Tivoli® Application Dependency Discovery Manager
Version 7 Release 2

# *Feature Guide for 7.2 Fix Pack 2*

IBM

Tivoli® Application Dependency Discovery Manager
Version 7 Release 2

# *Feature Guide for 7.2 Fix Pack 2*

IBM

**Note**

Before using this information and the product it supports, read the information in "Notices" on page 59.

**Edition notice**

This edition applies to version 7, release 2, modification 0 of IBM Tivoli Application Dependency Discovery Manager (product number 5724-N55) and to all subsequent releases and modifications until otherwise indicated in new editions.

# Contents

# About this information

This document provides the documentation updates for the features in the IBM® Tivoli® Application Dependency Discovery Manager (TADDM) Version 7 Release 2 Fix Pack 2.

This document supplements the information in the TADDM 7.2 information center at http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.2/welcome_page/welcome.html.

The sensor information center at http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddmsensors.doc_72/welcome_page/welcome.html contains information about the sensor updates that are delivered in TADDM 7.2 Sensor Interim Feature 1 and Fix Pack 2. This sensor information is more up-to-date than the sensor information in the TADDM 7.2 information center.

This TADDM fix pack includes the following new features:

Fix Pack 2 **Changes to application descriptors for ComputerSystem objects**

You can enable the automatic addition of ComputerSystem objects to business applications, through the use of application descriptors.

Fix Pack 2 **Updating the lifecycle state for business applications**

You can use the lifecycle state to filter objects for synchronization into IBM Tivoli Business Service Manager (TBSM) from TADDM. You can use the BusinessServiceLifecycle program to list information about a business service, or to set the lifecycle state of a business service.

TADDM Version 7 Release 2 Fix Pack 1 included the following features:

**Secure Product Console from TADDM Enterprise Domain Manager**

After you perform the appropriate configuration, you can launch a secure connection to the Product Console of any of the domains in the enterprise from the TADDM Enterprise Domain Manager.

**Configure different elevated privilege commands for different versions of Solaris**

When configuring the operating system commands that are used by TADDM to run on the target system, you can specify a particular Solaris version by appending the SunOS version number to the property name when setting the following properties:

- com.collation.platform.os.command.ps.SunOS
- com.collation.discover.agent.command.lsof.SunOS

**Externalization of the netstat command for Windows discoveries**

Using the com.collation.discover.agent.command.netstat.Windows property, you can specify a custom command to use instead of the netstat -nao command on a Windows® target when using the Generic server sensor.

**Launch in context for IBM Tivoli Monitoring using single sign-on**

Using launch in context for IBM Tivoli Monitoring using single sign-on, you can access the Query Home Page from IBM Tivoli Monitoring, without needing to provide a TADDM user name and password.

**Mark for comparison**

The Mark for Comparison feature extends the existing inventory component comparison functionality to the following reports:

- Custom query report
- Application summary report
- Services summary report

You can select two or more items in the report results and click **Mark for Comparison** to compare them.

You can use the Mark for Comparison functionality across multiple domains.

**Storage array information in Domain Manager topology graphs**

Information about the storage arrays that are used is displayed in the Domain Manager topology graphs.

**Predefined BIRT reports**

The following predefined BIRT reports have been added:

**Fibre Channel Network Report**

The Fibre Channel Network Report displays Fibre Channel connections between a selected Fibre Channel switch and other computer systems.

**Host Bus Adaptor Inventory Report**

The Host Bus Adaptor Inventory Report displays a list of all discovered host bus adaptors and the computer systems on which they are installed.

**Storage Array Consumers Report**

The Storage Array Consumers Report displays a list of the computer systems and application servers that use a specified storage array.

**Storage Arrays by Host Report**

The Storage Arrays by Host Report displays a list of the storage volumes and storage arrays that are used by a specified computer system.

**IBM Informix® sensor**

The IBM Informix sensor discovers IBM Informix Dynamic Servers.

**IBM Lotus® Domino® server sensor**

Documentation for the IBM Lotus Domino server sensor was updated to include a newer and more relevant configuration file example.

**Oracle sensor**

The Oracle sensor was updated to support queries to the v$sys_optimizer_env table and the discovery of DB link information.

