 属性集。 下表指定属性的值：

表 *35. hierarchyDomain* 和 *hierarchyType* 属性值。

连接端	hierachyDomain 属性值	hierarchyType 层级类型
本地	app.placeholder.client.local	发起连接的命令的名称，例如，Java

表 35. *hierarchyDomain* 和 *hierarchyType* 属性值。 (续)

连接端	**hierachyDomain 属性值**	**hierarchyType 层级类型**
远程	app.placeholder.server.remote	未知

通过使用这些值，您可以过滤业务应用程序的遍历配置中不想要的关系。 有关详细信息，请参阅《TADDM 用户指南》中的『遍历配置』主题。

在创建占位符，然后传感器创建等效的应用程序服务器，或者定制服务器模板时，PlaceholderCleanupAgent 将占位符与发现的应用程序服务器进行合并。

注：您可以在 TADDM 7.3.0.2 中创建占位符，但是存在限制。 因此，建议在 TADDM 7.3.0.3 和更高版本中使用占位符。 不支持将 FP2 中创建的占位符迁移到 FP3。

启用占位符创建

要启用占位符创建，请将以下属性添加到 collation.properties 文件中：

```
com.ibm.cdb.topomgr.topobuilder.agents.ConnectionDependencyAgent2
dependencyPlaceholders=true
```

缺省值为 false。

第一次将此属性设置为 true 时，必须重新启动 TADDM 以针对 LogicalConnection 和 SoftwareServer 类启用扩展属性。这些扩展属性是此功能正常运行所必需的。

如果上述属性设置为 true，那么不需要在 collation.properties 中显式设置下列属性，将会使用它们的缺省编码值。

```
com.ibm.taddm.dependencyPlaceholders.create.localClient.to.remoteServer
=true
```

缺省值为 true。

```
com.ibm.taddm.dependencyPlaceholders.create.remoteClient.to.localServer
=false
```

缺省值为 false。

注：在 collation.properties 中设置这些属性可能会更改占位符行为。

要点：在启用占位符创建时，业务应用程序可能显著增长，并且构建过程可能更长。 要避免此情况，您可以过滤业务应用程序的遍历配置中不想要的关系。

查看占位符

在将过滤器设置为 Placeholders 后，您可以在"库存摘要"窗格中查看占位符。 未发现的依赖关系的占位符位于**软件服务器**选项卡中。

创建定制服务器模板

您可以使用占位符来通过以下方式创建定制服务器模板：

- 通过使用 bizappscli 工具生成的占位符的相关信息。 有关详细信息，请参阅《TADDM 用户指南》中的『用于分析业务应用程序的内容的操作』主题。
- 通过使用"详细信息"窗格中**运行时**选项卡上针对类型为 app.placeholder.*.local 的占位符显示的命令行信息。

有关定制服务器模板的更多信息，请参阅《TADDM 用户指南》中的『创建和管理定制服务器模板』主题。

创建没有凭证的第 3 层应用程序服务器

Fix Pack 2

如果您想要发现有关基础结构元素的第 3 层基本信息，不需要提供访问列表中的凭证。可以使用传感器内部模板来创建应用程序服务器。这些模板可由 CustomAppServerTopoAgent 进行处理，或者在发现运行期间由定制服务器模板传感器进行处理。

关于此任务

通过创建没有凭证的应用程序服务器，您只能发现有关基础结构的基本信息，例如，所安装的软件种类。如果您不想为第 3 层发现提供凭证，但想要发现有关基础结构的基本信息，请选择此方式。

可使用两种方法来创建第 3 层应用程序服务器。可以运行定制服务器模板传感器，或启用 CustomAppServerTopoAgent。

过程

- 使用定制服务器模板传感器来运行发现
  请完成下列步骤：
  1. 在 collation.properties 文件中，将 com.collation.internaltemplatesenabled 属性设置为 true。该属性将启用第 3 层传感器的内部模板。缺省值为 false。
  2. 通过使用不包含传感器的概要文件来运行发现，通常会发现您想要使用定制服务器模板传感器来发现的信息。例如，如果您想要发现 DB2 服务器的基本信息，请选择第 2 层概要文件发现，或选择您自己的不包含 IBM DB2 传感器的概要文件。如果概要文件包含 IBM DB2 传感器，那么将会运行此传感器，而不是运行定制服务器模板传感器。
- 运行 CustomAppServerTopoAgent
  CustomAppServerTopoAgent 使用通用服务器传感器之前发现的运行时进程。您可以手动运行该代理程序，也可以将其设置为自动运行。请完成下列步骤：
  1. 对于该代理程序的手动方式和自动方式，请在 collation.properties 文件中将 com.collation.internaltemplatesenabled 属性设置为 true。该属性将启用第 3 层传感器的内部模板。缺省值为 false。
  2. 要手动启动 CustomAppServerTopoAgent，请运行以下命令：
     ```
     COLLATION_HOME/support/bin/runtopobuild.sh -a CustomAppServerTopoAgent
     ```
  3. 要设置该代理程序的自动运行，请设置 collation.properties 文件中的 com.ibm.cdb.topobuilder.groupinterval.discovery= 属性。
     该属性指定该代理程序的运行频率。缺省情况下未提供任何值，这表示该代理程序处于禁用状态。要将其启用，请指定以小时为单位的值，例如 com.ibm.cdb.topobuilder.groupinterval.discovery=4。
- 可选： 选择要从处理中排除的模板
  如果您只想启用第 3 层传感器的部分内部模板，可以使用以下属性进行控制：
  ```
  com.collation.discovery.ignoreTemplateList
  ```

该属性指定您不想处理的内部模板的列表。该属性的值是以分号分隔的模板名称的列表，例如 com.collation.discovery.ignoreTemplateList=DB2Unix;MSSQL。您可以在数据管理门户网站的**对象名称**字段中找到内部模板名称，该字段位于**其他数据库服务器**的**详细信息**面板中的**常规**选项卡内。

配置位置标记

位置标记指示每个配置项 (CI) 的创建位置。 这使您能够在 BIRT 报告和 API 查询中对配置项进行基于位置的过滤。

如果启用了位置标记，那么发现数据库中存储的每个所发现对象都将包含 **locationTag** 属性（字符串）。 在特定情况下，由拓扑代理程序创建的关系、聚集对象和继承对象之类的对象将包含位置标记数据：

- 一对一关系（例如依赖关系或网络连接）包含位置标记（如果所连接的两个对象的位置相同）。
- 聚集对象（例如集群）包含位置标记（如果所聚集的所有对象的位置相同）。

  注：对于定制集合，仅当其所有核心 CI 的位置标记的值都相同时，才会设置 **locationTag** 属性。 如果使用具有另一位置标记的核心 CI 扩展了定制集合，那么将清除这种定制集合的 **locationTag** 属性。
- 一个简单对象包含其所基于的对象中的位置标记。

在所有其他情况下，由拓扑代理程序创建的对象都不包含位置标记值。

要启用位置标记，请在 collation.properties 文件中设置以下属性：

```
com.ibm.cdb.locationTaggingEnabled=true
```

位置标记值可以是静态的（对特定的服务器或锚点指定），也可以是动态的（对特定的发现或 IdML 簿导入指定）。 位置标记值限长 192 个字符。 如果所指定位置标记的长度超过 192 个字符，那么系统会将其截断为要求的长度。

限制

运行第 1 层 L1 发现时，将不会更新已在数据库中存在的配置项。 因此，位置标记将仅分配给新发现的对象。

静态位置标记

静态位置标记根据 TADDM 或锚点服务器的静态配置对使用 IdML 簿导入功能发现或装入的所有对象指定 **locationTag** 属性。

TADDM 服务器

要为 TADDM 服务器上创建的 CI 配置位置标记值，请在 collation.properties 文件中指定以下属性：

```
com.ibm.cdb.locationTag=location
```

其中，**location** 是所要使用的位置标记值。

锚点

要为锚点处创建的 CI 配置位置标记值，请在 $COLLATION_HOME/etc/anchor.properties 文件中配置 **anchor_location_n** 属性。 anchor.properties 文件中的下列示例条目指示了如何设置锚点的位置信息：

```
anchor_host_1=192.168.1.13
anchor_scope_1=FIRST_SCOPE
anchor_zone_1=FIRST_ZONE
anchor_location_1=FIRST_LOCATION
anchor_host_2=192.168.2.22
anchor_scope_2=SECOND_SCOPE
anchor_location_2=SECOND_LOCATION
Port=8497
```

如果未对锚点指定位置标记，那么在锚点处创建的每个 CI 的位置都设置为对这些 CI 所连接的 TADDM 服务器指定的位置。

如果未对锚点或 TADDM 服务器指定位置标记值，那么没有为该 CI 设置位置信息。

动态位置标记

动态位置标记使用对特定发现或 IdML 簿导入指定的值来设置 **locationTag** 属性。

发现

要在发现期间指定位置标记值，请从命令行启动发现，并使用可选的 **-l** 或 **-myLocation** 选项来指定位置标记，如以下示例所示：

```
api.sh -u administrator -p collation discover start -n discovery1 -p myProfile -l myLocation myScope
```

其中，**locationTag** 是所要使用的位置标记值。 您指定的值将覆盖此特定发现期间创建的对象的任何静态位置标记值。

注：如果未在 collation.properties 文件中启用位置标记，那么在发现请求期间指定位置标记将引起发现异常。

IdML 簿导入

要在导入 IdML 簿时指定位置标记值，请使用可选的 -l 选项来指定位置标记，如以下示例所示：

```
loadidml.sh -f idml_book.xml -l locationTag
```

其中，**locationTag** 是所要使用的位置标记值。 如果要导入多个具有不同位置标记的 IdML 簿，那么各个簿必须分别装入。

访问列表

您可以创建指定了位置标记的访问列表条目。

位置标记属性是必需的，但以后可以更改。 凭证按位置进行过滤，这就是只使用特定位置的访问条目的原因。 这限制了窥探来自其他客户或位置的密码的可能性。如果在未指定位置标记的情况下运行发现，那么不会使用任何已标记的凭证。

如果添加了位置标记设置为星号字符 (*) 的新访问条目，那么与端点建立会话时，该条目将用作发现期间尝试的最后一个访问权条目。

星号字符 (*) 是缺省值，您可以通过设置以下参数进行更改：

```
com.ibm.cdb.locationTag.global=GLOBAL
```

在这种情况下，带有 GLOBAL 标记的访问条目是运行发现时尝试的最后一个条目。 上述位置标记仅用于访问列表，而不会影响向发现期间发现的 CI 分配的位置标记。

BIRT 报告

可以对 Business Intelligence and Reporting Tools (BIRT) 报告进行过滤，以便为特定的客户位置生成数据。

如果启用了位置标记，那么 BIRT 报告窗格的报告列表下方将显示一个文本字段。 您可以对任何位置标记运行 BIRT 报告，以使其只能看到属于该位置的数据。

没有任何现成报告能够处理位置标记。 如果您需要使用 BIRT 报告，那么必须以手动方式对其进行更新才能支持按位置标记进行过滤。

维护与调整

要使 TADDM 的性能达到最优化，可能需要执行其他配置步骤以及持续性的维护任务。

批量装入参数调整

可以通过在运行时指定特定参数或配置 bulkload.properties 文件来定制批量装入程序的行为。

通过批量装入程序装入数据时有三个不同阶段：

1. 分析对象和关系以确定数据中的图形。

   通常为执行时间的 1 - 5%
2. 构造模型对象和构建图形。

   通常为执行时间的 2 - 5%
3. 将数据传递给应用程序编程接口 (API) 服务器。

   通常为执行时间的 90 - 99%

装入数据有两个选项：

- 可以按一次一条记录的方式装入数据。 这是缺省方式。对于以下文件，必须一次装入一条记录：
  - 有错误的文件。
  - 具有扩展属性的文件。
- 可以批量装入数据。 这称作图形写入，因为装入的是整个图形，而不是单单一条记录。

  使用图形写入选项的批量装入快于一次一条记录的装入。 （请参阅"批量装入"度量值以获取详细信息）。 以下示例显示了图形写入选项，其中 -g=buffer，数据块传递给 API 服务器：

  ```
  ./loadidml.sh -g -f /home/configia/testfiles/sample.xml
  ```

  批量装入数据时，bulkload.properties 中的以下参数可用于提高性能：

```
com.ibm.cdb.bulk.cachesize=2000
```

cachesize 参数控制在使用图形写入选项批量装入时，要在单个写操作中处理的对象数。 增大高速缓存大小值将提高性能，但存在耗尽客户机或服务器上内存的风险。仅当提供了特定信息指示使用较大的高速缓存处理文件对性能有益时，才更改此数字。 缺省高速缓存大小值为 2000，最大高速缓存大小值为 40000。

```
com.ibm.cdb.bulk.allocpoolsize=1024
```

该值指定了可分配给"批量装入程序"进程的最大内存量。 它是传递给"批量装入程序"主 Java 类的一个 Xmx 值。 指定该值（以兆字节为单位）。

请确保 Java 虚拟机有足够的内存。 可以通过收集 TADDM 进程的线程转储并进行复审来执行此操作。 如果需要，请增加内存大小。

提示：在 ITNMIP 簿上运行的测试指示将批量装入流程属性和参数设置为以下值时可以实现最优性能：

```
com.ibm.cdb.bulk.cachesize=4000
com.ibm.cdb.bulk.allocpoolsize=4096
value-Xms768M|-Xmx1512M|-DTaddm.xmx64=6g|
```

在批量装入流程期间频繁运行 **RUNSTATS** 命令也是非常重要的。

数据库维护

要使系统保持最佳性能，您必须为 TADDM 数据库规划并完成所安排的定期维护和调整。

缺省数据库配置

TADDM 随附的缺省数据库配置足以支持 TADDM 的概念证明实现、技术证明实现以及小型试验性实现。

DB2 和 Oracle 数据库的调整准则

以下调整准则同时适用于 DB2 和 Oracle 数据库：

1. 请勿仅根据存储容量来限制可用于数据库的物理磁盘驱动器数。
2. 理想情况下，以下组件应该放在不同的磁盘驱动器或阵列上：
   - 应用程序数据（如表和索引）
   - 数据库日志
   - 数据库临时空间：用于对操作进行排序和连接
3. 使用可用于日志文件的速度最快的磁盘。
4. 在操作系统级别启用异步 I/O。

有关 DB2 和 Oracle 数据库调整的更多信息，请参阅位于 http://www.redbooks.ibm.com/redbooks/pdfs/sg245511.pdf 上的 *Database Performance Tuning on AIX*。

有关特定于 DB2 数据库调整的更多信息，另请参阅位于 http://www-01.ibm.com/support/docview.wss?uid=tss1wp100764 的 *Relational Database Design and Performance Tuning for DB2 Database Servers* 和位于 http://www.ibm.com/support/docview.wss?rs=71&uid=swg27009554 的 *DB2 UDB Version 8 Product Manuals*。

删除旧的数据库记录

表中的数据记录数随时间推移而增加，根据可用的存储空间量，有时您可能希望手动除去数据，以使各个表保持在较小的大小。 在清空 CHANGE_HISTORY_TABLE 之后，可以除去 CHANGE_CAUSE_TABLE 中的相应条目。 另外，还可以通过从 ALIASES_JN 表中删除旧记录来提高数据完整性工具的性能和易用性。

从 *CHANGE_HISTORY_TABLE* 和 *CHANGE_CAUSE_TABLE* 中删除记录：

可以除去旧记录以提高性能，并保持表处于较小的大小。 从 CHANGE_HISTORY_TABLE 中除去记录之后，可以安全地从 CHANGE_CAUSE_TABLE 中除去相应的条目。

要释放 TADDM 数据库中的存储空间，请使用 SQL 查询从 CHANGE_HISTORY_TABLE 中手动除去旧数据。 以下命令是此类 SQL 查询的示例，其中整数 1225515600000 表示日期 2008 年 11 月 1 日，其格式与 System.currentTimeMillis() Java 方法返回的格式相同，也即当前时间与 1970 年 1 月 1 日午夜 UTC 之间的差值（以毫秒为单位）：

```
DELETE FROM CHANGE_HISTORY_TABLE
WHERE PERSIST_TIME < 1225515600000 (this is the Java time stamp)
```

要将日期转换为 Java 时间戳记，请使用以下代码：

```
import java.util.*;
import java.text.*;
import java.sql.Timestamp;

public class DateToString {

  public static void main(String args[]) {
  try {
     String str = args[0];
     SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");
     Date date = formatter.parse(str);

    long msec = date.getTime();

    System.out.println("Date is " +date);
    System.out.println("MillSeconds is " +msec);

    } catch (ParseException e)
    {System.out.println("Exception :"+e);    }

  }
}
```

按如下所示运行代码：

```
java DateToString 1/11/2008
Date is Sat Nov 01 00:00:00 EST 2008
MillSeconds is 1225515600000
```

在 SQL 查询中使用生成的 Java 时间戳记。

如果 CHANGE_HISTORY_TABLE 中存在异常的记录数目，您可能希望执行递增删除（一次删除一部分记录）以防止填满数据库中的事务日志。

在清空 CHANGE_HISTORY_TABLE 之后，可以安全地除去 CHANGE_CAUSE_TABLE 中的相应条目。 CHANGE_CAUSE_TABLE 是用于传播更

改的链接表。 例如，如果向操作系统添加了新的软件组件，那么这个表将此更改链接到运行该操作系统的计算机系统。您可以使用以下命令从 CHANGE_CAUSE_TABLE 中除去记录：

```
delete from change_cause_table where cause_id not in (select id from change_history_table)
```

除去数据的时间范围

要限制数据库随时间增长，您可以管理 TADDM 存储的变更历史记录数据的大小。 在确定从变更历史记录表中除去数据的最佳时间范围时，请考虑该变更历史记录数据的用途以及该变更历史记录信息是否由其他应用程序使用。

如果变更历史记录信息正由另一应用程序使用，请确保您执行应用程序同步的频率高于 CHANGE_HISTORY_TABLE 中保留变更历史记录数据的周数。

以下示例演示了一些典型场景：

- 如果要使用变更历史记录数据来确定问题，并且希望调查五周前发生的问题，那么应在 CHANGE_HISTORY_TABLE 中至少保留五周的数据。
- 如果每周同步一次 Tivoli Business Service Manager (TBSM)，请在 TADDM 变更历史记录表中保留一周以上的变更历史记录数据。

请务必注意，在同步服务器部署中，如果域服务器上存在大量变更历史记录数据，将增加完全同步完成所花费的时间。

同步服务器部署中的数据维护

在域服务器部署中，您可以只根据域的数据需求来做出数据维护决策。 但在同步服务器部署中，您必须在每个域服务器数据库和该同步服务器数据库之间协调变更历史记录数据的除去操作，并且必须从所有这些数据库中除去该数据。

在同步服务器部署中，请使用以下数据维护准则：

- 在域级别保留变更历史记录数据的时间段应大于域服务器数据库与该同步服务器数据库之间各调度的同步之间的时间段。例如，如果同步调度为每周执行一次，请在每个域服务器数据库中保留至少两周的变更历史记录数据。
- 先从域服务器数据库中除去数据。 然后从同步服务器数据库中除去数据。
- 最佳实践是在所有 TADDM 数据库中保留相同周数的变更历史记录数据。 但同步服务器数据库中变更历史记录数据保留的时间段可以与这些数据在域服务器数据库中保留的时间段不同。
- 确定满足环境特定需求的数据除去操作的时间范围之后，最佳实践是在域服务器数据库和同步服务器数据库之间发生同步后立即除去数据。

从 *ALIASES_JN* 表中删除记录：

从 ALIASES_JN 表中删除旧记录后，可以提高数据完整性工具的性能和易用性，并可以释放数据库的更多空间。

关于此任务

ALIASES_JN 表包含对 ALIASES 表进行的更改的历史记录。 数据完整性工具需要收集的数据在数据库中查找可能的配置项过度合并。 随着时间推移，ALIASES_JN 表中的数据记录数将增加到相当大。 这个表的大小影响数据完整性工具的性能和易用性，并且增加 TADDM 数据库的存储空间需求。

拓扑代理程序 AliasesJnTableCleanup 会清除 ALIASES_JN 表。

缺省情况下，会移除 30 天之前的所有行。 通过配置 collation.properties 文件中的以下属性，可更改用于删除记录的天数限制：

```
com.ibm.cdb.topomgr.topobuilder.agents.AliasesJnTableCleanupAgent.removeOlderThanDays=30
```

如果将此属性设置为 -1，那么将禁用该代理程序。如果天数值设置过低，那么具有过度合并选项的验证数据工具可能无法生成完整的结果。

缺省情况下，该代理程序的运行时间不超过 1800 秒（30 分钟）。 如果此时间长度不够而无法移除所有旧行，那么在下次运行该代理程序时，会尝试删除剩余的旧行。通过配置 collation.properties 文件中的以下属性，可以设置代理程序的超时值：

```
com.ibm.cdb.topomgr.topobuilder.agents.AliasesJnTableCleanupAgent.timeout=1800
```

DB2 数据库维护

必须定期维护 TADDM DB2 数据库，以确保可接受的性能。

关于此任务

以下 DB2 实用程序可用：

REORG

在经过插入、删除和更新可变长度列等活动而对表数据造成的诸多更改后，逻辑上有序的数据可能会位于无序的物理数据页面上。 因此，数据库管理器必须执行额外的读操作才能访问数据。使用 **REORG** 实用程序可重新组织 DB2 表以清除分段并回收空间。 如果完成 **RUNSTATS** 的时间超过一般耗用时间，或 DB2 **REORGCHK** 命令指示需要 **REORG** 实用程序，那么请根据需要使用该实用程序。 请在运行 **REORG** 实用程序时关闭 TADDM 服务器，这是因为在脱机表或索引重组（数据碎片整理）期间，应用程序可以访问但不会更新表中的数据。 由于 TopologyBuilder 频繁运行，即使未运行发现，此类锁定也可能会在应用程序中产生不可预测的结果。

RUNSTATS（手动统计信息收集）

DB2 优化器使用 DB2 目录中的信息和统计信息根据提供的查询来确定对数据库的最佳访问。 运行 **RUNSTATS** 实用程序时，将为本地数据库中的特定表和索引收集统计信息。 如果大量添加或除去表中的行，或者为其收集统计信息的列中的数据已更新，那么必须使用 **RUNSTATS** 命令来更新统计信息。 为获得最佳性能，请每周完成一次 **RUNSTATS** 任务，或者在数据库活动频繁的情况下每日完成一次。 缺少更新的统计信息可能会导致 TADDM 中发生严重的性能下降。 您可以在 TADDM 服务器运行时运行 **RUNSTATS** 实用程序。 TADDM 需要特定的 **RUNSTATS** 格式（具体见下文），并且必须关闭 DB2 **AUTO_RUNSTATS** 选项。

AUTO_RUNSTATS（自动统计信息收集）

您可以启用自动化统计信息收集，也称作 **auto-runstats**，以让 DB2 决定是否必须更新 TADDM 数据库统计信息。 **RUNSTATS** 实用程序在后台运行，且数据库统计信息始终保持最新。

要启用自动统计信息收集，您必须将参数 AUTO_MAINT、AUTO_TBL_MAINT 和 AUTO_RUNSTATS 设置为 ON。 运行以下命令：

```
CONNECT TO <db alias>
UPDATE DB CONFIG USING AUTO_MAINT ON AUTO_TBL_MAINT ON AUTO_RUNSTATS ON
```

其中，*<db alias>* 是数据库的名称。

限制：仅当已安装 DB2 APAR IT05733，且设置 DB2_SELECTIVITY=DSCC 参数时，才能使用此实用程序。 DB2 APAR IT05733 包含在以下和更新版本的 DB2 发行版中：

- 9.7 Fix Pack 11
- 10.1 Fix Pack 6
- 10.5 Fix Pack 7

要在 DB2 V10.*x* 上设置 DB2_SELECTIVITY=DSCC，请运行以下命令：

```
db2set -immediate DB2_SELECTIVITY=DSCC
```

注：DB2 9.7 不支持 -immediate 参数。 要在此版本中设置 DB2_SELECTIVITY=DSCC 参数，请运行 **db2set DB2_SELECTIVITY=DSCC** 命令并重新启动 DB2。

注：如果 TADDM 用户在 TADDM 安装中升级 DB2 版本，那么还应更新兼容的驱动程序版本。您可以向您的 DBA 请求 TADDM DB2 服务器的 db2jcc.jar，或在此处下载您的 DB2 版本的相应的文件：http://www-01.ibm.com/support/docview.wss?uid=swg21363866 拥有文件后，请停止 TADDM，将文件复制到 dist/lib/jdbc/ 并确认权限正确以便 TADDM 用户能够读取文件，然后启动 TADDM。在您的环境中的所有 TADDM 服务器上重复该步骤。

DB2 HEALTH MONITOR

针对 TADDM 数据库运行 DB2 运行状况监视器是非常好的做法，它会主动监视情况是否发生变更，指示是否需要 **RUNSTATS**、**REORG** 或任何其他调整。 运行状况监视器可以向数据库管理员发出可能存在系统运行状况问题的警报。 运行状况监视器主动检测可能导致硬件故障或不可接受的系统性能或功能的问题。由于采取主动运行状况监视，您可以解决问题以避免影响系统性能。

DB2 PERFORMANCE ANALYSIS SUITE

当发现可疑的 DB2 问题时，Performance Analyst 工具可以快速分析在发生问题时捕获的 DB2 快照，并提出建议操作。 您可以在 https://www.ibm.com/developerworks/community/groups/community/perfanalyst 下载该工具。

要捕获 TADDM 的 DB2 快照，请完成以下步骤：

1. 从 DB2 服务器连接到 TADDM 数据库，然后运行以下命令：

   ```
   db2 -tf updmon.sql
   ```

   其中，updmon.sql 文件包含以下条目：

```
UPDATE MONITOR SWITCHES USING BUFFERPOOL ON ;
UPDATE MONITOR SWITCHES USING LOCK       ON ;
UPDATE MONITOR SWITCHES USING SORT       ON ;
UPDATE MONITOR SWITCHES USING STATEMENT  ON ;
UPDATE MONITOR SWITCHES USING TABLE      ON ;
UPDATE MONITOR SWITCHES USING UOW        ON ;
UPDATE MONITOR SWITCHES USING TIMESTAMP  ON ;
RESET MONITOR ALL
```

2. 在完成第 1 步后，运行"DB2 get monitor switches"命令以检查是否已全部设置。 其状态必须全部为 ON。
3. 运行该过程，您将遇到性能问题。
4. 当缓慢进程正在运行时，按合适的时间间隔从 DB2 运行以下命令：

   ```
   db2 get snapshot for all on <dbname> > <dbname>-dbsnap.out
   ```

   从步骤 1 中运行命令的同一窗口运行此命令。无法使用脚本运行此命令。
5. 通过每次使用不同的已添加时间戳记的输出文件来运行这些快照。 按进程中有三个或四个快照的时间间隔运行这些快照，但是运行之间的时间不超过 1 个小时。

收集快照后，使用 Performance Analyst 工具对其进行分析，从最后一个快照开始。 例如，运行多次的查询的语句选项卡上的高 CPU 和高平均执行时间通常指示使用 **RUNSTATS** 实用程序可以解决的优化问题。 表选项卡上的高溢出百分比可以指示需要 **REORG** 实用程序。 检查缓冲池选项卡以确保没有警报，太小的缓冲池可能会导致性能低下。

在开始之前

在执行了会导致模式更改的任何主要维护之后（例如，应用了修订包之后），必须在 TADDM 存储服务器上生成 TADDM_table_statistics.sql 文件。 必须定期执行的 **RUNSTATS** 数据库维护任务需要此文件。 由于在处理具有大量常用前缀（如在 TADDM 中广泛使用的类名）的列时存在 DB2 限制，因此 TADDM 需要特殊格式才能更新数据库统计信息。 因此，请勿使用 DB2 **AUTO_RUNSTATS** 选项，应使用通过完成以下步骤生成的 **RUNSTATS** 语法。 但是，如果已安装 DB2 APAR IT05733，且设置了 DB2_SELECTIVITY=DSCC 参数，则可以使用 **AUTO_RUNSTATS** 选项。

注：下列指示信息针对 Linux 和 UNIX 操作系统。 要在 Windows 操作系统上执行数据库维护，请使用相应的 .bat 脚本而非 .sh 脚本。

要生成 TADDM_table_stats.sql 文件，请完成以下步骤：

1. 运行以下命令：

   ```
   cd $COLLATION_HOME/bin
   ```
2. 运行以下命令，其中 *tmpdir* 是可创建此文件的目录：

   ```
   ./gen_db_stats.jy > tmpdir/TADDM_table_stats.sql
   ```

   在流式服务器部署中，对主存储服务器运行此命令。
3. 将此文件复制到数据库服务器，或向数据库管理员 (DBA) 提供此文件以运行"过程"中的第 2 步所示的 TADDM 数据库。 如果对任何表做出了较大更改，那么至少每周（或以更高频率）更新一次数据库统计信息。

过程

要在 DB2 数据库上执行维护，请完成以下步骤：

1. 要使用 **REORG** 实用程序，请完成以下步骤：
   a. 在数据库服务器上，将下列会生成 **REORG TABLE** 命令的 SQL 查询放入某个文件中：

      ```
      select  'reorg table '||CAST(RTRIM(creator) AS VARCHAR(40))||'.
       "'||substr(name,1,60)||'" ; ' from sysibm.systables where creator
       = 'dbuser' and type = 'T' and name not in ('CHANGE_SEQ_ID')
       order by 1;
      ```

      其中 *dbuser* 是来自 com.collation.db.user= 的值。

      注：请确保 *dbuser* 的字母大小写与数据库中 sysibm.systables 表的 creator 列中指定的值相同。
   b. 停止 TADDM 服务器。
   c. 在 DB2 命令行中，连接至数据库并运行以下命令：

      ```
      db2 -x -tf temp.sql > cmdbreorg.sql
      db2 -tvf cmdbreorg.sql > cmdbreorg.out
      ```
   d. 检查 cmdbreorg.out 文件中的错误，确保 **REORG** 实用程序成功。
   e. 启动 TADDM 服务器。
2. 要使用 **RUNSTATS** 实用程序，请完成以下步骤。 每周至少自动执行一次要运行的过程。
   a. 在数据库服务器上，通过使用先前生成的输出来运行特定于 TADDM 的 **RUNSTATS** 命令：

      ```
      db2 -tvf tmpdir/TADDM_table_stats.sql  > table_stats.out
      ```
   b. 通过检查 table_stats.out 文件来查找错误，确保 **RUNSTATS** 实用程序成功。

DB2 for z/OS 数据库维护

这些维护和调整准则适用于 IBM DB2 for z/OS® 数据库。

过程

这些准则假定 *DB_USER* 是主 DB2 数据库用户标识，*ARCHIVE_USER* 是辅助 DB2 数据库用户标识。

1. 使用发现管理控制台来运行发现。 此方法会在域数据库中填充数据。
2. 停止 TADDM 服务器。
3. 为 TADDM 使用的每个表空间生成并运行 REORG 控制语句。

   ```
   SELECT 'REORG TABLESPACE '||DBNAME||'.'||NAME FROM SYSIBM.SYSTABLESPACE
    WHERE CREATOR IN ('DB_USER', 'ARCHIVE_USER') ORDER BY 1;
   ```
4. 为 TADDM 使用的索引生成并运行 REORG 控制语句。

   ```
   SELECT 'REORG INDEX '||CREATOR||'.'||NAME FROM SYSIBM.SYSINDEXES
    WHERE CREATOR IN ('DB_USER', 'ARCHIVE_USER');
   ```
5. 为 TADDM 使用的表空间生成并运行 RUNSTATS 控制语句。

   ```
   SELECT 'RUNSTATS TABLESPACE '||DBNAME||'.'||NAME||' INDEX(ALL)
   SHRLEVEL REFERENCE' FROM SYSIBM.SYSTABLESPACE
   WHERE CREATOR IN ('DB_USER', 'ARCHIVE_USER') ORDER BY 1;
   ```
6. 为每个 TADDM DB 用户重新生成并运行 UPDATE 索引统计信息语句。

```
SELECT 'UPDATE SYSIBM.SYSINDEXES SET FIRSTKEYCARDF=FULLKEYCARDF
WHERE NAME = '||''''||CAST(RTRIM(name) AS VARCHAR(40))||''''||'
AND CREATOR = '||''''||CAST(RTRIM(creator) AS VARCHAR(40))||''''||'
AND TBNAME = '||''''||CAST(RTRIM(tbname) AS VARCHAR(40))||''''||'
AND TBCREATOR = '||''''||CAST(RTRIM(tbcreator) AS VARCHAR(40))||''''||';'
from sysibm.sysindexes a
where tbcreator in ('DB_USER', 'ARCHIVE_USER')
AND   NAME IN
(SELECT IXNAME
FROM SYSIBM.SYSKEYS B
WHERE A.CREATOR = B.IXCREATOR
AND A.NAME    = B.IXNAME
AND COLNAME = 'PK__JDOIDX')
AND TBNAME IN
(SELECT NAME
FROM SYSIBM.SYSTABLES C
WHERE A.TBCREATOR = C.CREATOR
AND A.TBNAME = C.NAME
AND CARDF > 0);
```

其中，DB_USER 是主 DB2 数据库用户标识，而 ARCHIVE_USER 是辅助 DB2 数据库用户标识。

7. 为每个 TADDM DB 用户重新生成并运行 UPDATE 列统计信息语句。

```
SELECT 'UPDATE SYSIBM.SYSCOLUMNS SET COLCARDF=(SELECT FULLKEYCARDF FROM
SYSIBM.SYSINDEXES WHERE NAME = '||''''||CAST(RTRIM(name)
AS VARCHAR(40))||''''||'
AND CREATOR = '||''''||CAST(RTRIM(creator) AS VARCHAR(40))||''''||'
AND TBNAME = '||''''||CAST(RTRIM(tbname) AS VARCHAR(40))||''''||'
AND TBCREATOR = '||''''||CAST(RTRIM(tbcreator) AS VARCHAR(40))||''''||')
WHERE NAME = '||''''||'PK__JDOIDX'||''''||' AND TBNAME = '||''''||
CAST(RTRIM(tbname) AS VARCHAR(40))||''''||'
AND TBCREATOR = '||''''||CAST(RTRIM(tbcreator) AS VARCHAR(40))||''''||';'
from sysibm.sysindexes a
where tbcreator in ('DB_USER', 'ARCHIVE_USER')
AND   NAME IN
(SELECT IXNAME
FROM SYSIBM.SYSKEYS B
WHERE A.CREATOR = B.IXCREATOR
AND A.NAME    = B.IXNAME
AND COLNAME = 'PK__JDOIDX')
AND TBNAME IN
(SELECT NAME
FROM SYSIBM.SYSTABLES C
WHERE A.TBCREATOR = C.CREATOR
AND A.TBNAME = C.NAME
AND CARDF > 0);
```

8. 根据 TADDM 的使用情况定期监视最大的表，并根据需要调整其存储属性。 特别是，监视以下数据库表的大小，这些表可能会变得非常大：
   - ALIASES
   - CHANGE_CAUSE_TABLE
   - CHANGE_HISTORY_TABLE
   - MSSOBJLINK_REL
   - PERSOBJ
   - SUPERIORS

   根据环境的需求，使用 ALTER 语句来修改 *PRIQTY* 和 *SECQTY* 属性。如果适用的话，请考虑将表移至其他表空间。

9. 在以下程序包上使用 REBIND 命令并搭配 KEEPDYNAMIC(YES) 选项：
   - SYSLH200
   - SYSLH201

- SYSLH202

Oracle 数据库维护

这些维护和调整准则适用于 Oracle 数据库。

1. 在数据库表上运行 dbms_stats 包。 Oracle 使用基于成本的优化器。 基于成本的优化器需要数据来确定存取方案，此数据由 dbms_stats 包生成。 Oracle 数据库取决于有关表和索引的数据。 如果没有此数据，那么优化器必须进行估计。

   重新构建索引并运行 dbms_stats 包对于实现 Oracle 数据库的最佳性能至关重要。填充数据库后，应按照定期调度基准（例如，每周）执行维护。

   - **REBUILD INDEX**：在经过插入、删除和更新活动而对表数据造成诸多的更改后，逻辑上有序的数据可能会位于无序的物理数据页面上，这样，数据库管理器必须执行其他读操作才能访问数据。 重新构建索引以帮助提高 SQL 性能。

     a. 通过在 Oracle 数据库上运行以下 SQL 语句来生成 **REBUILD INDEX** 命令，其中 *dbuser* 是来自 com.collation.db.user= 的值：

        ```
        select 'alter index dbuser.'||index_name||' rebuild tablespace '
        ||tablespace_name||';' from dba_indexes where owner = 'dbuser'
        and index_type not in ('LOB');
        ```

        此语句会生成需要运行的所有 **ALTER INDEX** 命令。

     b. 在 SQLPLUS 或某个同等设施中运行这些命令。 在大型数据库上重新构建索引需要 15 至 20 分钟时间。

2. **DBMS_STATS**：使用 Oracle RDBMS 来收集许多不同种类的统计信息以帮助提高性能。优化器使用字典中的信息和统计信息来基于提供的查询确定对数据库的最佳访问。运行 **DBMS_STATS** 命令时，会为本地数据库中的特定表和索引收集统计信息。如果大量添加或除去了表的行或者您为之收集统计信息的列中的数据已更新，请重新运行 **DBMS_STATS** 命令来更新统计信息。

   - $COLLATION_HOME/bin 目录中的 gen_db_stats.jy 程序会为 Oracle 或 DB2 数据库输出数据库命令，以更新 TADDM 表中的统计信息。 以下示例显示如何使用此程序：

     a. cd $COLLATION_HOME/bin

     b. 运行以下 SQL 语句，其中 *tmpdir* 是创建该文件所在的目录：

        ```
        ./gen_db_stats.jy > tmpdir/TADDM_table_stats.sql
        ```

        在流式服务器部署中，对主存储服务器运行此语句。

     c. 完成此语句后，请将该文件复制到数据库服务器并运行以下命令：

        – 要在 SQLPlus 中执行脚本文件，请输入 @，然后输入文件名：SQL > @{file}

     d. 在 SQLPLUS 或某个同等设施中运行这些命令。

3. 缓冲池：缓冲池或缓冲区高速缓存是 Oracle 系统全局区域 (SGA) 中用于每个实例的内存结构。 此缓冲区高速缓存用于对内存中的数据块进行高速缓存。从内存访问数据比从磁盘访问数据快很多。 块缓冲区调整的目标是对缓冲区高速缓存 (SGA) 中频繁使用的数据块进行有效地高速缓存，并提供更快的数据访问速度。 调整块缓冲区是任何 Oracle 调整启动计划中的关键任务，并且是持续调整和监视生产数据库过程中的一项任务。 Oracle 产品在 SGA 中为每个实例维护它自己的缓冲区高速缓存。 大小合适的缓冲区高速缓存通常可产生 90% 以上的高速缓存命中率，这意味着十分之九的请求无需进入磁盘就可得到执行。 如果缓冲区高速缓存太小，高速缓