**WebLogic sensor**

Documentation for the WebLogic sensor was updated to include newer and more relevant configuration file examples.

# Changes to application descriptors for ComputerSystem objects

Fix Pack 2 The following information has been added or updated for this feature.

## Application descriptors

You can use application descriptors to associate components to business applications and specify further details about the applications.

### Application descriptors overview

IBM Application descriptors provide complete automation of the process of discovering, creating, and maintaining business applications and their composition.

An application descriptor is an application tag that maps a module, server, or computer system to a business application. By using application descriptors, you can identify a component of a business application at development time. When application descriptors are discovered, they are used to automatically associate components with business applications, thus eliminating manual modeling and maintenance of business application compositions.

An application descriptor is an XML file placed in a specified location that specifies modules, servers (containers), or computer systems, and associates them with business applications. You can map multiple modules at one time or map an entire container (such as an IBM WebSphere® server).

The `TopologyBuilder.log` file contains log messages relevant to the processing of application descriptor files.

You can use the following strategies for creating and deploying application descriptors:

**During deployment**

Application definition during development and deployment is the recommended approach. With this approach, you can capture the most accurate and complete information about the packaging of modules in business applications.

**After deployment**

You can add application descriptors for deployed modules after initial deployment, by creating the descriptors and then deploying them to the file system on the target computer.

There are two types of application descriptors:

**Base application descriptor**

Contains general information about an application. The base application descriptor is optional.

**Component application descriptor**

Contains information about a specific computer system, application component, or module deployed within a server.

You must assign a unique application name in both the base application descriptor and the component application descriptor. This unique name is used to correlate all discovered application descriptors for a specific application.

## Base application descriptor

The base application descriptor contains general information about the application, such as the version, contact, and other information.

Because the base application descriptor contains general information, it is not required in order to discover an application.

**Important:** Business applications are created automatically even without a base application descriptor file as long as a component application descriptor file is provided and contains an app-instance-name tag in the file. The name that is used for the business application is the app-instance-name tag from the component application descriptor file.

You only need a single base application descriptor for each application. In cases when more than one descriptor is used, the system uses the one with the most recent time stamp.

The base application descriptor can be deployed to the descriptor directory of any one of the components of the application.

The following table describes the structure of the base application descriptor:

*Table 1. Base Application Descriptor elements and attributes*

| **Element** | **Description and attributes** | |
|---|---|---|
| base-app-descriptor | The root element for the base application descriptor. | |
| app-instance | The element for the application instance information. | |
| | name | (Required) The name of the application instance |
| | version | The application version |
| | description | A description of the application instance |
| | url | The URL pointing to the application |
| | contact | A contact name or other information for the application (This is not imported into TADDM.) |
| app-definition | The element for the application definition information. | |
| | name | (Required) The name of the application definition |
| | description | A description of the application instance |

The following XML snippet shows an example of the base application descriptor:

```
<base-app-descriptor>
   <app-instance
      name="application_name"
      version="application_version"
      description="application_description"
      url="application_url"
      contact="contact_name"/>
   <app-definition
      name="definition_name"
      description="definition_description"/>
</base-app-descriptor>
```

## Component application descriptor

The component application descriptor contains information about a specific computer system, server, or module deployed within a server, along with information about the participation of the component within the application.

Components can include computer systems, database servers, J2EE servers, or modules within servers. You can use a separate descriptor for each module, or a single descriptor for all modules within a server.

A component application descriptor must be deployed in the descriptor directory of each server that is a component of the business application, or contains modules that are components of the business application. Use component application descriptors instead of the WebSphere or Weblogic sensors to add the J2EE components to a business application for a finer granularity to the discovered dependencies. For more details, see Best Practices for Discovering Business Applications on the TADDM wiki at: http://www.ibm.com/developerworks/wikis/display/tivoliaddm/Best+Practices+for+Discovering+Business+Applications

An application descriptor is an XML with the following format:

```
<component-app-descriptor
   app-instance-name="instance_name">
   <component-descriptor
      type="component_type"
      name="component_name"
      functional-group="group_name"
      marker-module="true|false"/>
</component-app-descriptor>
```

The elements and attributes of the component application descriptor file are as follows:

*Table 2. Component application descriptor elements and attributes*

| Component descriptor element | Description and attributes | |
|---|---|---|
| component-app-descriptor | The root element for the component application descriptor. | |
| | app-instance-name | (Required) The name of the application instance. |

Table 2. Component application descriptor elements and attributes (continued)

| Component descriptor element | Description and attributes | |
|---|---|---|
| component-descriptor | (Required) The element for the component information. | |
| | type | (Required) A component descriptor can apply to a computer system (host), a server in its entirety, or to individual modules within a server. The type attribute specifies this relationship, and can assume either of the following values:<br>• host<br>• module<br>• server |
| | name | (Required) The name of the component. |
| | functional-group | (Required) The functional group that the component occupies within the application. Functional groups allow for the grouping of similar components within the application. They are used to compare applications to each other. |
| | marker-module | (Optional) A special type of module definition for J2EE domains. When a module is indicated as a marker module, J2EE-managed servers within the domain that include the marker module are treated as having all of its modules included in the application. You can specify the following values as the marker module:<br>• true<br>• false |

## Application descriptor locations

The location of the directory containing application descriptors depends upon the component type and the system configuration.

An application descriptor file is an XML placed in a specific location that depends upon the type of application descriptor. The file name is not significant, but it must end in the `.xml` extension.

**Note:** Make sure that the TADDM service account has access to the location of any application descriptor you want to use.

During discovery, the TADDM server checks for application descriptors as follows:

- For base application descriptors, the application descriptor file can be placed in the application descriptor directory of any component that is part of the application.
- For computer system (host) application descriptors, the application descriptor directory is specified by the **com.collation.platform.os.hostappdescriptorfiles.dir** parameter in the collation.properties configuration file on the TADDM server, as in the following example:

  ```
  com.collation.platform.os.hostappdescriptorfiles.dir="/home/taddm/hostappdesc"
  ```

  A host application descriptor file must contain only the host application descriptor. Do not combine a host application descriptor with any other application descriptors in the same file.

- For module and server application descriptors, the application descriptor directory is a subdirectory named `appdescriptors` in one of the following locations (listed in order of priority):
  1. The custom path specified by the COLL_APP_DESC_DIR environment variable (if set)
  2. The custom path specified by the COLL_APP_DESC_DIR command-line argument (if set)
  3. The default application descriptor location as specified in Table 3, if no custom path is specified.

*Table 3. Default Application Descriptor Locations*

| Server | Supported modules | Default directory and supported modules |
|---|---|---|
| WebSphere Application Server 5.1 | J2EE applications, Web, EJB, and connector modules | *WebSphere_home_dir*/appdescriptors |
| WebSphere Application Server version 6 | J2EE applications, Web, EJB, and connector modules | *WebSphere_home_dir*/profiles/*profile_name*/appdescriptors, where *profile_name* is what you configured for your WebSphere Application Server cell. The following example uses the `default` profile:<br>/opt/IBM/WebSphere/profiles/default/appdescriptors |
| WebLogic | J2EE applications, Web, EJB, and connector modules | *WebLogic_home_dir*/appdescriptors |
| JBoss | J2EE applications, Web, EJB, and connector modules | *JBoss_home_dir*/appdescriptors |
| IPlanet | Servlets, JSP pages | *IPlanet_home_dir*/appdescriptors |
| Apache | | *Apache_home_dir*/appdescriptors |
| Microsoft® IIS | Virtual hosts | *IIS_home_dir*/appdescriptors |
| Oracle | Users | *Oracle_home_dir*/instance_name/appdescriptors<br>**Note:** You must create the `instance_name` for this location. |
| Sybase/ Sybase IQ | Databases | *Sybase_home_dir*/appdescriptors |
| SQLServer | Databases | *SQLServer_home_dir*/appdescriptors |
| DB2® | | *DB2_home_dir*/instance_name/appdescriptors |
| Domino Server | | *Domino_server_home_dir*/appdescriptors |
| Microsoft Exchange Server 2003 | Exchange Servers, Exchange Protocol Virtual Servers | *exchange_server_home_dir*/appdescriptors |
| Custom Server | User-supplied through template definition | User-supplied through template definition |
| Veritas Cluster | | *VS_home_dir*/appdescriptors |

In the case of managed servers such as J2EE servers, which are managed by the J2EE domain, the location of the application descriptor directory is at the level of the Admin Server or Domain Manager. The contents specified in that directory are used as the superset of all possible mappings for all managed servers. For each managed server (depending on which modules are discovered as deployed), the application descriptor is processed for inclusion of those modules in the application.

## Application descriptors example

You can view application information for a sample application descriptor.

The following table describes details of a sample application:

*Table 4. Application description*

| | **Detail** |
|---|---|
| Application | Order Management |
| Instance | Staging |
| Containers | • Three Apache servers<br>• Two WebLogic servers (managed in one domain)<br>• One custom server (Order Fulfillment gateway)<br>• One custom Java™ process (Automailer)<br>• One Oracle Instance |
| Modules | • Static content<br>• WAR file<br>• EAR file<br>• RAR file (communicates with the gateway module)<br>• Virtual gateway module<br>• automailer jar<br>• DB schema |
| Computer systems | • Three Web server systems<br>• Two application server systems<br>• One Order Fulfillment gateway server system<br>• One Oracle database server system |

The base application descriptor for the sample application is stored in the `oms_coll_desc.xml` file, placed in the application descriptor directory of any application component:

```
<base-app-descriptor>
   <app-instance
         name="Order Management-Staging"
         version="1.5.1"
         description="Order Entry application - staging"
         url="http://orderentry.stage.lab.com"
         contact="John Public" />
   <app-definition
      name="Order Management"
      description="Order Entry and Tracking application" />
</base-app-descriptor>
```

The component application descriptor for the Apache server is stored in the `apache_coll_desc.xml` file in the *Apache_server_home_dir*/appdescriptors directory on the Apache server. Additional application descriptors are present on each Apache server that is part of the Order Management application:

```
<component-app-descriptor
   app-instance-name="Order Management-Staging">
   <component-descriptor
```

```
      type="module"
      name="/opt/apache13/htdocs/ordermgt/"
      functional-group="Web Tier"
      marker-module="false" />
</component-app-descriptor>
```

The component application descriptor for the computer system hosting the Apache server is stored in the `apache_host_coll_desc.xml` file in the location specified by the **com.collation.platform.os.hostappdescriptorfiles.dir** parameter in the collation.properties configuration file. Additional host application descriptors are present on each computer system included in the business application:

```
<component-app-descriptor
   app-instance-name="Order Management-Staging">
   <component-descriptor
      type="host"
      name="staging.example.com"
      functional-group="Web Tier"
      marker-module="true" />
</component-app-descriptor>
```

The WebLogic component application descriptor is stored in the `wls_coll_desc.xml` file in the *`WebLogic_home_dir`*`/appdescriptors` directory:

```
<component-app-descriptor app-instance-name="Order Management-Staging">
   <!-- order.war -->
   <component-descriptor
      type="module"
      name="WebLogicWebModule:order"
      functional-group="Order Processing Server"
      marker-module="false" />
   <!-- orderejb.ear -->
   <component-descriptor
      type="module"
      name="WebLogicWebModule:orderejb"
      functional-group="Order Processing Server"
      marker-module="false" />
   <!-- ofg.rar -->
   <component-descriptor
      type="module"
      name="WebLogicWebModule:ofg"
      functional-group="Order Processing Server"
      marker-module="false" />
</component-app-descriptor>
```

The Order Fulfillment Gateway component application descriptor is stored in the `ofg_coll_desc.xml` file in the directory specified in the custom server template:

```
<component-app-descriptor app-instance-name="Order Management-Staging">
   <component-descriptor
      type="server"
      name="n/a"
      functional-group="Order Fulfillment Gateway"
      marker-module="false" />
</component-app-descriptor>
```

The component application descriptor for the automailer Java process is stored in the `am_coll_desc.xml` file in the directory specified in the custom server template:

```
<component-app-descriptor app-instance-name="Order Management-Staging">
   <component-descriptor
      type="module"
      name="automailer.jar"
      functional-group="Order Processing Automailer"
      marker-module="false" />
   <component-descriptor
      type="module"
      name="login.jar"
      functional-group="Order Processing Automailer"
      marker-module="false" />
</component-app-descriptor>
```