存命中率将比较低，并产生更多的物理磁盘 I/O。 如果缓冲区高速缓存太大，那么缓冲区高速缓存的某些部分无法得到充分利用，将会浪费内存资源。

表 *36.* 缓冲池大小准则 *(db_cache_size)*

CI 数量	缓冲池大小准则
< 500,000	38000
500,000 - 1,000,000	60000
> 1,000,000	95000

4. 如果发现或批量装入花费的时间太长而无法完成并且 NRS 包含以下错误，那么可以使最大开放光标数加倍：

```
com.ibm.tivoli.namereconciliation.service.NrsService
getAliases(masterGuid)
SEVERE: NOTE *** SQL State = 60000.   SQL Code  = 604.   SQL Message =
ORA-00604: error occurred at recursive SQL level 1
ORA-01000: maximum open cursors exceeded
ORA-01000: maximum open cursors exceeded
```

5. 验证 Oracle JDBC 驱动程序和 Oracle 服务器的版本是否相同。 必要时，请替换以下位置中的 Oracle JDBC 驱动程序文件。

   注：仅当 BIRT 报告查看器处于启用状态时，这才适用。

   - TADDM 7.3.0 - $COLLATION_HOME/deploy-tomcat/birt-viewer/WEB-INF/platform/plugins/org.eclipse.birt.report.data.oda.jdbc_2.2.1.r22x_v20070919/drivers/
   - TADDM 7.3.0.1 及更高版本 - $COLLATION_HOME/apps/birt-viewer/WEB-INF/platform/plugins/org.eclipse.birt.report.data.oda.jdbc_2.2.1.r22x_v20070919/drivers/
   - $COLLATION_HOME/lib/jdbc/

数据库通信

当数据库不可用时，存储服务器将尝试重新建立连接。

当数据库与存储服务器之间没有连接时，存储服务器将按 com.ibm.cdb.db.timeout 属性所指定的时长进行等待，然后尝试连接到数据库。 重试建立连接的次数由 com.ibm.cdb.db.max.retries 属性指定。

要了解更多有关数据库属性的信息，请转至数据库属性一节。

发现性能调整

您可以在 collation.properties 文件中更新 com.collation.discover.dwcount、com.collation.discover.observer.topopumpcount 和 com.ibm.cdb.discover.observer.topopump.threshold 属性，以影响发现速率以及在 TADDM 数据库中存储发现结果的速率，或者限制负责存储数据的线程数。

有关这些属性的详细信息，请参阅第 80 页的『性能属性』。

如果增大了 com.collation.discover.dwcount 或 com.collation.discover.observer.topopumpcount 属性值，那么可能也必须通过提高下列 Java 虚拟机 (JVM) 的最大堆大小设置来增加已安装的内存量：

对于 dwcount 属性：

- 在流式服务器部署中：
  - 发现
  - DiscoverService
- 在域服务器部署中：
  - 发现

对于 topopumpcount 属性：

- 在流式服务器部署中：
  - StorageService
- 在域服务器部署中：
  - 拓扑

有关更多信息，请参阅第 132 页的『Java 虚拟机：IBM 参数调整』。

有关调整发现性能的更多信息，请参阅标题为调整发现性能的文档（位于以下网址：http://www.ibm.com/software/brandcatalog/ismlibrary/）。

发现率调整

发现率属性是最有可能进行调整的区域。 对性能影响最大的属性是发现工作程序线程数。 您也可以使用进行中的传感器来监视性能或通过指定会话池大小来对性能进行改进。

发现工作程序线程是运行传感器的线程。 以下属性指定了发现工作程序最大线程数：

```
com.collation.discover.dwcount=32
```

如果服务器有足够的闲置容量，您可以增加此数字，并允许更多的传感器并行运行。

进行中的传感器

要监视性能，您可以查看进行中的传感器。 进行中的传感器可处于以下三个执行阶段中的任何一个阶段：

启动阶段
该阶段的传感器正在发现一个或多个 CI。

发现阶段
该阶段的传感器已完成发现一个或多个 CI，但是正在等待要保存到数据存储器中的结果。

存储阶段
该阶段的传感器会将其发现结果保存到数据库中。

要按执行阶段对进行中的传感器进行排序，请单击"描述"列。

通过观察发现运行情况并比较处于启动阶段的进行中的传感器数和处于发现或存储阶段的进行中的传感器数，评估对于特定的环境，属性发现是快于还是慢于属性存储。 与对 collation.properties 文件进行的所有更改一样，必须重新启动服务器才能使更改生效。

示例：

进行中的传感器：启动阶段、发现阶段和存储阶段。

如果发现阶段和存储阶段的传感器总数小于启动阶段的传感器数目，可能会指示该发现为性能瓶颈。

如果发现阶段和存储阶段的传感器总数超过启动阶段的传感器数目，可能会指示该存储为性能瓶颈。

会话和网关池大小

要发现特定 CI 的属性，传感器必须与其主机具有 SSH 或 WMI 会话。 要改善性能，必须汇聚这些会话并对其进行缓存。 在大多数情况下，缺省池大小是足够的；但是如果缺省池大小不够，它们可以限制发现率。您可以将以下属性更改为 true 以监视此情况：

```
com.collation.platform.session.ExtraDebugging=false
```

必须重新启动发现服务器才能使更改生效。 运行发现后，您可以搜索 DiscoverManager 日志，以获取与会话池相关的等待时间问题。 要执行此操作，请搜索 pool lock 的日志。 因会话池争用而引起性能降级的示例如下所示：

```
2006-08-04 16:11:50,733 DiscoverManager [DiscoverWorker-34]
WindowsComputerSystemAgent(192.168.16.181)
INFO session.SessionClientPool -
Session client [3x ssh2:/admlxz@151.179.84.85]#9612508
waited 158.682 seconds for pool lock
```

如果会话的等待时间过长，您可以增加池大小。 有两种方法可增加池大小。 通过编辑 collation.properties 文件中的以下属性对每个主机全局更改会话的池大小：

```
com.collation.platform.session.PoolSize=3
```

但是，争用不太会涉及环境中所有或大多数主机的会话。 争用很可能局限于许多传感器所用的数量较少的大型主机。 发现服务器使用作用域属性，这表示 collation.properties 文件中的许多属性对常规目标使用一个值，而对特定目标使用另一个值。 您可以通过添加 IP 地址或添加发现服务器作用域名称来调整该属性，如以下示例所示：

```
com.collation.platform.session.PoolSize.10.10.250.1=20
```

在此情况下，10.10.250.1 的池大小为 20，而所有其他主机的池大小为 3。您可以查看日志消息（如 DiscoverManager 日志中的消息），并确定哪个主机的缺省会话池大小不够，然后对 collation.properties 文件进行相应更改。

相关设置为网关池大小。 它用于设置发现服务器和 Windows 网关之间所允许的会话数。 您可以通过编辑以下属性来进行指定：

```
com.collation.platform.session.GatewayPoolSize=10
```

如果您的环境主要由 Windows 计算机系统组成，请增加该属性值，从而使其与发现工作程序线程的数目相等。

存储调整

存储是进行调整的第二个主要区域。 如果存储阶段的传感器数接近用于指定并行存储线程数的属性值，那么发现结果的存储将是性能瓶颈。 为了提高性能，您还可以限制负责存储数据的线程数。

以下属性指定了并行存储线程数。 它是控制发现存储性能的其中一个主要设置：

com.collation.discover.observer.topopumpcount

为了提高拓扑代理程序处于运行状态时的存储性能，您可以限制负责在发现期间存储数据的线程数。 这样，发现完成将花费较少时间。 要指定处于运行状态的线程数限制，请在 collation.properties 文件中编辑以下属性：

com.ibm.cdb.discover.observer.topopump.threshold
此属性指定要限制的存储线程数。

com.ibm.cdb.discover.observer.topopump.threshold.<agentGroupName>
此属性指定运行指定的代理程序组时要限制的存储线程数。

下表显示 com.ibm.cdb.discover.observer.topopump.threshold 属性可以提高发现性能的程度。 计算涉及具有 76 000 个配置项的数据库。

阈值属性值	改善时间百分比
0.2	55
0.5	33
0.7	13
1	0

Java 虚拟机：IBM 参数调整

可以设置 Java 虚拟机 (JVM) 参数以减少 Java 堆分段，从而可以帮助提高性能。

当处理的对象数增加时，会出现 Java 堆分段。 您可以设置若干参数来帮助减少堆中的分段。

- kCluster 是专用于类块的存储区。 它的大小足可容纳 1280 个条目。 每个类块为 256 个字节长。 此缺省值通常太小，可能会导致堆分段。 请按如下所示设置 kCluster 参数 **-Xk** 以帮助减少堆分段。 这些是起始值，可能必须在您的环境中进行调整。 最好使用堆转储分析来确定最佳大小。
  - 拓扑：-Xk8300
  - EventsCore：-Xk3500
  - DiscoverAdmin：-Xk3200
  - 代理：-Xk5700
  - 发现：-Xk3700

  通过在"JVM 供应商特定设置"部分中添加条目来在 collation.properties 文件中实现这些更改。 例如，要为拓扑服务器实现这些更改，请添加以下行：

  com.collation.Topology.jvmargs.ibm=-Xk8300
- 解决分段问题的另一个选项是专门为大于 64K 的大对象分配一些空间。 请使用 **-Xloratio** 参数。例如：
  - **-Xloratio0.2**

  该命令将活动 Java 堆的 x%（不是 -Xmx 的 x%，而是 Java 堆当前大小的 x%）保留为仅分配给大对象（≥64 KB)。 如果更改了此参数，那么也应该更改 -Xmx 以确保不会减少小对象区域的大小。 最好使用堆转储分析来确定此参数的最佳设置。

还可以设置几个影响 Java 性能的参数。 要将现有 JVM 选项更改为其他值，请编辑以下某个文件：

- 对于 TADDM 7.3.0 中的域服务器，$COLLATION_HOME/deploy-tomcat/ROOT/WEB-INF/cmdb-context.xml 文件。
- 对于 TADDM 7.3.0.1 及更高版本中的域服务器，$COLLATION_HOME/apps/ROOT/WEB-INF/cmdb-context.xml 文件。
- 对于 TADDM 7.3.0 中的同步服务器，$COLLATION_HOME/deploy-tomcat/ROOT/WEB-INF/ecmdb-context.xml 文件。
- 对于 TADDM 7.3.0.1 及更高版本中的同步服务器，$COLLATION_HOME/apps/ROOT/WEB-INF/ecmdb-context.xml 文件。
- 对于 TADDM 7.3.0 中的发现服务器，$COLLATION_HOME/deploy-tomcat/ROOT/WEB-INF/discovery-server-context.xml 文件。
- 对于 TADDM 7.3.0.1 及更高版本中的发现服务器，$COLLATION_HOME/apps/ROOT/WEB-INF/discovery-server-context.xml 文件。
- 对于 TADDM 7.3.0 中的存储服务器，$COLLATION_HOME/deploy-tomcat/ROOT/WEB-INF/storage-server-context.xml 文件。
- 对于 TADDM 7.3.0.1 及更高版本的存储服务器，$COLLATION_HOME/apps/ROOT/WEB-INF/storage-server-context.xml 文件。

要编辑其中某个文件以更改某项 TADDM 服务的设置，请先在文件中找到该服务。 以下示例显示了 XML 文件中服务定义的开头部分：

```
<bean id="Discover"
 class="com.collation.platform.service.ServiceLifecycle" init-method="start"
 destroy-method="stop">
 <property name="serviceName">
  <value>Discover</value>
 </property>
```

在该定义中，有一些控制 JVM 参数的元素和属性。 例如：

```
<property name="jvmArgs">
 <value>-Xms8M;-Xmx512M;
 -Djava.nio.channels.spi.SelectorProvider=sun.nio.ch.PollSelectorProvider
 </value>
</property>
```

JVM 参数可在以下元素中设置为以分号分隔的列表：

```
<property name="jvmArgs"><value>
```

您还可以修改 collation.properties 文件中的 JVM 属性。这些属性可以具有下列其中一种格式：

com.collation.JVM.jvmargs.VENDOR
这类属性将添加到从 *-config.xml 文件读取的值中。

com.collation.jvmargs.VENDOR
这类属性将添加到所有 TADDM JVM 中。

com.collation.JVM.jvmargs
这类属性将覆盖 *-config.xml 文件中指定的所有值。

其中

- JVM 是 Proxy、Topology、EventsCore、ExcmdbCore、DiscoverAdmin、StorageService 和 DiscoveryService
- VENDOR 是 ibm 或 sun

Java 虚拟机属性调整

在 collation.properties 文件中，应用于 TADDM发现管理控制台的 Java 虚拟机 (JVM) 属性的缺省值以环境中的服务器等效 (SE) 数目为基础。

适用于发现管理控制台的 JVM 属性的缺省值

- 小型环境（少于 1000 个 SE）：
  - com.collation.gui.initial.heap.size=128m
  - com.collation.gui.max.heap.size=512m
- 中型环境（1000-2500 个 SE）：
  - com.collation.gui.initial.heap.size=256m
  - com.collation.gui.max.heap.size=768m
- 大型环境（2500-5000 个 SE）：
  - com.collation.gui.initial.heap.size=512m
  - com.collation.gui.max.heap.size=1024m

网络调整

实施系统后，应该监视网络以确保其带宽使用率不超过 50%。

网络可能影响应用程序的总体性能，如果在以下情境中存在时间延迟，网络通常是造成性能问题的一个因素：

- 客户机系统向服务器发送请求和服务器接收到该请求之间的延迟
- 服务器系统将数据回发给客户机系统和客户机系统接收到该数据之间的延迟

DNS 调优

TADDM 对部署的 DNS 基础结构的性能非常敏感。 即使 DNS 性能对于其他应用程序而言已足够，也仍可能需要进行某些配置以优化 TADDM 的性能。

TADDM 执行大量 DNS 查找查询以解析组件和事件的有意义显示名。 与大多数其他应用程序不同，TADDM 主要执行逆向查找（将 IP 地址映射到名称），而不是执行正向查找（将名称映射到 IP 地址）。

这种用法模式决定了，与 DNS 性能相关的问题对 TADDM 性能产生的影响可能甚于对其他应用程序的影响。例如，DNS 响应时间 500 毫秒可能不会对典型应用程序产生显著影响，但可能由于 TADDM 执行大量 DNS 查询而导致 TADDM 出现明显的性能问题。 另外，由于其他应用程序只执行正向查找，因此，与逆向查找相关的性能问题不会影响大多数应用程序，但将影响 TADDM。

通常，应该解决与 DNS 基础结构相关的性能问题，以使 DNS 服务的所有使用者从中受益。 如果无法解决问题，那么可以通过多种方式缓解 DNS 性能问题对 TADDM 的影响：

- 确保正确配置 in-addr.arpa 逆向查找委托。 委托问题可能会导致 TADDM 服务器尝试访问不存在的服务器，从而导致逆向查找期间出现长时间的暂停或挂起。 此类配置问题仅影响执行逆向查找的应用程序（例如 TADDM）。
- 在 TADDM 服务器系统上至少设置一个高速缓存/转发 DNS 服务器，并将 TADDM 服务器配置为使用该 DNS 服务器进行查找。 这使得 DNS 查找结果可以根据区域的 TTL 规则在本地 TADDM 环境中进行高速缓存。 此类服务器是无状态服务器，因此只要求进行最低限度的维护，并且只添加很小的开销。
- 在 TADDM 服务器系统上至少设置一个 DNS 从属服务器，并将 TADDM 服务器配置为使用该 DNS 服务器进行查找。这使得 DNS 查找能够在本地 TADDM 环境中执行，而不必与更广泛的 DNS 基础结构进行通信。 DNS 从属服务器自动维护状态，因此只要求进行最低限度的维护，并且只添加很小的开销。
- 使用替代查找方法（例如 hosts 文件）来代替 DNS。 （此方法可能要求进行大量维护工作。）

注：请勿更改 java.security 文件中的缺省 DNS 高速缓存参数。虽然高速缓存参数可能影响 DNS 性能，但是，应用 TADDM 维护修订时不会保留对此配置文件进行的更改。 而是，请使用本主题中描述的其中一种方法来优化 DNS 性能。

同步服务器调整

同步服务器的性能高度依赖于数据库处理，因此也高度依赖于数据库维护和调整。 如果在同步处理时遇到性能问题，请参阅有关数据库调整的信息，尤其要注意 DB2 数据库的缓冲池设置、Oracle 数据库的缓冲区高速缓存设置和有关数据库维护的信息。

特别是对于同步服务器，请输入以下命令来更新 DB2 数据库配置：

```
UPDATE DATABASE CONFIG FOR TADDM USING
   UTIL_HEAP_SZ 5000
   LOGBUFSZ 1024
   LOCKLIST 20000
   SORTHEAP 2048
   PCKCACHESZ AUTOMATIC
;
```

Windows 系统调整

要为 TADDM 服务分配更多内存，请调整 Windows 系统。

请完成以下任务：

- 系统调页文件不得与操作系统位于同一个驱动器上。 如果可能，请将系统调页文件放在单独的磁盘驱动器上。
- 配置数据库和应用程序服务器，以使联网应用程序的数据最大化。

报告

通过使用外部报告查看器、JSP 报告查看器或 BIRT 报告系统，可以创建并向数据管理门户网站添加定制报告。

外部报告查看器

使用外部报告查看器可以运行用于生成报告的外部程序。 外部程序通过命令行使用 TADDM API 来访问数据。 接着，报告将显示在用户界面中。

创建外部报告查看器逻辑

可以在任何可执行程序内实现外部报告。 例如，Perl 脚本、shell 脚本或 Java 程序。外部程序必须通过标准输出输出一个有效的 HTML 文件，以便产生的报告能够显示在数据管理门户网站中。

关于此任务

外部报告查看器的典型实现使用 shell 脚本来查询 TADDM API，并将查询的 XML 结果输出到临时文件。 然后，shell 脚本将启动 XSLT 处理器以将查询结果转换为输出到 STDOUT 的 HTML 输出。

要点： 使用 TADDM API 的外部报告查看器必须提供命令行程序 api.sh 脚本（在 Linux 和 UNIX 上）和 api.bat 文件（在 Windows 上）的凭证。 这是因为，凭证是 api.sh 脚本和 api.bat 文件的命令行自变量，系统的其他用户可通过进程列表查看这些凭证。 为了防止泄露敏感密码，设置对应该在外部生成的报告中显示的对象具有读许可权的虚拟帐户可能十分有用。

以下示例是一个简单的外部报告 bourne shell 脚本实现。 请将以下内容复制到新文件 $COLLATION_HOME/sdk/bin/appServers.sh 中，并使该文件可供运行 TADDM 服务器的用户读取和执行：

```
#!/bin/sh
# Set environment variables for called scripts
export COLLATION_HOME=/opt/ibm/taddm/dist

# Invoke the query via API and output to $COLLATION_HOME/sdk/bin/appServers.xml
# NOTE: Change 'restrictedUser' and 'restrictedPassword' to your dummy account
#       credentials.
sh $COLLATION_HOME/sdk/bin/api.sh -l log -H localhost -u restrictedUser -p
restrictedPassword \ find AppServer > $COLLATION_HOME/sdk/bin/appServers.xml

# Invoke the XSLT processor
sh $COLLATION_HOME/sdk/bin/xslt.sh -XSL $COLLATION_HOME/sdk/bin/appServers.xsl
```

以下示例是用于转换 shell 脚本生成的 appServers.xml 文件的 appServers.xsl 样式表。 此报告将显示应用程序服务器名称及其产品版本。 请将这些内容复制到新文件 $COLLATION_HOME/sdk/bin/appServers.xsl 中，并使该文件可供运行 TADDM 服务器的用户读取。

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version = '1.0' xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:coll="urn:www-collation-com:1.0" xmlns:xhtml="http://www.w3.org/1999/xhtml">
        <xsl:variable name="nl">
        <xsl:text>
        </xsl:text>
        </xsl:variable>

        <xsl:variable name="pageheadertext">
                Simple Application Server report
        </xsl:variable>

        <xsl:variable name="pagefootertext">
                End Simple Application Server report
        </xsl:variable>


        <xsl:template match="/">
        <html>
                <head>
                <link rel="stylesheet" type="text/css" media="all"
```

```
       href="styles.css" />
                </head>
         <body>
                <h3>
                        <xsl:value-of select="$pageheadertext"/>
                </h3>
    <table border="1" width="100%">
                 <tr>
                     <th>Product Version/th>
                     <th>Name/th>
                        </tr>

      <xsl:apply-templates select="document('appServers.xml')/coll:results"/>
    </table>
                <xsl:value-of select="$nl"/>
        </body></html>
        </xsl:template>

        <xsl:template match="coll:AppServer">
                        <tr>
                          <td>xsl:value-of select="coll:productVersion"/></td>
                          <td>xsl:value-of select="coll:displayName"/></td>
                        </tr>
        </xsl:template>
</xsl:stylesheet>
```

要测试报告逻辑，请从命令行运行 appServer.sh 脚本。 此时将显示有效的 HTML 输出。

将外部报告查看器添加到数据管理门户网站

通过修改 reports.xml 文件可将报告添加到数据管理门户网站。 reports.xml 文件位于 $COLLATION_HOME/etc/cdm/xml/ 目录。

过程

要将外部报告查看器添加到数据管理门户网站，请完成下列步骤：

1. 使用文本编辑器打开 $COLLATION_HOME/etc/cdm/xml/reports.xml 文件。
2. 在 reports.xml 文件中，指定报告定义的报告描述符、报告组、报告名称以及外部脚本。 以下示例显示如何创建名为 Application Servers 并且位于 Inventory Reports 组中的外部报告，并指定 sdk/bin/appServers.sh 文件：

   ```
   <bean class="com.collation.cdm.reports.viewer.ExternalReportViewer" id="AppServers1">
    <property name="reportGroup"><value>Inventory Reports</value></property>
    <property name="reportName"><value>Application Servers</value></property>
    <property name="script"><value>sdk/bin/appServers.sh</value></property>
   </bean>
   ```

3. 保存 $COLLATION_HOME/etc/cdm/xml/reports.xml 文件。
4. 该报告现在显示在数据管理门户网站中。

JSP 报告查看器

JSP 报告查看器为了解如何编写 Java Server Pages (JSP) 的用户提供了更多的灵活性和安全性。 报告逻辑（包括任何 API 访问）置于一个 JSP 页面中，然后，数据管理门户网站显示该页面。 使用 JSP 报告查看器时，将自动继承已登录的用户的安全凭证。

创建 JSP 报告查看器逻辑

JSP 报告查看器的逻辑包含在由数据管理门户网站调用的 JSP 中。 JSP 报告的典型实现使用名为 TMSDataHelper 的 Java 帮助程序类来查询 TADDM API。 查询结果是

可以使用 Java 方法进行处理的对象。 有关 TADDM API 和模型的更多信息，请参阅 $COLLATION_HOME/sdk/doc 中的 SDK 文档。

关于此任务

以下示例是一个简单的 JSP 报告查看器实现。 将以下内容复制到新文件 $COLLATION_HOME/deploy-tomcat/reports.war/WEB-INF/view/custom.jsp（如果使用的是 TADDM 7.3.0）或 $COLLATION_HOME/apps/reports.war/WEB-INF/view/custom.jsp（如果使用的是 TADDM 7.3.0.1 及更高版本）中，并且使运行 TADDM 服务器的用户能够读取和执行该文件。

以下示例显示用于转换 shell 脚本生成的 appServers.xml 文件的 appServers.xsl 样式表。 此报告将显示应用程序服务器名称及其产品版本。 请将这些内容复制到新文件 $COLLATION_HOME/sdk/bin/appServers.xsl 中，并使该文件可供运行 TADDM 服务器的用户读取。

```
<%@ page language="java" %>
<%@ page import="com.collation.cdm.common.util.TMSDataHelper" %>
<%@ page import="java.lang.StringBuffer" %>
<%@ page import="com.collation.cdm.reports.util.ReportsParser" %>
<%@ page import="com.collation.cdm.common.util.TMSReportingTransformer" %>
<%@ page import="com.collation.platform.model.AttributeNotSetException" %>
<%@ page import="com.collation.platform.model.ModelObject" %>
<%@ page import="com.collation.platform.model.topology.sys.ComputerSystem" %>
<%@ page import="com.collation.platform.model.topology.process.BusinessProcess" %>
<%@ page import="com.collation.platform.model.topology.process.Activity" %>
<%@ taglib prefix="x" uri="http://java.sun.com/jstl/xml" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page import="com.collation.platform.util.Props" %>
<%@ page import="java.util.ArrayList"%>
<%@ page import="com.collation.cdm.common.messages.CdmLocalizedMessages"%>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%
java.util.Locale locale =
com.collation.cdm.common.util.CDMUtil.checkLocale(request.getLocale());
    if (null == session.getAttribute(org.apache.struts.Globals.LOCALE_KEY)) {
session.setAttribute(org.apache.struts.Globals.LOCALE_KEY, locale);
    }
%>
<%
//TMSDataHelper is a utility class for running MQL queries against the DB
TMSDataHelper tms = new TMSDataHelper(locale);

//Perform a query for all ComputerSystems
ModelObject dataIn[] = tms.doModelObjectQuery("SELECT * FROM ComputerSystem",null);

  //Build an HTML report based on the API output
  StringBuffer output = new StringBuffer();
  output.append("<p>");
  output.append("<table border=\"1\">");
                int c = 0;
                int s = dataIn.length;
                while (cs) {
                        ComputerSystem tmo = (ComputerSystem)dataIn[c];
                        String csName = null;
                        String csLabel = null;
                        if (tmo.hasName()) {
                                try {
                                        csName = tmo.getName();
                                } catch (AttributeNotSetException e) {
                                        csName = "unknown";
                                }
```

```
                        }
                        if (tmo.hasSignature()) {
                                try {
                                        csLabel = tmo.getSignature();
                                } catch (AttributeNotSetException e) {
                                        csLabel = "";
                                }
                        }
                        output.append("<tr><td colspan=\"2\" bgcolor=\"#9999FF\">");
                        output.append("ComputerSystem" + "<br>");
                        output.append(" Name: " + csName + "<br>");
                        output.append("</td><td>");
                        output.append("Signature: " + csLabel);
                        output.append("</td></tr>");
                        c++;
                }
output.append("</table>");
String bpstring = output.toString();
%>
<html>
<body>
<h1>Sample JSP Report/h1>
<%=bpstring%>
</body>
</html>
```

将 JSP 报告查看器添加到数据管理门户网站

通过修改 reports.xml 文件可将报告添加到数据管理门户网站。 reports.xml 文件位于 $COLLATION_HOME/etc/cdm/xml/ 目录。

过程

要将 JSP 报告查看器添加到数据管理门户网站，请完成下列步骤：

1. 使用文本编辑器打开 $COLLATION_HOME/etc/cdm/xml/reports.xml 文件。
2. 在 reports.xml 文件中，指定报告定义的报告描述符、报告组、报告名称以及外部脚本。 以下示例显示如何创建名为 Custom Report 并且位于 Inventory Reports 组中的外部报告，并指定 /WEB-INF/view/custom.jsp 脚本：

   ```
   <bean class="com.collation.cdm.reports.viewer.JSPReportViewer" id="CustomReport">
    <property name="reportGroup"><value>Inventory Reports</value></property>
    <property name="reportName"><value>Custom Report</value></property>
    <property name="script"><value>/WEB-INF/view/custom.jsp</value></property>
   </bean>
   ```

3. 保存 $COLLATION_HOME/etc/cdm/xml/reports.xml 文件。
4. 报告现在应该显示在数据管理门户网站中。

使用 Tivoli Common Reporting 进行报告

由于在 BIRT 报告查看器中查看 BIRT 报告并不安全，并且缺省情况下禁用了此查看器，因此您可以将 TADDM 的 BIRT 报告导入到 Tivoli Common Reporting 中。这将启用包含 TADDM 数据的跨产品报告。 您还可以使用报告调度之类的 Tivoli Common Reporting 功能，或者使用 Tivoli Common Reporting 作为报告的中央存储库。

对于某些任务，必须完成的步骤根据您所使用的 Tivoli Common Reporting 版本不同或者所使用的数据库不同而有所变化。

Fix Pack 1 如果您具有 TADDM 7.3 FP1 或更高版本，另请参阅最佳实践指南 The enhanced Cognos model in TADDM 7.3 FP*x*。

Tivoli Common Reporting 概述

Tivoli Common Reporting 工具是某些 Tivoli 产品随附的报告功能部件，它提供了一种集中方法，来查看和管理多个产品之间外观一致的报告。

Tivoli Common Reporting 包含用于存储和组织报告的数据存储器，以及用于管理、运行、安排和查看报告的界面。 Tivoli Common Reporting 同时使用 Cognos® 和 BIRT r运行时引擎。

要点：IBM Jazz™ for Service Management 安装光盘上提供 Tivoli Common Reporting。 如果不打算安装 IBM Jazz for Service Management，那么可以使用集成的 BIRT 报告功能。

如果系统上已经安装了 Tivoli Common Reporting，那么您可以选择性地导入与 Tivoli Common Reporting 兼容的预定义 TADDM 报告。 然后，可以使用 Tivoli Common Reporting 作为 Tivoli 产品报告的中央存储库。 您还可以使用增强型报告选项，包括跨产品报告、基于角色的安全性以及报告计划安排。

要查看产品的受支持版本，请转至第 171 页的『受支持的版本』部分。

注：如果要将 TADDM 与 IBM Tivoli Change and Configuration Management Database (CCMDB) 或 IBM SmartCloud Control Desk 配合使用，请参阅 CCMDB 或 IBM SmartCloud Control Desk 文档，以了解有关支持哪些 Tivoli Common Reporting 版本的信息。

有关 Tivoli Common Reporting 的更多信息，请访问 https://www.ibm.com/developerworks/community/groups/service/html/communityview?communityUuid=9caf63c9-15a1-4a03-96b3-8fc700f3a364。

安装 Tivoli Common Reporting 和 IBM Cognos Framework Manager

必须安装 Tivoli Common Reporting 和 IBM Cognos Framework Manager。

过程

要安装 Tivoli Common Reporting 和 IBM Cognos Framework Manager，请完成以下步骤：

1. 使用提供的缺省选项来安装 Tivoli Common Reporting。 如果您使用的是 Oracle 数据库，那么必须使用 Tivoli Common Reporting 2.1 或 3.1。
2. 安装 `CognosModeling` 文件夹中提供的 IBM Cognos Framework Manager 软件包。 请使用提供的缺省选项。
3. 此外，还应安装 `CognosModelingFix` 文件夹中提供的安全补丁（如果有）。 请使用提供的缺省选项。

安装和配置数据库客户机

如果您将 Tivoli Common Reporting 安装到 TADDM 数据库服务器以外的计算机上，那么必须安装数据库客户机以连接到数据库。 您可以使用 DB2 或 Oracle 数据库客户机，具体取决于 TADDM 数据库类型。 如果已将 Tivoli Common Reporting 安装到 TADDM 数据库所在的服务器上，那么无需安装数据库客户机。

过程

要安装和配置数据库客户机，请完成下列步骤：

完成下列其中一个任务：

- 如果要使用 DB2 数据库客户机，请完成下列步骤：
  1. 使用提供的缺省选项在安装了 TCR 的机器上安装 DB2 客户机。
  2. 确保已对 TADDM 数据库进行编目。 如果要 Tivoli Common Reporting 使用 DB2 客户机成功连接到 DB2 服务器，此步骤是必需的。
- 如果要使用 Oracle 数据库客户机，请使用 Oracle Universal Installer 向导和 Oracle Net Configuration Assistant 向导通过完成下列步骤来安装并配置该客户机：
  1. 在 Oracle Universal Installer 向导的"选择安装类型"页面中，选择**管理员**作为安装方式。
  2. 在"指定主页详细信息"页面中，指定安装名称以及要将该产品安装到的位置路径。
  3. 在"特定于产品的先决条件检查"页面中，确保每项安装和配置要求均已满足。 直到每项检查都处于**成功**状态之后，才应继续进行安装。
  4. 在 Oracle Net Configuration Assistant 向导的"欢迎"页面中，确保未选中**执行典型配置**复选框。
  5. 在"命名方法配置，选择命名方法"页面中，设置**本地命名**作为命名方法。
  6. 在"网络服务名称配置，服务名称"页面中，输入远程 Oracle 数据库服务器的服务名称，例如 ORCL。
  7. 在"网络服务名称配置，选择协议"页面中，选择 **TCP** 作为用于连接到数据库的协议。
  8. 在"网络服务名称配置，TCP/IP 协议"页面中，输入运行数据库的计算机的主机名。请选中**使用标准端口号 1521**。
  9. 在"网络服务名称配置，测试"页面中，选择**是，执行测试**。

     如果数据库用户名和密码正确，那么将显示以下文本：

     正在连接... 测试成功。

     如果未能成功地连接到数据库，那么您可能需要更改登录凭证。 要更改数据库登录凭证，请单击**更改登录**并指定有效的数据库用户名和密码。
  10. 在"网络服务名称配置，网络服务名称"页面中，接受缺省服务名称，此名称应该是您之前指定的服务名称。
  11. 创建名为 TNS_ADMIN 的 Windows 系统变量，并将值设置为 tnsnames.ora 文件所在文件夹的完整路径。在安装期间，将在 %ORACLE_HOME%/client_1/NETWORK/ADMIN 文件夹中创建 tnsnames.ora 文件，例如 C:/oracle/product/10.2.0/client_1/NETWORK/ADMIN。
  12. 在 startTCRserver.sh/bat 脚本中，将 TNS_ADMIN 变量设置为指向 tnsnames.ora 文件的位置，例如 %ORACLE_HOME%/client_1/NETWORK/ADMIN。
  13. 重新启动计算机，以确保新系统变量可用。

配置 IBM Cognos Framework Manager

您必须更新 IBM Cognos 10 Framework Manager 属性，使其具有适当的值。

关于此任务

注：以下过程适用于为配置 IBM Cognos 10 Framework Manager for Tivoli Common Reporting 3.1。 但是，该过程也同样适用于 IBM Cognos 8 Framework Manager for Tivoli Common Reporting 2.1。

安装 Tivoli Common Reporting 时，将安装 IBM Cognos Configuration 程序并更新一些属性值。 安装 IBM Cognos 10 Framework Manager 时，将安装另一版本的 IBM Cognos Configuration 程序，但不会更新所有属性。 您必须以手动方式将某些属性值从 Tivoli Common Reporting 版本的 IBM Cognos Configuration 复制到 IBM Cognos 10 Framework Manager 版本的 IBM Cognos Configuration。

过程

要配置 IBM Cognos 10 Framework Manager，请完成以下步骤：

1. 打开 Tivoli Common Reporting 所安装的 IBM Cognos Configuration 版本。要打开此程序，请单击**开始** > **程序** > **Tivoli Common Reporting 3.1** > **IBM Cognos Configuration**。
2. 打开 IBM Cognos 10 所安装的 IBM Cognos Configuration 版本。 要打开此程序，请单击**开始** > **程序** > **IBM Cognos 10** > **IBM Cognos Configuration**。
3. 对于每个版本的 IBM Cognos Configuration，单击**局部配置** > **环境**。
4. 将**网关 URI** 属性的值从 Tivoli Common Reporting 版本的 IBM Cognos Configuration 复制到 IBM Cognos 10 版本的 IBM Cognos Configuration 的**网关 URI** 属性。 URI 语法是： `http://tcrhost:16310/tarf/servlet/dispatch`。
5. 将**外部分派器 URI** 属性的值从 Tivoli Common Reporting 版本的 IBM Cognos Configuration 复制到 IBM Cognos 10 版本的 IBM Cognos Configuration 的**外部应用程序的分派器 URI** 属性。 URI 语法是： `http://tcrhost:16310/tarf/servlet/dispatch`。
6. 保存对 IBM Cognos 10 版本的 IBM Cognos Configuration 所做的更改。

生成 TADDM 模型

Fix Pack 1

您可以生成 TADDM 模型来获取 TADDM 数据库内容的最新快照，其中包括所有扩展属性的定义。 如果不使用扩展属性，那么可以跳过此过程，然后使用预构建的 TADDM Cognos 模型文件（`$COLLATION_HOME/etc/reporting/tcr/model.xml` 文件）。

开始之前

生成的 TADDM 模型包括 TADDM 数据库中存储的 TADDM 和扩展属性定义所支持的所有"公共数据模型"类。 您可以在 Tivoli Common Reporting 服务器中发布 TADDM 模型，并将其用于 Cognos 报告。 可多次生成 TADDM 模型。 每次重新生成的模型包含更新的 TADDM 数据库内容。

注意：

- 在 Tivoli Common Reporting 服务器中发布 TADDM 模型后，如果将扩展属性定义从 TADDM 数据库中移除，使用那些扩展属性定义的 Cognos 报告可能会停止运行。

- 如果使用的是 Windows 操作系统，请将以下过程中使用的脚本扩展名从 .sh 更改为 .bat。

过程

1. 在 TADDM 服务器上，打开 $COLLATION_HOME/bin 目录。
2. 通过完成以下步骤来更新扩展属性视图：
   a. 如果已创建扩展属性视图，请运行以下命令将其移除：
      ```
      ./extattr_views.sh remove
      ```
   b. 通过运行以下命令来生成具有扩展属性视图定义的 SQL 脚本：
      ```
      ./extattr_views.sh scripts
      ```
   c. 通过运行以下命令，可使用生成的 SQL 脚本来创建扩展属性视图：
      ```
      ./extattr_views.sh create
      ```
3. 要生成 Cognos 模型文件，请运行以下命令：
   ```
   ./genCognosModel.sh
   ```
   生成的 TADDM 模型将存储在 model.xml 文件中，而该文件位于 $COLLATION_HOME/etc/reporting/tcr 目录。 命令的日志消息位于 $COLLATION_HOME/log/genCognosModel.log 文件中。

下一步做什么

通过使用 IBM Cognos Framework Manager，可在 Tivoli Common Reporting 服务器中发布生成的 TADDM 模型。 有关更多信息，请参阅第 147 页的『使用 IBM Cognos Framework Manager 来发布模型』。

有关扩展属性视图的更多信息，请参阅 TADDM *SDK Developer's Guide* 中的 *Extended attributes views* 主题。

将模型和样本报告导入 Tivoli Common Reporting 中

可以将样本 TADDM 报告导入 Tivoli Common Reporting V2.1 和 V3.1 中。

关于此任务

此过程适用于 **Tivoli Common Reporting V2.1**。

过程

要将模型和样本报告导入 Tivoli Common Reporting 2.1 中，请完成以下步骤：

1. 将 $COLLATION_HOME/etc/reporting/TADDMPackage.zip 软件包从 TADDM 服务器复制到 Tivoli Common Reporting 服务器的 TCRComponent/cognos/deployment 文件夹中。
2. 打开 Tivoli Common Reporting 主页。
3. 单击**报告** > **Common Reporting**。
4. 从**启动**菜单中选择**管理**。 这将显示"管理"窗格。
5. 单击**配置**选项卡。
6. 单击**新建导入**图标。 此时将显示"新建导入"向导。
7. 从可用软件包列表中选择 **TADDMPackage**。单击**下一步**。

8. 可选： 在**描述**字段中，输入软件包的描述。 单击**下一步**。
9. 选中软件包名称旁边的复选框。
10. 在**选项**部分中，单击**源中的所有者**以及**新条目和现有条目**。 从**记录级别**菜单中选择**基本**。 单击**下一步**。
11. 单击**保存并运行一次**。 单击**下一步**。

关于此任务

此过程适用于 **Tivoli Common Reporting V3.1**。

过程

要将模型和样本报告导入 Tivoli Common Reporting 3.1 中，请完成以下步骤：

1. 将 `$COLLATION_HOME/etc/reporting/TADDMPackage.zip` 软件包从 TADDM 服务器复制到 JazzSM 安装的 `reporting/cognos/deployment` 文件夹。
2. 打开 Tivoli Common Reporting 主页。
3. 单击**报告** > **Common Reporting**。
4. 在**启动**菜单中选择 **IBM Cognos 管理**。 这将显示"管理"窗格。
5. 单击**配置**选项卡。
6. 转至"内容管理"。 单击**新建导入**图标。 此时将显示"新建导入"向导。
7. 从可用软件包列表中选择 **TADDMPackage**。单击**下一步**。
8. 可选： 在**描述**字段中，输入软件包的描述。 单击**下一步**。
9. 选中软件包名称旁边的复选框。 单击**下一步**。
10. 在**条目所有权**部分中，单击**源中的所有者**以及**新条目和现有条目**。 从**部署记录**部分中的**记录级别**菜单中，选择**基本**。单击**下一步**。
11. 验证所提供的值是否正确。 单击**下一步**。
12. 单击**保存并运行一次**。 单击**完成**。
13. 单击**运行**。

TADDM 模型中的数据视图

可以根据 TADDM 数据模型文件（`model.xml` 文件）生成报告。

数据模型由多个名称空间构成。 名称空间是逻辑容器，其中的所有名称都是唯一的。 每个名称空间都包含查询主题、查询项和对象。 导入 TADDM `model.xml` 文件后，存在下列名称空间：

Fix Pack 1 **CDM 名称空间**

这些视图包含大部分"公共数据模型"类（包括发现相关的类）的查询主题，而这些公共数据模型按照其软件包名称分为多个名称空间。 软件包名称按照字母顺序排序。 "简化模型"类可根据名称空间名称中的 `simple` 前缀进行区分。 您可以使用此数据来生成包含不同类型的 CDM 对象的报告。

CDM 名称空间中的查询主题包含与属性 `Parent` 相关的预定义关系。 例如，`app.j2ee.J2EEDomain` 类具有类型为 `app.j2ee.J2EEServer[]` 的属性 `Servers`。 `app.j2ee.J2EEServer` 类具有类型为 `app.j2ee.J2EEDomain` 的属性 `Parent`。因此，在 CDM 类的所有兼容对之间存在预定义关系，例如：

- `app.j2ee.J2EEDomain [0..1] - [0..n] app.j2ee.J2EEServer`

- app.j2ee.J2EEDomain [0..1] - [0..n] app.j2ee.jboss.JBossServer
- app.j2ee.J2EEDomain [0..1] - [0..n] app.j2ee.weblogic.WebLogicServer
- app.j2ee.jboss.JBossDomain [0..1] - [0..n] app.j2ee.jboss.JBossServer
- app.j2ee.websphere.WebSphereCell [0..1] - [0..n] app.j2ee.websphere.WebSphereServer

在 TADDM 7.3.0.1 中，已为阵列类型的非持久性属性定义了 CDM 名称空间中的部分查询主题。 在 TADDM 7.3.0.2 中，已为阵列类型的所有属性定义了查询主题。 它们的名称具有以下格式："*[name of the class declaring thearray attribute]-->[array attribute name]*"。 例如，simple.SGroup 类具有类型为 ModelObject[] 的 GroupMembers 属性，因此查询主题为 "SGroup-->GroupMembers"。这些查询主题包含所描述的阵列属性与具有这些属性的所有 CDM 类之间的预定义关系。 例如，对于上述提到的 GroupMembers 属性，其他类中已定义了以下关系：

- simple.SGroup [1..1] - [0..n] simple."SGroup-->GroupMembers"
- simple.SBaseCollection [1..1] - [0..n] simple."SGroup-->GroupMembers"
- app.biztalk.BizTalkGroup [1..1] - [0..n] simple."SGroup-->GroupMembers"
- app.hacmp.HACMPResourceGroup [1..1] - [0..n] simple."SGroup-->GroupMembers"

要使用阵列类型的属性，必须通过使用阵列类型的 PK_C 属性或者（针对阵列类型的非持久属性 (ModelObject[])）使用其 Guid 属性来定义阵列类型的属性与所需 CDM 类之间的关系。例如：

- Fix Pack 2 要创建将 sys.zOS.ZReportFile 对象显示为 sys.ComputerSystem 对象的 ZReportfiles 的 Cognos 报告，必须在 IBM Cognos Report Studio 中的以下列之间定义一个连接点：