The component application descriptor for the Oracle database schema is stored in the `ora_coll_desc.xml` file in the `$ORACLE_HOME/appdescriptors` directory:

```
<component-app-descriptor app-instance-name="Order Management-Staging">
   <component-descriptor
      type="module"
      name="ORDER"
      functional-group="Order Processing DB"
      marker-module="false" />
   <component-descriptor
      type="module"
      name="OFG"
      functional-group="Order Processing DB"
      marker-module="false" />
   <component-descriptor
      type="module"
      name="ADMIN"
      functional-group="Order Processing DB"
      marker-module="false" />
</component-app-descriptor>
```

## Discovery settings

You can edit the properties for discovery. When you modify the file, you must save the file and restart the server for the change to take effect.

The following list identifies extra details for the properties for discovery:

**com.collation.discover.agent.exchange.command.timeout=***600000*
Specifies the timeout for the Exchange Server sensor. The default timeout in milliseconds is *600000*. If you change the default, make sure that you specify an integer.

**com.collation.discover.anchor.forceDeployment=***true*
Specifies if all anchors for the discovered scope are to be deployed during discovery startup. Valid values are *true* and *false*. The default is *true*. If you change the default to *false*, anchors are deployed only if any IP address from the scope cannot be pinged, or if port 22 cannot be reached on any of the discovered IP addresses.

**com.collation.discover.anchor.lazyDeployment=***false*
Specifies when data is copied, during an anchor deployment, or when the sensor requiring the files is about to start. For both cases files are copied. Valid values are *true* and *false*. The default is *false*.

For example, the WebSphere Application Server sensor has dependencies in the `dist/lib/websphere` directory. The size of the directory is 130 MB. If the flag is set to *false*, this data is copied to the target host when the anchor is deployed. If the flag is set to *true*, the data is copied when the WebSphere Application Server sensor is about to be run on the anchor. If no WebSphere Application Server sensor is run through the anchor, 130 MB is not sent to the remote host.

**com.collation.discover.DefaultAgentTimeout=***600000*
Specifies the default timeout for sensors in milliseconds, which is 10 minutes.

The default for all sensors is 10 minutes. The default can be changed. It can also be specified by individual sensors.

To override the timeout for a particular sensor, add the following line to the `collation.properties` file:

```
com.collation.discover.agent.<sensorName>Agent.timeout=
  <timeInMilliseconds>
```

For example,

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

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

This property defines how IP addresses discovered during a Level 1 discovery are assigned to generated subnets. A Level 1 discovery does not discover subnets; instead, IpNetwork objects are generated to contain any interfaces that are not associated with an existing subnet discovered during a Level 2 or Level 3 discovery. This configuration property defines which IpNetwork objects are created, and how many nodes each subnet contains. (It also applies to any interface discovered during a Level 2 or Level 3 discovery that for any reason cannot be assigned to a discovered subnet.)

The value for this property consists of a single line containing one or more entries separated by commas. Each entry describes an IP address range in IPv4 dotted decimal format, along with a subnet mask specified as an integer 8 - 31. Discovered interfaces in the specified range are then placed in created subnets no larger than the size specified by the subnet mask.

For example, the following value defines two subnet address ranges with different subnet masks:

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

The specified address ranges can overlap. If a discovered IP address matches more than one defined range, it is assigned to the first matching subnet as they are listed in the property value.

After you create or change this configuration property and restart the TADDM server, any subsequent Level 1 discoveries use the defined subnets. To reassign existing IpInterface objects in the TADDM database, go to the `$COLLATION_HOME/bin` directory and run one of the following commands:

- adjustL1Networks.sh (Linux® and UNIX® systems)
- adjustL1Networks.bat (Windows systems)

**Note:** If the value is not specified correctly then the appropriate messages are displayed only when running the command-line utility adjustL1Networks.sh (Linux and UNIX systems) or adjustL1Networks.bat (Windows systems). Otherwise the messages are placed in the `TopologyBuilder.log` file in the `$COLLATION_HOME/log/services` directory.

This script reassigns all IpInterface objects discovered during Level 1 discoveries to the appropriate subnets as described in the configuration property. Any generated IpNetwork object that contains no interfaces is then deleted from the database. After the script is completed, the TADDM interface might show multiple notifications of changed components because of the modified objects. You can clear these notifications by refreshing the window.

**Note:** Before you use this command, make sure that the TADDM server is running, and that no discovery or bulk load operation is currently in progress. This script is not supported on the Enterprise Domain Server.

**com.collation.rediscoveryEnabled=***false*
: Valid values are *true* and *false*. The default value is *false*. Change the value to *true* to enable the rediscovery function. In addition to enabling the rediscovery function, setting the property to *true* also ensures that information is stored during the rediscovery.

**com.collation.ChangeManager.port=***19431*
: Specifies the firewall port used by the change manager.

**com.collation.platform.os.hostappdescriptorfiles.dir="***path***"**
: Specifies the fully qualified path to the directory where component application descriptor files for computer systems (hosts) are deployed. This property is required if you want to add computer systems to business applications using application descriptors. You can scope this property to a specific host name or IP address in order to specify a different location for each host. The following examples show how to specify the host application descriptor path:

- Linux and UNIX systems: `/home/taddm/hostappdescriptors`
- Windows systems: `c://taddm//hostappdescriptors`

## Extending custom servers

You can create a custom server template for an application to categorize the application and subsequently display it as part of the topology. You can also view details about the application, including the listening port, runtime information, and any configuration files or application descriptors that were collected.

### About this task

In some cases, however, this might not be sufficient. For example, you might also need to access the product version. By default, TADDM cannot collect version information for arbitrary custom server applications.

You can, however, extend custom server templates to collect additional information, as required, using the following approaches:

- Run commands on the target system to populate any attribute in the IBM model for the component.

  You can use this approach to set the productVersion attribute, for example.
- Run commands on the target system and store the result as a Config file for the component.

  One common use of this approach is to extract information from the Windows Registry.
- Run a Jython script on the server.

  You can change any information about a component. The difference between this and the first approach is where the code runs.

### Procedure

To define a custom server, complete the following steps:

1. Launch the IBM Product Console.

2. Create a custom server template for the application by completing the following steps:
   a. Click **Discovery** → **Custom Servers** in the sidebar.
   b. Click **Add** to define a new custom server template.
   c. Configure the general information and criteria for the custom server template.
   d. Configure the custom server configuration files. You can configure the capture of the following types of files:
      - configuration files
      - application descriptors
      - software modules

      For more information on application descriptors, see "Application descriptors" on page 1. Software modules represent deployed application modules such as executable code or scripts that are running inside the custom server. This is an optional level of detail that you can add to the discovery of your custom server. If applicable, you can include these software modules in Business Applications or Business Services for a higher level of visibility into the composition of your Business Applications or Business Services.
3. Create a directive file that contains the commands to run, and add commands and scripts to the directive file, as required.

   Use the format described in Table 5 on page 12 when specifying commands in the directive file. Table 6 on page 13 outlines the environment variables that you can use in the directive file.

   Keep commands as simple as possible. If the command stops during execution, the sensor will time out and the component is not discovered.
4. Save the directive file.

   The directive file must have the same name as the custom server template, and must be stored in the following directory: `$COLLATION_HOME/etc/templates/commands`. TADDM triggers directives in this directory using the name of the custom server template.

## What to do next

Table 5 describes the command format for directive files.

*Table 5. Directive file format*

| Directive | Description |
|---|---|
| CMD:variable= path/command | You can define an inline command. For example:<br>`CMD:productVersion=/usr/sbin/postconf \|awk '/^mail_version/ {print $3}'`<br>You must always specify absolute paths to commands, and you must double quotation marks (") commands or arguments containing spaces.<br>You can use environment variables associated with the process, specified by $VARIABLE$. For example,<br>`CMD:productVersion=grep versionNum $TOMCAT_HOME$/config/config.props \| awk '{print $2}'` |
| CMD:NOP= path/command | You can run the command without assigning results to a variable. For example:<br>`CMD:NOP=reg export HKLM\Software\Microsoft\InetStp c:\windows\temp\iis.reg /y` |
| CMD:CONFCONTENT. filename= path/command | You can run a command and store the results in the custom configuration file specified by filename. For example:<br>`CMD:CONFCONTENT.iisREG=cmd.exe /c type c:\windows\temp\iis.reg`<br>For more information, see the section on executing commands to create a custom configuration file. |
| SCRIPT: path/script | You can initiate Jython (.py) scripts. For example:<br>`SCRIPT:path/command.py`<br>When the path starts with (/) TADDM assumes an absolute path; otherwise the path is relative from `$COLLATION_HOME`. For more information, see the section on executing Jython scripts. |

Table 6 on page 13 describes the environment variables available for use in directive files.