  ```
  sys."ComputerSystem-->ZReportfiles".PK__ZReportfiles_C
   [0..n]-[0..1] sys.zOS.ZReportFile.PK_C
  ```

- 要创建将 app.AppServer 对象显示为 simple.SBaseCollection 对象的 GroupMembers 的 Cognos 报告，必须在 IBM Cognos Report Studio 中的以下列之间定义一个连接点：

  ```
   simple."SGroup-->GroupMembers".GroupMembersGuids [0..n]-
  [0..1] app.AppServer.Guid
  ```

Fix Pack 2 在许多情况下，无需为阵列类型的属性手动创建连接，因为存在从属对象的对应 Parent 属性。 Cognos 模型包含这些属性之间的关系。 例如，无需手动创建连接以创建将 sys.FileSystem 对象显示为 sys.ComputerSystem 对象的 FileSystems 的报告，因为 sys.FileSystem 对象具有指向 sys.ComputerSystem 对象的 Parent 属性。

Fix Pack 3 在 TADDM 7.3.0.3 和更高版本中，Cognos 模型包含时间戳记类型的所有属性的 Data Time 类型的查询项。 例如，sys.aix.AixUnitaryComputerSystem 查询主题包含以下查询项：

- 类型为 Int64 的 LastStoredTime，其指向构建块视图中的 LASTSTOREDTIME_C 列。 列中的示例值：1445417251307。

- 类型为 Data Time 的 LastStoredTimeT，其指向构建块视图中的 LASTSTOREDTIME_T 列。 列中的示例值：Oct 21，2015 10:47:31 AM。

LastStoredTimeT 查询项等同于 LastStoredTime 查询项，仅以全球标准时间格式表示，而不是 UNIX epoch（长整数）。 包含 T 后缀的查询项是等同于原始长整数属性的时间戳记。

WebSphere 名称空间

注： Fix Pack 1 在 TADDM 7.3.0.1 及更高版本中，不推荐使用 WebSphere 名称空间。

此视图包含 WebSphere 环境的主要查询主题。您可以使用此数据来生成特定于 WebSphere 的报告，例如列出 WebSphere 服务器的属性或 JVM 设置。 这个 WebSphere Server 查询主题链接到共享名称空间中包含的 AppServer 查询主题。 WebSphere cluster 和 WebSphere cell 查询主题链接到共享名称空间中包含的 AppServer cluster 和 J2EE Domain 查询主题。

共享名称空间

注： Fix Pack 1 在 TADDM 7.3.0.1 及更高版本中，不推荐使用共享的名称空间。

此视图包含被视为键类并可用作在不同名称空间之间连接数据的桥梁的查询主题。 共享名称空间包含有关计算机系统和集合类的信息。 您可以使用此数据来创建库存报告。

业务应用程序名称空间

注： Fix Pack 1 在 TADDM 7.3.0.1 及更高版本中，不推荐使用业务应用程序名称空间。

此视图包含业务应用程序的查询主题，即 Application 和 FunctionalGroup 查询主题。 功能组查询主题通过 Collection 查询主题链接到共享名称空间。 您可以使用此数据来创建用于显示业务应用程序及其成员的报告。

数据库名称空间

注： Fix Pack 1 在 TADDM 7.3.0.1 及更高版本中，不推荐使用数据库名称空间。

此视图包含与数据库和数据库服务器相关的查询主题。 您可以使用 All Databases 查询主题来生成通用数据库报告，而不是特定于供应商的数据库报告。数据库内容通过 AppServers 查询主题链接到共享名称空间。

依赖关系和关系名称空间

注： Fix Pack 1 在 TADDM 7.3.0.1 及更高版本中，不推荐使用依赖关系名称空间。

此视图包含表示生成的关系和依赖关系（例如 IP 依赖关系或者交换机与设备的关系）的查询主题。您可以使用通用的 relationship(unlinked) 查询主题在创建报告或查询时创建手动链接。 SwitchToDevice 查询主题将交换机连接到共享名称空间中的计算机系统对象。 存在三个与服务器亲缘关系相关的查询主题。 Server 查询主题显示数据库中所有计算机系统、应用程序服务器和服务对象的并集。 Affinity (target-linked) 查询主题将每种亲缘关系连接到 Server 查询主题中相应的目标。 Affinity (source-linked) 查询主题将每种亲缘关系连

接到 Server 查询主题中相应的源。 服务器内容通过计算机系统主题、应用程序服务器主题和服务查询主题链接到共享名称空间。您可以使用此数据来生成用于显示网络中配置项之间的关系的通用报告。

使用 IBM Cognos Framework Manager 来发布模型

如果要向 TADDM 数据模型（model.xml 文件）添加对象，那么必须编辑此文件，然后使用 IBM Cognos 10 Framework Manager 导入此文件。

关于此任务

以下过程适用于 IBM Cognos 10 Framework Manager。 但是，该过程也同样适用于 IBM Cognos 8 Framework Manager。

过程

要使用 IBM Cognos 10 Framework Manager 来导入数据模型，请完成以下步骤：

1. 启动 IBM Cognos 10 Framework Manager。
2. 创建新项目。
3. 在系统提示您输入 Tivoli Common Reporting 服务器的凭证时，执行此操作。 系统可能会多次提示您输入这些凭证。
4. 关闭 IBM Cognos 10 Framework Manager。
5. 将以下文件从 TADDM 服务器复制到 Cognos Framework 项目文件夹中：

   ```
   $COLLATION_HOME/etc/reporting/tcr/model.xml
   ```

   请覆盖 Cognos Framework 项目文件夹中的现有 model.xml 文件。
6. 启动 IBM Cognos 10 Framework Manager 并打开先前创建的项目。
7. 在"项目查看器"窗格中，单击**数据源** > ***content_manager_data_source_name***。
8. 如果您使用的 DB2 数据库的名称与 Cognos 数据库源中定义的名称不同，请将**模式**字段的内容替换为用于 TADDM 数据库的 DB2 实例名称。
9. 保存项目。
10. 在"项目查看器"窗格中，单击**包**。
11. 右键单击包名，并选择**发布包**。 验证和发布 TADDM Cognos 模型可能需要几分钟的时间。

在 Tivoli Common Reporting 中配置数据源

您可以使用 Tivoli Common Reporting 来配置数据源。

开始之前

请确保存在下列其中一种情况：

- 对 TADDM 数据库进行了本地编目。
- Tivoli Common Reporting 在主管 TADDM 数据库的服务器上运行。

如果您使用的是 DB2 数据库，请检查模式名称是否与 DB2 实例名称匹配。 模式名称指定用来授权访问指定数据库的 DB2 数据库名称。 您已在安装 TADDM 时指定了 DB2 实例名称。 TADDM model.xml 文件中指定的缺省实例名称为 DB2INST1。 如有需要，请更改模式名称。

如果您使用的是 Oracle 数据库，请确保模式名称为空。

过程

要使用 Tivoli Common Reporting 来配置数据源，请完成下列步骤：

1. 打开 Tivoli Common Reporting 主页。
2. 单击**报告** > **Common Reporting**。
3. 根据要使用的 Tivoli Common Reporting 版本，在**启动**菜单中选择以下某个菜单项：
   - V2.1 - **管理**。
   - V3.1 - **IBM Cognos 管理**。

   这将显示"管理"窗格。
4. 单击**配置**选项卡。
5. 单击**新建数据源**图标。 这将显示"新建数据源"向导。
6. 在**名称**字段中，输入 CMDBTCR。 在数据模型中引用了名称 CMDBTCR，因此必须对新数据源指定同一名称。
7. 从**类型**菜单中选择要使用的数据库类型。
8. 完成下列其中一个步骤：
   - 如果数据库类型为 DB2，请在 **DB2 数据库名称**字段中输入 TADDM 数据库名称或者已编目的 TADDM 数据库的别名。
   - 如果数据库类型为 Oracle，请在 **SQL*Net 连接字符串**字段中输入 Oracle 数据库的服务名称，例如 ORCL。 您已在配置 Oracle 数据库客户机时指定了 Oracle 数据库服务名称。 可以在 %TNS_ADMIN%/tnsnames.ora 文件中检查 Oracle 数据库服务名称。请搜索以下字符串：

     ```
     SERVICE_NAME =
     ```
9. 在**登录**部分中，指定数据库用户名和密码。
10. 要测试数据库连接，请单击**测试**。 在"新建数据源"向导的"查看结果"页面中，将显示测试状态。

将 TADDM 报告包导入到 Tivoli Common Reporting 中

要将预定义的 TADDM 报告导入到 Tivoli Common Reporting 中，您可以导入 TADDM 报告包。

开始之前

必须先在系统上安装 Tivoli Common Reporting 功能部件。 Tivoli Common Reporting 随某些 Tivoli 产品一起提供，但当前未由 TADDM 随附。

关于此任务

Tivoli Common Reporting 报告包是一个 .zip 文件，其中包含格式可由 Tivoli Common Reporting 使用的一个或多个报告或报告设计及其所需资源。 TADDM 的预定义 BIRT 报告在一个报告包中提供，您可以将此报告包导入到 Tivoli Common Reporting 中。

对于某些 BIRT 报告，根据运行报告的服务器不同，提供了同一报告的不同版本，例如域服务器或存储服务器上的 TADDM_SNAPSHOT_CHANGE 报告以及同步服务器上

的 TADDM_SNAPSHOT_SYNC_CHANGE 报告。 通常，只提供了适当版本的报告，但将 BIRT 报告导入到 Tivoli Common Reporting 之后，可能会同时提供这两个版本的报告。 请确保仅使用适合于要运行报告的服务器的报告版本。

将 BIRT 报告导入到 Tivoli Common Reporting 之后，可能会提供一些在名称中包含"仅深入查看"文本的报告。 这些报告是通过在另一报告中深入查看所选数据来运行的，而无法单独运行。

"按作用域分组的服务器亲缘关系报告"无法导入到 Tivoli Common Reporting 中。

有关导入报告包的更多信息，请参阅 Tivoli Common Reporting 文档。

过程

要导入 TADDM 报告，请完成下列步骤：

1. 如果您使用的是 Tivoli Common Reporting 1.3，请完成下列步骤：
   a. 在 Tivoli Common Reporting 报告导航窗口中，转到**导航**选项卡。
   b. 右键单击导航树的根节点（报告集）。
   c. 单击**导入报告包**。
   d. 在"导入报告包"窗口中，指定 TADDMReports.zip 报告包文件的位置。 此文件位于 $COLLATION_HOME/etc/reporting 目录中。
   e. 展开**高级选项**并执行以下操作：
      1) 选中**覆盖**复选框。 这将确保覆盖先前安装的所有报告副本。
      2) 在**安全集**字段中，输入要将报告包内容导入到的安全集的名称。
   f. 单击**导入**。 该 TADDM 报告包将导入到 Tivoli Common Reporting 数据存储器中。
2. 如果您使用的是 Tivoli Common Reporting 2.1，请完成下列步骤：
   a. 打开命令行并浏览到 TIP_install_dir/tipv2Components/TCRComponent/bin。
   b. 运行 import 命令：

      ```
      trcmd -user userID -password password -import
      -bulk pkgFile
      ```

      其中 *pkgFile* 是从 TADDM 服务器上的 $COLLATION_HOME/etc/reporting 复制到 Tivoli Common Reporting 服务器的 TADDMReports.zip 报告包文件的路径。
   c. 该 TADDM 报告包将导入到 Tivoli Common Reporting 数据存储器中。
3. 如果使用的是 Tivoli Common Reporting 3.1，请完成以下步骤：
   a. 打开命令行并浏览到 JazzSM_install_dir/reporting/bin。
   b. 运行 import 命令：

      ```
      trcmd -user userID -password password -import
      -bulk pkgFile
      ```

      其中 *pkgFile* 是从 TADDM 服务器上的 $COLLATION_HOME/etc/reporting 复制到 Tivoli Common Reporting 服务器的 TADDMReports.zip 报告包文件的路径。
   c. 该 TADDM 报告包将导入到 Tivoli Common Reporting 数据存储器中。

下一步做什么

导入 TADDM 报告之后，必须为各个报告重新配置 JDBC 数据源。

在 Tivoli Common Reporting 中配置 TADDM BIRT 报告

将 TADDM 报告导入 Tivoli Common Reporting 之后，必须配置每个报告所使用的 JDBC 数据源。

开始之前

在配置 JDBC 访问之前，请确保在 Tivoli Common Reporting 的 drivers 目录中安装相应的 JDBC 驱动程序文件。对于 Tivoli Common Reporting 1.3，这些文件位于以下目录中：

tcr_install_dir/products/tcr/lib/birt-runtime-2_2_1/ReportEngine/plugins/
org.eclipse.birt.report.data.oda.jdbc_2.2.1.r22x_v20070919/drivers

对于 Tivoli Common Reporting 2.1，这些文件位于以下目录中：

tip_install_dir/tip21Components/TCRComponent/lib/birt-runtime-2_2_2/ReportEngine/plugins/
org.eclipse.birt.report.data.oda.jdbc_2.2.2.r22x_v20071206/drivers

对于 Tivoli Common Reporting 3.1，这些文件位于以下目录中：

JazzSM_install_dir/reporting/lib/birt-runtime-2_2_2/ReportEngine/plugins/
org.eclipse.birt.report.data.oda.jdbc_2.2.2.r22x_v20071206/drivers

如果您使用的是 Oracle 数据库，请确保 ojdbc14.jar 或 ojdbc5.jar 包括在此目录中。

关于此任务

导入的报告最初配置为使用缺省数据源。 您必须修改每个 TADDM 报告的数据源属性，以使用存储了发现数据的数据库。 TADDM 报告不使用共享数据源。 因此，请完成以下步骤以配置所有 TADDM 报告的数据源属性。

过程

要为 Tivoli Common Reporting 配置 JDBC 数据源，请完成下列步骤：

1. 如果您使用的是 Tivoli Common Reporting 1.3，请完成下列步骤：
   a. 在 Tivoli Common Reporting 的**报告**表中，右键单击要配置的 TADDM 报告。
   b. 在弹出菜单中，单击**数据源**。
   c. 在**报告数据源**窗口中，输入 JDBC 驱动程序、URL、用户标识和密码信息。 您可以在 $COLLATION_HOME/etc 目录中的 collation.properties 文件中找到这些设置的正确值。
   d. 对每个要配置的 TADDM 报告重复上述步骤。
2. 如果使用的是 Tivoli Common Reporting 2.1 或 3.1，请完成以下步骤：
   a. 打开命令行并浏览到 Tivoli Common Reporting 2.1 的 tip_install_dir/tip21Components/TCRComponent/bin，或浏览到 Tivoli Common Reporting 3.1 的 JazzSM_install_dir/reporting/bin。
   b. 药配置所有报告的所有 JDBC 源，请在一行上运行 **modify** 命令：

要点：以下命令包含目录的名称，此目录包含 BIRT 报告 'IBM Tivoli Products'。 此名称适用于 TADDM 7.3.0.1 和更高版本。 如果使用 TADDM 7.3.0，请将此名称替换为 'Tivoli Products'。

```
trcmd -user userID -password password -modify
-datasources -reports -reportname "/content/package[@name='IBM Tivoli
Products']/folder[@name='TADDM Reports']//report" -setdatasource
odaDriverClass=driverClass odaURL=jdbcUrl
odaUser=dbUser odaPassword=dbPassword
```

例如，如果使用 DB2 数据库，那么请在一行上输入以下命令：

```
trcmd -user tipadmin -password tipadmin -modify -datasources -reports
-reportname "/content/package[@name='IBM Tivoli Products']/folder[@name=
'TADDM Reports']//report" -setdatasource
odaDriverClass=com.ibm.db2.jcc.DB2Driver
odaURL=jdbc:db2://100.101.102.103:50000/SAMPLEDB
odaUser=db2inst1 odaPassword=db2inst1
```

例如，如果使用 Oracle 数据库，那么请在一行上输入以下命令：

```
trcmd -user tipadmin -password tipadmin -modify -datasources -reports
-reportname "/content/package[@name='IBM Tivoli Products']/folder[@name=
'TADDM Reports']//report" -setdatasource
odaDriverClass=oracle.jdbc.driver.OracleDriver
odaURL=jdbc:oracle:thin:@192.168.0.1:1521:orcl
odaUser=taddm_dev odaPassword=taddm_dev
```

验证 TADDM 报告

您可以进行检查，以确保在 Tivoli Common Reporting 中正确地显示 TADDM 报告。

过程

要验证是否在 Tivoli Common Reporting 中正确地显示了 TADDM 报告，请完成下列步骤：

1. 打开 Tivoli Common Reporting 主页。
2. 单击**报告** > **Common Reporting**。
3. 确保显示了 **TADDM** 和 **Tivoli 产品**文件夹。
4. 单击 **TADDM**。
5. 单击**运行**图标以运行其中一个报告。 这将显示该报告。
6. 确保该报告正确完整地显示。
7. 使用面包屑跟踪返回到**公用文件夹**。
8. 单击 **Tivoli 产品** > **TADDM 报告**。 单击**运行**图标以运行其中一个报告。 这将显示该报告。
9. 确保该报告正确完整地显示。

使用 BIRT 进行报告

您可以使用 Business Intelligence and Reporting Tools (BIRT) 报告功能根据 TADDM 数据库中的数据来运行预定义报告和定制报告。

BIRT 报告概述

除数据管理门户网站提供的内置报告以外，您还可以设计、开发和安装基于开放式源代码 Business Intelligence and Reporting Tools (BIRT) 系统的报告。

要点：使用 BIRT 报告查看器（BIRT 运行时引擎）查看数据管理门户网站中的 BIRT 报告并不安全，因此已将此查看器禁用。 最好是将 TADDM 报告导入到 Tivoli Common Reporting (TCR) 中，然后再使用 TCR 来查看 BIRT 报告。

如果您了解这些风险，可以复原 BIRT 报告查看器并按下列段落中指定的方式进行使用。

TADDM 包括开放式源代码 BIRT 运行时引擎作为集成组件。 另外，TADDM 还提供了数百个预定义数据库视图和预定义报告。 除预定义报告以外，您还可以使用 BIRT 设计器工具来创建要与 TADDM BIRT 运行时引擎配合使用的新报告。 这些报告可以使用通过预定义数据库视图抽取数据的 JDBC 数据源。

数据管理门户网站界面提供了管理这些 BIRT 报告的方法。 您可以添加新报告、下载选定的报告、删除已上载的报告以及运行报告。 另外，还打包了预定义报告，以使其能够与 Tivoli Common Reporting 工具配合使用。

Business Intelligence and Reporting Tools

Business Intelligence and Reporting Tools (BIRT) 是基于 Eclipse 系统的开放式源代码，用于设计、开发和运行报告。 您可以为 TADDM 开发 BIRT 报告，从而将这些报告设计成使用预定义数据库视图的 JDBC 数据源和 SQL 查询。

要点：BIRT 报告不得使用直接从 TADDM 数据库表获取的数据。 相反，您始终应将报告设计为使用 JDBC 数据源和《TADDM SDK 开发者指南》中描述的 TADDM 数据库视图。

BIRT 系统包括两个主要组件：

- BIRT 设计器，这是一个图形工具，用于设计和开发新报告
- BIRT 运行时引擎，此组件支持运行报告以及显示所发布的报告输出

TADDM 提供了 BIRT 运行时引擎，后者可用于运行预定义报告。 如果您要创建自己的 BIRT 报告，那么必须下载与 TADDM 随附的 BIRT 运行时引擎版本（当前为 V2.2.1）匹配的 BIRT 设计器工具。

有关 BIRT 项目的更多信息（包括如何下载 BIRT 设计器工具），请参阅 http://www.eclipse.org/birt。

相关任务:

第 167 页的『复原 BIRT 报告查看器』
如果您了解与安全性相关的风险，并且仍希望使用 BIRT 报告查看器，那么可以将其复原。

预定义的 BIRT 报告

TADDM 随附的预定义 BIRT 报告提供有关发现的系统、操作系统和服务器进程的信息。

应用程序服务器库存报告：

"应用程序服务器库存报告"提供 TADDM 发现的所有应用程序服务器。 运行此报告时，您可以指定一个参数值，用来将此报告限制为只显示特定类型的应用程序服务器。 此报告按系统对发现的应用程序服务器进行分组，并且按标准主机名列出这些应用程序服务器。

此报告的数据从 CM_APP_SERVERS_PER_HOST_V 数据库视图中获取。

计算机系统库存报告：

"计算机系统库存报告"列出 TADDM 数据库中所有已被赋予 IP 地址的计算机系统，这些计算机系统按标准主机名列出。 此报告没有参数。

此报告设计成可以导出为一个以逗号分隔的文件，该文件可以导入到电子表格应用程序中。 如果系统不具有 IP 地址，那么不会包括在此报告中。 同一计算机系统名称可能在此报告中列示多次，即，针对每个唯一 IP 地址（包括 127.0.0.1 回送地址）列示一次。

此报告的数据从 CM_COMPUTER_SYSTEMS_V 数据库视图中获取。

按操作系统类型分组的计算机系统库存报告：

"按操作系统类型分组的计算机系统库存报告"列出所有已发现的计算机系统（其操作系统也已发现）。 此报告没有参数。

此报告设计成可以导出为一个以逗号分隔的文件，该文件可以导入到电子表格应用程序中。 同一计算机系统名称可能在此报告中列示多次，即，针对每个唯一 IP 地址（包括 127.0.0.1 回送地址）列示一次。 操作系统必须与 TADDM 数据库中的某个系统相关联才会包括在此报告中。 同样，不会包括任何不具有 TADDM 数据库中定义的操作系统的系统。

单击此报告中的系统名称可以打开该系统的详细"库存详细信息报告""。

此报告的数据从以下数据库视图中获取：
- DP_UNITARY_COMP_GENERAL_V
- DP_UNITARY_COMP_OS_V
- DP_UNITARY_COMP_IP_INTERFACE_V
- BB_OPERATINGSYSTEM62_V

***ITNM IP* 报告：**

此报告提供有关已安装的 Network Manager 产品实例的信息，并列出所有与计算机系统相关的 Network Manager 资源。

"Network Manager 库存报告"在"TADDM 域管理器"控制台中提供。 此报告包含下列部分：

服务器摘要

此部分提供了已安装的 Network Manager 产品实例的信息，包括安装的 Network Manager 版本、安装了 Network Manager 的服务器的主机地址、以及用于访问 Network Manager GUI 的 URL。

资源摘要

此部分列出了所有与计算机系统相关的 Network Manager 资源，包括有关其 IP 地址、制造商、资源类型（例如路由器）以及 Network Manager 数据库中唯一标识的信息。

简明计算机系统库存报告：

"简明计算机系统库存报告"使您能够查看通过使用第 1 层发现概要文件发现的 IP 地址。对于每个 IP 地址，此报告还显示相关联的计算机系统名称，以及操作系统或控制软件名称（如果发现了此信息）。

虽然"简明计算机系统库存报告"计划在第 1 层发现之后使用，但也可在第 3 层发现之后使用。 但是，在使用凭证执行发现后，其他报告（例如"计算机系统库存报告"）可以提供更多详细信息。

光纤通道网络报告：

"光纤通道网络报告"显示所选光纤通道交换机与其他计算机系统之间的光纤通道连接。

要运行此报告，请指定光纤通道交换机的全球名称 (WWN)，以查看此交换机与其他计算机系统之间的光纤通道连接。 在"参数"窗口中，请输入此名称 (WWN) 或者从发现的光纤通道交换机的下拉列表中进行选择。

在此报告中，对于每个已连接的计算机系统，将显示以下信息：

- 计算机系统（显示名；对于光纤通道交换机，这是 WWN）
- 制造商
- 型号
- 序列号

您可以在此报告中单击计算机系统显示名，以打开另一个"光纤通道网络报告"。 此报告将显示所选计算机系统与其他计算机系统之间的光纤通道连接。

主机总线适配器库存报告：

"主机总线适配器库存报告"显示所有已发现的主机总线适配器以及这些适配器所安装在的计算机系统的列表。

对于每个已发现的主机总线适配器，此报告将显示以下信息：

主机总线适配器名称

主机总线适配器的名称。

标准域名

主机总线适配器所安装在的计算机系统的标准域名。

主机是否使用存储阵列

此布尔值指示主管计算机系统是否使用了存储阵列中的任何存储卷。

库存摘要：

"库存摘要"包括已发现的计算机系统上安装的操作系统的饼图，该图以 TADDM 发现的作用域为基础。 该图的每个部分表示一种操作系统类型，并指示运行该操作系统的已发现服务器的整体计数。 此报告没有参数。

单击该图的任何部分都可打开所选操作系统类型的深入钻取的"计算机系统库存"报告。

此报告的数据从 BB_OPERATINGSYSTEM62_V 数据库视图中获取。

监视覆盖范围报告：

监视覆盖范围报告显示关于环境中各种组件的详细信息。 您可以生成关于环境中的操作系统、数据库、Microsoft 应用程序、VMware 服务器和 System p 组件的报告。 这些组件受 IBM Tivoli Monitoring 6.1 或更高版本代理程序监视。 可以从数据管理门户网站的"BIRT 报告"窗格运行此报告。

表 37 列出了可用的覆盖范围报告。操作系统的报告监视覆盖范围可由 IBM® Tivoli® Monitoring Scope 传感器填充。 但是，剩余的报告需要 IBM® Tivoli® Monitoring 发现库适配器 (DLA) 来填充报告。

这些报告包含三个部分：

按类型分组的覆盖范围

此部分显示按报告类型分组的受监视实例数、未受监视实例数和总实例数。 "覆盖范围详细信息"窗口将显示以下统计信息的图形表示：

- 整体覆盖范围
- 按平台分组的覆盖范围

覆盖范围详细信息

此部分显示按报告类型分组的标准域名、受管系统名称和监视状态。 监视状态随代理程序版本信息（如果受监视）一起列出。单击受监视系统的 MSN 将打开"代理程序详细信息"窗口。

代理程序详细信息

此部分显示有关代理程序及其上运行的操作系统的详细信息。 显示的信息取决于代理程序是否受到监视。 亲缘关系和源标记信息随"在上下文中启动"链接一起提供，此链接指向 IBM Tivoli Monitoring 的 Tivoli Enterprise Portal 视图。

"管理软件系统"部分提供已安装的 IBM Tivoli Monitoring 代理程序的清单，以及指向 IBM Tivoli Monitoring 中工作空间的"在上下文中启动"链接。 "监视覆盖范围摘要"提供受监视的以及未受监视的系统的列表，您可以使用此列表来监视和维护监视代理程序。

第 1 层发现可以使用 IBM Tivoli Monitoring Scope 传感器来填充操作系统报告的监视覆盖范围。其他报告必须由 IBM Tivoli Monitoring 发现库适配器 (DLA) 填充。 请参阅《TADDM 管理员指南》，以了解有关 IBM Tivoli Monitoring DLA 的信息。

表 37 列出了可用的覆盖范围报告。

表 *37.* 监视覆盖范围报告

报告名称	描述
操作系统的监视覆盖范围	此报告显示环境中操作系统的详细信息。
数据库的监视覆盖范围	此报告显示环境中 DB2 实例和 SQL 服务器的详细信息。

表 37. 监视覆盖范围报告 (续)

报告名称	描述
Microsoft 应用程序的监视覆盖范围	此报告显示已启用的 Active Directory、Cluster Server、Exchange Server、Host Integration Server、Hyper-V 服务器角色以及 Internet Information Services 服务器的详细信息。
VMware 的监视覆盖范围	此报告显示 VMware ESX 服务器和 VMware Virtual Center 服务器的详细信息。
System p 的监视覆盖范围	此报告显示 System p、硬件管理控制台、Virtual I/O Server 和 AIX 逻辑分区的详细信息。

传感器报告：

预定义的传感器报告用于整理针对传感器度量值收集到的信息。

表 38 列出了可用的预定义传感器报告。

表 38. 预定义的传感器报告

报告名称	描述
TADDM_SENSORS_WEEKLY_METRICS_ALL TADDM_SENSORS_WEEKLY_METRICS	此报告显示第 1 层发现概要文件、第 2 层发现概要文件或第 3 层发现概要文件中启用的传感器的每周成功率百分比。 将显示以下信息： • 日期 • 第 1 层 (L1) 成功百分比 • 第 2 层 (L2) 成功百分比 • 第 3 层 (L3) 成功百分比 • L1 和 L2 成功百分比 • 所有成功百分比 第二个"TADDM_SENSORS_WEEKLY_METRICS"报告包含同一信息，但使用条形图来显示此信息。

表 *38.* 预定义的传感器报告 *(续)*

报告名称	描述
TADDM_SENSORS_SUMMARY_TOTAL TADDM_SENSORS_SUMMARY	此报告显示已运行并成功完成的传感器的总数。 将显示以下信息： • 层次 • 对存储的 CI 运行次数 • 成功次数 • 失败次数 另外，还将显示摘要，该摘要显示发现概要文件层次以及每一层的总体成功百分比和失败百分比。 "TADDM_SENSORS_SUMMARY"报告显示发现期间各个传感器的成功百分比或失败百分比。 将显示以下信息： • 层次 • 传感器 • 运行次数 • 成功次数 • 失败次数 • 成功百分比 • 失败百分比
TADDM_SENSORS_SERVER_SCANS_IP	此报告显示通过指定 IP 地址扫描服务器之后的状态。 将显示以下信息： • 周 • 状态 此报告开头显示有关 IP 地址、主机名、标准域名、首次扫描日期及状态以及最近一次扫描日期及状态的摘要信息。
TADDM_SENSORS_SERVER_SCANS_HOSTNAME	此报告显示通过指定主机名扫描服务器之后的状态。 将显示以下信息： • 周 • 状态 此报告开头显示有关主机名、IP 地址、标准域名、首次扫描日期及状态以及最近一次扫描日期及状态的摘要信息。
TADDM_SENSORS_MONTHLY_COVERAGE	此报告显示一个条形图，该图显示"会话"传感器的每月覆盖范围。 包括有关扫描运行次数以及成功或未成功扫描次数的信息。 "会话"传感器在 TADDM 服务器与目标计算机系统之间创建会话。
TADDM_SENSORS_METRICS_LEVEL_1_AND_2 TADDM_SENSORS_METRICS_LEVEL3	此报告显示一个条形图，该图显示在指定星期执行第 1 层和第 2 层发现时，各个传感器的成功百分比。 报告"TADDM_SENSORS METRICS_LEVEL3"的条形图显示执行第 3 层发现时各个传感器的度量值。
TADDM_SENSORS_FAILED_LEVELS_1_2_3 TADDM_SENSORS_FAILED_LEVEL	此报告根据执行第 1 层、第 2 层或第 3 层发现时发生的失败次数显示指定星期的饼图。 图表的各个部分分别表示会话问题、传感器问题、连接问题和其他问题。 "TADDM_SENSORS_FAILED_LEVEL"报告的饼图显示指定发现层的度量值。

表 *38.* 预定义的传感器报告 *(续)*

报告名称	描述
TADDM_SENSORS_EVENTS_SENSOR_IP TADDM_SENSORS_EVENTS_SENSOR TADDM_SENSORS_EVENTS_IP TADDM_SENSORS_DONE_EVENTS_RUN	此报告显示指定传感器和 IP 地址的事件数据。 将显示以下信息： • 日期 • 传感器详细信息 • 严重性 • 描述 "TADDM_SENSORS_EVENTS_SENSOR"报告包含同一信息，但显示指定传感器的事件数据。 "TADDM_SENSORS_EVENTS_IP"报告包含同一信息，但显示指定 IP 地址的事件数据。 "TADDM_SENSORS_DONE_EVENTS_RUN"报告包含同一信息，但显示指定发现运行的事件数据。

按作用域分组的服务器亲缘关系：

"按作用域分组的服务器亲缘关系报告"显示服务器之间的关系，并根据每个关系的源和目标对这些关系进行排列。 第一个表显示指定作用域内所有作为关系源的服务器，以及从这些服务器到其他服务器的连接。 第二个表显示指定作用域内所有作为关系目标的服务器，以及从其他服务器到这些服务器的连接。

"按作用域分组的服务器亲缘关系报告"仅在域服务器部署中可用。

要查看显示了服务器之间的通信的图，请单击**启动亲缘关系图**。 此图将显示计算机系统之间的事务依赖关系和服务依赖关系，各依赖关系由系统之间绘制的链接指示。 此图显示所有至少涉及发现作用域内的一个系统的依赖关系链接，并以黄色突出显示作为作用域成员的系统。

亲缘关系图中显示的链接可以表示事务关系或服务关系。 链接方向指示了哪个系统是依赖关系的源，哪个系统是目标。 源对象和目标对象可以是以下多种类型，具体取决于关系：

- 计算机系统
- 应用程序服务器
- 服务

图中的链接始终在计算机系统之间绘制。对于涉及应用程序服务器或服务的关系，链接将连接到主计算机系统。 要了解更多有关依赖关系的信息（包括涉及的源、目标、命令名和端口号），请将鼠标指针移动到图中的链接上。

"按作用域分组的服务器亲缘关系报告"无法导入到 Tivoli Common Reporting 中。

快照报告：

预定义的快照报告对一个或多个快照捕获的信息进行整理。

快照是已发现的计算机信息的副本，并在特定时间点创建。 有关创建快照的更多信息，请参阅第 167 页的『使用快照工具』。

具体报告名称取决于您在哪个服务器上运行和查看 BIRT 报告。 如果您是在域服务器或存储服务器上使用数据管理门户网站，请运行标准报告，例如 TADDM_SNAPSHOT_CHANGE。如果您是在同步服务器上使用数据管理门户网站，请运行名称中带有"SYNC"的报告，例如 TADDM_SNAPSHOT_SYNC_CHANGE。下列报告是例外情况，它们在所有服务器上都具有同一名称：

- TADDM_SNAPSHOT_FRAME
- TADDM_SNAPSHOT_HOST

将 BIRT 报告导入 Tivoli Common Reporting 后，显示的报告名称有所变化，例如，TADDM_SNAPSHOT_SYNC_SESSION_FAILED 报告将显示为"TADDM：有关失败会话的详细信息（企业）"。

表 39 列出了可用的预定义快照报告。

表 *39.* 预定义的快照报告

报告名称	描述
TADDM_SNAPSHOT_FRAME	显示有关已发现的服务器的下列详细信息： • 机架名称 • 序列号 • 制造商 • 型号 • CPU 类型 • CPU 速度 • CPU 数目 • 内存 • 位置 • 支持区域 • 最近一次发现时间
TADDM_SNAPSHOT_HOST	显示有关物理和虚拟服务器的下列详细信息： • 机架名称 • 系统名称 • IP 地址 • 操作系统类型 • 主机类型 • 受管系统名称 • 最近一次发现时间
TADDM_SNAPSHOT_SESSION_FAILED TADDM_SNAPSHOT_SYNC_SESSION_FAILED	显示由于会话失败而导致 TADDM 无法获取其相应 L2 信息的已发现服务器的相关名称和 IP 地址信息。

表 *39.* 预定义的快照报告 *(续)*

报告名称	描述
TADDM_SNAPSHOT_CHANGE TADDM_SNAPSHOT_SYNC_CHANGE	对不同时间创建的两个快照进行比较。对于在这两个快照的创建时间之间添加或除去的每个服务器，此报告将显示以下信息： • 名称 • IP 地址 • 虚拟 另外，还显示这两个快照的创建时间之间物理服务器与虚拟服务器的比率变化信息。
TADDM_SNAPSHOT_DISCOVERY_ERROR TADDM_SNAPSHOT_SYNC_DISCOVERY_ERROR	显示有关任何发现期间发生的错误的信息。
TADDM_SNAPSHOT_FQDN_OS_CHANGES TADDM_SNAPSHOT_SYNC_FQDN_OS_CHANGES	显示任何在这两个快照的创建时间之间更改了标准域名 (FQDN) 或操作系统信息的服务器的相关信息。
TADDM_SNAPSHOT_REFERENCE TADDM_SNAPSHOT_SYNC_REFERENCE	将快照与参考列表进行比较。 此报告显示在参考列表中存在但在快照中不存在的服务器的相关信息，以及在快照中存在但在参考列表中不存在的服务器的相关信息。
TADDM_SNAPSHOT_RECONCILIATION_SUMMARY TADDM_SNAPSHOT_SYNC_RECONCILIATION_SUMMARY	此报告将提示您指定快照，并显示有关已发现的服务器的以下摘要信息： • 基线主机名 • 基线 IP 地址 • TADDM 主机名 • TADDM IP 地址 • 状态 • 错误原因

表 *39.* 预定义的快照报告 *(续)*

报告名称	描述
TADDM_SNAPSHOT_RECONCILIATION_DETAIL TADDM_SNAPSHOT_SYNC_RECONCILIATION_DETAIL	此报告将提示您指定快照，并显示有关已发现的服务器的以下详细信息： • 基线主机名 • 基线 IP 地址 • TADDM 主机名 • TADDM IP 地址 • 状态 • 错误原因 • 错误描述 • 作用域名称 • 过滤掉的排除项 • TADDM 机架 • TADDM 主机名 • TADDM FQDN • TADDM 名称 • TADDM 显示名 • TADDM JDO 类 • TADDM 派生操作系统 • TADDM 操作系统名称 • TADDM IP 地址 • TADDM 序列号 • TADDM 制造商 • TADDM 型号 • TADDM 主机类型 • TADDM 虚拟 • TADDM 类型 • TADDM 发现日期

按主机分组的存储阵列报告：

"按主机分组的存储阵列报告"显示指定计算机系统使用的存储卷和存储阵列的列表。

运行此报告时，系统将提示您输入要查看其存储器信息的计算机系统的主机名。在"参数"窗口中，请输入主机名或者从下拉列表中进行选择。

在此报告中，将显示以下信息：

- 存储卷
- 存储阵列
- 制造商
- 型号
- 序列号

- 可用容量
- 已分配的容量

存储阵列使用者报告：

"存储阵列使用者报告"显示使用了指定存储阵列的计算机系统和应用程序服务器的列表。

运行此报告时，系统将提示您输入存储阵列的名称。 在"参数"窗口中，请输入存储阵列的名称或者从下拉列表中进行选择。

此报告将以如下三个表的形式显示：

使用了存储阵列 *storage_array_name* 的计算机系统
此表列出发现的所有使用了指定存储阵列的计算机系统。

使用了存储阵列 *storage_array_name* 的应用程序服务器
此表列出发现的所有使用了指定存储阵列的应用程序服务器。

使用了存储阵列 *storage_array_name* 的业务应用程序
此表列出发现的所有使用了指定存储阵列的业务应用程序。

系统连接拓扑报告：

"系统连接拓扑报告"显示与其他计算机系统建立了网络连接的计算机系统的文本报告。运行此报告时，必须输入要针对其运行此报告的配置项，并且必须指定该配置项是计算机系统还是业务应用程序。

如果针对计算机系统运行此报告，那么所有与所选计算机系统建立了网络连接的计算机系统都将与每个网络连接的度量值一起显示在表中。 如果针对业务应用程序运行此报告，那么所有与所选业务应用程序建立了网络连接的计算机系统都将显示在表中。

通过单击报告中的系统名称，您可以查看每个计算机系统的系统连接拓扑。

系统利用率每小时峰值报告：

"系统利用率每小时峰值报告"显示指定作用域内的系统在指定日期的每小时系统利用率峰值。

利用率度量值包括以下信息：

- 每小时 CPU 利用率 95%
- 每小时内存利用率百分比峰值
- 每小时网络带宽利用率峰值
- 每小时磁盘 I/O 利用率峰值

系统利用率报告：

"系统利用率报告"显示通用服务器操作系统配置以及相关联的利用率信息。

服务器操作系统配置数据包括下列信息：

- CPU
- 内存
- 文件系统

这是可供 TADDM 使用的最新服务器配置信息。 服务器利用率数据包括以下内容：

- CPU
- 内存
- 网络
- 磁盘

未知服务器报告：

"未知服务器报告"显示 TADDM 无法识别的所有已发现的服务器进程。 此报告按系统对发现的服务器进程进行分组，并且按标准主机名列出这些服务器进程。 此报告没有参数。

完成发现后，拓扑构建代理程序将标识未知的服务器。 拓扑构建代理程序根据配置的频率值定期在后台运行，因此，在发现完成后，可能无法立即识别未知的服务器。 拓扑构建代理程序的缺省运行频率为每 4 小时运行一次。

因此，如果在拓扑构建代理程序完成前运行"未知服务器报告"，那么此报告可能不会列出所有的未知服务器。

在此报告中，将显示以下信息：

名称 正在运行未知服务器进程的计算机的名称。

上下文 IP
正在运行未知服务器进程的计算机的 IP 地址。

PID 未知服务器进程的进程标识。

PPID 未知服务器进程的父进程的进程标识。

命令行
用来运行未知服务器进程的命令。

此报告的数据从 BB_RUNTIMEPROCESS15_V 数据库视图中获取。

运行 BIRT 报告

可以使用数据管理门户网站的"分析"部分来运行 BIRT 报告。

关于此任务

要点：只有在启用 BIRT 报告查看器后，才能在数据管理门户网站中运行 BIRT 报告。 BIRT 报告查看器由于安全性问题而处于禁用状态。 另一种查看 BIRT 报告的方法是，先将 TADDM 报告导入到 Tivoli Common Reporting (TCR) 中，然后使用 TCR 来查看这些报告。 如果您了解这些风险，那么可以复原 BIRT 报告查看器。

过程

要运行 BIRT 报告，请完成下列步骤：

1. 在"功能"窗格中，单击**分析**。
2. 在"分析"部分中，单击 **BIRT 报告**。 这将打开 **TADDM BIRT 报告**列表，该列表显示所有可用的 BIRT 报告。
3. 在 **TADDM BIRT 报告**列表中，单击以突出显示要运行的报告。

4. 可选： 指定位置标记值。 COLLATION_HOME/etc/collation.propertes 文件中的 com.ibm.cdb.locationTaggingEnabled 值必须设置为 true。 将仅显示此特定位置标记的报告数据。

   注：TADDM 随附的 BIRT 报告当前不支持不进行其他定制的位置过滤。
5. 单击**运行报告**。 如果报告具有任何参数，那么系统将提示您指定参数值。 完成指定参数值之后，请单击**确定**。

结果

格式化的报告将显示在"BIRT 报告查看器"窗口中。 单击报告顶部的图标可以在报告中向前和向后翻页、打印报告或者将报告导出到文件。要打开显示了有关部分报告数据的更多详细信息的深入钻取报告，请单击报告中的链接。

注：.doc 格式的导出报告与 Microsoft Word 2003 或更高版本兼容。

从命令行界面运行 BIRT 报告

可以从 TADDM 服务器的命令行界面运行 BIRT 报告。

过程

要从命令行界面运行 BIRT 报告，请完成下列步骤：

1. 打开命令提示符，然后根据使用的 TADDM 版本浏览至以下某个目录：
   - 7.3.0：$COLLATION_HOME/deploy-tomcat/birt-viewer/WEB-INF/resources
   - 7.3.0.1 及更高版本：$COLLATION_HOME/apps/birt-viewer/WEB-INF/resources
2. 设置 BIRT_HOME 变量。 执行下列其中一项操作：
   - 在 Linux 上，根据使用的 TADDM 版本运行以下某个命令：
     - 7.3.0：

       ```
       export BIRT_HOME=$COLLATION_HOME/deploy-tomcat/birt-viewer
       ```
     - 7.3.0.1 及更高版本：

       ```
       export BIRT_HOME=$COLLATION_HOME/apps/birt-viewer
       ```
   - 在 Windows 上，根据使用的 TADDM 版本运行以下某个命令：
     - 7.3.0：

       ```
       set BIRT_HOME=%COLLATION_HOME%/deploy-tomcat/birt-viewer
       ```
     - 7.3.0.1 及更高版本：

       ```
       set BIRT_HOME=%COLLATION_HOME%/apps/birt-viewer
       ```
3. 运行 BIRT 报告。 执行下列其中一项操作：
   - 在 Linux 上，运行以下命令：

     ```
     ./genReport.sh -f format -o output -F parameters report
     ```
   - 在 Windows 上，运行以下命令：

     ```
     genReport.bat -f format -o output -F parameters report
     ```

   以下命令行选项可以与 **genReport** 程序配合使用：

   format

   报告文件的输出格式。 有效值为 PDF 和 HTML。

output

要生成的报告文件的路径。 例如，/home/cognos/utilization.pdf（对于 Linux）或 C:\data\utilization.pdf（对于 Windows）。

parameters

（可选）属性文件的路径，其中每个属性都表示报告所需的一个参数。 例如，/home/cognos/utilization.properties（对于 Linux）或 C:\data\utilization.properties（对于 Windows）。

以下文本是属性文件内容的示例：

```
scope=All Windows Machines
metric=ALL
operator=N/A
value1=N/A
value2=N/A
appdeps=N/A
```

必须确保使用反斜杠字符对参数名称中的空格进行转义。 例如，如果参数名称为 Snapshot ID Parameter，那么属性文件中的条目应该为

```
Snapshot\ ID\ Parameter=my_id
```

report

要运行的报告的路径（对名称追加了字符串"compiled"）。 例如：

- 在 Linux 和 TADDM 7.3.0 上： $COLLATION_HOME/deploy-tomcat/birt-viewer/WEB-INF/report/taddm_server_utilization.rptdesigncompiled。
- 在 Linux 和 TADDM 7.3.0.1 及更高版本上： $COLLATION_HOME/apps/birt-viewer/WEB-INF/report/taddm_server_utilization.rptdesigncompiled。
- 在 Windows 和 TADDM 7.3.0 上： %COLLATION_HOME%\deploy-tomcat\birt-viewer\WEB-INF\report\taddm_server_utilization.rptdesigncompiled。
- 在 Windows 和 TADDM 7.3.0.1 及更高版本上： %COLLATION_HOME%\apps\birt-viewer\WEB-INF\report\taddm_server_utilization.rptdesigncompiled

结果

注：genReport 命令不会生成深入报告。 因此，生成的报告中的链接不起作用。

导入 BIRT 报告

您可以使用数据管理门户网站通过导入 BIRT 报告设计来添加定制报告。

开始之前

要添加定制报告，必须先使用 BIRT 设计器工具来设计和开发报告。 必须将报告设计保存到可从客户机系统访问的 .rptdesign 文件中。

注： Fix Pack 3 在 TADDM 7.3.0.3 以及更高版本中，扩展属性数据库视图中的列具有特定数据类型，例如，VARCHAR。 在先前 TADDM 发行版中，列仅具有 CLOB 类

型。 因此，在升级到 FP3 后，使用扩展属性的 BIRT 报告可能会停止工作。 例如，如果扩展属性列未强制转换为特定数据类型，例如，VARCHAR，那么可能生成错误。

过程

要导入 BIRT 报告，请完成下列步骤：

1. 在数据管理门户网站的"功能"窗格中，单击**分析**。
2. 在"分析"部分中，单击 **BIRT 报告**。 这将打开 **TADDM BIRT 报告**列表，该列表显示了所有可用的 BIRT 报告。
3. 单击**新建**。
4. 出现系统提示时，指定新报告的详细信息，包括报告设计文件的名称、描述和位置。 名称和描述用于标识 **TADDM BIRT 报告**列表中的报告。
5. 单击**确定**。

结果

报告设计将上载到服务器，新报告将可从数据管理门户网站中获得。

注：如果该报告在服务器上已存在，那么导入将失败。 即使现有报告未显示在数据管理门户网站中，也会发生这种情况。（例如，在同步服务器上不支持"服务器亲缘关系报告"，因此，该报告不会显示在数据管理门户网站中，即使它在服务器上存在也是如此。）

删除 BIRT 报告

可以使用数据管理门户网站从服务器中删除 BIRT 报告。

开始之前

从服务器中删除报告将从该服务器的报告目录中删除该报告所使用的 .rptdesign 文件。如果要保存报告设计以供今后使用，请确保在删除报告之前创建了 .rptdesign 文件的备份副本。

过程

要删除 BIRT 报告，请完成下列步骤：

1. 在"功能"窗格中，单击**分析**。
2. 在"分析"部分中，单击 **BIRT 报告**。 这将打开 **TADDM BIRT 报告**列表，该列表显示所有可用的 BIRT 报告。
3. 选择要删除的报告。
4. 单击**删除**。
5. 要刷新 **TADDM BIRT 报告**列表，请单击**刷新**。

结果

所选报告将从服务器中删除，并且不再显示在数据管理门户网站中的 **TADDM BIRT 报告**中。 此外，还将从 TADDM 服务器的报告目录中删除该报告的 .rptdesign 文件。

导出 BIRT 报告设计

可以使用数据管理门户网站从服务器导出 BIRT 报告设计。

关于此任务

如果要使用现有报告作为新定制报告的基础，或者如果要将报告设计导入其他服务器中，那么您可能希望导出报告设计。

过程

要导出 BIRT 报告设计，请完成下列步骤：

1. 在"功能"窗格中，单击**分析**。
2. 在"分析"部分中，单击 **BIRT 报告**。 这将打开 **TADDM BIRT 报告**列表，该列表显示所有可用的 BIRT 报告。
3. 选择要导出的报告。
4. 单击**下载**。
5. 出现浏览器提示时，请指定要保存文件并指定位置。

结果

所选报告使用的设计将作为 .rptdesign 文件保存在您指定的位置中。 可以使用 BIRT 设计器工具打开并修改此文件。

复原 BIRT 报告查看器

如果您了解与安全性相关的风险，并且仍希望使用 BIRT 报告查看器，那么可以将其复原。

过程

1. 在 `collation.properties` 文件中，将 `com.ibm.taddm.birtviewer.enabled` 属性设置为 `true`：

   ```
   com.ibm.taddm.birtviewer.enabled=true
   ```

2. 重新启动 TADDM 服务器。

   注：对于 TADDM 服务器升级，此标志在缺省情况下将设置为 `false`。

使用快照工具

使用快照工具可以生成在创建快照时存在的计算机系统信息和发现事件以及正在运行的服务器应用程序的副本。

另外，还可以使用快照工具来装入协调处理中使用的信息，例如：

- 装入所需服务器的列表（也称为参考列表）
- 装入排除的服务器的列表

您可以使用报告来查询快照工具所捕获的信息，例如：

- 添加或除去了哪些服务器
- 物理服务器与虚拟服务器的比率
- 由于未能成功建立 SSH 会话而无法完全发现哪些服务器
- 已发现的服务器的列表与预期服务器列表之间的差异

限制：请在发现和拓扑代理程序完成运行之后创建快照。 如果在拓扑代理程序完成处理发现的信息之前创建快照，那么某些快照报告（例如"快照会话失败报告"）可能不完整。

snapshot.sh 命令语法：

您可以使用 snapshot.sh 命令来创建系统以及相关事件和服务器的快照。 snapshot.sh 命令位于 $COLLATION_HOME/bin 目录中。

可以在 TADDM 服务器上运行 snapshot.sh 命令。 在流式服务器部署中，必须在主存储服务器上运行 snapshot.sh 命令。

命令语法

snapshot.sh *action* [*action_parameter*]

参数

addexclude *filename* [*exclude_list*]
向文件中添加排除列表，或者替换文件中的现有排除列表实例。

addref *filename* [*reference_list*]
向文件中添加参考列表，或者替换文件中的现有参考列表实例。

clear
清除所有快照数据并删除表。

compare [*snapshot_A snapshot_B*]
根据主机名显示最后两个快照（即 snapshot_A 和 snapshot_B）之间的差异。

compareref [*snapshot_A reference_list*]
显示快照与参考列表之间的差异。

comparesig [*snapshot_A snapshot_B*]
根据主机名或操作系统的更改特征符，显示最后两个快照（即 snapshot_A 和 snapshot_B）之间的差异。

compsys
显示计算机系统。

detail [*snapshot_A*]
显示最新快照（即 snapshot_A）中的计算机系统的所有详细信息。

detailos [*snapshot_A*]
显示最新快照（即 snapshot_A）中的计算机系统的操作系统信息。

help
显示有关 **snapshot.api** 命令的详细用法帮助。

list [*snapshot_A*]
显示最新的快照，即 snapshot_A。

listall [*default*]
显示所有快照。

listexclude [*exclude_list*]
显示最新的排除列表或者由名称指定的排除列表。

listref [*reference_list*]
显示最新的参考列表或者由名称指定的参考列表。

listallexclude
显示所有排除列表。

listallref
显示所有参考列表。

nosession [*snapshot_A*]
显示未能主管最新快照（即 snapshot_A）中的会话的计算机系统。

remove *snapshot_A* [*type*]
除去快照 A，或者除去指定类型的所有快照。

removeexclude *exclude_list*
除去由名称指定的排除列表。

removeref *reference_list*
除去由名称指定的参考列表。

session [*snapshot_A*]
显示主管最新快照（即 snapshot_A）中的会话的计算机系统。

sensorerror [*snapshot_A*]
显示最新快照（即 snapshot_A）中的所有传感器错误。

take [*type*] [*desciption*]
创建快照，包括类型和描述信息（如果已指定这些信息）。

使用快照工具帮助减少物理服务器的数目：

使用快照工具，可以通过运行虚拟服务器减少物理服务器的数目。

过程

要获取在尝试减少使用的物理服务器数目时有用的信息，请完成下列步骤：

1. 对所有已知系统执行发现。
2. 使用快照工具创建快照：

   ```
   snapshot.sh take
   ```

   （可选）可以向快照中添加类型和描述信息：

   ```
   snapshot.sh take type description
   ```
3. 在数据管理门户网站中，运行 TADDM_SNAPSHOT_SESSION_FAILED 报告。 此报告将返回由于未能建立 SSH 会话而无法发现的系统的相关信息。
4. 确保可以与所有系统建立 SSH 会话。可能需要在 TADDM 中更新认证详细信息。
5. 仅对第一次发现过程中未访问的系统执行发现，以确保解决了所有连接问题。
6. 在适当的时间段（例如，1 个月）过后，对所有已知系统执行发现。
7. 在数据管理门户网站中，运行 TADDM_SNAPSHOT_CHANGE 报告。 此报告将返回有关自创建快照以来可见的新系统、不再存在的系统以及物理服务器与虚拟服务器的百分比比率的信息。

使用快照工具对预期系统列表和实际系统列表进行协调：

您可以使用快照工具和预定义报告来验证网络中的服务器列表是否与预期服务器列表匹配。

过程

要对预期系统和实际系统进行协调，请完成下列步骤：

1. 准备一个包含预期服务器列表的参考列表。 参考列表是"以逗号分隔值"(CSV) 格式的文本文件，其中包含下列字段：
   - 主机名
   - IP 地址
   - 机架
   - 操作系统
   - 主机类型
   - 注释
   - 支持区域
   - 位置

   有关参考文件的语法的更多信息，请运行带有 help 参数的 **snapshot.sh** 命令：

   ```
   snapshot.sh help
   ```
2. 根据需要，准备一个包含要在协调过程中忽略的服务器列表的排除列表。 排除列表是 CSV 格式的文本文件，其中包含下列字段：
   - 主机名
   - 排除类型

   有关排除文件的语法的更多信息，请运行带有 help 参数的 **snapshot.sh** 命令：

   ```
   snapshot.sh help
   ```
3. 使用快照工具创建快照：

   ```
   snapshot.sh take
   ```

   （可选）可以向快照中添加类型和描述信息：

   ```
   snapshot take type description
   ```
4. 在数据管理门户网站中，运行下列其中一个 BIRT 报告：
   - TADDM_SNAPSHOT_RECONCILIATION_SUMMARY
   - TADDM_SNAPSHOT_RECONCILIATION_DETAIL

在同步服务器部署中使用快照报告：

您可以通过运行预定义快照报告的企业版本在同步服务器部署中收集信息。

过程

要在同步服务器部署中运行预定义的快照报告，请完成下列步骤：

1. 如果尚未创建快照表，请进行设置。 为此，请完成下列步骤：
   a. 在每台 TADDM 服务器上，运行不带参数的 snapshot.sh 命令。
   b. 在每台域服务器和同步服务器上重新启动 TADDM。

此过程将创建快照表（如果这些表尚未存在）。 对于每个 TADDM 环境，仅需设置一次快照表。

2. 根据需要，在每个 TADDM 域中运行发现，从而对每个域创建快照。
3. 在同步服务器上执行同步。 请确保包括所有的域。
4. 创建企业快照。 在同步服务器上，运行以下命令：

   ```
   snapshot.sh take
   ```

5. 在每个域中运行报告。 请使用每个快照报告的常规版本，例如 TADDM_SNAPSHOT_CHANGE。
6. 在同步服务器上运行报告。 请使用每个快照报告的企业版本，例如 TADDM_SNAPSHOT_SYNC_CHANGE。

将 TADDM 与其他 Tivoli 产品集成

为了在管理 IT 环境时使用扩展的功能，您可以将 IBM Tivoli Application Dependency Discovery Manager (TADDM) 与其他 Tivoli 产品集成，这些产品包括 IBM Tivoli Business Service Manager、IBM Tivoli Monitoring 和事件管理系统（例如 IBM Tivoli Netcool/OMNIbus）。

受支持的版本

您可以使用下表来查看可与 TADDM 集成的产品受支持的版本。

下表显示了可与 TADDM 集成的产品受支持的版本。

表 *40.* 受支持的产品版本。

产品名称	受支持的版本
上下文菜单服务和数据集成服务 (CMS/DIS)	
IBM Control Desk (ICD)	• 7.6
IBM SmartCloud® Control Desk (SCCD)	• 7.5.1 - 使用最新提供的修订包级别
IBM Tivoli Business Service Manager (TBSM)	• 4.2.1 - 使用最新提供的修订包级别 • 6.1.0 - 使用最新提供的修订包级别 • 6.1.1 - 使用最新提供的修订包级别
IBM Tivoli Change And Configuration Management Database (CCMDB)	• 7.2.1
IBM Tivoli Integration Composer (ITIC)	• 7.5.1 - 使用最新提供的修订包级别
IBM Tivoli Monitoring (ITM)	• 6.2.1 • 6.2.2 - FP3 • 6.2.3 • 6.3
IBM Tivoli Netcool®/OMNIbus	• 7.3 • 7.4 • Fix Pack 1 8.*x* - 受 TADDM 7.3.0.1 和更高版本的支持

表 40. 受支持的产品版本。 (续)

产品名称	受支持的版本
IBM Tivoli Network Manager IP (ITNMIP)	• 3.9 • 4.1
Jazz for Service Management (JazzSM)	• 1.1
Tivoli Common Reporting (TCR)	• 1.3 • 2.1.1 • 3.1
Tivoli Directory Integrator (TDI)	• 7.0 • 7.1 • 7.1.1
Tivoli Netcool/IMPACT	• 7.1
Tivoli Workload Scheduler (TWS)	• 8.5.1 • 8.6

有关与 TADDM 集成的产品的更多信息，请参阅相应的文档：

- 有关上下文菜单服务和数据集成服务 (CMS/DIS) 的信息，请参阅《TADDM 安装指南》中的『配置上下文菜单服务和数据集成服务』主题
- IBM Control Desk (ICD)
- IBM SmartCloud Control Desk (SCCD)
- IBM Tivoli Business Service Manager (TBSM)
- IBM Tivoli Change and Configuration Management Database (CCMDB)
- IBM Tivoli Integration Composer (ITIC)
- IBM Tivoli Monitoring (ITM)
- IBM Tivoli Netcool/OMNIbus
- IBM Tivoli Network Manager IP (ITNMIP)
- Jazz for Service Management (JazzSM)
- Tivoli Common Reporting (TCR)
- Tivoli Directory Integrator (TDI)
- Tivoli Netcool/Impact
- Tivoli Workload Scheduler (TWS)

通过 OSLC 自动化将 TADDM 与 IBM Tivoli Monitoring 相集成

通过使用 OSLC 自动化可将 TADDM 与 IBM Tivoli Monitoring 相集成。 如果要将 TADDM 与 IBM Tivoli Monitoring 6.3 相集成，建议使用 OSLC 自动化。 不推荐使用 IBM Tivoli Monitoring Scope 传感器进行集成，在后续发行版中将移除这个老方法。

TADDM 可按照以下两种方式来使用 IBM Tivoli Monitoring 基础结构：

- TADDM 可通过 OSLC 自动化会话从 Tivoli Enterprise Portal 服务器获取 IBM Tivoli Monitoring 端点列表。

- 在第 2 层和第 3 层发现中，TADDM 不仅在传感器的目标系统上运行 CLI 命令，还捕获这些命令的输出。

如果您遇到任何问题，请参阅《TADDM 故障诊断指南》中的『ITM OSLC Execute Automation Service Provider 问题』主题。

Fix Pack 5

先决条件：

如果使用 Windows 7 及更高版本，您需要：

1. PowerShell V2+
2. TEMS SOAP URL
3. 检查您可以连接 TEMS 和 TEPS

必须完成下表中提供的步骤，才能通过 OSLC 自动化成功将 TADDM 与 IBM Tivoli Monitoring 相集成。

表 *41.* 通过 *OSLC* 自动化将 *TADDM* 与 *IBM Tivoli Monitoring* 相集成

步骤	详细信息
配置 ITM Tivoli Enterprise Monitoring Server (TEMS) 和 ITM TEPS 主机	第 177 页的『配置 ITM Tivoli Enterprise Monitoring Server (TEMS) 和 ITM TEPS 主机』
在 IBM Tivoli Monitoring 上安装 OSLC Execute Automation Service Provider **注：**请确保满足 第 177 页的『在 IBM Tivoli Monitoring 上安装 OSLC Execute Automation Service Provider 的先决条件』 中指定的所有先决条件。	第 180 页的『在 IBM Tivoli Monitoring 上安装 OSLC Execute Automation Service Provider』
将 TADDM 配置为使用 OSLC Execute Automation Service Provider	第 184 页的『将 TADDM 配置为使用 OSLC Execute Automation Service Provider』
为发现配置 TADDM： • 配置 `collation.properties` 文件中的自动化属性。 • 在访问列表中新建类型为<"集成">"OSLC 自动化"的访问列表条目。	第 99 页的『进行配置以通过 OSLC 自动化会话执行发现』

完成上述步骤后，可以使用 ITM OSLC Execute Automation Service Provider 来运行发现。

相关概念:

第 183 页的『通过 OSLC 自动化将 TADDM 与其他产品集成』
可以使用生命周期协作开放式服务 (OSLC) 自动化将 TADDM 与其他产品集成。TADDM 将连接到 OSLC Execute Automation Service Provider，后者提供有关其他产品基础结构的数据，而这些数据可以由 TADDM 使用 OSLC 自动化会话发现。

ITM OSLC Execute Automation Service Provider

ITM OSLC Execute Automation Service Provider 用于将 IBM Tivoli Monitoring 所管理的有关端点 IP 地址的数据导入 TADDM 中，并通过 OSLC 自动化会话来发现 IBM Tivoli Monitoring 端点。

图 1. 显示 TADDM 连接到 ITM OSLC Execute Automation Service Provider，后者使用 KT1 命令来收集有关 ITM 所管理的基础结构的数据。

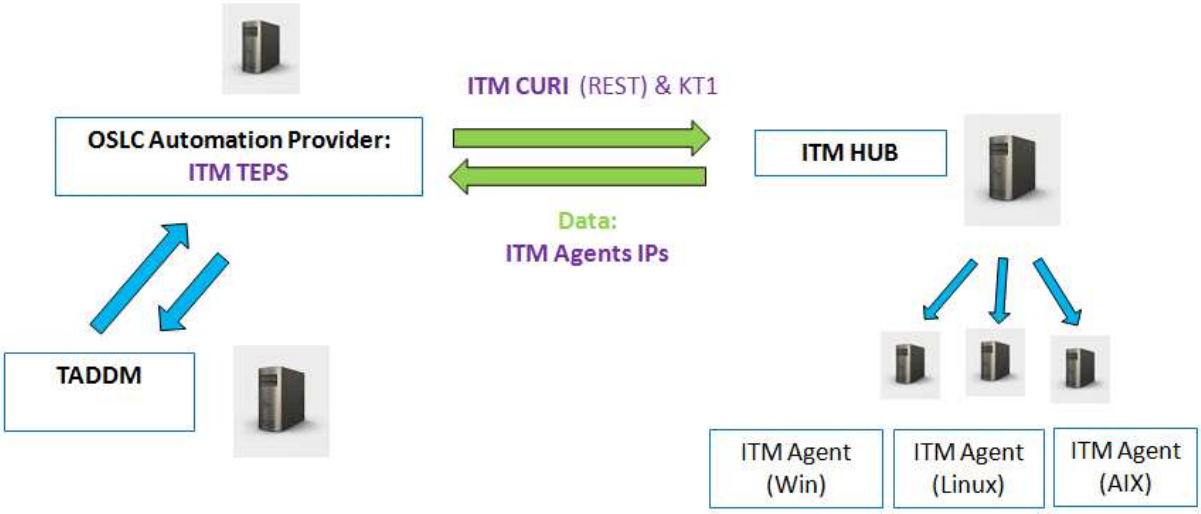

图 *4. TADDM* 连接到 *ITM OSLC Execute Automation Service Provider*，后者使用 *KT1* 命令来收集有关 *ITM* 所管理的基础结构的数据。

TADDM 从 Jazz SM Registry Services 或 `collation.properties` 文件中获取 OSLC Execute Automation Service Provider 的目标。 图 2. 显示 TADDM 使用 Jazz SM Registry Services 获取 OSLC Execute Automation Service Provider 的地址。

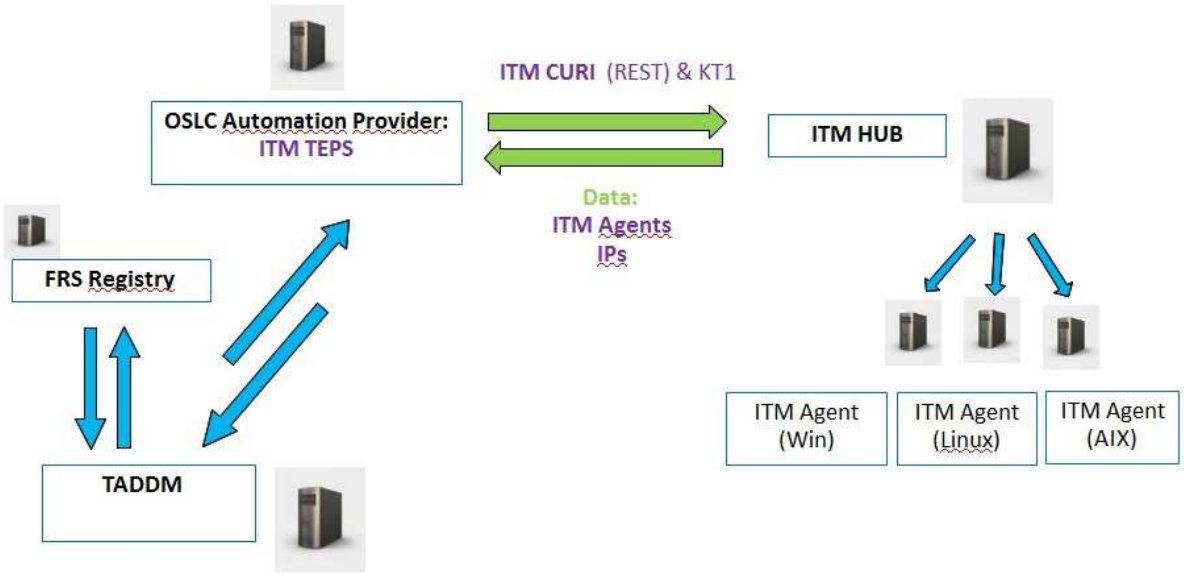

图 *5. TADDM* 使用 *Jazz SM Registry Services* 获取 *OSLC Execute Automation Service Provider* 的地址。

TADDM 可以直接连接到多个 OSLC Execute Automation Service Provider 和单个 Jazz SM Registry Services（在该 Jazz SM Registry Services 中，可以注册多个提供程序）。 图 3. 显示 TADDM 从 Jazz SM Registry Services 下载多个 ITM TEPS（门户网站服务器）上部署的 OSLC Execute Automation Service Provider 的地址。

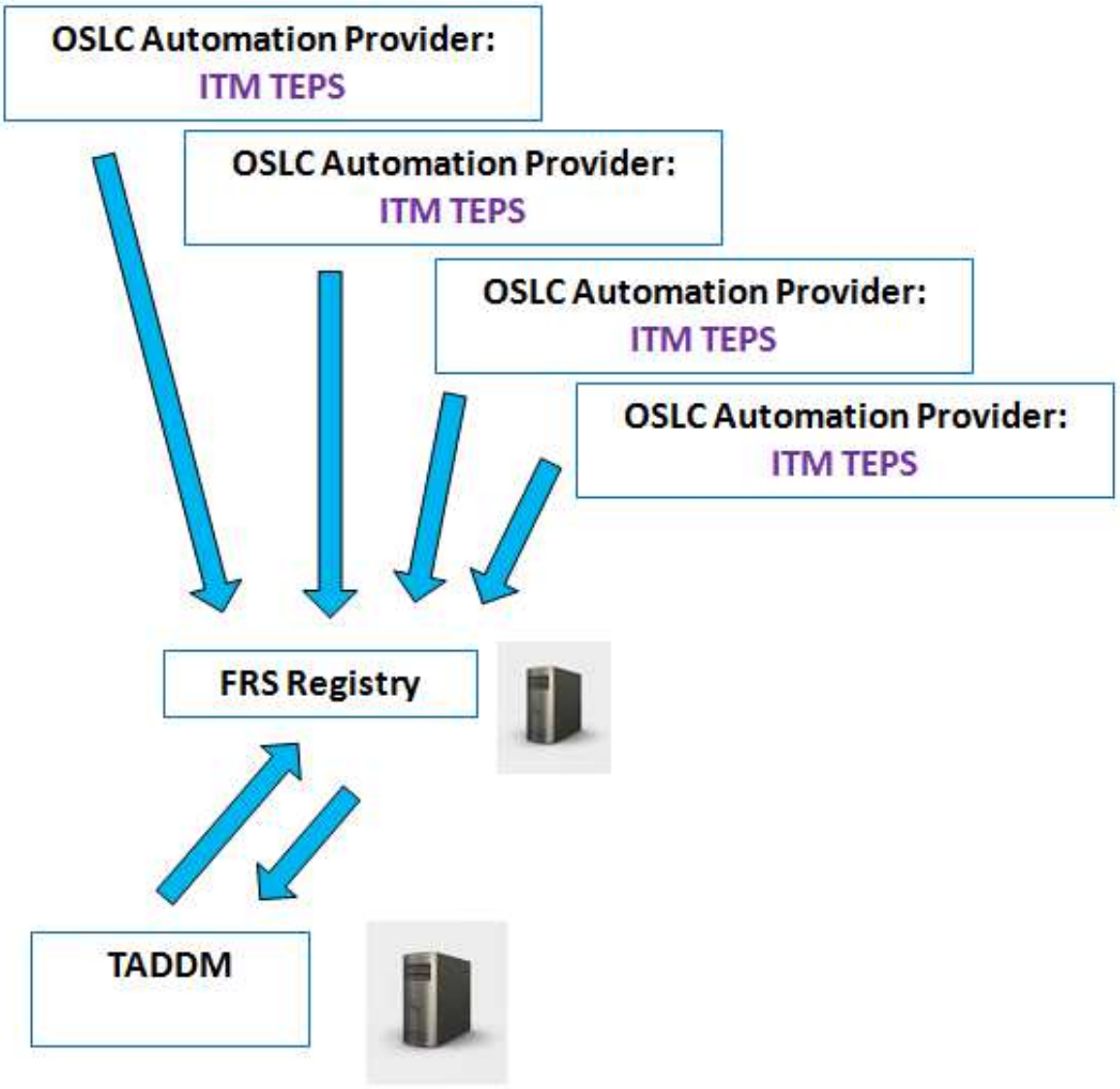

图 6. *TADDM* 从 *Jazz SM Registry Services* 下载多个 *ITM TEPS*（门户网站服务器）上部署的 *OSLC Execute Automation Service Provider* 的地址。

相关概念:

第 184 页的『OSLC Execute Automation Service Provider』
OSLC Execute Automation Service Provider 用于将其他产品所管理的端点的 IP 地址相关数据填充到 TADDM 中。 这些数据用于通过 OSLC 自动化会话来发现端点。

ITM OSLC Execute Automation Service Provider 安装

要将受管于 IBM Tivoli Monitoring (ITM) 的端点的 IP 地址数据导入 TADDM，或要运行发现，必须在 IBM Tivoli Monitoring 上安装 OSLC Execute Automation Service Provider。

如果您遇到任何问题，请参阅《TADDM 故障诊断指南》中的『ITM OSLC Execute Automation Service Provider 问题』主题。

在 *IBM Tivoli Monitoring* 上安装 *OSLC Execute Automation Service Provider* 的先决条件：

必须先将环境配置为满足所有先决条件，然后才能在 IBM Tivoli Monitoring (ITM) 上安装 OSLC Execute Automation Service Provider。

ITM OSLC Execute Automation Service Provider 必须安装在 ITM Tivoli Enterprise Portal Server (TEPS) 主机上。 受支持的 IBM Tivoli Monitoring 版本为 IBM Tivoli Monitoring 6.3。

配置 ITM Tivoli Enterprise Monitoring Server (TEMS) 和 ITM TEPS 主机

Fix Pack 5 **步骤 1- 重新配置 TEMS 和 TEPS**

最好使用 Manage Tivoli Enterprise Monitoring Services (MTEMS) GUI 进行 TEMS 和 TEPS 配置。

对于 Windows 操作系统，请启动 ITM **kinconfig .exe** 进程以启动 Manage Tivoli Enterprise Monitoring Services (MTEMS) GUI。

对于 Unix/Linux，请使用 CLI 命令"**./itmcmd manage**"启动 MTEMS GUI。

对于两个 ITM 组件（TEMS 和 TEPS）中的每一个：

- 在 MTEMS UI 上突出显示组件
- 右键单击并选择"重新配置"

然后 TEMS 或 TEPS 配置窗口会显示。

请正确提供所有 TEMS 参数（主要配置设置中的 TEMS 类型、TEMS 名称和协议以及"高级"设置中的主机名/IP 地址和端口）并选择 "**OK**"。

在 TEPS 配置窗口中选择"**Enable the dashboard data provider**"。

Fix Pack 5 **步骤 2 - 执行 TADDM 脚本和配置**

此步骤包含执行两个主要的 TADDM 脚本以帮助设置到 ITM 的 TADDM 集成。

1. 配置 provider.properties 文件

ITM TEMS 主机

通过在 ITM TEMS 主机上运行下列其中一个脚本，启用 KT1 命令 (**tacmd get/put/execute**)：

- 对于 Linux 操作系统：