*Table 6. Directive file environment variables*

| Variable | Description |
|---|---|
| $COLL_PROGPATH$ | This variable expands to the name of the directory where the program is located. For example, if the command line is `/usr/local/bin/foobar -c /etc/foobar.conf`, the $COLL_PROGPATH$ variable expands to `/usr/local/bin`.<br>You can use this variable to insulate your directive file in cases when a command is located in different directories on multiple computers. |
| $COLL_PROGNAME$ | This variable expands to the fully qualified executable name. For example, if the command line is `/usr/local/bin/foobar -c /etc/foobar.conf`, the $COLL_PROGNAME$ variable expands to `/usr/local/bin/foobar`. You can use this variable in cases when you write a custom server template that matches an argument on the command line, but the actual command might be Java or JRE. To run the appropriate command, you can use $COLL_PROGPATH$/ $COLL_PROGNAME$. |
| $COLL_CMDLINE$ | This variable expands to the entire command line, including any arguments. For example, if the command line is `/usr/local/bin/foobar -c /etc/foobar.conf`, the $COLL_CMDLINE$ variable expands to `/usr/local/bin/foobar -c /etc/foobar.conf`.<br>You can use this variable to find the version of the secure shell daemon (sshd) running on a system without having to know where it is installed, using the following command:<br>`CMD:productVersion=$COLL_PROGPATH$/ sshd -V 2>&1 \|awk '/version/ {print $3}'` |

# Updating the lifecycle state for business applications

Fix Pack 2 The following information has been added or updated for this feature.

## Integration with IBM Tivoli Business Service Manager

Depending on the specific tasks that you must do in your IT environment, you can use the integration capabilities that are available between the IBM Tivoli Application Dependency Discovery Manager (TADDM) and IBM Tivoli Business Service Manager (TBSM).

### Generating explicit relationship information on discovered resources

To generate explicit relationship information on discovered resources, run the `explicitrel.sh` script or call the generateExplicitRelationships API. Using either the script or calling the API can take a long time.

The `explicitrel.sh` script is located in the following directory:

- For Linux, Solaris, AIX®, and Linux on System z® operating systems: `$COLLATION_HOME/bin`
- For Windows operating systems: `%COLLATION_HOME%\bin`

The `explicitrel.sh` script takes one optional parameter. There are three options for specifying the parameter:

- If the parameter is not supplied, the program runs in delta mode. In delta mode, explicit relationships are created only from the data that was added since the last time the program ran.
- If the parameter supplied is *0*, the program runs in full refresh mode. In full refresh mode, each time the program runs, it deletes all explicit relationships and creates new instances of the explicit relationships in the database.
- If the parameter supplied is *1*, the program runs in delta mode. This mode is the same as providing no parameter.

There are two APIs:

1. generateExplicitRelationships()
   - This API defaults to the delta operation, as if the parameter supplied is *true*.
2. generateExplicitRelationships(boolean deltaGen)
   - If *true* is passed in, a delta operation is performed, as if the parameter supplied is *1*.
   - If *false* is passed in, a full refresh operation is performed, as if the parameter supplied is *0*.

If you need to generate explicit relationship data, but do not need to call methods programmatically, you can run the included `explicitrel.sh (explicitrel.bat)` script from the command line.

### Updating the lifecycle state for business applications

You can use the lifecycle state to filter objects for synchronization into IBM Tivoli Business Service Manager (TBSM) from TADDM. You can use the

BusinessServiceLifecycle program to list information about a business service, or to set the lifecycle state of a business service.