```
TADDM_CD_ISO/itm-discovery-support/configure_tems.sh <-i <ITM_HOME>>
 [-t <TEMP-DIR>
```

- 对于 Windows 操作系统：

```
TADDM_CD_ISO/itm-discovery-support/configure_tems.ps1 <-i <ITM_HOME>>
 [-t <TEMP-DIR>
```

其中，*<ITM_HOME>* 是 ITM TEPS 安装目录（例如 /opt/IBM/ITM），*<TEMP-DIR>* 是临时文件目标目录。 *<TEMP-DIR>* 参数的缺省值为 /var/log/automation_provider。

Fix Pack 5 完成后，您应该会在脚本执行的末尾看到以下代码行。

```
INFO: Stopping ITM TEPS...
INFO: ITM TEPS has stopped.
INFO: Starting ITM TEPS...
INFO: ITM TEPS has started.
INFO: Checking if TEPS is running...
INFO: Checking if OSLC Automation Provider is installed...
INFO: Installation of OSLC Automation Provider successfully finished.
```

注：要进行初始用户标识验证 (**tacmd get/put/execute**)，请使用 **http:1920** 或 **https:3661**，然后对从自动化提供程序到所发现目标的 KT1 工作使用 **ip.pipe:1920** 或 **ip.spipe:3660**。应该在 ITM 中启用这些协议才能完成发现。

ITM TEPS 主机

验证是否已安装或启用 ITM 仪表板数据提供程序。 如果尚未安装或启用，您可以选择性地进行安装或启用。 请参阅 IBM Tivoli Monitoring 文档中的验证是否已启用仪表板数据提供程序主题。

要点：如果使用的是 64 位 Windows 操作系统，请确保系统路径（或 kfwenv 文件中的路径条目）指向 64 位 TMAITM6 目录。 如果未指向该目录，请进行手动添加。 例如，如果将 ITM 安装在 C:\IBM\ITM\ 目录中，那么在系统路径环境变量中或 C:\IBM\ITM\CNPS\kfwenv 文件的 PATH 伪指令中必须指定 C:\IBM\ITM\TMAITM6_x64。

Fix Pack 5

2. 从 ITM TEPS 主机运行 automation_provider 脚本

在服务器上执行 **automation_provider** 脚本：

- 对于 Linux 操作系统：

  ```
  automation_provider.sh install  -t /tmp/log  -i /opt/IBM/ITM -c
  /tmp/provider.properties
  ```

- 对于 Windows 操作系统：

  ```
  automation_provider.ps1 install  -t /tmp/log  -i /opt/IBM/ITM -c
  /tmp/provider.properties
  ```

此处需要注意的很重要的一点是尽管在运行 automation_provider 脚本前已创建 provider.properties 文件，会忽略该文件并在 ITM TEPS 配置路径 ($ITM_HOME/iw/profiles/ITMProfile/installedApps/ITMCell/itmautomationprovider.ear/itmautomationprovider.war/WEB-INF/provider.properties) 中创建"缺省"provider.properties。

必须找到该文件，手动修改参数，然后重新启动 TEPS 以使文件生效。

需要的文件

在要从中运行安装脚本的目录中，必须存在下列文件：

- 安装脚本：
  - 对于 Linux 和 AIX 操作系统：TADDM_CD_ISO/itm-discovery-support/automation_provider.sh 及其子模块位于 TADDM_CD_ISO/itm-discovery-support/mod/sh/。
  - 对于 Windows 操作系统：TADDM_CD_ISO/itm-discovery-support/automation_provider.ps1 及其子模块位于 TADDM_CD_ISO/itm-discovery-support/mod/ps/。

- itmautomationprovider.ear - ITM OSLC Execute Automation Service Provider 所在的软件包。 此文件的准确位置为 TADDM_CD_ISO/itm-discovery-support/ear/itmautomationprovider.ear。
- provider.properties - ITM OSLC Execute Automation Service Provider 的示例配置文件。 您可以手动配置此文件，并将其作为参数传递给安装脚本。如果未传递此文件，那么安装期间必须提供所需的参数。 此文件的准确位置为 TADDM_CD_ISO/itm-discovery-support/template_provider.properties。
- KT1 支持适用于相应操作系统及其体系结构（32 位或 64 位）的库。
  - 对于 Linux 操作系统：

    TADDM_CD_ISO/itm-discovery-support/linux32
    TADDM_CD_ISO/itm-discovery-support/linux64
  - 对于 AIX 操作系统：

    TADDM_CD_ISO/itm-discovery-support/aix32
    TADDM_CD_ISO/itm-discovery-support/aix64
  - 对于 IBM System Z (zLinux) 上的 Linux：

    TADDM_CD_ISO/itm-discovery-support/linuxz32
    TADDM_CD_ISO/itm-discovery-support/linuxz64
  - 对于 Windows 操作系统：

    TADDM_CD_ISO/itm-discovery-support/win32
    TADDM_CD_ISO/itm-discovery-support/win64

配置 provider.properties 文件

您可以通过设置下列参数来选择性地配置 provider.properties 文件：

- com.ibm.automationprovider.registration.host=*http://localhost:15210* - 启用 ITM 连接。此值指定 TEPS 的公用 URL。 此参数的缺省值为 http://localhost:15210。

  注：请修改 localhost，以使其表示 TEPS 服务器的主机名或 IP 地址。
- com.ibm.automationprovider.itm.curi.url=*http://localhost:15210* - 指定 ITM CURI (REST) 提供程序的 URL 地址。缺省值为 http://localhost:15210。

  注：请修改此示例中的 localhost，以使其表示您的 TEPS 服务器的主机名或 IP 地址。
- com.ibm.automationprovider.itm.soap.url=*http://localhost:1920///cms/soap* - 指定 ITM SOAP 的 URL 地址。 缺省值为 http://localhost:1920///cms/soap。

  注：请修改此示例中的 localhost，以使其表示您的中心 TEMS 服务器的主机名或 IP 地址，中心 TEMS 服务器可能与您的 TEPS 位于同一主机上，也可能位于不同主机上（大多数生产环境应使 TEMS 和 TEPS 位于不同服务器上）。

注：如果您采用非缺省 ITM CURI 或 ITM SOAP 配置，并且/或者在 ITM TEPS 上配置了 SSL 安全性，请确保对 com.ibm.automationprovider.itm.curi.url 和 com.ibm.automationprovider.itm.soap.url 属性指定正确的 URL 地址。

provider.properties 文件中指定的参数值优先于从命令行定义的参数值。

Fix Pack 5 如果您想要在 RTEMS 上运行发现，那么请确保已在环境文件中启用 "KT1_TEMS_SECURE=YES"参数。

在 Jazz SM Registry Services (FRS) 中注册 OSLC Execute Automation Service Provider

可以选择性地在 Jazz SM Registry Services 中注册 OSLC Execute Automation Service Provider。请选择以下某个方法：

- 向 provider.properties 文件添加下列参数：
  - com.ibm.automationprovider.frs.url - 指定用于注册 OSLC Execute Automation Service Provider 的 FRS URL 地址。 需要集合的完整 URL 地址，例如 http://9.122.100.100:9083/oslc/pr/collection。
  - com.ibm.automationprovider.frs.user - 指定用于连接到 FRS 的用户名。
  - com.ibm.automationprovider.frs.password - 指定用于连接到 FRS 的密码。
  - com.ibm.automationprovider.registration.initialdelay=5000 - 指定从 OSLC Execute Automation Service Provider 启动开始直到第一次尝试在 FRS 中进行注册为止的间隔时间。 缺省值为 5000，并且以毫秒为单位。要禁止注册，请将此值设置为 -1。
- 在命令行中添加 -f 选项（例如 ./automation_provider.sh -f），并且在提供程序安装期间出现提示时提供所需的参数。

在 *IBM Tivoli Monitoring* 上安装 *OSLC Execute Automation Service Provider*:

要在 IBM Tivoli Monitoring (ITM) 上安装 OSLC Execute Automation Service Provider，必须运行 automation_provider 脚本。 可以采用非交互方式或交互方式来安装 OSLC Execute Automation Service Provider。

过程

要安装 OSLC Execute Automation Service Provider，请从 ITM TEPS 主机运行以下 automation_provider 脚本：

- 对于 Linux 操作系统：

```
TADDM_CD_ISO/itm-discovery-support/automation_provider.sh install
[-i <ITM-HOME>] [-t <TEMP-DIR>] [[-c <CONFIG-FILE> | [-h <TEPS-IP>]
 [-p <TEPS-PORT>]] [-f]
```

- 对于 Windows 操作系统：

```
TADDM_CD_ISO/itm-discovery-support/automation_provider.ps1 install
[-i <ITM-HOME>] [-t <TEMP-DIR>] [[-c <CONFIG-FILE> | [-h <TEPS-IP>]
 [-p <TEPS-PORT>]] [-f]
```

其中：

-i *<ITM-HOME>*
是 ITM TEPS 的安装目录，例如 /opt/IBM/ITM。

-t *<TEMP-DIR>*
是临时文件目标目录。 缺省值为 /var/log/automation_provider。

-h *<TEPS-IP>*
是 ITM TEPS 主机的 IP 地址。

-p *<TEPS-PORT>*
是 ITM TEPS 的 HTTP 端口。

-c *<CONFIG-FILE>*
是包含 OSLC Execute Automation Service Provider 配置的 provider.properties 文件的目标。

-f 是一个标志，可用于在安装期间提示您提供在 Jazz SM Registry Services 中注册 OSLC Execute Automation Service Provider 时需要的参数。

要点：安装脚本的所有参数都是可选的。 您可以按任意顺序指定这些参数。
示例：

```
automation_provider.sh install -t /tmp/log -i /opt/IBM/ITM -h 9.100.100.200 -p 15210
automation_provider.ps1 install -i /opt/IBM/ITM
```

- 可以采用非交互方式来安装 OSLC Execute Automation Service Provider。 请完成下列步骤：
  1. 配置 provider.properties 文件。 请参阅第 179 页的『配置 provider.properties 文件』一节。
  2. 从 ITM TEPS 主机运行以下 automation_provider 脚本：
     - 对于 Linux 操作系统：

       ```
       automation_provider.sh install  -t /tmp/log
       -i /opt/IBM/ITM -c /tmp/provider.properties
       ```

     - 对于 Windows 操作系统：

       ```
       automation_provider.ps1 install  -t /tmp/log
       -i /opt/IBM/ITM -c /tmp/provider.properties
       ```

     注：如果您采用非缺省 ITM CURI 或 ITM SOAP 配置，并且/或者在 ITM TEPS 上配置了 SSL 安全性，请以非交互方式安装 OSLC Execute Automation Service Provider。请确定对 com.ibm.automationprovider.itm.curi.url 和 com.ibm.automationprovider.itm.soap.url 属性指定正确的 URL 地址。
- 可以采用交互方式来安装 OSLC Execute Automation Service Provider。 安装期间，请为第 179 页的『配置 provider.properties 文件』一节中指定的必需参数提供值。

验证 *OSLC Execute Automation Service Provider* 安装：

您可以验证 OSLC Execute Automation Service Provider 是否已成功地安装在 IBM Tivoli Monitoring 上。

过程

1. 通过运行下列命令，确保 ITM TEMS 具有任何处于运行状态的 Windows、Linux 或 UX 代理程序：

   ```
   /opt/IBM/ITM /bin/tacmd login -u admin -p password -s localhost
   /opt/IBM/ITM /bin/tacmd listSystems
   ```

2. 确保每个 ITM TEMS 都有一个自动化计划。 这些计划必须包含 ITM 端点的 IP 地址。 在 Web 浏览器中打开以下 Web 地址：

   ```
   http://<ITM_TEPS>:<ITM_PORT>/itmautomationprovider
   http://<ITM_TEPS>:<ITM_PORT>/itmautomationprovider/services/plans
   ```

   示例

   ```
   http://9.100.200.100:15210/itmautomationprovider/services/plans
   ```

检查 *ITM OSLC Execute Automation Service Provider* 的状态：

您可以检查 ITM OSLC Execute Automation Service Provider 安装的状态。

过程

运行以下 automation_provider 脚本：

- 对于 Linux 操作系统：

  automation_provider.sh status [i- <*ITM-HOME*>] [t- <*TEMP-DIR*>]

- 对于 Windows 操作系统：

  automation_provider.ps1 status [i- <*ITM-HOME*>] [t- <*TEMP-DIR*>]

其中：

i- *<ITM-HOME>*
是 ITM TEPS 的安装目录，例如 /opt/IBM/ITM。

t- *<TEMP-DIR>*
是临时文件目标目录。 缺省值为 /var/log/automation_provider。

要点：此脚本的所有参数都是可选的。 您可以按任意顺序指定这些参数。

示例

```
automation_provider.sh status
automation_provider.ps1 status -i /opt/IBM/ITM
automation_provider.sh status -t /tmp/log -i /opt/IBM/ITM
```

卸载 *OSLC Execute Automation Service Provider*：

您可以通过运行 automation_provider 脚本来卸载 ITM OSLC Execute Automation Service Provider。

过程

运行以下 automation_provider 脚本：

- 对于 Linux 操作系统：

  automation_provider.sh uninstall [i- <*ITM-HOME*>] [t- <*TEMP-DIR*>]

- 对于 Windows 操作系统：

  automation_provider.ps1 uninstall [i- <*ITM-HOME*>] [t- <*TEMP-DIR*>]

其中：

i- *<ITM-HOME>*
是 ITM TEPS 的安装目录，例如 /opt/IBM/ITM。

t- *<TEMP-DIR>*
是临时文件目标目录。 缺省值为 /var/log/automation_provider。

要点：此脚本的所有参数都是可选的。 您可以按任意顺序指定这些参数。

示例

```
automation_provider.sh uninstall
automation_provider.ps1 uninstall -i /opt/IBM/ITM
automation_provider.sh uninstall -t /tmp/log -i /opt/IBM/ITM
```

为 ITM OSLC Execute Automation Service Provider 配置发现

使用 ITM OSLC Execute Automation Service Provider 时，您可以设置以下属性来配置发现进程。

com.collation.discover.dwcount=32

缺省值为 32。

此属性是 TADDM 服务器属性，它定义了发现工作程序线程的数量。

要获得最佳结果，请将 com.collation.discover.dwcount 和 KT1_RPC_THREADS 属性设置为相同的值。

com.ibm.automationprovider.kt1.concurenttasks.limit=100

缺省值为 100。

此属性是可在 provider.properties 文件中编辑的 ITM OSLC Execute Automation Service Provider 属性。 它定义了此提供程序向 TEMS 发出的并发请求的数量。 过多的请求在提供程序级别排队。

注：仅当需要在 TADDM 和 TEMS 之间执行附加调速，或者设置了超过 100 个 KT1 工作程序线程时，更改此属性的值。

KT1_RPC_THREADS=10

缺省值为 10。

这是可在 ITM_HOME/config/kbbenv.ini 文件中编辑的 ITM TEMS 属性。 它定义了响应 KT1 请求的工作程序线程的数量。

要获得最佳结果，请将 KT1_RPC_THREADS 和 com.collation.discover.dwcount 属性设置为相同的值。

通过 OSLC 自动化将 TADDM 与其他产品集成

可以使用生命周期协作开放式服务 (OSLC) 自动化将 TADDM 与其他产品集成。TADDM 将连接到 OSLC Execute Automation Service Provider，后者提供有关其他产品基础结构的数据，而这些数据可以由 TADDM 使用 OSLC 自动化会话发现。

通过使用 OSLC Execute Automation Service Provider 执行发现是一个通用过程，此过程可以增强为包括发现其他实现了自己的 OSLC Execute Automation Service Provider 的产品。 在发现期间，将为每个 OSLC Execute Automation Service Provider 主机和/或 Jazz SM Registry Services 打开一个端口。 这可以确保更好地进行安全性控制。

下表中列示的主题包含更多有关通过 OSLC 执行发现的信息。

表 *42.* 包含更多有关通过 *OSLC* 执行发现的信息的主题。

信息	位置
针对发现进行配置	第 99 页的『进行配置以通过 OSLC 自动化会话执行发现』
TADDM 服务器属性	第 72 页的『对使用 OSLC 自动化会话的发现应用的属性』
支持使用 OSLC 自动化会话来执行发现的传感器	请参阅 TADDM *Sensor Reference* 中的 *Sensors that support discovery using OSLC Automation Session* 主题。

OSLC Execute Automation Service Provider

OSLC Execute Automation Service Provider 用于将其他产品所管理的端点的 IP 地址相关数据填充到 TADDM 中。 这些数据用于通过 OSLC 自动化会话来发现端点。

TADDM 可以从 Jazz SM Registry Services 或 `collation.properties` 文件中获取 OSLC Execute Automation Service Provider 的目标。

TADDM 可以直接连接到多个 OSLC Execute Automation Service Provider 和单个 Jazz SM Registry Services 实例（在该实例中，可以注册多个 OSLC Execute Automation Service Provider）。每个 OSLC Execute Automation Service Provider 都存储与 TADDM 集成的特定产品实例的相关信息，例如 ITM HUB 的相关信息。

相关参考:

第 174 页的『ITM OSLC Execute Automation Service Provider』
ITM OSLC Execute Automation Service Provider 用于将 IBM Tivoli Monitoring 所管理的有关端点 IP 地址的数据导入 TADDM 中，并通过 OSLC 自动化会话来发现 IBM Tivoli Monitoring 端点。

将 *TADDM* 配置为使用 *OSLC Execute Automation Service Provider*:

为了能够使用 OSLC 自动化会话来运行发现，必须将 TADDM 配置为使用 OSLC Execute Automation Service Provider。

过程

要将 TADDM 配置为使用 OSLC Execute Automation Service Provider，请完成下列步骤：

1. 确保 OSLC Execute Automation Service Provider 已安装且处于运行状态。
2. 将 TADDM 连接到 OSLC Execute Automation Service Provider。可以通过两种方法完成此任务：直接连接，或者使用 Jazz for Service Management Registry Services 进行连接。 如果存在多个 OSLC Execute Automation Service Provider，那么可以将这些方法结合使用。
   - 要将 TADDM 直接连接到 OSLC Execute Automation Service Provider，请将 OSLC Execute Automation Service Provider 的地址添加到 `collation.properties` 文件中的 `com.ibm.cdb.topobuilder.integration.oslc.automationprovider` 属性。
   - 要使用 Jazz for Service Management Registry Services 将 TADDM 连接到 OSLC Execute Automation Service Provider，请允许 TADDM 在 Jazz SM Registry Services 中查找 OSLC Execute Automation Service Provider。 请完成下列步骤：
     a. 确保 Jazz SM Registry Services 处于运行状态。
     b. 确保 OSLC Execute Automation Service Provider 已连接到 Jazz SM Registry Services。
     c. 在 `collation.properties` 文件中配置下列其中一个属性，以便向 Jazz SM Registry Services 提供地址：

        ```
        com.ibm.cdb.topobuilder.integration.oslc.frsurl
        com.ibm.cdb.topobuilder.integration.oslc.automation.frsurl
        ```
3. 重新启动 TADDM 服务器。

结果

配置 TADDM 之后，可以使用 OSLC 自动化会话来运行发现。

相关参考:

第 72 页的『对使用 OSLC 自动化会话的发现应用的属性』
这些属性应用于使用 OSLC 自动化会话的发现。

OSLCAutomationAgent 的命令行界面

OSLCAutomationAgent 用于从 OSLC Execute Automation Service Provider 中收集数据。 您可以使用命令手动运行该代理程序，并可以使用命令来刷新或更新该代理程序所创建的作用域集。

OSLC Execute Automation Service Provider 的地址是在 collation.properties 文件中进行配置，并且/或者从 Jazz SM Registry Services 下载。 该代理程序将连接到每个 OSLC Execute Automation Service Provider，以获取与 TADDM 兼容的自动化计划的列表。 自动化计划包含该代理程序用来缓存和创建发现作用域集的 IP 地址。 例如，将 TADDM 与 IBM Tivoli Monitoring 集成后，自动化计划将包含 IBM Tivoli Monitoring 所管理的 ITM TEMS 服务器及端点（代理程序）的 IP 地址。 OSLC 自动化代理程序使用 IBM Tivoli Monitoring 代理程序的 IP 地址来进行高速缓存并创建作用域集。 每个 ITM TEMS 都有一个单独的作用域集。

OSLC 自动化代理程序会在集成代理程序组中定期运行。

您可以对 OSLCAutomationAgent 使用以下命令。

- 要手动运行该代理程序，请使用以下命令：

  ```
  /taddm/dist/support/bin/runtopobuild.sh -a OSLCAutomationAgent
  ```

- 要刷新作用域集，请使用以下命令：

  ```
  /taddm/dist/support/bin/runtopobuild.sh -a OSLCAutomationAgent -s true
  ```

  注：仅当 ITM Automation Provider 的自动化计划有所变化时，才会刷新作用域集。 要强制刷新作用域集，请使用以下命令：

  ```
  /taddm/dist/support/bin/runtopobuild.sh -a OSLCAutomationAgent
  --forceScopeSetRefresh true
  ```

  这些作用域集将显示在发现管理控制台的**作用域**窗格中。

- 要显示已缓存的作用域集，请使用以下命令：

  ```
  /taddm/dist/support/bin/runtopobuild.sh -a OSLCAutomationAgent -d true
  /taddm/dist/support/bin/runtopobuild.sh -a OSLCAutomationAgent
   --displayCache true
  ```

  这些作用域集将显示在以下 TADDM 日志文件中：

  – <COLLATION_HOME>/dist/log/services/TopologyBuilder.log

  – <COLLATION_HOME>/dist/log/agents/OSLCAutomationAgent.log

  以下示例显示可以在 <COLLATION_HOME>/dist/log/agents/OSLCAutomationAgent.log 文件中找到的输出：

  ```
  2014-07-22 11:42:54,660 TopologyBuilder [pool-1-thread-1]  DEBUG
  oslc.OSLCAutomationAgent - OSLCAutomationAgent:displaying cache
  2014-07-22 11:42:54,669 TopologyBuilder [pool-1-thread-1]  INFO
   oslc.OSLCAutomationAgent - <AGENT_IP_2> http://9.120.100.100:15210/
  ```

```
itmautomationprovider/services/plans/2 1406009933764
2014-07-22 11:42:54,669 TopologyBuilder [pool-1-thread-1]  INFO
 oslc.OSLCAutomationAgent - <AGENT_IP_2> http:///9.120.100.100:15210/
itmautomationprovider/services/plans/2 1406009933764
2014-07-22 11:42:54,675 TopologyBuilder [pool-1-thread-1]  DEBUG
 oslc.OSLCAutomationAgent - OSLCAutomationAgent:cache end
```

相关概念:

第 13 页的『拓扑构建过程概述』

TADDM 定期运行拓扑构建过程。 在发现或批量装入操作后一直到拓扑构建过程完成前，TADDM 数据库中可能存在未调和的对象，拓扑关系也可能不完整。

将 TADDM 与 IBM Tivoli Monitoring（老方法）相集成

根据在 IT 环境中必须执行的特定任务，您可以使用 IBM Tivoli Application Dependency Discovery Manager (TADDM) 和 IBM Tivoli Monitoring 之间可用的集成功能。 您可以使用 IBM Tivoli Monitoring Scope 传感器将 TADDM 与 IBM Tivoli Monitoring 集成。

新的集成方法

要点： 从 TADDM V7.3.0 开始，建议使用 OSLC 自动化将 TADDM 与 IBM Tivoli Monitoring 6.3 集成。 不推荐使用 IBM Tivoli Monitoring Scope 传感器进行集成，在后续发行版中将移除这个老方法。

在 第 172 页的『通过 OSLC 自动化将 TADDM 与 IBM Tivoli Monitoring 相集成』中可了解更多有关使用 OSLC 自动化将 TADDM 与 IBM Tivoli Monitoring 集成的信息，在 TADDM *Sensor Reference* 的使用 *OSLC* 自动化来支持发现的传感器主题中可了解更多有关使用 OSLC 自动化来支持发现的传感器的信息。

旧的集成方法

以下所有部分适用于旧的集成方法。 您仍然可以使用旧的集成方法，但请记住，我们不推荐使用此方法，在后续发行版中将移除此方法。

表 1. 列出了对于应该使用的集成功能可能需要执行的一些任务，其余部分提供了有关这些集成功能的概述。

表 *43.* 用户任务以及要使用的相应集成功能

任务	要使用的集成功能
通过查看操作系统设置、应用程序设置以及 IBM Tivoli Monitoring 监视的系统的更改历史记录，可以了解可用性。	• 第 187 页的『使用 IBM Tivoli Monitoring 的发现』 • 第 188 页的『在上下文中启动』
请确保监视了 TADDM 所发现的操作系统的可用性。	• 第 187 页的『使用 IBM Tivoli Monitoring 的发现』 • 第 188 页的『监视覆盖范围报告』
查看 TADDM 发现的系统的可用性和性能。	• 第 187 页的『IBM Tivoli Monitoring DLA』 • 第 188 页的『监视覆盖范围报告』

表 43. 用户任务以及要使用的相应集成功能（续）

任务	要使用的集成功能
监视业务应用程序的配置更改。	• 『使用 IBM Tivoli Monitoring 的发现』 • 第 188 页的『更改事件』 • 第 188 页的『在上下文中启动』

使用 IBM Tivoli Monitoring 的发现

TADDM 可使用 IBM Tivoli Monitoring 6.2.1 或更高版本的基础结构来执行第 1 层、第 2 层和某些第 3 层发现。 TADDM 发现 IBM Tivoli Monitoring 环境中的配置项时，仅需要使用 Tivoli Enterprise Portal Server 的凭证，而不使用该门户网站服务器所监视的每台计算机的凭证。

TADDM 通过以下两种方式利用 Tivoli Monitoring 基础结构：

- TADDM 从 Tivoli Enterprise Portal Server 获取 Tivoli Monitoring 端点的列表，以创建基本的第 1 层发现信息，和创建更深的第 2、3 层发现的作用域。
- TADDM 使用 Tivoli Monitoring 基础结构在目标系统上为第 2、3 层发现中的传感器运行 CLI 命令，以及捕获这些命令的输出

此功能提供以下优势：

- 在现有 Tivoli Monitoring 环境中快速部署 TADDM
- 不需要 TADDM 锚点和网关服务器
- 不需要定义包含要扫描的计算机的作用域集。 仅需要具有 Tivoli Enterprise Portal Server 的单个条目的作用域。
- 不需要为发现目标定义访问列表（操作系统凭证）
- 仅需要 Tivoli Enterprise Portal Server GUI 登录的单个访问列表条目。

表 44. 包含使用 *IBM Tivoli Monitoring* 的发现的更多相关信息的主题

信息	信息的位置
使用 IBM Tivoli 配置发现	第 97 页的『使用 IBM Tivoli Monitoring（老方法）配置发现』
适用于使用 IBM Tivoli Monitoring 的发现的 TADDM 服务器属性	第 70 页的『使用 IBM Tivoli Monitoring（老方法）的发现属性』
• 支持使用 IBM Tivoli Monitoring 的发现的传感器 • IBM Tivoli Monitoring Scope 传感器，包括关于配置传感器的信息，以及在部署或使用传感器时可能发生的任何已知问题的故障诊断信息	TADDM *Sensor Reference*

IBM Tivoli Monitoring DLA

IBM Tivoli Monitoring 发现库适配器 (DLA) 从 Tivoli Monitoring 抽取有关 Tivoli Monitoring 监视的计算机系统和数据库的配置数据。 DLA 的输出是格式化的 XML 文

件，它包含这些组件及其关系。 DLA 的输出还包括表示 Tivoli Monitoring 代理程序的数据以及用于从 TADDM 启动可用性视图的数据。 有关将 DLA 导出的数据装入到 TADDM 的详细信息，请参阅《TADDM 用户指南》中的『批量装入程序』主题。

要运行 DLA，请完成下列步骤：

1. 按照 http://www-01.ibm.com/support/knowledgecenter/SSTFXA_6.2.2.1/com.ibm.itm.doc_6.2.2fp1/discoverylibraryadapter_tms.htm?lang=en 上使用 *Tivoli Management Services Discovery Library Adapter* 主题中指定的方式在 ITM 上生成 DLA。
2. 将 DLA 输出文件复制到 TADDM 主机。
3. 使用批量装入程序将 DLA 从 ITM 装入到 TADDM。 请使用以下命令：

   ```
   $COLLATION_HOME/bin/loadidml.sh -u user -p password -f path_to _DLA
   ```

如果您安装新的 Tivoli Monitoring 代理程序，它们可以对 Tivoli Monitoring DLA 提供更多支持。 这些代理程序可提供用于填充监视覆盖范围报告的信息，但操作系统报告的监视覆盖范围不需要 DLA。

如果您安装代理程序，必须为这些代理程序启用应用程序支持，以确保代理程序参与 DLA 生成的输出。 并非所有代理程序都支持 Tivoli Monitoring DLA。

有关为非标准代理程序配置应用程序支持的信息，请参阅相应的文档。 要验证代理程序是否支持 Tivoli Monitoring DLA，请参阅 IBM Tivoli Composite Application Manager 代理程序的文档。

监视覆盖范围报告

监视覆盖范围报告显示关于环境中各种组件的详细信息。 您可以生成关于环境中的操作系统、数据库、Microsoft 应用程序、VMware 服务器和 System p 组件的报告。 这些组件受 IBM Tivoli Monitoring 6.1 或更高版本代理程序监视。

有关监视覆盖范围报告的更多信息，请参阅《TADDM 用户指南》。

更改事件

您可以将 TADDM 配置为在检测到发现的资源发生更改时通知 IBM Tivoli Monitoring。

表 *45.* 包含有关更改事件的更多信息的主题

信息	信息的位置
• 配置 TADDM 发送更改事件 • 配置 IBM Tivoli Monitoring 数据提供程序 • 配置业务系统的更改事件	第 196 页的『将更改事件发送到外部系统』

在上下文中启动

使用在上下文中启动，您可以在 IBM Tivoli Monitoring 的 Tivoli Enterprise Portal 视图中查看 TADDM 数据。

通过将拓扑视图配置为在 Tivoli Enterprise Portal 中显示，您可以在 Tivoli Enterprise Portal 可用性视图中查看物理基础结构、应用程序基础结构和业务系统拓扑。

表 *46.* 包含有关在上下文中启动的更多信息的主题

信息	信息的位置
拓扑视图显示所需的 URL	第 193 页的『在上下文中启动的配置』
配置在上下文中启动以查看操作系统设置、应用程序设置以及入局更改事件的更改历史记录的指示信息	第 205 页的『在 IBM Tivoli Monitoring 中的配置更改事件报告中创建详细信息链接』

为上下文菜单服务和数据集成服务注册配置项

如果要使用上下文菜单服务 (CMS) 和数据集成服务 (DIS) 来启用灵活的跨产品启动点，必须在 CMS/DIS 数据库中注册 TADDM 配置项 (CI)。

开始之前

必须先设置 CMS/DIS 数据库，然后才能使用上下文菜单服务和数据集成服务。

关于此任务

有两种方式可在 CMS/DIS 数据库中注册 TADDM CI：

- 使用 CMS/DIS 注册脚本进行初始注册
- 使用 CMSDISAgent 拓扑构建器代理程序进行定期自动更新

运行初始注册

要完成上下文菜单服务和数据集成服务数据库中 TADDM 配置项 (CI) 的初始注册，您必须手动运行 **run_cms_dis_registration** 脚本。此初始注册完成后，CMSDISAgent 拓扑构建器代理程序才会自动更新 CI 注册。

关于此任务

如果要使用流式服务器部署，请在主存储服务器上运行注册脚本。 如果要使用同步服务器部署，请在同步服务器上运行注册脚本。

过程

要完成 TADDM CI 的初始注册：

1. 在命令提示符下，转至 `$COLLATION_HOME/bin` 目录。
2. 针对您的操作系统运行 **run_cms_dis_registration** 脚本：
   - Linux 和 UNIX 系统：

     ```
     ./run_cms_dis_registration.sh [ register [guid] |
                                     clean [guid [classtype]] |
                                     re-register-all | register-menu |
                                     help ]
     ```

   - Windows 系统：

     ```
     run_cms_dis_registration.bat [ register [guid] |
                                     clean [guid [classtype]] |
                                     re-register-all | register-menu |
                                     help ]
     ```

其中：

register [*guid*]

在上下文菜单服务和数据集成服务数据库中注册 TADDM 数据。 （可选）可以指定要注册的模型对象的全局唯一标识 (GUID)。

在使用 register 选项而不指定 GUID 的情况下首次运行脚本时，所有 TADDM 数据都将在数据库中注册，且所有启动点都会通过上下文菜单服务注册。 以后使用该选项运行时将只注册前一次运行以来对 TADDM 数据所进行的更改。 此选项为缺省选项。

如果指定 GUID，将只注册带有指定 GUID 的模型对象。

注：初始注册所有 TADDM 数据可能花费很长时间。

clean [*guid* [*classtype*]]

注销数据库中当前的 TADDM 数据。

如果不指定 GUID，将注销所有 TADDM 数据。 如果指定 GUID，将只注销带有指定 GUID 的模型对象。如果带有指定 GUID 的模型对象在 TADDM 中不再可用，还必须指定模型对象类型。

re-register-all

注销所有 TADDM 数据和启动点，然后重复初始注册。 此选项等同于先使用 clean 选项运行脚本，然后再使用 register 选项运行脚本。

register-menu

仅更新在"上下文菜单服务"数据库中注册的菜单定义。 如果注册了 TADDM 数据，但您希望仅更新菜单定义，请使用此选项。

help

显示脚本的帮助信息。

示例

- 在此示例中，首次运行时使用上下文菜单服务和数据集成服务注册所有 TADDM 数据；而在以后运行时，注册上次运行以来进行的所有更改：

  ```
  ./run_cms_dis_registration.sh
  ```

- 此示例只注册带有指定 GUID 的模型对象。

  ```
  ./run_cms_dis_registration.sh register 3950DF835FA0337A829D864415CC1384
  ```

- 此示例除去所有已注册的 TADDM 数据：

  ```
  ./run_cms_dis_registration.sh clean
  ```

- 此示例除去带有指定 GUID 和模型对象类型的对象：

  ```
  ./run_cms_dis_registration.sh clean 3950DF835FA0337A829D864415CC1384
    LinuxUnitaryComputerSystem
  ```

- 此示例除去所有已注册的 TADDM 数据，然后重复注册过程：

  ```
  ./run_cms_dis_registration.sh re-register-all
  ```

下一步做什么

如果以后要再次运行注册脚本，请先禁用 CMSDISAgent 拓扑构建器代理程序以停止增量更新。 要禁用该代理程序，请编辑 $COLLATION_HOME/etc/collation.properties 文件并设置以下属性：

```
com.ibm.cdb.DisCmsIntegration.enabled=false
```

脚本完成后，随后您必须将该属性设置为 true 以重新启用该代理程序。

配置 CMSDISAgent

CMSDISAgent 作为拓扑构建器代理程序定期运行，并更新上下文菜单服务和数据集成服务数据库中 TADDM 配置项 (CI) 的注册，从而注册任何新的或已修改的 CI 并注销任何已删除的 CI。

关于此任务

如果已启用，那么 CMDDISAgent 会在使用 **run_cms_dis_registration** 脚本完成 TADDM CI 的初始注册后开始运行。可以修改代理程序配置以更改代理程序的运行方式。

过程

- 要启用或禁用 CMSDISAgent，请编辑 $COLLATION_HOME/etc/collation.properties 文件并设置以下属性：

  ```
  com.ibm.cdb.DisCmsIntegration.enabled=value
  ```

  其中 *value* 为 true 或 false。 如果将值设为 true，该代理程序会在完成初始注册后定期运行。 （该属性并不影响 **run_cms_dis_registration** 脚本的操作，该脚本可以随时运行。）
- 要定制数据库中注册的 CI，请修改 $COLLATION_HOME/etc/cmsdis 目录中的以下文件：

  classtype-changehistory.list
  列示满足以下条件的 CI 的模型对象类型：针对这些 CI，TADDM 支持变更历史记录报告在上下文中启动。

  classtype-detailPanel.list
  列示满足以下条件的 CI 的模型对象类型：针对这些 CI，TADDM 支持详细信息面板在上下文中启动。

  可以除去其他产品在上下文中启动 TADDM 时不需要的任何模型对象类型。 请不要向这些文件添加任何类型；TADDM 可能不支持其他类型的上下文中启动。 修改类类型列表文件后，禁用该代理程序，然后再次运行 **run_cms_dis_registration** 脚本，同时指定 re-register-all 选项。

创建发现库存储

发现库存储是数据中心内计算机上的目录或文件夹，它表示用于编写包含资源信息的 XML 文件的所有发现库适配器 (DLA) 的公共位置。 要批量装入到 TADDM 系统的 XML 数据文件放在发现库存储中。 要使用批量装入程序，您必须创建发现库存储。

开始之前

DLA 是从源应用程序（例如 IBM Tivoli Monitoring 或 IBM Tivoli Business Service Manager）抽取数据的软件程序。

每个 DLA 都以称为身份标记语言 (IdML) 的特定 XML 格式编写包含资源信息的 XML 文件。 任意以 IdML 格式撰写的 XML 文件通常都称为簿。 要查看可将其他 Tivoli 产品中的数据装入 TADDM 数据库的 Tivoli 簿集合，请参阅 http://www.ibm.com/software/brandcatalog/ismlibrary/。

DLA 专用于特定产品，因为每个产品都有访问环境中的资源的专用方法。 DLA 的配置和安装对于每个应用程序各不相同。 系统上安装了典型的 DLA，该 DLA 具有特定应用程序的数据的访问权。 例如，计算机上安装了 IBM Tivoli Monitoring 的 DLA，可访问 IBM Tivoli Monitoring 企业管理系统数据库。 所有 DLA 都使用命令行界面运行，并且可以使用环境中任何类型的调度程序（例如 cron）来调度运行。

可创建 DLA 来从环境中的现有产品或数据库抽取信息。

有关如何创建 DLA 的信息、IdML 规范的信息或发现库存储的其他详细信息，请参阅《TADDM 发现库适配器开发者指南》。

关于此任务

通常，发现库存储位于 TADDM 服务器上。 如果未在 TADDM 服务器上设置发现库存储，那么必须确保在 TADDM 服务器上运行的 TADDM 批量装入程序可访问发现库存储。其他应用程序也可在托管发现库存储的计算机上运行。

过程

要创建"发现库"存储，请完成下列步骤：

1. 以特定目录名称（例如：`c:\IBM\DLFS`）创建一个目录，以便在计算机上存储 XML 文件。（可选）您可以在主发现库存储中为要使用的每个 DLA 创建子目录。
2. 使用至少一个用户标识来设置文件传输协议服务器 (FTP)。 该用户标识必须对存储发现库 XML 文件的目录具有以下许可权：写、重命名和读访问。 如果未使用 FTP 来将 XML 文件传输到发现库存储，请确保您使用的工具和用于运行该工具的用户标识对发现库存储目录具有写许可权。
3. 确保各种发现库适配器对托管发现库存储的系统的名称（主机名）具有访问权。 大多数发现库适配器会将 XML 文件复制到发现库存储中。
4. 确保多种"发现库适配器"具有用于连接 FTP 服务器的用户标识和密码。
5. 如果 DLA 未使用 FTP，请将您希望批量装入程序访问的 XML 文件（簿）复制到该共享目录中。该共享目录必须可由批量装入程序进行访问。

   簿编写者和管理员不负责将簿存入发现库存储。 例如，设置 cron 作业来将生成的 IdML 簿通过 FTP 发送到发现库存储中。

下一步做什么

如果要创建发现库存储且希望设置 TADDM 数据库来包含 DLA 簿，那么域服务器上的本地驱动器就可以充当联网的发现库存储。 此目录必须在装入数据的域服务器上的 `$COLLATION_HOME/etc/bulkload.properties` 文件中定义。 如果有多个域服务器，请配置正确的批量装入程序来访问相应的共享目录。 批量装入程序不会从发现库存储中删除 XML 文件。 您必须维护发现库存储中的文件。 确保服务器上有足够的磁盘空间可用于目录中的文件。 如果经常会有新的 XML 文件添加到目录中，那么您必须定期清除该目录。

如果具有同步服务器部署，必须从以下选项中进行选择：

- 如果在簿中引用的资源包含在某一域服务器上定义的作用域定义中，则将该簿装入相应的域服务器中。
- 如果在簿中引用的资源**未**包含在某一域服务器上定义的作用域定义中，则将所有簿装入同步服务器中。

在上下文中启动的配置

要查看环境中组件的更详细信息，您可以从其他 Tivoli 应用程序启动 TADDM 视图。要配置应用程序以在上下文中启动 TADDM 视图，您必须指定 URL。

可以从其他 Tivoli 应用程序启动的视图

您可以从其他 Tivoli 应用程序启动数据管理门户网站视图。 您也可以为指定的配置项 (CI) 启动详细信息和更改历史记录报告。

您可以在数据管理门户网站视图中查看有关以下组件分组的更多信息：

- 业务应用程序
- 业务服务
- 集合

如果 TADDM 服务器和要启动 TADDM 的应用程序都未进行单点登录配置，那么将显示登录窗口。 您必须提供用户名和密码，才能在数据管理门户网站中查看其他信息。

指定用于启动 TADDM 视图的 URL

要从其他 Tivoli 应用程序在上下文中启动 TADDM 视图，您必须指定 URL。

在上下文中启动的 URL 格式为：

Protocol://*TADDMHostname*:*TADDMPort*/*ContextRoot*/?*queryString*

以下列表描述了 URL 格式中每个变量的有效值：

Protocol
要使用的 Web 协议。 有效值为 `http` 或 `https`。

TADDMHostname
要启动到的 TADDM 服务器的主机名。

TADDMPort
要启动到的 TADDM 服务器的端口号。缺省值为 `9430`。

ContextRoot
以下值有效：

cdm/servlet/LICServlet
在 Apache Tomcat 服务器（针对 TADDM 7.3.0）中以及在 WAS Liberty 概要文件服务器（针对 TADDM 7.3.0.1 及更高版本）中部署的 Java servlet 的相对路径。

cdm/queryHomePage.do
这是在使用单点登录功能并指定了搜索文本的情况下，从 IBM Tivoli Monitoring 中启动的"查询主页"的相对路径。

queryString

包含以分隔符定界的名称/值对参数。名称/值对的格式为 name=value。 请使用 = 来分隔名称和值，并使用 & 来分隔名称/值对。

以下列表描述了可在 *queryString* 变量中使用的有效名称/值对：

view

指定要显示更改历史记录。

唯一的有效值为 changehistory。

days_previous

指定要显示特定配置项的更改历史记录的时间段（过去的天数）。

有效值为正整数。

hoursback

指定要显示特定配置项的更改历史记录的时间段（过去的小时数）。

有效值为正整数。

guid

指定配置项的全局唯一标识 (GUID)。

对于域服务器和同步服务器，表 47 列示了 graph 参数的有效值，并根据相应的图形值指示 guid 参数是可选的还是必需的。

如果对 graph 参数指定了下列任何值，那么 guid 参数是可选的：

- businessapplications
- applicationinfrastructure
- physicalinfrastructure

如果对 graph 参数指定了任何其他类型的拓扑图，那么 guid 参数是必需的。

有效值是 GUID 的有效字符串表示，如以下示例中所示：

```
BA2842345F693855A3165A4B5F0D8BDE
```

您应为每个在上下文中启动的 URL 请求仅指定一个 GUID。

graph

指定要启动的拓扑图的类型。

如果您还通过在 guid 参数中提供配置项 GUID 指定了配置项，并且在此 graph 参数中指定的拓扑图中找到所请求的配置项，那么将选中此配置项。

对于域服务器和同步服务器，表 47 列示了 graph 参数的有效值，并根据相应的图形值指示 guid 参数是可选的还是必需的。

表 *47.* 有效图形值及其与 *guid* 参数之间的关系

	有效值	**对于此图形值，guid 参数是可选的还是必需的?**
域服务器	businessapplications	可选
	applicationinfrastructure	可选
	physicalinfrastructure	可选
	对于定制集合对象： • ba_infrastructure	必需

表 47. 有效图形值及其与 *guid* 参数之间的关系 (续)

	有效值	对于此图形值，guid 参数是可选的还是必需的?
同步服务器	businessapplications	可选
	physicalinfrastructure	可选
	对于定制集合对象： • ba_infrastructure	必需

注：不推荐使用先前 TADDM 发行版中用来按 GUID 呈示特定分组实体的其他图形类型。 但是，为了确保与旧版本兼容，如果您使用 GUID 指定了旧图形类型，那么该请求将重定向到新拓扑类型。

username
指定用于登录到 TADDM 的用户名。

password
指定用于登录到 TADDM 的密码。

launchsource
唯一的有效值为 ITM。 它始终与 searchtext=*search_term* 名称/值对一起使用。

搜索范围限定为 ComputerSystem 和 TMSAgent 类型的配置项，列示于 $COLLATION_HOME/etc/cdm/xml/itm_query_components.xml 配置文件中。

从"查询主页"结果中，对于所列的每个配置项，可以启动以下视图：

- "变更历史记录"窗格
- "详细信息"窗格
- 显示了"详细信息"窗格的数据管理门户网站

searchtext
指定搜索词汇。 它始终与 launchsource=ITM 名称/值对一起使用。

如何指定 URL 的示例

以下示例显示如何指定用于启动 TADDM 视图的 URL：

用于启动数据管理门户网站的 URL，无需单独输入授权信息

```
http://home.taddm.com:9430/cdm/servlet/LICServlet?username=administrator
&password=adminpwd&guid=BA2842345F693855A3165A4B5F0D8BDE
```

如果使用的是可信连接，那么您必须仅将凭证用作 URL 的一部分以在上下文中启动，因为用户名和密码都未加密。

这是使用单点登录功能并搜索与搜索文本匹配的配置项时，用于为 IBM Tivoli Monitoring 启动"查询主页"窗口的 URL

```
http://home.taddm.com:9430/cdm/queryHomePage.do?launchsource=itm&searchtext=127.0.0.1
```

这是用于显示 guid 参数所指示定制集合的拓扑的 URL

```
http://home.taddm.com:9430/cdm/servlet/LICServlet?username=administrator
&password=adminpwd&graph=ba_infrastructure&guid=BA2842345F693855A3165A4B5F0D8BDE
```

将更改事件发送到外部系统

您可以将 TADDM 配置为在检测到发现的资源发生更改时通知外部事件处理系统。

要从 TADDM 发送更改事件，您必须已安装以下一个或多个事件处理系统：

- IBM Tivoli Monitoring 6.2.1 修订包 2 或更高版本
- IBM TivoliNetcool/OMNIbus，包括事件集成工具 (EIF) 探针

要查看产品的受支持版本，请转至第 171 页的『受支持的版本』部分。

发现完成时，TADDM 将检查外部事件处理系统跟踪的项的更改。 如果检测到任何更改，将使用 EIF 将这些更改直接发送到 IBM Tivoli Netcool/OMNIbus，并使用 Universal Agent 将其发送到 IBM Tivoli Monitoring。

Universal Agent 将收到的通知转换为异步事件并将数据转发到 IBM Tivoli Monitoring 的 IBM Tivoli Enterprise Monitoring Server 组件。 IBM Tivoli Monitoring Server 将存储事件并通过它们来评估情境。 事件随后将传递到 IBM Tivoli Enterprise Portal 以进行显示。

IBM Tivoli Netcool/OMNIbus 服务器根据自己的内部规则来处理接收到的事件并显示这些事件。

要将更改事件从 TADDM 发送到外部事件处理系统，您必须在 TADDM 中启用更改事件并对每个外部接收方进行适当配置以处理入局事件。

配置 TADDM 发送更改事件

要发送更改事件，您必须使用要向其发送更改事件的事件处理系统的相关信息来配置 TADDM。

关于此任务

根据 TADDM 部署的类型，对以下 TADDM 服务器进行如下更改：

- 在域服务器部署中，对域服务器进行更改。
- 在同步服务器部署中，对同步服务器进行更改。
- 在流式服务器部署中，对主存储服务器进行更改。

过程

要启用更改事件信息的发送，请完成下列步骤：

1. 要启用更改事件，请在 `$COLLATION_HOME/etc/collation.properties` 文件中设置以下属性：`com.ibm.cdb.omp.changeevent.enabled=true`
2. 要配置跟踪哪些资源以发现其更改以及向哪些事件处理系统发送事件，请编辑 `$COLLATION_HOME/etc/EventConfig.xml` 文件。

   有关在 `EventConfig.xml` 文件中指定信息时必须使用的格式的信息，请参阅第 197 页的『TADDM OMP 变更事件模块配置』。

   升级 TADDM 时，将保留 TADDM 前发行版中的 `EventConfig.xml` 文件以确保不会丢失已配置的定制设置。 可在 `$COLLATION_HOME/etc/EventConfigDefault.xml`

文件中获取有关新功能及其使用方法的信息。 EventConfigDefault.xml 文件仅供参考。 如果要使用任何新功能，必须根据 EventConfigDefault.xml 中的相应示例更新 EventConfig.xml。

3. 如果在 EventConfig.xml 文件中指定了 IBMTivoliNetcool/OMNIbus 事件处理系统，请为该系统类型创建相应的 EIF 属性文件。 为此，请完成下列步骤：
   a. 创建 $COLLATION_HOME/etc/omnibus.eif.properties 属性文件。
   b. 定制 omnibus.eif.properties 文件。有关定制 EIF 属性文件的更多信息，请参阅 IBM Tivoli Netcool/OMNIbus 文档中的 *Configuring support for TADDM events in your integrated environment* (http://www-01.ibm.com/support/knowledgecenter/SSSHTQ_7.4.0/com.ibm.netcool_OMNIbus.doc_7.4.0/omnibus/wip/install/task/omn_con_ext_configuringtaddmevents.html?lang=en)。

***TADDM OMP* 变更事件模块配置：**

为了能够发送变更事件，您必须编辑 EventConfig.xml 文件以定义事件侦听器和接收方。

事件侦听器

可以通过提供执行 TADDM 查询所需的条件来定义侦听器。 每次执行发现后，将在此查询所选择的结果对象中查找变更。 可以有多个侦听器。 侦听器和相应的接收方块必须都存在才能传递事件。

请使用以下格式来指定侦听器。

```
<listener object="[OBJECT_TYPE]"
          enabled="true|false">
          sendCauses="true|false"
          sendOriginGuid="true|false">
  <alert recipient="[RECIPIENT_SYSTEM_NAME]"/>
  <attribute name="[ATTRIBUTE_NAME]" operator="[OPERATOR]">
    <value>
    [ATTRIBUTE_VALUE]
    </value>
  </attribute>
  <causeFilter object="[CAUSEFILTER_OBJECT_TYPE]"
               sendOriginGuid="true|false"/>

 </listener>
```

其中：

[OBJECT_TYPE]
是 TADDM 中表示的模型对象类型，例如 ComputerSystem 或 ITSystem。要获取更多示例，请参阅 TADDM 数据字典 (http://*taddmserverhost*:9430/cdm/datadictionary/model-object/index.html)。

enabled
是用于允许发送事件的属性。 要激活侦听器，必须将值设置为 true。

sendCauses
是一个可选属性，用于定义侦听器是否发送关于传播到模型对象的变更的事件。 例如，如果对 Windows 操作系统进行的变更导致对 ComputerSystem 对象进行变更，并且 ComputerSystem 侦听器的 sendCauses 属性设置为 true，

那么该侦听器将向 ComputerSystem 和 Windows 操作系统都发送一个变更事件。 sendCauses 属性的缺省值为 false。

sendOriginGuid

是与 sendCauses 属性配合使用的可选属性。当 sendOriginGuid 属性设置为 true 时，与侦听器匹配的对象将被视为传播到该对象的变更的逻辑源。 发送的关于所传播变更的事件包含源对象的唯一标识。 例如，如果对 ConfigFile 对象进行的变更导致对 ComputerSystem 对象进行变更，并且 ComputerSystem 侦听器的 sendCauses 和 sendOriginGuid 属性都设置为 true，那么关于 ConfigFile 变更的事件不仅包含 ConfigFile 对象的唯一标识，还包含 ComputerSystem 对象的唯一标识。 此功能仅对 Netcool/OMNIbus 事件接收方可用。 sendOriginGuid 属性的缺省值为 false。

[RECIPIENT_SYSTEM_NAME]

是警报接收方。 请参阅第 200 页的『事件接收方』。

[ATTRIBUTE_NAME]

是所查询的 [OBJECT_TYPE] 的属性名称。

[OPERATOR]

是 TADDM MQL 查询的运算符名称。 允许使用下列值。

表 *48. TADDM MQL* 查询的运算符名称。.

运算符	等效的 TADDM MQL
contains-with	contains
ends-with	ends-with
equals	equals
greater-or-equal	>=
greater-than	>
less-or-equal	<=
less-than	<
not-equals	not-equals
starts-with	starts-with

[ATTRIBUTE_VALUE]

是对属性求值时使用的依据值。

<causeFilter>

此属性用于对 sendCauses 属性处于启用状态时传递的原因事件的对象类型进行过滤。 如果指定了此属性，那么将只发送所指定对象类型的原因事件。 但是，仍将发送所传播的事件，例如，属于侦听器中指定的对象类型的事件。 如果未指定 causeFilter 属性，那么侦听器找到的所有原因事件都将发送到接收方。

例如，对 WindowsService 进行变更将导致对 Windows 操作系统进行变更，并因此导致对 ComputerSystem 进行变更。 如果您将 causeFilter 属性设置为 WindowsService，那么将只显示 ComputerSystem 和 WindowsService 变更，而不显示 Windows 操作系统变更。

如果设置了 causeFilter 属性，那么可以选择性地为 sendOriginGuid 属性设置值。缺省情况下，causeFilter 属性将继承作为 causeFilter 属性父代的侦

听器的 sendOriginGuid 设置。在 causeFilter 属性中使用 sendOriginGuid 属性时，只有该 causeFilter 属性的侦听器设置会被覆盖。

如果要对其变更传播到较高级别对象（例如 ComputerSystem）的对象（例如 WindowsService 或 ConfigFile）进行更新，请组合使用 sendCauses 和 causeFilter 属性（而不要使用单独的侦听器）来捕获这种对象。

[CAUSEFILTER_OBJECT_TYPE]

是 CDM 中定义的对象的类名。 您可以使用全名（例如 com.collation.platform.model.topology.sys.windows.WindowsService），也可以使用短名称（例如 WindowsService）。

事件侦听器示例

在以下示例中，对于其 FQDN 包含字符串 "mycompany" 的任何 ComputerSystem，在其中检测到的变更都将发送到接收方 "enterprise-eventhost-itm"。

```
<listener object="ComputerSystem" enabled="true">
  <alert recipient="enterprise-eventhost-itm"/>
  <attribute name="fqdn" operator="contains-with">
    <value>
     mycompany
    </value>
  </attribute>
</listener>
```

在以下示例中，将检测对指定类型的所有对象进行的变更。

```
  <attribute name="guid" operator="not-equals">
    <value>
     0
    </value>
  </attribute>
```

在以下示例中，在类型为 ComputerSystem 的任何对象中检测到的变更都将发送到接收方 "enterprise-eventhost-omnibus"。

```
<listener object="ComputerSystem" enabled="true">
  <alert recipient="enterprise-eventhost-omnibus"/>
  <attribute name="guid" operator="not-equals">
    <value>
     0
    </value>
  </attribute>
</listener>
```

在以下示例中，仅发送 Linux 计算机系统上的 ConfigFile 中的变更所引起的变更。

```
<listener object="ITSystem" enabled="true" sendCauses="true">
  <alert recipient="enterprise-eventhost-itm"/>
  <attribute name="name" operator="ends-with">
  <value>
    ShoppingCart
  </value>
  </attribute>
  <causeFilter object="ConfigFile" />
  <causeFilter object="LinuxUnitaryComputerSystem" />
</listener>
```

事件接收方

事件接收方是 IBM Tivoli Monitoring 或 OMNIBus 的实例，它可以接收来自变更事件模块的事件。 变更侦听器找到变更后，将向相应的接收方发送通知。 您可以同时定义多个具有不同或相同类型的接收方。 侦听器和相应的接收方块必须都存在才能传递事件。

请使用以下格式来指定接收方。

```
<recipient name="[RECIPIENT_NAME]" type="[RECIPIENT_TYPE]">
  <address>[RECIPIENT_FQDN]</address>
  <port>[EVENT_ROUTING_PORT]</port>
  <config>[PATH_TO_EIF_CONFIGURATION]</config>
</recipient>
```

其中：

[RECIPIENT_NAME]
是侦听器中引用的系统的名称。

[RECIPIENT_TYPE]
是用于接收事件的软件的类型。 支持下列类型：

- `itm` - 带有 Universal Agent POST Data Provider 的 IBM Tivoli Monitoring 6。
- `omnibus` - 带有 EIF 适配器的 Netcool/OMNIbus。

[RECIPIENT_FQDN]
（仅限 IBM Tivoli Monitoring）是 Universal Agent 所在主机的标准域名。

[EVENT_ROUTING_PORT]
（仅限 IBM Tivoli Monitoring）是 Universal Agent POST Data Provider 在 `KUMENV` 中指定为 `KUMP_POST_DP_PORT` 的端口。

[PATH_TO_EIF_CONFIGURATION]
（仅限 OMNIBUS）是从属性文件中读取的 EIF 配置路径。 请使用此文件的完整路径。

事件接收方示例

以下示例定义 Netcool/OMNIbus 事件接收方。

```
<recipient name="enterprise-eventhost-omnibus" type="omnibus">
  <config>/opt/IBM/taddm/dist/etc/omnibus.eif.properties</config>
</recipient>
```

以下示例定义 IBM Tivoli Monitoring 事件接收方。

```
<recipient name="enterprise-eventhost-itm" type="itm">
  <address>itm-ua.mycompany.com</address>
  <port>7575</port>
</recipient>
```

配置 IBMTivoliNetcool/OMNIbus

可以配置 IBMTivoliNetcool/OMNIbus V7.3 或更高版本以接收 TADDM 发送的更改事件。 您可以对先前版本 Tivoli Netcool/OMNIbus 中显示的事件数据进行聚集和定制，并可以定义事件处理逻辑。

开始之前

要将 IBM Tivoli Netcool/OMNIbus V7.3 或更高版本配置为接收由 TADDM 发出的更改事件，请参阅 IBM Tivoli Netcool/OMNIbus 文档（位于 http://www-01.ibm.com/support/knowledgecenter/SSSHTQ/landingpage/NetcoolOMNIbus.html?lang=en）中的启用 *TADDM* 事件支持主题。 Tivoli Netcool/OMNIbus 文档还包含有关 `tivoli_eif_taddm.rules` 文件的信息。该文件包含的逻辑可用于处理在 TADDM 发现期间检测到的配置更改的详细信息。

在使用高可用性计算或故障转移计算的环境中，可配置 TADDM 支持自动故障转移。此支持在 TADDM 事件发送到 Tivoli Netcool/OMNIbus 时发生。您可以在 EIF 属性文件中指定主要和次要 EIF 探针地址及其关联端口。 以下示例显示在何处添加这些属性：

```
# Hostname where the NetCool/OMNIbus EIF probe resides.  Specify up to 8 locations.
# Each location should be separated by a comma.
# The event is sent to the first available probe in the list.
# Example:
#     ServerLocation=netcool.mycompany.com,netcool2.mycompany.com
ServerLocation=netcool.mycompany.com,netcool2.mycompany.com

# Port the NetCool/OMNIbus EIF probe is listening on.
# There must be a port entry for each probe specified under ServerLocation.
# Example:
#     ServerPort=9998,9998
ServerPort=9998,9998
```

输入的每个探针地址都必须有在 *ServerPort* 属性中指定的关联端口。 如果没有为每个探针地址指定端口，在发送事件时会出错。 如果无法将事件发送到主端口，则会将其发送到列表上第一个可用的端口。 在 *ServerLocation* 属性中最多可指定八个探针地址。

关于此任务

IBMTivoliNetcool/OMNIbus V7.3 之前的版本中，缺省行为是将事件模块中的所有事件组合为一个事件，并将计数属性设置为显示该组合事件中包含的事件数。 下列步骤描述如何更改该缺省行为。

过程

1. 在 TADDM 服务器上，打开以下文件以进行编辑：`$COLLATION_HOME/etc/omnibus.eif.properties`
2. 设置以下 TADDMEvent_Slot 属性的属性值：

```
TADDMEvent_Slot_object_name=$TADDM_OBJECT_NAME
TADDMEvent_Slot_change_type=$TADDM_CHANGE_TYPE
TADDMEvent_Slot_change_time=$TADDM_CHANGE_TIME
TADDMEvent_Slot_class_name=$TADDM_CLASS_NAME
TADDMEvent_Slot_attribute_name=$TADDM_ATTRIBUTE_NAME
TADDMEvent_Slot_old_value=$TADDM_OLD_VALUE
TADDMEvent_Slot_new_value=$TADDM_NEW_VALUE
TADDMEvent_Slot_host=$TADDM_HOST
TADDMEvent_Slot_port=$TADDM_PORT
TADDMEvent_Slot_guid=$TADDM_GUID
TADDMEvent_Slot_origin=$TADDM_ORIGIN
```

下一步做什么

如果配置 IBM Tivoli Netcool/OMNIbus 时遇到任何问题，请参阅《故障诊断指南》中的『将 TADDM 与其他产品集成的问题』主题。

配置 IBM Tivoli Monitoring 数据提供程序

您可以配置 Universal Agent 初始化文件来定义新的数据提供程序。

开始之前

如果您使用的是 Tivoli Monitoring V6.2.2 或更低版本，请确保 KUMPOST 配置文件中不存在跳格或空格字符。

过程

要配置 IBM Tivoli Monitoring 数据提供程序，请完成下列步骤：

如果您要在 Windows 系统上运行 Universal Agent，请完成下列步骤：

1. 在安装了 Universal Agent 的 Windows 系统上，单击**开始** > **IBM Tivoli Monitoring** > **管理 Tivoli Monitoring Services**。
2. 右键单击 Universal Agent 并单击**重新配置**。
3. 在两个"代理程序高级配置"窗口中，分别单击**确定**。
4. 要更新 Universal Agent 初始化文件，请单击**是**。 将在系统文本编辑器中打开 KUMENV 文件。
5. 将 KUMA_STARTUP_DP 值设置为 POST：

   ```
   KUMA_STARTUP_DP=POST
   ```

   注： 如果 Universal Agent 已配置为使用另一个数据提供程序，请指定以逗号分隔的两个值，如下例所示：

   ```
   KUMA_STARTUP_DP=ASFS,POST
   ```

6. 将必需的 POST 参数信息添加到 KUMENV 文件：

   ```
   *----------------------------------------*
   * TADDM POST DP Parameters *
   *----------------------------------------*
   KUMP_POST_DP_PORT=7575
   KUMP_POST_GROUP_NAME=TADDM
   KUMP_POST_APPL_TTL=14400
   ```

7. 保存并关闭 KUMENV 文件。
8. 要配置代理程序，请单击**是**。
9. 在"管理 Tivoli Enterprise Monitoring Services"窗口中，单击 **Universal Agent** > **启动**。
10. 在系统文本编辑器中，创建文本文件。 在文件中输入以下信息：

    ```
    //APPl CONFIGCHANGE
    //NAME dpPost E 3600
    //ATTRIBUTES ';'
    Post_Time T 16 Caption{Time}
    Post_Origin D 32 Caption{Origination}
    Post_Ack_Stamp D 28 Caption{Event time stamp}
    Comp_Type D 512 Caption{Component type}
    Comp_Name D 512 Caption{Component name}
    Comp_Guid D 512 Caption{Component GUID}
    Change_Type D 512 Caption{Change type}
    Chg_Det_Time D 512 Caption{Change detection time}
    Chg_Attr D 512 Caption{Changed attribute}
    Srv_Addr D 512 Caption{TADDM server}
    Srv_Port D 16 Caption{TADDM port}
    ```

11. 将文件另存为 %ITM_HOME%\TMAITM6\metafiles\KUMPOST。

注：确保文件名 KUMPOST 使用大写字母拼写，如此处所示。

12. 打开 Windows 命令提示符并浏览到 %ITM_HOME%\TMAITM6 文件夹。
13. 运行 KUMPCON.exe 程序以验证并导入 KUMPOST 元文件。
14. 在"管理 Tivoli Monitoring Services"窗口中，右键单击 Universal Agent 并选择**重新启动**。

如果您要在 Linux 或 UNIX 系统上运行 Universal Agent，请完成下列步骤：

1. 使用以下命令重新配置 Universal Agent：

   ```
   itmcmd config -A um
   ```

   提示您输入数据提供程序时，请输入 POST。

   注： 如果 Universal Agent 已配置为使用另一个数据提供程序，请指定以逗号分隔的两个值（例如，ASFS,POST）。

2. 在 $ITM_HOME/config 目录中，创建 um.ini 文件的备份副本，然后将以下条目添加到文件的原始副本中：

   ```
   # TADDM POST DP Parameters
   KUMP_POST_DP_PORT=7575
   KUMP_POST_GROUP_NAME=TADDM
   KUMP_POST_APPL_TTL=14400
   ```

3. 在 $ITM_HOME/*interp*/um/metafiles 目录中，创建文本文件。 在文件中输入以下信息：

   ```
   //APPl CONFIGCHANGE
   //NAME dpPost E 3600
   //ATTRIBUTES ';'
   Post_Time T 16 Caption{Time}
   Post_Origin D 32 Caption{Origination}
   Post_Ack_Stamp D 28 Caption{Event time stamp}
   Comp_Type D 512 Caption{Component type}
   Comp_Name D 512 Caption{Component name}
   Comp_Guid D 512 Caption{Component GUID}
   Change_Type D 512 Caption{Change type}
   Chg_Det_Time D 512 Caption{Change detection time}
   Chg_Attr D 512 Caption{Changed attribute}
   Srv_Addr D 512 Caption{TADDM server}
   Srv_Port D 16 Caption{TADDM port}
   ```

4. 将文件另存为 KUMPOST。

   注：确保文件名 KUMPOST 使用大写字母拼写，如此处所示。

5. 使用以下命令重新启动 Universal Agent：

   ```
   itmcmd agent stop um

   itmcmd agent start um
   ```

6. 要验证和刷新 KUMPOST 元文件，请完成下列步骤：
   a. 运行具有以下参数的 $ITM_HOME/bin/um_console 命令：

      ```
      um_console -h <ITM directory>
      ```

   b. 在命令行，键入以下文本：

      ```
      validate KUMPOST
      ```

      会显示类似以下内容的消息：

      ```
      KUMPS001I Console input accepted.
      KUMPV025I Processing input metafile /opt/IBM/ITM//lx8266/um/metafiles/KUMPOST
      KUMPV026I Processing record 0001 -> //APPl CONFIGCHANGE
      ```

```
KUMPV148I Note: APPL names starting with letters A-M are designated for
Best Practices and Business Partner UA solutions.
KUMPV026I Processing record 0002 ->     //NAME dpPost E 3600
KUMPV026I Processing record 0003 ->     //ATTRIBUTES ';'
KUMPV026I Processing record 0004 -> Post_Time T 16 Caption{Time}
KUMPV026I Processing record 0005 -> Post_Origin D 32 Caption{Origination}
KUMPV026I Processing record 0006 -> Post_Ack_Stamp D 28 Caption{Event time stamp}
KUMPV026I Processing record 0007 -> Comp_Type D 512 Caption{Component type}
KUMPV026I Processing record 0008 -> Comp_Name D 512 Caption{Component name}
KUMPV026I Processing record 0009 -> Comp_Guid D 512 Caption{Component GUID}
KUMPV026I Processing record 0010 -> Change_Type D 512 Caption{Change type}
KUMPV026I Processing record 0011 -> Chg_Det_Time D 512 Caption{Change detection time}
KUMPV026I Processing record 0012 -> Chg_Attr D 512 Caption{Changed attribute}
KUMPV026I Processing record 0013 -> Srv_Addr D 512 Caption{TADDM server}
KUMPV026I Processing record 0014 -> Srv_Port D 16 Caption{TADDM port}
KUMPV000I Validation completed successfully
KUMPV090I Application metafile validation report saved in file
/opt/IBM/ITM//lx8266/um/metafiles/KUMPOST.rpt.
```

c. 当系统提示您指定要对元文件执行的操作时，键入以下文本：

```
Refresh
```

d. 输入 `Yes` 以确认。

下一步做什么

要验证 Universal Agent 配置是否已成功，请检查 Tivoli Enterprise Portal 中的更改事件报告。

要使用 IBM Tivoli Monitoring 6.2.1 或更高版本来打开更改事件报告，请完成下列步骤：

1. 浏览到已配置为发送和接收 TADDM 的事件通知的 Universal Agent。
2. 展开 CONFIGCHANGE 节点。
3. 单击 **DPPOST** 节点。

在 IBM Tivoli Monitoring 中创建配置更改情境

您可以在 Tivoli Enterprise Portal 中使用"情境"功能来监视更改事件，并触发基于更改事件中的信息的情境。

过程

要在 IBM Tivoli Monitoring 中创建配置更改情境，请完成下列步骤：

要在使用 IBM Tivoli Monitoring 6.2.1 的情况下创建配置更改情境，请完成下列步骤：

1. 在 IBM Tivoli Enterprise Portal 的"导航器"窗格中，浏览到已配置为发送和接收 TADDM 的事件通知的 Universal Agent。
2. 展开 CONFIGCHANGE 节点。
3. 右键单击 DPPOST 节点。 单击**情境**。
4. 在"*node_name* 的情境"窗口中，右键单击 **Universal Data Provider**。 单击**新建**。 此时将显示"创建情境或规则"窗口。
5. 在**名称**字段中，输入情境的名称。 例如，`ConfigurationChanged`。
6. 在**描述**字段中，输入情境的描述。 例如，`TADDM 检测到已跟踪对象的更改`。
7. 从**受监视的应用程序**列表中，选择 **Universal Data Provider**。
8. 确保**跨受管系统关联情境**复选框已清除。
9. 单击**确定**。 此时将显示"选择条件"窗口。

10. 从**属性组**列表，选择 **DPPOST**。
11. 从**属性项**列表，选择**组件名称**。
12. 单击**确定**。 将显示情境的**公式**选项卡。
13. 配置情境，以便在组件名与要监视的环境中资源的名称相匹配时触发该情境。
14. 单击**确定**。

要在使用 IBM Tivoli Monitoring 6.2.2 或更高版本的情况下创建配置更改情境，请完成下列步骤：

1. 在 IBM Tivoli Enterprise Portal 的"导航器"窗格中，浏览到已配置为发送和接收 TADDM 的事件通知的 Universal Agent。
2. 展开 CONFIGCHANGE 节点。
3. 右键单击 DPPOST 节点。 单击**情境**。
4. 在"*node_name* 的情境"窗口中，单击**创建新情境**。 此时将显示创建情境窗口。
5. 在**名称**字段中，输入情境的名称。 例如，`ConfigurationChanged`。
6. 在**描述**字段中，输入情境的描述。 例如，`TADDM 检测到已跟踪对象的更改`。
7. 从**受监视的应用程序**列表中，选择 **Universal Data Provider**。
8. 单击**确定**。 此时将显示"选择条件"窗口。
9. 从**属性组**列表，选择 **DPPOST**。
10. 从**属性项**列表，选择**组件名称**。
11. 单击**确定**。 将显示情境的**公式**选项卡。
12. 配置情境，以便在组件名与要监视的环境中资源的名称相匹配时触发该情境。
13. 单击**确定**。
14. 在 IBM Tivoli Enterprise Portal 的"导航器"窗格中，右键单击包含更改事件报告的节点。 单击**情境**。
15. 在"*node_name* 的情境"窗口中，右键单击所创建的 **ConfigurationChanged** 情境，然后单击**启动情境**。

结果

收到配置更改事件时，将检查它们的组件名。 如果组件名与您在情境公式中指定的组件名相匹配，那么会触发配置的情境。

在 IBM Tivoli Monitoring 中的配置更改事件报告中创建详细信息链接

您可以在报告表中创建到工作空间的链接来直接显示 TADDM 服务器中的更改历史记录和详细信息。 这些链接提供的信息比在报告中显示的信息更加详细。

过程

要在配置更改事件报告中创建指向更详细的更改事件信息的链接，请完成下列步骤：

1. 要创建工作空间以显示信息，请完成下列步骤：
   a. 在"导航器"窗格中，右键单击要包含工作空间的节点。 单击**文件** > **将工作空间另存为**。 此时将显示"将工作空间另存为"窗口。
   b. 在**名称**字段中，输入工作空间的名称。 例如，`ConfigChangeDetails`。
   c. 在**描述**字段中，输入工作空间的描述。 例如，更改事件表的一般工作空间。

d. 选中**仅可选作工作空间链接的目标**复选框。

e. 单击**确定**。

2. 要使用 IBM Tivoli Monitoring 6.2.1 或更高版本来配置工作空间，请完成下列步骤：

   a. 将工作空间配置为具有一个导航器窗格和两个浏览器窗格。

   b. 单击**编辑** > **属性**。

   c. 在"浏览器"窗格中，选择**入门**的第一个实例。

   d. 在"样式"窗格中，选择**使用提供的位置**。

   e. 单击**确定**。

   f. 在其中一个浏览器窗格的**位置**字段中，输入 TADDM 中"更改历史记录"视图的 URL。在一行上输入 URL 之后，不要按 **Enter** 键。

   ```
   http://$taddm_server$:$taddm_port$/cdm/servlet/
   LICServlet?view=changehistory&hoursback=10000&console=web&guid=$taddm_guid$
   ```

   hoursback 参数指定显示更改事件的小时数。 例如，将 hoursback 设置为 6 会显示过去六个小时内的所有更改事件。

   g. 在"浏览器"窗格中，选择**入门**的第二个实例。

   h. 在"样式"窗格中，选择**使用提供的位置**。

   i. 单击**确定**。

   j. 在另一个浏览器窗格的**位置**字段中，输入 TADDM 中"对象详细信息"视图的 URL。 在一行上输入 URL 之后，不要按 **Enter** 键。

   ```
   http://$taddm_server$:$taddm_port$/cdm/servlet/LICServlet?console=web
   &guid=$taddm_guid$
   ```

   k. 要保存新的工作空间，请单击**文件** > **保存**。

   将 URL 输入到**位置**字段中后，不要马上按 **Enter** 键，而要保存工作空间。

3. 打开 IBM Tivoli Enterprise Portal。 在"报告"窗格中，右键单击**报告**表中的某行。
4. 单击**链接至** > **链接向导**。 此时将显示"工作空间链接向导"的"欢迎"页面。
5. 单击**创建新链接**。 单击**下一步**。 此时将显示"工作空间链接向导"的"链接名称"页面。
6. 在**名称**字段中，输入链接的名称。 例如，显示详细信息。
7. 在**描述**字段中，输入链接的描述。 例如，链接至详细信息。
8. 单击**下一步**。 此时将显示"工作空间链接向导"的"链接类型"页面。
9. 单击**绝对**。 单击**下一步**。 此时将显示"工作空间链接向导"的"目标工作空间"页面。
10. 在"导航器"面板中，选择包含您已创建的工作空间的节点。 在"工作空间"面板中，选择您创建的工作空间。
11. 单击**下一步**。 此时将显示"工作空间链接向导"的"参数"页面。
12. 您必须添加三个符号："taddm_server"、"taddm_port"和"taddm_guid"。 要添加符号，请完成下列步骤：

    a. 单击**添加符号**。 此时将显示"添加符号"窗口。

    b. 在**符号**字段中，输入符号的名称。

    c. 单击**确定**。

13. 对于创建的每个符号，您必须将其链接到表示报告中正确列的属性。

- 将"taddm_server"符号链接至 TADDM 服务器属性。
- 将"taddm_port"符号链接至 TADDM Web 控制台端口号。
- 将"taddm_guid"符号链接至组件 GUID 属性。

要将符号链接至属性，请完成下列步骤：

a. 在"工作空间链接向导"的"参数"页面中，选择要链接至报告列的符号。
b. 单击**修改表达式**。 此时将显示"表达式编辑器"窗口。
c. 单击**符号**。 此时将显示"符号"窗口。
d. 浏览到**属性**，并选择要链接到符号的属性。 单击**确定**。
e. 在"表达式编辑器"窗口中，单击**确定**。 此时将显示"工作空间链接向导"的"参数"页面。

14. 单击**下一步**。 此时将显示"工作空间链接向导"的"摘要"页面。
15. 单击**完成**。

结果

如果您在更改事件报告中具有活动事件，那么在每个表行旁边将显示一个链接图标。 要移至目标工作空间，请单击链接图标并选择**显示详细信息**。在表行中，值将替换符号。 在工作空间中，将在上下文中启动"更改历史记录"和"对象详细信息"面板。

配置业务系统的更改事件

只要业务系统发生更改，您就可以使用更改事件功能来发送更改事件。

关于此任务

缺省情况下，TADDM 不会在其依赖的某台计算机发生更改时指出业务系统已更改。

过程

要为业务系统启用更改事件的发送，请完成下列步骤：

1. 在适当的编辑器中打开 `$COLLATION_HOME/etc/propagationserver.xml`。
2. 在"计算机系统"部分中，对于应用程序和业务系统关系元素，将 `enabled` 属性的值设置为 `true`。 例如：

```
<relationship enabled="true" source="sys.ComputerSystem" attribute="groups"
target="app.Application" targetAttribute="true"
collectionType="app.FunctionalGroup" radius="1"/>

<relationship enabled="true" source="sys.ComputerSystem" attribute="components"
target="sys.BusinessSystem" targetAttribute="true"/>
```

3. 重新启动 TADDM。
4. 在更改事件配置 `$COLLATION_HOME/etc/EventConfig.xml` 中为业务系统创建侦听器。在以下示例中，事件接收方为 `mycompany-itm`，业务系统名称为 `MyBiz`。

```
<listener object="ITSystem" enabled="true">
 <alert recipient="mycompany-itm"/>
 <attribute name="name" operator="equals">
  <value>MyBiz</value>
 </attribute>
</listener>
```

使用 IBM Tivoli Workload Scheduler 调度作业

您可以使用 IBM Tivoli Workload Scheduler 来调度 TADDM 中的作业。 IBM Tivoli Workload Scheduler 是一款软件自动化工具，用于提供自动化工作负载管理及监视主干。

请使用 IBM Tivoli Workload Scheduler 8.5.1 或更高版本。 您必须在 TADDM 服务器上安装主域管理器和容错代理程序。 有关如何安装和配置 Tivoli Workload Scheduler 的信息，请参阅http://www-01.ibm.com/support/knowledgecenter/SSGSPN_8.5.1.1/com.ibm.tivoli.itws.doc_8.5.1.1/ic-homepage.html?lang=en。 调度对象可以通过 composer 命令行程序进行管理，并存储在 Tivoli Workload Scheduler 中。

Tivoli Workload Scheduler 作业使用 `invokejob.sh` 脚本来运行所需的操作。 `invokejob.sh` 脚本由 TADDM 安装提供。

以下参数是该脚本的所有用法中的公共参数：

必需：`-u` *user*
该值指定运行 API 命令的用户。

必需：`-p` *password*
该值指定对用户进行认证的密码。

必需：`--profile` *profile*
该值定义发现概要文件。

可选：`-H` *host*
该值指定 TADDM 服务器主机名。 缺省名称为 `localhost`。如果使用 **`-T`** 参数，那么还必须指定 **`-H`** 参数。

可选：`-P` *port*
该值指定 TADDM 服务器端口。 缺省值为 `9433`。

可选：`-v` *version*
该值指定版本名称或版本号。 缺省值为 `0`。

可选：`-t` *timeout*
该值指定作业自动中断前的时间长度。

可选： `-T | --truststorefile` *truststore*
此值指定信任库文件 `jssecacerts.cert` 的位置以及用于 TADDM 服务器连接的证书。 此参数是到 TADDM 的安全连接所必需的。 如果使用此参数，那么还必须指定 **`-H`** 参数。

要调度作业，请完成以下步骤：

1. 根据 Tivoli Workload Scheduler，在编辑文件中输入 TADDM 作业的定义文件。以下示例显示了模板作业定义：

```
WORKSTATION_ID#TADDM_JOB
 SCRIPTNAME "./opt/IBM/taddm/dist/bin/invokejob.sh -u
^TADDM_USERNAME^ -p ^TADDM_PASSWORD^ command [parameters]"
 STREAMLOGON taddmuser
 TASKTYPE UNIX
 RECOVERY STOP
```

^*TADDM_USERNAME*^ 和 ^*TADDM_PASSWORD*^ 是必须在 Tivoli Workload Scheduler 中定义的变量。 这些变量映射到存储在数据库中的值。 出于安全性考虑，请使用变量（尤其是对密码编码时）来确保其值并不作为开放式文本而可视。

2. 使用组合器将该编辑文件添加到数据库中。
3. 将作业添加到作业流中，并调度该作业流运行。 IBM Tivoli Workload Scheduler 代理程序将启动并监视 `invokejob.sh` 脚本的操作。

调度发现作业

以下示例在作用域 127.0.0.1 上运行发现：

```
dist/bin/invokejob.sh -u USER -p PASSWORD --timeout 60000 discover start
--profile  "Level 3 Discovery" 127.0.0.1
```

以下示例在 MyScopeSet 作用域集上运行发现，该作用域集必须已经存在于作用域列表中：

```
dist/bin/invokejob.sh -u USER -p PASSWORD --timeout 60000 discover start
--profile "Level 3 Discovery" MyScopeSet
```

在之前的示例中，最后一个参数指定在发现运行时要包含的作用域元素或作用域集。 **profile** 参数是必需参数。 **name** 参数是可选参数，表示所运行发现的名称。

以下命令是演示如何停止当前正在运行的发现的示例：

```
dist/bin/invokejob.sh -u USER -p PASSWORD --timeout 60000 discover stop
```

discover stop 命令不带其他任何参数。

调度域同步作业

以下示例显示了 TADDM 脚本 `invokejob.sh` 的命令行语法和选项，用于在同步服务器部署中运行域同步：

```
dist/bin/invokejob.sh -u USER -p PASSWORD --timeout 60000 sync start TestDomain
```

sync start 和 **sync stop** 这两个命令都需要一个参数，用于指示要启动或停止同步作业的域的名称。

将 TADDM 与 IBM Tivoli Business Service Manager 集成

根据 IT 环境中必须执行的特定任务，您可以使用在 TADDM 与 IBM Tivoli Business Service Manager 之间可用的集成功能。要使用这些功能，您必须具备 IBM Tivoli Business Service Manager 4.2.1 IF3，但不需要其他 TADDM 配置。

更新业务应用程序的生命周期状态

可以使用生命周期状态来过滤对象，以将其从 TADDM 同步到 IBM Tivoli Business Service Manager。可以使用 **BusinessServiceLifecycle** 程序来列出有关业务应用程序的信息，或设置业务应用程序的生命周期状态。

IBM Tivoli Business Service Manager ITsystems 应用程序仅包括业务应用程序。 因此，**BusinessServiceLifecycle** 程序仅支持业务应用程序。

BusinessServiceLifecycle 程序位于以下位置：

- 对于 Linux 和 UNIX 操作系统，BusinessServiceLifecycle 脚本位于 $COLLATION_HOME/bin 目录下。
- 对于 Windows 操作系统，BusinessServiceLifecycle.bat 批处理文件位于 %COLLATION_HOME%\bin 文件夹中。

使用带以下命令行选项的 **BusinessServiceLifecycle** 程序：

```
BusinessServiceLifecycle -u TADDM_username -p TADDM_password -l | -s guid state
```

使用 -l 选项列出业务应用程序的生命周期信息，或者使用 -s 选项以及 GUID 参数和状态码参数来设置生命周期状态。 不能同时使用 -l 选项和 -s 选项。

下表列出了有效的状态码：

表 *49.* 状态码

代码	状态
0	未知
1	其他
2	已排序
3	已接收
4	测试中
5	已测试
6	已安装
7	已启用
8	已禁用
9	维护
10	已引退
11	已归档
12	已接受
13	构建
14	开发
15	草稿
16	库存
17	脱机
18	生产后
19	生产
20	生产就绪
21	弃用
22	验证

将 TADDM 与 Jazz for Service Management 集成

TADDM 支持与 Open Services for Lifecycle Collaboration (OSLC) 平台集成。 当 OSLC 与 TADDM 配合使用时，您可以获取以标准资源定义的形式显示的发现数据。 Jazz for Service Management 平台是一个基于 OSLC 开放式社区规范的 IBM 集成工具。

Jazz for Service Management 为所有 Tivoli 产品（但不限于）提供单点配置和管理。Jazz for Service Management 显示了 IT 资源、应用程序和业务关系的端对端视图。

TADDM OSLC REST 通信

TADDM 表述性状态转移 (REST) 服务在若干 OSLC REST 订阅源中提供 OSLC 集成。 该服务指定了运行时所返回的媒体类型，并描述了与该服务相连接的安全性问题。

Common Resource Type Vocabulary (CRTV) 是一种由 IBM 和 OSLC 社区所定义的受 TADDM 支持的数据模型，可以与 Tivoli Common Data Model (CDM) 相结合。 TADDM 对 OSLC 的支持使 CDM 发现数据能够以 CRTV 定义的资源形式提供。

OSLC REST 接口：

在 TADDM 中，为 Open Services Lifecycle Collaboration (OSLC) 提供了 REST 接口。 您可以使用 OSLC REST 接口来获取有关注册配置项 (CI)、其属性及变更历史记录的信息。

仅当 Common Resource Type Vocabulary (CRTV) 或 TADDM 词汇表支持 CI 属性时才能获取有关这些属性的信息。

每个有效的请求必须提供用于识别具体 CI 的 GUID。

有两种服务类型：

配置服务
该服务提供了一个接口，可检索 CRTV 资源的扩展属性。

变更历史记录服务
该服务提供了一个接口，可检索指定时间段内 CRTV 资源的变更历史记录。

对于每个服务，您都可以查看以下三类内容：

- RDF 表示
- OSLC 紧凑视图
- HTML 预览

以下 URL 为基地址：

`http[s]://`*`taddm_host`*`:`*`port`*`/cdm/oslc/`*`provider_name`*`/`*`ci_guid`*

其中

- *port* 是 Tomcat 服务器 (TADDM 7.3.0) 或 WAS Liberty 概要文件服务器（TADDM 7.3.0.1 及更高版本）进行侦听所在的端口。 缺省值为 9430。
- *provider_name* 是以下两个值中的任何一个（取决于要使用的服务）：
  - `configuration`
  - `changehistory`
- *ci_guid* 是 TADDM 中 CI 的标识

要查看 CI 的 HTML 预览，请使用以下 URL：

- `http[s]://`*`taddm_host`*`:`*`port`*`/cdm/oslc/`*`provider_name`*`/`*`ci_guid`*`/preview`

OSLC REST 接口仅接受 HTTP-GET 请求。您可以使用 HTTP 接受头来指定返回的内容类型。

要查看给定 CI 的 OSLC 紧凑视图，请指定以下接受头：

```
application/x-oslc-compact+xml
```

查看给定 CI 的 RDF 表示，指定以下接受头：

```
application/rdf+xml
```

如果未向接受头提供任何值，那么这就是缺省行为。

***OSLC* 紧凑视图：**

OSLC 紧凑视图是用 XML 表示的目标资源。

OSLC 紧凑视图是 OSLC REST 接口所提供的预览。 要查看目标资源的预览，提供程序必须提供如 OSLC 规范中所定义的资源表示。

通过向具有 `application/x-oslc-compact+xml` 访问头的目标资源的 URI 发出 HTTP GET 请求，您可以获取该资源表示。

如果提供程序支持预览机制，那么会以紧凑表示作出响应。紧凑表示包含了可供使用者显示链接和目标资源预览的信息。

***Jazz for Service Management HTML* 预览：**

Jazz for Service Management Registry Services 提供了 HTML 用户接口，可传递来自连接的外部系统的注册元素的信息。

具有 TADDM 所提供的数据的所有元素都有 HTML 预览功能，该预览会直接从 TADDM 服务器为所选元素数据提供快速概述。

TADDM 为 Jazz for Service Management 提供了订阅源服务，地址如下所示：

```
http[s]://host_name:port/cdm/oslc/configuration/guid/preview
```

其中 *host_name* 和 *port* 是 TADDM 服务器的主机名和端口号，*guid* 是唯一的元素标识。

URL 显示了含有所选元素概述信息的页面。 该页面会自动显示在 Jazz for Service Management 用户界面中。

该页面内容与 TADDM 数据管理门户网站中可用的"库存摘要详细信息"视图中的"常规"选项卡相似。

安全性：

可以配置 TADDM，从而在访问 OSLC REST 接口所提供的订阅源时需要进行认证。

要访问 REST 接口，必须使用以下一个方法进行认证：

基本 HTTP 认证
必须将凭证放在认证请求头中。该头的值必须符合基本 HTTP 认证规则。

单点登录

在使用单点登录时，所有提交给 REST 接口的请求必须持有轻量级第三方认证 (LTPA) 令牌。 要验证令牌，TADDM 必须配置为将 WebSphere Virtual Member Manager (VMM) 用作用户资源库。

有关配置 VMM 的更多信息，请参阅第 23 页的『将 TADDM 服务器配置为使用 WebSphere 联合存储库』。

要提供无需认证就可显示的请求订阅源，必须在 collation.properties 文件中使用有效的 Registry Services URL 配置以下属性：

```
com.ibm.cdb.topobuilder.integration.oslc.frsurl
```

然后，如果未随请求一起提供有效的凭证，那么将使用预配置的用户名和密码。

用户名和密码取自"Common Data Model"Web 应用程序的 web.xml 部署描述符文件。您可以使用以下 OSLCFilter init 参数来配置该定制：

OSLC_LOGIN_OFF

如果将该参数设置为 true，当入站请求未包含其自身有效的凭证时，会使用 OSLC_USER 和 OSLC_PASSWORD 所指定的用户名和密码。

如果将该参数设置为 false，那么入站请求必须包含有效凭证。
缺省值为 true。

OSLC_USER

当请求中未包含有效凭证时，将该参数设置为所用的用户名。 如果需要，可以更改所用的用户名。

缺省值为 administrator。

OSLC_PASSWORD

当请求中未包含有效凭证时，将该参数设置为所用的密码。 如果通过 TADDM UI 更改管理员的密码，必须用该参数更新密码值集。

缺省值为 collation。

使用 OSLCAgent 将数据导出至 Registry Services

您可以使用 OSLCAgent 拓扑代理程序将配置项 (CI) 信息导出至 Registry Services。

OSLCAgent 是一个将数据从 TADDM 导出至 Registry Services 的自动化解决方案。代理程序会定期执行以下任务：

- 查询可在 Registry Services 中注册的对象
- 将这些对象转换为 RDF 格式的消息
- 使用 HTTP 发布这些消息

OSLCAgent 从属于"集成"组。 在 collation.properties 文件的下列条目中指定运行的时间间隔：

```
com.ibm.cdb.topobuilder.groupinterval.integration
```

OSLCAgent 可以充当配置提供程序和变更历史记录提供程序。 可以分别启用这两个角色。 要启用配置提供程序角色，请将以下属性设置为 true：

```
com.ibm.cdb.topobuilder.integration.oslc.enable.configurationsp
```

要启用变更历史记录提供程序角色，请将以下属性设置为 `true`：

`com.ibm.cdb.topobuilder.integration.oslc.enable.changehistorysp`

要将 OSLCAgent 配置为连接到 Registry Services，必须指定 Registry Services 地址并访问条目详细信息。

在以下属性中配置 Registry Services 地址：

`com.ibm.cdb.topobuilder.integration.oslc.frsurl`

以下列格式指定 Registry Services 地址：

protocol://*fqdn_or_ip_or_hostname*:*port*

例如，`http://192.0.2.24:9081`

注：为了与其他产品保持一致并避免出现集成问题，最好使用标准域名 (FQDN) 或标准主机名，而不要使用 IP 地址。 但是，如果与 TADDM 配合使用的所有其他产品都使用 IP 地址，那么您必须指定 IP 地址。 如果没有任何产品与 TADDM 配合使用，那么仍最好使用 FQDN，以备将来添加其他产品。

创建类型为**集成/注册表服务**的访问列表条目。 指定 Registry Services 的用户名和密码。

通过使用以下属性，您可以对 OSLCAgent 的运行方式进行调整：

com.ibm.cdb.topobuilder.integration.oslc.maxtimeperrun
: 该属性指定了允许 OSLCAgent 运行的最长时间（以分钟计）。 对于每个提供程序，在超时事件之前提交到池中的作业所耗用的时间长度可以超过此时间。 如果未配置该属性，或将该属性设置为 `-1`，那么允许 OSLCAgent 单次运行的时间将没有限制。

com.ibm.cdb.topobuilder.integration.oslc.jobspoolsize
: 该属性指定了允许同时运行的作业最大数。 每个作业注册一个 CI。 如果未配置该属性，那么缺省值为 `10`。

com.ibm.cdb.topobuilder.integration.oslc.frshttptimeout
: 此属性指定 HTTP 连接的超时值（以毫秒计）。 缺省值为 `5000`。

com.ibm.cdb.topobuilder.integration.oslc.frsfailfastafter
: 此属性指定了代理程序认为 Registry Services 不可用并停止尝试连接之前的连续 HTTP 连接超时数。 缺省值为 `5`。

com.ibm.cdb.topobuilder.integration.oslc.unregisterableIPs
: 此属性指定由于不是公共地址并且可能导致过度合并资源记录而不得注册的 IP 地址。此属性可以包含正则表达式。缺省值为 `127.0.0.1,01111111000000000000000000000001,0:0:0:0:0:0:0:1,0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001`。

com.ibm.cdb.topobuilder.integration.oslc.unregisterableFQDNs
: 此属性指定由于不是公共标准域名并且可能导致过度合并资源记录而不得注册的标准域名。 此属性可以包含正则表达式。缺省值为 `localhost.localdomain,localhost,loopback6(.?\u005c\w*)*`。

com.ibm.cdb.topobuilder.integration.oslc.enablecrtvtype.*CRTVType*

该属性指定了 OSLCAgent 仅处理类型为特定 Common Resource Type Vocabulary (CRTV) 的 CI。 如果启用了该属性，那么仅会对相应 CRTV 类型的 CI 进行注册、更新及注销。

例如，设置

```
com.ibm.cdb.topobuilder.integration.oslc.enablecrtvtype.ComputerSystem=true
```

指定了仅处理计算机系统。

允许设置以下 CRTV 类型：

- ComputerSystem
- 数据库
- Fix Pack 2 SAPSystem
- ServiceInstance
- SoftwareModule
- SoftwareServer

com.ibm.cdb.topobuilder.integration.oslc.history.days_previous

此属性指定特定天数，使用"在上下文中启动"(LIC) URL 将提供所指定天数的变更历史记录信息。 缺省值为 5。

您可以使用 days_previous 属性来控制管理变更历史记录信息所需的存储空间量和处理时间。

LIC URL 列示在 OSLC 变更历史记录 HTML 预览中。

如果值大于 0，那么 days_previous 参数将应用于 LIC URL，以限制所显示的变更历史记录量。

如果值等于或小于 0，那么 LIC URL 将不包含 days_previous 参数，并且您可以查看该 CI 的整个变更历史记录。

OSLCAgent 命令行界面

您可以使用 OSLCAgent 命令行界面 (CLI) 以手动方式将配置项 (CI) 信息导出到 Registry Services。

对于 OSLCAgent，您可以将命令和开关的组合传递到 runtopobuild 脚本或批处理文件。 每个命令和开关都具有较短的单字母格式以及较长而更具描述性的格式。 您可以使用命令和开关格式的任意组合。

可用的命令如下所示：

- `-R | -refreshAll true|false`

  此命令用于注册所有合格的 CI，即使它们已注册也是如此。

- `-r | -refreshGuid` *guid*

  此命令用于注册具有指定 GUID 的 CI，即使该 CI 已注册也是如此。

- `-l | -refreshIgnored true|false`

  如果发现 CI 所在位置的深度不够，那么可能未正确形成命名规则。 缺省情况下，OSLCAgent 将忽略这类 CI。 此命令会强制 OSLCAgent 再次处理这些 CI。

要指定特定的操作，可以使用任何命令来传递开关。 共有两类可用的开关。

可以使用以下开关来启用或禁用特定 CRTV 类型的处理：

- -c | --enableComputerSystem true|false
- -d | --enableDatabase true|false
- -i | --enableServiceInstance true|false
- -m | --enableSoftwareModule true|false
- -s | --enableSoftwareServer true|false

例如，如果您不希望重新注册计算机系统，请使用 -c false 开关。

可以使用下列开关来启用或禁用配置和变更历史记录角色：

- -h | --enableChangeHistoryProvider true|false
- -p | --enableConfigurationProvider true|false

例如，如果您不希望重新注册为变更历史记录提供者，请使用 -h false 开关。

如果您希望在运行 runtopobuild 脚本或批处理文件时未传递命令或开关的情况下使用缺省值，请在 collation.properties 文件中配置以下属性：

- com.ibm.cdb.topobuilder.integration.oslc.refreshAll=true|false
- com.ibm.cdb.topobuilder.integration.oslc.refreshGuid=*guid*
- com.ibm.cdb.topobuilder.integration.oslc.enablecrtvtype.*crtv_type*

要获取可用参数和开关的完整列表，请转到 $COLLATION_HOME/support/bin 并运行带有 -H 开关的 runtopobuild 脚本或批处理文件。 例如，

```
./runtopobuild.sh -H
```

向 Registry Services 注册配置项

本主题列示了在 Registry Services 中查询注册的 TADDM 所发现的配置项 (CI)，并列示了所设置的属性及详细的映射信息。

如果特定的 CI 未注册，那么每个注册线程会生成有关 CI 未注册原因的日志信息。日志中会列示复原命名规则属性的列表。 要配置正确的日志记录级别，请在 collation.properties 文件中设置以下属性值：

```
com.collation.log.level.vm.Topology=DEBUG
```

以下属性对每个 CRTV 类型通用：

guid 设置为 CI 的 GUID 值。

name 设置为名称值、标签值或 displayName 属性值。

description
设置为描述属性值。

lastDiscoveredTime
设置为 lastModifiedTime 属性值。

SoftwareServer

CRTV SoftwareServer 类型包括以下 TADDM 类和属性：

- WebSphereServer
  - host
  - node
  - node.cell
- Db2Instance
  - home
  - host
- MQQueueManager
  - displayName | label | name
- AppServer
  - displayName | label | name
  - host
- CommunityServer
  - displayName | label
- SametimeServer
  - displayName | label
- MeetingServer
  - displayName | label
- SpecialityServer
  - displayName | label | name
- AgentManager
  - displayName | label
- SharePointRole
  - displayName | label | name

按以下方式将 TADDM 属性映射到 CRTV 属性中：

TADDM 属性	CRTV 属性	其他信息
PrimarySAP	crtv:serverAccessPoint	使用 crtv:ipAddress 来注册 serviceAccessPoint 资源及其指向的 IpAddress 资源。
version	crtv:version	
vendorName	crtv:manufacturer	
host	crtv:runsOn	crtv:runsOn points to ComputerSystem
home	crtv:instancePath	仅适用于 DatabaseServer 和 Db2Instance。
dataPath	crtv:instancePath	仅适用于 MQQueueManager。

将 rdf:type 设置为以下某个值：

- J2EEServer
- WebSphereServer
- IBMHTTPServer
- WebServer
- Db2Instance
- OracleInstance
- MQQueueManager
- WebServer
- DatabaseInstance
- CICSRegion

ComputerSystem

CRTV ComputerSystem 类型包括以下 TADDM 类和属性：

- ComptuerSystem

  设置以下某个属性组合：

  – systemId &VMID
  – systemId
  – serialNumber & model & manufacturer &VMID
  – serialNumber & model & manufacturer
  – systemBoardUUID
  – ipInterfaces

按以下方式将 TADDM 属性映射到 CRTV 属性中：

TADDM 属性	CRTV 属性	其他信息
label 或 displayName	crtv:name	
OSVersion 或 OSRunning	crtv:version	
hostSystem	crtv:dependsOn	
fqdn	crtv:fqdn	
name	crtv:shortHostname	如果设置名称，该名称应该是有效的主机名。 仅适用于 SunSPARCComputerSystem。
ipInterface	crtv:ipAddress	将这些 IP 地址的所有 FQDN 合并到 crtv:fqdn 中。

将 crtv:type 设置为以下某个值

- Generic
- SunFire
- SunSPARC
- SystemP
- Unitary

- Virtual
- WPAR

对于 LinuxUnitaryComputerSystem，按以下方式映射其他属性：

TADDM 属性	CRTV 属性	其他信息
manufacturer	crtv:manufacturer	
model	crtv:model	
serialNumber	crtv:serialNumber	
VMID	crtv:vmid	如果设置了 CPUType 和 Model： • 对于 intel，将 VMID 设置为 null，并尝试将 crtv:systemBoardUUID 设置为 systemBoardUUID 或 convertedUUID。 • 对于 power，如果已经设置了 VMID，那么将忽略 CS。

对于 SunSPARCUnitaryComputerSystem，按以下方式映射其他属性：

TADDM 属性	CRTV 属性	其他信息
systemId	crtv:hostid	
VMID	crtv:vmid	

对于任何其他计算机系统，按以下方式映射其他属性：

TADDM 属性	CRTV 属性	其他信息
manufacturer	crtv:manufacturer	
model	crtv:model	
serialNumber	crtv:serialNumber	
VMID	crtv:VMID	如果将 OSRunning 设置为 WindowsOperatingSystem，那么将 VMID 设置为 null。 如果将 OSRunning 设置为 HpUx，那么将 VMID、model 和 serialNumber 设置为 null。
systemBoardUUID 或 convertedUUID	crtv:systemBoardUUID	
worldWideName	crtv:hostid	仅适用于 FCSwitch、TapeLibrary 和 TapeMediaChanger。

数据库

"CRTV 数据库"类型包括以下 TADDM 类和属性：

- Db2Database
  - name | displayName
- IDSDatabase
  - name | displayName
- IMSDatabase
  - name | displayName
- OracleDatabase
  - name | displayName
- SqlServerDatabase
  - name | displayName
- SybaseDatabase
  - name | displayName
- DominoDatabase
  - name | displayName

按以下方式将 TADDM 属性映射到 CRTV 属性中：

TADDM 属性	CRTV 属性	其他信息
name	crtv:name	
fileName	crtv:name	仅适用于 DominoDatabase。
parent	crtv:dbInstance	

ServiceInstance

根据是否启用了早期版本兼容性，CRTV ServiceInstance 类型包含下列 TADDM 类和属性：

- 如果启用了早期版本兼容性：
  - BusinessSystem
    - name
  - Application
    - name
  - ServiceInstance
    - name
  - ServiceInfrastructure
    - name
  - SAPSystem
    - SAPSystemSID | systemHome
- 如果禁用了早期版本兼容性：
  - CustomCollection（类型仅限"BusinessApplication"）

- collectionId

按以下方式将 TADDM 属性映射到 CRTV 属性中：

TADDM 属性	CRTV 属性	其他信息
name	crtv:name	
SAPSystemSID:systemHome	crtv:name	如果 name 和 displayName 都未设置。 仅适用于 SAPSystem。
parentGUID 或 NULL	crtv:parentServiceInstance	
collectionId	crtv:name	

SoftwareModule

CRTV SoftwareModule 类型包括以下 TADDM 类和属性：

- SoftwareModule
  - fileName
  - name
  - parent.name
- MQQueue
  - name
  - queueManager

按以下方式将 TADDM 属性映射到 CRTV 属性中：

TADDM 属性	CRTV 属性	其他信息
parent	deployedTo	
fileName	crtv:fileName	

将 rdf:type 设置为以下某个值

- `J2EEApplication`
- `MQQueue`

对 OSLC 进行故障诊断

本主题描述了 OSLC 会发生的常见问题，并介绍了这些问题的解决方案。

已配置的 TADDM URL 未包含端口号

问题 配置在 `collation.properties` 文件中的 TADDM URL 属性 `taddmURL` 必须包含端口号。

如果该属性未配置端口号，那么必须更新 TADDM URL 以使其包含端口号、清除 Registry Services 信息或特定提供程序、并清除 TADDM 时间戳记。

解决方案

要更新 TADDM URL 以使其包含端口号，请完成以下步骤：

1. 在 `collation.properties` 文件中，按以下方式设置 `taddmURL` 属性：

   `taddmURL=http://server.domain:port`

2. 在具有 Registry Services 的计算机上，完成以下步骤：
   a. 转至 /opt/IBM/JazzSM/registry/etc。
   b. 在 CLI.properties 文件中，将凭证配置为以下属性：
      - ds.jdbc.user
      - ds.jdbc.password
      - appserver.user
      - appserver.password
   c. 转至 /opt/IBM/WebSphere/AppServer/bin。
   d. 运行 stopServer.sh 脚本以停止 WebSphere Application Server。

      ```
      ./stopServer.sh server_name -user user_name -p password
      ```

      例如，

      ```
      ./stopServer.sh server1 -user wasadmin -p passw0rd
      ```
   e. 转至 /opt/IBM/JazzSM/registry/bin。
   f. 以相应的参数运行 frs.sh 脚本：

      ```
      ./frs.sh uninstall -type db -properties ../etc/CLI.properties
      ```
   g. 检查数据库是否已断开。 如果尚未断开，请运行以下命令：

      ```
      db2 drop db db_name
      db2 create db db_name
      ```

      其中 *db_name* 是 Registry Services 数据库的名称。
   h. 转至 /opt/IBM/JazzSM/registry/bin。
   i. 以相应的参数运行 frs.sh 脚本：

      ```
      ./frs.sh install -type db -properties ../etc/CLI.properties
      ```
   j. 转至 /opt/IBM/WebSphere/AppServer/bin。
   k. 运行 startServer.sh 脚本以启动 WebSphere Application Server。

      ```
      ./startServer.sh server_name -user user_name -p password
      ```

      例如，

      ```
      ./startServer.sh server1 -user wasadmin -p passw0rd
      ```
   l. 以相应的参数运行 frs.sh 脚本：

      ```
      ./frs.sh uninstall -type container -properties ../etc/CLI.properties
      ```
   m. 以相应的参数运行 frs.sh 脚本：

      ```
      ./frs.sh install -type container -properties ../etc/CLI.properties
      ```

   通过使用以下命令可以将项目从特定提供程序的 Registry Services 中移除：

   ```
   ./frs.sh deleteProvider -providerUrl url - properties cli.properties
   ```
3. 在具有 TADDM 数据库的计算机上，完成下列步骤：
   a. 转至 $COLLATION_HOME/support/bin。
   b. 运行带有相应参数的 runtopobuild 脚本或批处理文件，例如：

      ```
      ./runtopobuild.sh -a OSLCAgent -R
      ```

Tivoli Directory Integrator

购买 IBM Tivoli Application Dependency Discovery Manager (TADDM) 时，您还会收到 Tivoli Directory Integrator，您可以通过它将 TADDM 与其他数据源集成。

Knowledge Center 中的 Tivoli Directory Integrator 文档

http://www-01.ibm.com/support/knowledgecenter/SSCQGF_7.1.0/KC_ditamaps/welcome.html?lang=en

Tivoli Application Dependency Discovery Manager Wiki 中的 TADDM 集成方案

https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Application%20Dependency%20Discovery%20Manager/page/Integration%20Scenarios

与早期版本之间的业务实体兼容性

引入了一项新的功能可支持 TADDM 与以下产品的集成：使用 DataApi 从 TADDM 中读取数据的产品，或者使用 SQL 直接从 TADDM 数据库中读取数据的产品。 IBM Tivoli Business Service Manager (TBSM)、IBM SmartCloud Control Desk (SCCD) 和 Tivoli Directory Integrator (TDI) 都是这种产品的示例。 当前业务应用程序数据模型依赖于 CustomCollection 接口，此接口与以前的 Application 和 ITSystem 接口没有任何共同点。新功能支持与其他产品进行集成，且不会对这些系统引入修改。

在 TBSM 和 SCCD 的将来版本中，将引入具有新功能的业务应用程序模型。 目标是生成先前的业务实体，这些实体是定制集合实例的副本。

提供了早期版本兼容性的新功能具有下列特征。

运行 BizAppsAgent 时的附加步骤

此附加步骤将生成与代理程序所生成的每个定制集合的早期版本兼容的业务实体（服务、应用程序和集合）。

为了激活此步骤，向 `collation.properties` 文件添加了新属性 `com.ibm.cdb.serviceinfrastructure.earlier.ver.compatibility`。 此属性的缺省值为 *TRUE*（对于升级方案）和 *FALSE*（对于全新安装方案）。

OSLC 支持

OSLC 代理程序已进行了修改，现在能够注册旧业务实体或新定制集合。 兼容性标志设置为 *TRUE* 时，将注册旧业务实体。否则，将使用定制集合为 Jazz for Service Management (JazzSM) 生成内容。

以后，当集成产品开始使用新模型对象（定制集合和节点）来装入数据时，将需要完全重新装入业务实体。 旧业务应用程序（应用程序）和新业务应用程序（定制集合）不得具有相同的 GUID。 为了避免重复，在装入新定制集合之前，用户必须除去旧业务应用程序。

创建功能组

与旧业务应用程序不同，新业务应用程序没有功能组。 但是，引入了一项新的层功能来实现相似目的。 对于每个唯一的层，为了确保与早期版本兼容，将创建名称与层名称对应的功能组。

有关更多信息，请参阅《TADDM 用户指南》中的『业务应用程序层』主题。

集成 BigFix

Fix Pack 5

IBM 致力于开发 TADDM 支持，以发现安全服务器/机器而不必使用锚点和网关，其利用 BigFix 基础结构作为基础。

注：任何时候路径显示为相对路径时，都假定路径相对于 $COLLATION_HOME (/opt/ibm/taddm/dist) 或 %COLLATION_HOME% (E:\ibm\taddm\dist)。

简介

IBM/Aricent 致力于开发 TADDM 支持，以发现安全服务器/机器而不必使用锚点和网关，其利用 BigFix 基础结构作为基础。

注：任何时候路径显示为相对路径时，都假定路径相对于 $COLLATION_HOME (/opt/ibm/taddm/dist) 或 %COLLATION_HOME% (E:\ibm\taddm\dist)。

目的：

TADDM 使用锚点和网关来发现防火墙之后的机器/应用程序/网络。当前，可使用 IBM Netcool 监控工具 (ITM) 来避免使用锚点/网关。此外，可利用 TADDM 与 BigFix 体系结构的集成以避免使用锚点和网关。BigFix 体系结构有 BigFix 服务器（BES 服务器）和多个 BigFix 端点（BES 客户机）组成，其中，BES 客户机是可通过 BES 服务器进行访问的安全机器。可通过 BES 服务器，复用/利用 BigFix 基础结构来自动在 BES 客户机上运行 TADDM 脚本软件包。

此集成对于 TADDM 管理员的主要优点包括：

1. 能够在不使用锚点的情况下发现受防火墙防护的区域。
2. 能够复用 BigFix 体系结构（例如，打开的安全端口）来访问端点，并可节省使用标准 TADDM 方法发现相同目标的设置时间。
3. 与 TADDM 的基于脚本的传感器的战略方向保持一致。
4. 使 TADDM 管理员的所需的干预降至最低。
5. 对于受防火墙保护区域的机器，提供无需使用 TADDM-ITM 集成的备用发现方法。

参考：

TADDM 文档

下表显示了可与 TADDM 集成的产品受支持的版本。

有关与 TADDM 集成的产品的更多信息，请参阅相应的文档：

- TADDM 7.3 和传感器 Knowledge Center（官方文档）http://www-01.ibm.com/support/knowledgecenter/SSPLFC_7.3.0/com.ibm.taddm.doc_7.3/welcome_page/kc_welcome-444.html?lang=en
- TADDM 7.3 传感器和支持的目标系统 https://www.ibm.com/developerworks/community/groups/service/html/communityview?communityUuid=7d5ebce8-2dd8-449c-a58e-4676134e3eb8#fullpageWidgetId=Wea1cb2531f10_4ccd_99d7_6ab0334cb21f&file=e70bf323-31f1-45ba-8992-4cb491feab4a

- TADDM 配置异步脚本发现 (ASD) https://www.ibm.com/support/knowledgecenter/SSPLFC_7.3.0/com.ibm.taddm.doc_7.3/SensorGuideRef/r_cmdb_async_script_sensors.html#sensorsthatcanbescripted
- IBM BigFix 配置指南 https://www.ibm.com/support/knowledgecenter/SSPLFC_7.3.0/com.ibm.taddm.doc_7.3/SensorGuideRef/r_cmdb_async_script_sensors.html#sensorsthatcanbescripted
- TADDM 支持 Web 站点 http://www.ibm.com/software/sysmgmt/products/support/IBMTivoliApplicationDependencyDiscoveryManager.html
- TADDM Wiki https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Application%20Dependency%20Discovery%20Manager/page/Home 这是 TADDM 最新信息和最佳实践的优秀来源。请将此页面加入书签以方便使用。
- TADDM 论坛 http://www.ibm.com/developerworks/forums/forum.jspa?forumID=1547&categoryID=15&ca=drs-fo
- 改进请求社区 http://www.ibm.com/developerworks/rfe/?BRAND_ID=90 应使用此社区以直接向 IBM 开发人员请求产品增强功能。

解决方案体系结构：

TADDM-BigFix 集成基于增强和自动化异步脚本发现 (ASD)（其需要 TADDM 管理员的手动干预）的当前行为。此集成利用 BigFix 基础结构提供的指向受防火墙保护区域的机器的连通性，通过 TADDM 脚本软件包运行发现。

在 ASD 中，TADDM 管理员必须手动执行以下步骤：

1. 在 TADDM 服务器上执行脚本以创建包含要在目标上运行的所有传感器的发现软件包。
2. 将此软件包传输到目标系统。
3. 在目标系统执行发现软件包。
4. 将在目标系统生成的结果文件传输回 TADDM 服务器。

利用当前解决方案，自动化了手动步骤，因此解决方案也称为自动异步脚本发现 (AASD)。TADDM 管理员仅需执行脚本以在 TADDM 服务器上启动发现，其余步骤将自动执行。

BigFix 发现中的步骤：

Big Fix 发现步骤详细信息

步骤 1：Big Fix 集成脚本

开发了脚本"runBigFixDiscovery.sh"，此脚本将从 TADDM 发现服务器启动 AutoASD 发现 (AASD)。可随需执行脚本。此脚本获取发现作用域和发现概要文件名称作为输入（在 BigFix 访问凭证旁边）并支持以下方式：

- 发现方式 - 用于启动 BigFix 发现
- 轮询方式 - 用于轮询 BigFix 发现的结果
- 清除方式 - 用于随需从 BES 根服务器清除发现结果软件包
- 重新发现方式 - 用于重新运行先前的发现

a) 创建 AutoASD 传感器软件包

- 指定的发现概要文件用于访存传感器列表，并且 AASD 脚本软件包创建仅考虑有效的脚本化传感器子集。请参阅附录 D 以获取 TADDM 支持的所有脚本化传感器的完整列表，但是此功能仅支持标准 ASD 方式支持的这些传感器的子集。
- 忽略发现概要文件中的其他非脚本化传感器
- AASD 软件包是操作系统不可知 - 因此，如果不存在，那么在 BigFix 端点上某些传感器可能失败
- 使用 /api/upload REST API 将生成的 AASD 脚本软件包上载到 BigFix 根服务器

b) 创建 BigFix 任务

- 使用指定的发现作用域来创建 BigFix 理解的"相关性"XML
- 使用"相关性"和哑元"ActionScript"生成 BigFix 任务 XML
- 基于当前日期时间生成任务标题
- 使用 /api/tasks/custom/TADDM REST API 以在 BigFix 服务器上名为"TADDM"的定制站点下创建 BigFix 任务

c) 启动 BigFix 任务

- 使用 <SourcedFixletAction> 以针对以上创建的 BigFix 任务启动操作执行
- /api/actions BigFix REST API 用于在目标端点上启动"ActionScript"执行

步骤 2：运行脚本

- 作为"ActionScript"执行的一部分，将解压缩 TADDM AASD 软件包，并且将在 BigFix 端点执行包含的传感器脚本（基于发现概要文件）。

步骤 3：收集 Zip

- 在"ActionScript"执行结束时，会将通过执行 TADDM AASD 软件包在 BES 客户机上生成的结果软件包复制到 BES 根服务器

步骤 4：将结果导回 TADDM

- TADDM 将持续轮询 BigFix 服务器数据库以检查上载到 BES 服务器的结果文件
- 如果数据库显示存在新的结果文件，那么 TADDM 将生成 HTTP 请求以访存加密的结果文件、进行解密并保存
- 然后，TADDM 将根据配置的作用域和概要文件处理这些结果文件，并在数据库中存储发现的对象

TADDM-Bigfix Integration (Limited Availability)

BigFix 增强型 TADDM 发现解决方案的 Limited Availability 发行版关注于 Windows、Linux、AIX 和 Solaris 操作系统和相关传感器（针对多个端点）发现的端到端功能执行，并且通过 BigFix REST API 在 TADDM 和 BigFix 之间支持 SSL 通信。可在 TADDM 发现服务器上启动发现，并且应在 TADDM GUI 上自动检索和显示发现结果

假设：

在运行服务时考虑以下假设：

1. BigFix 服务器和客户机具有第 2.1 节中提及的版本。

2. BigFix 客户机具有相应的权限来执行 BigFix 服务器上载的发现任务/操作脚本。

3. collation.properties 中配置的 BigFix SQL 数据库用户必须具有 BFEnterprise 数据库的读访问权。

4. 通过 BigFix 代理程序执行的传感器脚本软件包需要配置的 TEMP 目录（例如，"C:\Windows\Temp"）的写访问权。可在 collation.properties 中配置 Temp 目录，并且假定目录路径不含空格。

5. 不在 TADDM 发现服务器上处理脚本请求软件包的清除，并且假定管理员进行管理。

6. 因为 TADDM-BigFix Integration 基于 TADDM 中的当前现有 ASD 框架，因此该进程的性能特征将基于 ASD 框架基准测试。

7. 仅可使用"taddmusr"在 TADDM 发现服务器上运行 BigFix 发现脚本，并且不允许 root 用户。

8. Bigfix 根服务器清除 - 将每次 TADDM 启动时调用清除，而且还将根据配置的持续时间（com.collation.bigfix.root.cleanup.interval = Default 1 day）定期调用。这将删除超过配置的时间的结果文件 (com.collation.bigfix.root.cleanup.days = Default 5 days)。

9. TADDM 服务器清除 - 将处理以下清除：在发现期间在 TADDM 服务器上创建/复制的所有结果文件，这些文件的名称包含"taddmasd"且以"_DONE"结尾。（至少在附录 A 中指定一个阈值，项目符号 6 应配置为在 TADDM 服务器上启用清除）。

10. 端点清除 - 缺省情况下启用发现端点清除，并且可通过配置以下属性设置进行控制：

- a) 设置为值"N"的"com.collation.bigfix.endpoint.cleanup"将禁用发现端点上的清除。

先决条件：

在从 TADDM 服务器启动发现之前，必须满足以下先决条件：

1. 在发现概要文件创建期间，必须选择 ASDSensor、ASDPingSensor 和 Generic Server Sensor 并取消选中 PingSensor、PortSensor 和 SessionSensor。

2. 已完成第 2.3 节中指定的所有配置步骤。

3. Bigfix 操作脚本已使用本机 powershell 命令来解压缩 Windows 端点上的请求软件包，在 Linux 上使用本机 tar 命令

注：根据特定需求，可进行 ActionScript 定制。通过更新 `$COLLATION_HOME/etc/ folder` 中的客户可修改文件 ActionScript_Pre_Post.txt，支持此定制；例如，要启用下载和使用定制解压缩软件（BigFix 根服务器上的可执行文件分发版）。以下提供样本示例片段：

```
%WIN_PRE_START%
if {not exists file "C:\Windows\System32\unzip.exe"}
prefetch unzip.exe sha1:e1652b058195db3f5f754b7ab430652ae04a50b8
```

```
size:167936 http://10.160.161.199:52311/Uploads/Unzip/unzip.exe

// Make sure that environment is set appropriately and "unzip"
utility is available in the windows PATH
copy "__Download\unzip.exe" "C:\Windows\System32\unzip.exe"

endif
%WIN_PRE_END%

%WIN_POST_START%
%WIN_POST_END%

%LIN_PRE_START%
%LIN_PRE_END%
...
```

4. 运行发现的用户应具有结果文件夹的读/写许可权。

5. 应提供足够的磁盘空间、处理容量和内存以满足要 TADDM 服务器、BigFix 根服务器和发现目标上进行处理的请求和结果软件包。

6. Bigfix 代理程序（端点计算机）应配置足够的"_BESClient_ArchiveManager_MaxArchiveSize"值，以便成功将结果上载到 BigFix 根服务器。

7. 必须设置正确的作用域和概要文件（请参阅第 4.1 节以设置作用域和概要文件）。

8. 必须在 BigFix 服务器上配置并且存在站点名称"TADDM"。

9. 标准 ASD 传感器脚本所需的所有先决条件也适用于 BigFix 发现。

- a) 对于涉及 Windows2003 端点的发现，应正确安装并配置 Powershell 可执行文件。

限制：

以下限制与当前发行版相关：

1. "发现历史记录"中将不显示发现期间指定的不可从 BigFix 根服务器访问的任何发现目标。

2. 在选择和启用其他传感器时将自动启用 PingSensor、PortSensor 和 SessionSensor，并且必须在发现概要文件创建期间手动进行这些传感器。

3. 根据设计，不支持 TADDM 发现服务器/BigFix 根服务器上的请求软件包清除。请注意，在重新发现期间可使用（由重新发现方式触发）。

4. 仅在启动原始发现的相同 TADDM 发现服务器上支持重新发现。

5. 这是 Limited Availability 发行版，因此翻译、联机文档支持等不可用。

6. BigFix Integration 不支持定制服务器发现。

配置：

请遵循本部分中提到的步骤以设置期望的配置。

Bigfix 发现的基本配置：

1. 在 $COLLATION_HOME/etc/collation.properties 中设置以下必需的属性

a) BigFix Integration 功能部件设置

```
com.collation.bigfix.enabled=true
```

b) BigFix 服务器设置

- com.collation.bigfix.host=<BigFix 服务器的 ip 或 fqdn>
- com.collation.bigfix.port=<port_no>
- com.collation.bigfix.uid=<BigFix 服务器控制台访问的用户标识>
- com.collation.bigfix.pwd=<用于访问 BigFix 服务器控制台的密码>

c) BigFix 数据库设置

- com.collation.bigfix.db.type=<MSSQL/DB2>
- com.collation.bigfix.db.host=<BigFix 服务器数据库的 IP 或 FQDN>
- com.collation.bigfix.db.port=<TADDM 用于连接 BigFix 数据库的端口>
- com.collation.bigfix.db.dbname=<BigFix 数据库名称>
- com.collation.bigfix.db.domain=<用户域> 可选参数 - 仅在针对 BigFix 数据库配置基于 Windows 的认证时需要>
- com.collation.bigfix.db.uid=<用于访问 BigFix 数据库的用户标识>
- com.collation.bigfix.db.pwd=<用于访问 BigFix 数据库的密码>

d) 结果处理线程设置

- com.ibm.cdb.discover.asd.ProcessUnreachableIPs=true
- com.ibm.cdb.discover.asd.autodiscovery.enabled=true

2. 设置 $COLLATION_HOME/etc/collation.properties 中的以下属性（仅限在 BigFix 服务器上启用 SSL 配置时）：

e) Bigfix 证书

- com.collation.bigfix.certificate.type=<PKCS12/JKS>
- com.collation.bigfix.certificate.file=<证书文件的完整路径>
- com.collation.bigfix.certificate.pwd=<用于使用证书的密码>

注：除这些配置的必需属性外，还有更多属性。请参阅附录 A 以获取属性及其详细信息的完整列表。

注：不支持使用空密码生成的证书。

3. 执行脚本"encryptprops.sh"以加密属性（请参阅附录 C 以检查用于运行此脚本的格式）。否则，发现脚本 (runBigFixDiscovery.sh/.bat) 将失败，发生缺少自变量或无效错误（请参与附录 E 以获取错误代码详细信息），因为不接受非加密密码。

4. 在 TADDM 中创建 $COLLATION_HOME/var/asdd 文件夹来存储结果文件。如果不使用 var/asdd 文件夹，那么必须使用特定文件夹（管理员想要下载结果文件的位置）设置属性"com.ibm.cdb.discover.asd.AsyncDiscoveryResultsDirectory"。

5. 重新启动 TADDM。

6. 使用第 2.1 节中列出的必需传感器创建发现概要文件。除要发现的其他传感器外，概要文件应具有必需的传感器。

7. 使用需要运行发现的 BigFix 目标端点创建发现作用域。

其他配置：

在 BigFix 服务器上，将创建站点名称"TADDM"。

1. 打开"BigFix 控制台" -> 转至"工具"选项卡 -> 选择"创建定制站点"-> 提供站点名称"TADDM"。

2. 单击"TADDM" -> 选择"计算机预订"选项卡 -> 根据需求预订计算机（应包含 BigFix 服务器想要通过 TADDM 连接到的所有计算机）。

日志文件和故障诊断提示：

如果在发现期间遇到任何故障，可检查以下事项。确认满足并遵循所有先决条件：

TADDM 发现服务器

- 检查 Bigfix Discovery 脚本执行日志
  - `$COLLATION_HOME/log/BigFixDiscovery.log`
- 检查结果文件的日志是否到达 TADDM 服务器：
  - `$COLLATION_HOME/log/services/ApiServer.log`（搜索关键字"BigfixDiscoveryServerController"和"AASDiscoveryServerController"）

BigFix Root Server

要使用 **BigFix Root Server** 上的 `IBM BigFix Console` 检查发现和操作执行的状态

1. 打开 **IBM Bigfix Console**。
2. 选择**站点**（定制 -> TADDM）-> **Fixlet** 和**任务**。
3. 选择**任务**（在脚本执行期间指定）。
4. 检查**详细信息**和**操作历史记录**。
5. 选择**特定操作历史记录** -> **报告的计算机**。
6. **双击**行以按行执行检查行的状态。
7. 单击**确定**以返回至"发现概要文件"窗口。

BigFix Agent/发现目标

- 检查 `%wintemp%"/taddm7.3.0.4/asd` 文件夹中存在结果文件（仅在 com.collation.bigfix.endpoint.cleanup 属性设置为"N"时）
- 在传感器脚本执行期间，可针对任何错误引用 `allErrors.txt` 文件（位于 `%wintemp%"/taddm7.3.0.4/asd` 中）

运行 *Bigfix* 发现：

运行 Bigfix 发现

创建作用域：

打开 TADDM 服务器 GUI 以创建发现作用域。作用域应包含所有 BigFix 目标和端点。目标端点可设置为个别主机或者指定域/网络作用域。

创建概要文件：

应通过 TADDM 服务器 GUI 创建"发现"概要文件。概要文件应包含管理员想要 TADDM 发现的这些应用程序的传感器。请参阅第 2.1 节以获取有关需要在创建的发现概要文件中强制包含/排除的传感器的详细信息。

运行脚本：

要运行发现，请从 $COLLATION_HOME/bin 执行脚本"runBigFixDiscovery.sh"。可通过 4 种方式执行脚本："发现"、"轮询"、"清除"和"重新发现"。在发现方式下，启动发现。在轮询方式下，将访存当前发现状态。在清除方式下，将在 BigFix 根服务器上触发结果文件清除，并且在重新发现方式下，可重新执行先前运行的发现。

1. 发现方式 -

TADDM 为 Jazz for Service Management 提供了订阅源服务，地址如下所示：

```
./runBigFixDiscovery.sh -d -o <output dir> -s <scope> -p <profile>
Where,
-d – for discovery mode
-o - output directory where discovery package would be created
-s – input scope, having BigFix endpoints target that are to be discovered.
-p – input profile, having sensors to be run
```

在发现方式下执行命令后，启动发现。这将向您显示所执行的步骤的状态并且给出"操作"的标识。可在轮询方式下使用此操作以检查每个 BigFix端点上操作的状态

注：

- 针对轮询之目的，可复用针对发现创建的"ActionID"（在控制台输出中显示，例如，以下示例中的 2090）。
- 请保留与给定的作用域和概要文件相关联的新创建的 bigfix 任务名称（在控制台输出中显示"TaskName"，例如，20180130125432），此名称可在重新发现时复用。

2. 轮询方式 -

```
./runBigFixDiscovery.sh -p -r <repetition> -i <Action id>

Where,
-p – for poll mode
-r – no of times polling will be done to BigFix server
-i - Action id obtained from discovery mode command.
```

3. 清除方式 -

```
./runBigFixDiscovery.sh -c -d <No of Days>
Where,
-c – for cleanup mode
-d – Files older than specified number of days to be removed
```

4. 重新发现方式 -

```
./runBigFixDiscovery.sh  -r/--rediscover -i <TaskName>
Where,
-r – for rediscover mode
-i – TASK NAME corresponding to the previous discovery, that needs to be run again
```

注：可在附录 B 中获取有关此命令以及所有可能的选项的更多信息信息，并且可在附录 C 中获取运行命令的示例。

发现结果处理：

"runBigFixDiscovery.sh"命令的轮询方式给出在每个 BigFix 端点上执行的操作的状态。根据状态，将创建和处理结果文件。

1. 在针对端点成功完成操作后，将在配置的结果文件夹中下载此端点的结果文件（缺省情况下为 var/asdd。请检查第 2.3.1 节以配置结果文件夹）。

2. 在针对端点成功完成操作后，将在配置的结果文件夹中下载此端点的结果文件（缺省情况下为 var/asdd。请检查第 2.3.1 节以配置结果文件夹）。

3. 在成功处理结果文件后，可在 TADDM GUI 的"历史记录"选项卡上查看结果。

4. 处理的结果数据将存储在 TADDM 数据库中并在 TADDM 的数据管理门户网站或 PSS 上提供。

可能的故障情况：

如果在发现、轮询、清除或重新发现方式期间遇到任何故障，那么可检查以下事项：

1. 确认遵循第 2.2 节中提及的所有先决条件：

2. 检查以下路径中的 TADDM 日志：

- `$COLLATION_HOME/log/BigFixDiscovery.log` - 发现和脚本执行日志
- `$COLLATION_HOME/log/services/ApiServer.log` - 结果访存和解析日志

3. 检查 BigFix 服务器日志和 BigFix 控制台以查找任何故障状态。

4. 如果"**操作轮询**"从"**根服务器**"和"**端点**"收到故障，那么可验证 **Bigfix ActionScript** 执行日志。

附录 A. 在集成中使用 Collation.Properties

1. BigFix 功能已启用

表 *50.*

属性名称	可能的值	描述	必需
com.collation. bigfix.enabled	true/false	如果为 true，那么将启用 BigFix 功能。在将此属性设置为 true 后，需要重新启动 TADDM。	Y

2. BigFix 服务器

表 51.

属性名称	可能的值	描述	必需
com.collation. bigfix.host	IP 或 FQDN	BigFix 服务器的 IP 或 FQDN。	Y
com.collation. bigfix.port	<port no> 缺省值 = 52311	TADDM 将向 BigFix 服务器发送请求的端口。	N
com.collation. bigfix.uid	<userid>	用于访问 BigFix 服务器控制台的用户标识。	Y
com.collation. bigfix.pwd	<password>	用于访问 BigFix 服务器控制台的密码。将存储为加密形式。	Y
com.collation. bigfix.connectTo	<time_period> 缺省值 = 20 秒	在 HTTP/RestAPI 连接超时前，TADDM 将等待此时间段（以秒为单位）。	N
com.collation. bigfix.responseTo	<time_period> 缺省值 = 20 秒	在 HTTP 响应超时前，TADDM 将等待此时间段（以秒为单位）。	N
com.collation. bigfix.site.type	<site_type> 缺省值 = custom	TADDM 将连接的 BigFix 服务器的站点类型。	N
Visibility.Control. Automation	<site_name> 缺省值 = TADDM	TADDM 将连接的 BigFix 服务器的站点名称。	N
com.collation. bigfix. aasdpkgmaxsize	<Request Package size> 缺省值 = 1024	BigFix 发现脚本生成的最大允许的请求软件包数。	N

3. 大服务器证书

表 52.

属性名称	可能的值	描述	必需
com.collation. bigfix.certificate.type	<PKCS12/JKS> 缺省值 = JKS	支持的客户机证书类型。	N
com.collation. bigfix.certificate.file	<Path>	客户机证书文件位置	N
com.collation. bigfix.certificate.pwd	<Password>	客户机证书的密码	N

4. BigFix 服务器数据库

表 53.

属性名称	可能的值	描述	必需
com.collation. bigfix.db.type	MSSQL 或 DB2	BigFix 服务器使用的数据库的类型；MSSQL 用于基于 Windows 的 BigFix 服务器，DB2 用于基于 Linux 的服务器。	Y

表 53. (续)

属性名称	可能的值	描述	必需
com.collation. bigfix.db.host	IP 或 FQDN	BigFix 数据库的 IP 或 FQDN。	Y
com.collation. bigfix.db.port	<port no>	TADDM 用于连接 BigFix 数据库的端口。	Y
com.collation. bigfix.db.dbname	<db name>	BigFix 数据库的名称。	Y
com.collation. bigfix.db.domain	<user domain>	用户的域，如果使用基于 Windows 的认证，那么是必需的。	N
com.collation. bigfix.db.domain	<userid>	用于访问 BigFix 数据库的用户标识。	Y
com.collation. bigfix.db.pwd	<password>	用于访问 BigFix 数据库的密码；将存储为加密形式。	Y

3. 检查 BigFix 服务器日志和 BigFix 控制台以查找任何故障状态。

注：

- 如果 TADDM-DB 连接中断，那么 TADDM 将按照"com.collation.bigfix.result.wait" 设置尝试重新连接。
- 如果以上设置中有任何更改，那么需要重新启动 TADDM。

5. TADDM - 结果访存/处理线程

表 54.

属性名称	可能的值	描述	必需
com.collation. bigfix.result.wait	<value in sec> 缺省值 = 60 秒	在启用 "com.collation.bigfix.enabled" 时，"结果访存线程"将衍生为定期从 BigFix 服务器访存发现结果文件。"结果访存线程"将根据配置的周期（以秒定义）从 BigFix 服务器访存结果软件包。	N
com.ibm.cdb. discover.asd. autodiscovery.enabled	True/false	如果为 true，那么支持线程处理存储的 ASD 结果文件	Y
com.ibm.cdb. discover.asd. ProcessUnreachableIPs	True/false	线程将针对不可访问的目标处理 ASD 结果。	Y
com.ibm.cdb. discover.asd. AsyncDiscovery ResultsDirectory	路径缺省值 = var/asdd	要保留结果文件的路径。路径可配置，但是缺省情况下设置为 var/asdd。	N
com.ibm.cdb. discover.asd. autodiscovery. asdScope	<scope name> 缺省值 = ASD	线程将选取此作用域中提及的目标以处理结果文件。如果未提及此属性，那么将处理缺省 ASD 作用域。	N

表 *54. (续)*

属性名称	可能的值	描述	必需
com.ibm.cdb.discover.asd.autodiscovery.asdProfile	<profile name> 缺省值 = ASD	线程将选取此概要文件中提及的传感器以处理结果文件。如果未提及此属性，那么将处理缺省 ASD 概要文件。	N
com.ibm.cdb.discover.asd.autodiscovery.filesThreshold	<file threshold> 缺省值 = 20	线程开始处理文件时所需的最大文件数。如果满足文件阈值或时间阈值，那么线程将处理结果。	N
com.ibm.cdb.discover.asd.autodiscovery.timeThreshold	<time threshold> 缺省值 = 60 秒	时间阈值，在此时间后线程将处理结果文件，即使未满足"文件阈值"。	N

注：1. BigFix 发现结果将异步到达 TADDM 服务器，并且在满足其中一个属性（com.ibm.cdb.discover.asd.autodiscovery.filesThreshold 或 com.ibm.cdb.discover.asd.autodiscovery.timeThreshold）时，将处理可用的结果文件群，并且将创建全新的"发现历史记录"条目。将根据特定需求微调这些属性以控制"发现历史记录"条目的数量。

6. 清除

表 *55.*

S.No	资源	TADDM 服务器		BES 根服务器		BES 端点	
		已创建	Cleanup	已创建	Cleanup	已创建	Cleanup
1.	请求软件包	Y	N[1]	Y	N[2]	Y	Y
2.	任务	-	-	Y	Y [4]	-	-
3.	操作	-	-	Y	Y [3]	-	-
4.	结果软件包	Y	Y	Y	Y	Y	Y
5.	文件集	-	-	-	-	Y	Y

注：

- 不支持 TADDM 服务器上的请求软件包清除
- 不支持 BES 根服务器上的请求软件包清除。（在重新发现期间可复用请求软件包）
- 仅针对由 TADDM 创建且处于"已到期"状态的操作考虑清除（使用名称 TADDMCLEANUP 创建的操作除外）。
- 将除去 TADDM 创建的任务，并且仅限已除去与此任务相关联的所有操作时。
- 将除去 TADDM 创建的任务，并且仅限已除去与此任务相关联的所有操作时。
  - 要从清除中排除特定任务及其关联的操作（针对重新发现支持），可根据以下详细信息使用 retainBigFixTask.sh/.bat：

  用法：./retainBigFixTask.sh <TaskName> <enable/disable>

TADDM 服务器上的清除

表 56.

属性名称	可能的值	描述	必需
com.collation. bigfix.taddm. cleanup.volume	<带后缀的限制大小>	<限制大小，在到达此大小后将除去旧的已处理文件，例如，50MB、2GB 等>	N
com.collation. bigfix.taddm. cleanup.time	<带后缀的限制时间>	<用于检查时间超过配置的单元的已处理文件，例如，1D、5H 和 30M 等>	N
com.collation. bigfix.taddm. cleanup.runtime	分钟数	TADDM 清除线程将在运行后等待配置的分钟数。	N

注：

- 将执行 TADDM 服务器上的结果文件清除，仅限至少了配置了一个属性（com.collation.bigfix.taddm.cleanup.volume 或 com.collation.bigfix.taddm.cleanup.time）时。

端点上的清除

表 57.

属性名称	可能的值	描述	必需
com.collation. bigfix.endpoint. cleanup	<Y 或 N> <缺省值 = Y>	在设置为 Y 时，将从端点除去请求软件包大小、抽取的请求包目录以及新创建的结果软件包 zip。	N

BigFix 根服务器上的清除

表 58.

属性名称	可能的值	描述	必需
com.collation. bigfix.root. cleanup.interval	<no of days> <Default=1>	定期运行清除任务以从 BES 根服务器除去结果软件包、任务和已到期的操作。	N
com.collation. bigfix.root. cleanup.days	<no of days> <Default=5>	超过给出天数的结果文件将被视为除去。	N

7. 定制相关性

表 59.

属性名称	可能的值	描述	必需
com.collation. bigfix.relevance. appendscope	true/false <Default=true>	如果为 true，那么除指定的作用域外，将使用定制相关性查询 如果为 false，那么仅使用定制相关性查询，代替指定的作用域	N

表 59. (续)

属性名称	可能的值	描述	必需
com.collation. bigfix. relevance	True/false	用于针对给定的发现标识一组端点的相关性查询。	N

8. Bigfix 软件包 Temp 路径设置

表 60.

属性名称	可能的值	描述	必需
com.collation. bigfix.action. enable.os	<action script for the configured os><Default= Windows, AIX, Linux, SunOS>	Bigfix ActionScript 中将包含配置的操作系统的操作脚本	N
com.collation. bigfix.temp. Windows	请求软件包路径 <Default= C:\Windows\ Temp>	将用于 ASD 请求软件包的路径。*注：在 Windows 路径中，"\"需要指定为"\\"。	N
com.collation. asd.temp.Windows	结果软件包路径 <Default= C:\Windows\ Temp>	将用于 ASD 结果软件包的路径。	N
com.collation. asd.temp.Unix	结果软件包路径 <Default= /tmp>	将用于 ASD 结果软件包的路径。	N
com.collation. bigfix.temp.Linux	结果软件包路径 <Default=/tmp>	将用于 ASD 请求软件包的路径。	N
com.collation. bigfix.temp. SunOS	结果软件包路径 <Default= /tmp>	将用于 ASD 请求软件包的路径。	N
com.collation. bigfix.temp. AIX	结果软件包路径 <Default= /tmp>	将用于 ASD 请求软件包的路径。	N

注：根据以上配置的 Temp 路径，如果不存在，那么将在目标端点上创建文件夹。例如，在 Windows 2003 中，将使用缺省 Temp 路径"C:\Windows\Temp\"，并且将在发现期间创建此文件夹。

附录 B. 不同方式的脚本参数帮助

./runBigFixDiscovery.sh - 用于运行 BigFix 增强型发现或查询现有发现操作的 TADDM 工具。

方式：发现

用法：bin/runBigFixDiscovery.sh -d/--discover [-c <arg>] [-freq <arg>] [-h] [-intr <arg>] -o <arg> -p <arg> -s <arg>

其中，

表 61.

-c，--compressMethod <arg>	[缺省值：ZIP] 可能的值：[ZIP, TAR]。
-freq，--frequency <arg>	[缺省值：1] 需要运行发现的次数。
-h，--help	显示帮助。
-intr，--interval <arg>	[缺省值：P1D] 重新运行发现之间的时间间隔。受支持的值：[PT15M、PT30M、PT1H、PT2H、PT4H、PT6H、PT8H、PT12H、P1D、P2D、P3D、P5D、P7D、P15D 和 P30D]。
-o，--output <arg>	必需：将生成 Bigfix 发现程序包的输出目录。
-p，--profile <arg>	必需：概要文件名称将用于发现程序包软件以包含传感器。
-s，--scope	必需：Scope/ScopeGroup 名称（逗号分隔。用引号括起包含空格的名称）。

方式：轮询

用法：bin/runBigFixDiscovery.sh -p/--poll [-h] -i <arg> [-r <arg>] [-t <arg>]

其中，

表 62.

-d，--detail <arg>	[缺省值：true] 每个端点的轮询结果
-h，--help	显示帮助。
-r，--repeat <arg>	[缺省值：1] 轮询操作状态的次数
-i，--id <arg>	必需：要轮询的操作标识
-t，--timeout <arg>	[缺省值：1] 连续轮询之间的时间间隔（以秒为单位）。

方式：清除

用法：bin/runBigFixDiscovery.sh -c/--cleanup [-d <arg>] [-h]

其中，

表 63.

-h，--help	显示帮助。
-d，--days <arg>	[缺省值：5] 清除超过指定的天数的结果文件

方式：重新发现

用法：bin/runBigFixDiscovery.sh -r/--rediscover [-freq <arg>] [-h] [-intr <arg>]

其中，

表 64.

-freq，--frequency <arg>	[缺省值：1] 需要运行发现的次数

表 64. (续)

-h，--help	显示帮助。
-intr，--interval <arg>	[缺省值：P1D] 重新运行发现之间的时间间隔。受支持的值：[PT15M、PT30M、PT1H、PT2H、PT4H、PT6H、PT8H、PT12H、P1D、P2D、P3D、P5D、P7D、P15D 和 P30D]。

5. TADDM - 结果访存/处理线程

表 65.

属性名称	可能的值	描述	必需
com.collation.bigfix.result.wait	<value in sec> 缺省值 = 60 秒	在启用 "com.collation.bigfix.enabled" 时，"结果访存线程"将衍生为定期从 BigFix 服务器访存发现结果文件。"结果访存线程"将根据配置的周期（以秒定义）从 BigFix 服务器访存结果程序包。	N
com.ibm.cdb.discover.asd.autodiscovery.enabled	True/false	如果为 true，那么支持线程处理存储的 ASD 结果文件	Y
com.ibm.cdb.discover.asd.ProcessUnreachableIPs	True/false	线程将针对不可访问的目标处理 ASD 结果。	Y
com.ibm.cdb.discover.asd.AsyncDiscoveryResultsDirectory	路径缺省值 = var/asdd	要保留结果文件的路径。路径可配置，但是缺省情况下设置为 var/asdd。	N
com.ibm.cdb.discover.asd.autodiscovery.asdScope	<scope name> 缺省值 = ASD	线程将选取此作用域中提及的目标以处理结果文件。如果未提及此属性，那么将处理缺省 ASD 作用域。	N
com.ibm.cdb.discover.asd.autodiscovery.asdProfile	<profile name> 缺省值 = ASD	线程将选取此概要文件中提及的传感器以处理结果文件。如果未提及此属性，那么将处理缺省 ASD 概要文件。	N
com.ibm.cdb.discover.asd.autodiscovery.filesThreshold	<file threshold> 缺省值 = 20	线程开始处理文件时所需的最大文件数。如果满足文件阈值或时间阈值，那么线程将处理结果。	N
com.ibm.cdb.discover.asd.autodiscovery.timeThreshold	<time threshold> 缺省值 = 60 秒	时间阈值，在此时间后线程将处理结果文件，即使未满足"文件阈值"。	N

注：1. BigFix 发现结果将异步到达 TADDM 服务器，并且在满足其中一个属性（com.ibm.cdb.discover.asd.autodiscovery.filesThreshold 或 com.ibm.cdb.discover.asd.autodiscovery.timeThreshold）时，将处理可用的结果文件群，并且将创建全新的"发现历史记录"条目。将根据特定需求微调这些属性以控制"发现历史记录"条目的数量。

6. 清除

表 66.

S.No	资源	TADDM 服务器		BES 根服务器		BES 端点	
		已创建	Cleanup	已创建	Cleanup	已创建	Cleanup
1.	请求程序包	Y	N[1]	Y	N[2]	Y	Y
2.	任务	-	-	Y	Y [4]	-	-
3.	操作	-	-	Y	Y [3]	-	-
4.	结果程序包	Y	Y	Y	Y	Y	Y
5.	文件集	-	-	-	-	Y	Y

注：

- 不支持 TADDM 服务器上的请求程序包清除
- 不支持 BES 根服务器上的请求程序包清除。（在重新发现期间可复用请求程序包）
- 仅针对由 TADDM 创建且处于"已到期"状态的操作考虑清除（使用名称 TADDMCLEANUP 创建的操作除外）。
- 将除去 TADDM 创建的任务，并且仅限已除去与此任务相关联的所有操作时。
- 将除去 TADDM 创建的任务，并且仅限已除去与此任务相关联的所有操作时。
  - 要从清除中排除特定任务及其关联的操作（针对重新发现支持），可根据以下详细信息使用 retainBigFixTask.sh/.bat：

  用法：./retainBigFixTask.sh <TaskName> <enable/disable>

TADDM 服务器上的清除

表 67.

属性名称	可能的值	描述	必需
com.collation.bigfix.taddm.cleanup.volume	<带后缀的限制大小>	<限制大小，在到达此大小后将除去旧的已处理文件，例如，50MB、2GB 等>	N
com.collation.bigfix.taddm.cleanup.time	<带后缀的限制时间>	<用于检查时间超过配置的单元的已处理文件，例如，1D、5H 和 30M 等>	N
com.collation.bigfix.taddm.cleanup.runtime	分钟数	TADDM 清除线程将在运行后等待配置的分钟数。	N

注：

- 将执行 TADDM 服务器上的结果文件清除，仅限至少了配置了一个属性（com.collation.bigfix.taddm.cleanup.volume 或 com.collation.bigfix.taddm.cleanup.time）时。

端点上的清除

表 *68.*

属性名称	可能的值	描述	必需
com.collation. bigfix.endpoint. cleanup	<Y or N><Default=Y>	在设置为 Y 时，将从端点除去请求程序包大小、抽取的请求包目录以及新创建的结果程序包 zip。	N

BigFix 根服务器上的清除

表 *69.*

属性名称	可能的值	描述	必需
com.collation. bigfix.root. cleanup.interval	<no of days> <Default=1>	定期运行清除任务以从 BES 根服务器除去结果程序包、任务和已到期的操作。	N
com.collation. bigfix.root. cleanup.days	<no of days> <Default=5>	超过给出天数的结果文件将被视为除去。	N

7. 定制相关性

表 *70.*

属性名称	可能的值	描述	必需
com.collation. bigfix.relevance. appendscope	true/false <Default=true>	如果为 true，那么除指定的作用域外，将使用定制相关性查询 如果为 false，那么仅使用定制相关性查询，代替指定的作用域	N
com.collation. bigfix.relevance	True/false	用于针对给定的发现标识一组端点的相关性查询。	N

8. Bigfix 程序包 Temp 路径设置

表 *71.*

属性名称	可能的值	描述	必需
com.collation. bigfix.action. enable.os	<action script for the configured os> <Default=Windows, AIX, Linux, SunOS>	Bigfix ActionScript 中将包含配置的操作系统的操作脚本	N
com.collation. bigfix.temp. Windows	请求程序包路径 <Default=C:\Windows\ Temp>	将用于 ASD 请求程序包的路径。*注：在 Windows 路径中，"\"需要指定为"\\"。	N

表 71. (续)

属性名称	可能的值	描述	必需
com.collation.asd.temp.Windows	结果程序包路径 <Default=C:\Windows\Temp>	将用于 ASD 结果程序包的路径。	N
com.collation.asd.temp.Unix	结果程序包路径 <Default=/tmp>	将用于 ASD 结果程序包的路径。	N
com.collation.bigfix.temp.Linux	结果程序包路径 <Default=/tmp>	将用于 ASD 请求程序包的路径。	N
com.collation.bigfix.temp.SunOS	结果程序包路径 <Default=/tmp>	将用于 ASD 请求程序包的路径。	N
com.collation.bigfix.temp.AIX	结果程序包路径 <Default=/tmp>	将用于 ASD 请求程序包的路径。	N

注：根据以上配置的 Temp 路径，如果不存在，那么将在目标端点上创建文件夹。例如，在 Windows 2003 中，将使用缺省 Temp 路径"C:\Windows\Temp\"，并且将在发现期间创建此文件夹。

附录 C. 检查脚本执行的示例

1. 执行脚本 - encryptprops.sh

```
/opt/IBM/taddm/dist/bin/encryptprops.sh $COLLATION_HOME
```

2. 执行脚本 - runBigFixDisocvery.sh

```
TADDM Server  – 9.167.42.227 (Linux)
BigFix server     – 10.160.161.195 (windows)
BigFix endpoints        – 10.160.161.196 (windows)
                          10.160.161.212 (windows)
Scope       – ASD (having both BigFix endpoints)
Profile           - ASD (having sensors mentioned in section 2.2)
Configuration     - done as per section 3.
```

a. 启动发现：

```
[taddmusr@nc042227 bin]$ ./runBigFixDiscovery.sh -d -o /tmp -p ASD -s ASD
BigFix Action will be applied total [1] times with [P1D] interval

Task created on BES server with Name [20170828083852] and Action created with ID [633]

DISCOVER: LAUNCH OK
The Bigfix Discovery script exited successfully.
```

b. 任务名称 - 20170828083852，操作标识 - 633

c. 启动轮询：

```
[taddmusr@nc042227 bin]$ ./runBigFixDiscovery.sh -p -i 633
Repeatedly poll the BigFix Action [1] number of times for every [1] seconds
Total [2] Computers returned for Action with ID [633] has status: Open
```

```
Total [1] computers with status : The action executed successfully.
[Hostname] [Apply Count] [Line Number] [Start Time] [End Time]
[PNC161196] [1] [98] [Mon, 28 Aug 2017 14:43:56 +0000] [Mon, 28 Aug 2017 14:44:11 +0000]

Total [1] computers with status : The action failed.
[Hostname] [Apply Count] [Line Number] [Start Time] [End Time]
[PRODUCTIONWASB] [1] [37] [Mon, 28 Aug 2017 07:40:26 +0000] [Mon, 28 Aug 2017 07:40:26 +0000]

POLL FINISHED
The Bigfix Discovery script exited successfully
```

启动清除：

```
[taddmusr@nc042227 bin]$ ./runBigFixDiscovery.sh -c
CLEANUP TASK FOUND: TADDMCLEANUP with ID: 2067

Cleanup Action created with ID: [2068]

CLEANUP: LAUNCH OK
The Bigfix Discovery script exited successfully
```

启动重新发现：

```
[taddmusr@nc042227 bin]$ ./runBigFixDiscovery.sh –r –i 20171117085907

TASK FOUND : 20171117085907 with ID : 2075

Action created with ID: [2086]

REDISCOVERY: LAUNCH OK
The Bigfix Discovery script exited successfully.
```

附录. 错误代码和描述

表 72.

消息标识	M：消息，C：原因，E：影响
CTJTD1260E	M：Bigfix 发现未启用，请在 collation.properties 中配置 com.collation.bigfix.enabled E：Bigfix 发现脚本将不会执行，并且将不会调用用于访存结果的线程
CTJTD1261E	M：缺少参数或者不正确 C：尝试使用发现、轮询、清除或重新发现之外的方式执行脚本 E：脚本将不执行
CTJTD1262E	M：提供的数字格式不正确 C：以字符串格式指定属性或自变量，但是需要数字格式 E：Bigfix 发现脚本将不执行，并且用于访存结果的线程无法正确运行。
CTJTD1263E	M：无法解析命令行属性 <property name> C：不支持运行脚本时传递的自变量 E：将不会调用脚本方式
CTJTD1264E	M：collation.properties 中缺少 <Property name> C：在执行脚本时，缺少必需的属性或无效（请检查附录 A） E：Bigfix 发现脚本将不执行，并且用于访存结果的线程无法正确运行。
CTJTD1265I	M：仅使用定制相关性，代替指定的作用域

表 72. (续)

消息标识	M：消息，C：原因，E：影响
CTJTD1266I	M：除指定的作用域外，将使用定制相关性
CTJTD1267E	M：指定了空作用域，找不到任何元素 C：给定的作用域/作用域组不包含任何元素来定义端点 E：将不会调用发现
CTJTD1268E	M：指定的概要文件中不存在或者未启用传感器 C：给定的概要文件不包含任何传感器 E：将不会调用发现
CTJTD1269E	M：AASD 请求软件包不存在 C：创建请求软件包时发生问题，或者没有权限或不存在而无法上载 E：将不会调用发现
CTJTD1270E	M：AASD 软件包大小超过配置的 com.collation.bigfix.aasdpkgmaxsize 阈值 C：创建的请求软件包大小超过配置的值 E：将不会调用发现
CTJTD1271E	M：无法连接到 BigFix，原因：<reason> C：由于无效参数/无效证书，发生 Bigfix Web Service 连接问题。 E：将不会上载软件包，将不会调用发现
CTJTD1272E	M：设置时捕获错误：<reason> C：意外场景，无法处理代码 E：脚本执行将不会正确运行
CTJTD1273I	M：主脚本（重新）运行 BigFix 发现，或者轮询指定的发现操作，或者执行手动清除

声明

本信息是为在美国国内供应的产品和服务而编写的。 IBM 可能在其他国家或地区不提供本文档中讨论的产品、服务或功能特性。 有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。 任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。 只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。 但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。 提供本文档并未授予用户使用这些专利的任何许可。 您可以用书面方式将许可查询寄往：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

有关双字节 (DBCS) 信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

本条款不适用于英国或任何这样的条款与当地法律不一致的国家或地区：

International Business Machines Corporation"按现状"提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。

某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。 此处的信息将定期更改；这些更改将编入本资料的新版本中。 IBM 可以随时对本资料中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。 那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：(i) 允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及 (ii) 允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。 因此，在其他操作环境中获得的数据可能会有明显的不同。 有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。 此外，有些测量是通过推算而估计的， 实际结果可能会有差异。 本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。 IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。 有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息包含在日常业务操作中使用的数据和报告的示例。 为了尽可能完整地说明这些示例，示例中可能会包括个人、公司、品牌和产品的名称。 所有这些名字都是虚构的，若现实生活中实际业务企业使用的名字和地址与此相似，纯属巧合。

如果您正以软拷贝格式查看本信息，图片和彩色图例可能无法显示。

商标

IBM、IBM 徽标和 ibm.com 是 International Business Machines Corp. 在全球多个管辖区域中注册的商标或注册商标。 其他产品和服务名称可能是 IBM 或其他公司的商标。 Web 站点 http://www.ibm.com/legal/copytrade.shtml 上的"Copyright and trademark information"提供了 IBM 商标的最新列表。

Java 和所有基于 Java 的商标和徽标是 Oracle 和/或其附属公司的商标或注册商标。

Linux 是 Linus Torvalds 在美国和/或其他国家或地区的注册商标。

Microsoft 和 Windows 是 Microsoft Corporation 在美国和/或其他国家或地区的商标。

UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。

其他公司、产品和服务名称可能是其他公司的商标或服务标记。

Printed in China