The BusinessServiceLifecycle program is in the following location:

- For Linux and UNIX operating systems, the `BusinessServiceLifecycle` script is in the `$COLLATION_HOME/bin` directory.
- For Windows operating systems, the `BusinessServiceLifecycle.bat` batch file is in the `%COLLATION_HOME%\bin` folder.

Use the BusinessServiceLifecycle program with the following command-line options:

```
BusinessServiceLifecycle -u TADDM_username -p TADDM_password -l | -s guid state
```

Use the `-l` option to list business service lifecycle information, or use the `-s` option, along with a guid parameter and a state code parameter, to set a lifecycle state. You cannot use the `-l` option and the `-s` option at the same time.

The following table lists the valid state codes:

*Table 7. State codes*

| Code | State |
|---|---|
| 0 | Unknown |
| 1 | Other |
| 2 | Ordered |
| 3 | Received |
| 4 | In Test |
| 5 | Tested |
| 6 | Installed |
| 7 | Enabled |
| 8 | Disabled |
| 9 | Maintenance |
| 10 | Retired |
| 11 | Archived |
| 12 | Accepted |
| 13 | Build |
| 14 | Development |
| 15 | Draft |
| 16 | Inventory |
| 17 | Offline |
| 18 | Postproduction |
| 19 | Production |
| 20 | Production Ready |
| 21 | Sunset |
| 22 | Validate |

# Secure Product Console from TADDM Enterprise Domain Manager

The following information has been added or updated for this feature.

## Starting a Product Console in secure mode

You can launch a secure connection to the Product Console of any of the domains in the enterprise from the TADDM Enterprise Domain Manager, once they have been configured to do so.

### Procedure

To configure SSL connection settings and start a Product Console in secure mode for a domain in your enterprise, complete the following steps:

1. Click **Domain Management** → **Domain Summary** . The Domain Summary pane displayed.
2. In the Domain Summary pane, select the domain for which you want to start the Product Console in secure mode.
3. Optional: If necessary, configure the SSL connection settings. You must complete this step in either of the following situations:
   - The SSL connection settings have not been configured. This step must be completed before you can start the Product Console in secure mode. (If you have not yet configured the SSL connection settings, the **Start in Secure Mode** button is not enabled.)
   - The list of managed domains has changed. In this situation, you must repeat the SSL connection settings configuration to update the truststore (especially if any domains have been added).

   To configure the SSL settings, complete the following steps:

   a. Click **SSL Connection Settings**. The SSL Connection Settings window is displayed.

   b. Click **Download Trust Store**. Take note of the directory to which you save the truststore file.

      **Important:** Do not change the name of the truststore file.
      The truststore file is downloaded using a secure connection.

   c. In the **Directory for the trust store** field, type the directory to which you saved the truststore file, without the trailing path separator. For example, if you saved the truststore file as

      ```
      C:\domain_certs\Domain.cert
      ```

      enter the directory for the truststore as

      ```
      C:\domain_certs
      ```

   d. Click **OK**. The truststore file information is saved in a browser cookie and is used when you start a secure connection.

   To disable non-secure connections and force the use of SSL connections, set **com.collation.security.enforceSSL** to *true*. The default value for this property is *false*.

4. Optional: If you are using Firefox version 3.0 or later, make sure the TLS 1.0 encryption protocol is enabled in your browser settings. To enable TLS 1.0 in Firefox 3.6:
   a. Click **Tools** → **Options**.
   b. In the Options window, click **Advanced**.
   c. Click the Encryption tab.
   d. Select **Use TLS 1.0**.
5. In the Domain Summary pane, ensure that you have selected the domain for which you want to start the Product Console in secure mode.
6. Click **Start in Secure Mode**. If you are prompted to accept a security certificate, you must do so. You are asked to accept a security certificate only the first time that you connect to the domain. Your browser attempts to open `confignia.jnlp` .

   **Note:** If the **Start in Secure Mode** button is not enabled, you must configure the SSL connection settings as described in step 3 on page 17.
7. If your browser prompts you to specify how you want to use `confignia.jnlp`, do one of the following:
   a. Open `confignia.jnlp` with Java Web Start 5.0, or later.
   b. Save the `confignia.jnlp` file locally. To launch the Product Console at a later time, open `confignia.jnlp` with Java Web Start 5.0, or later.

   The Product Console for the selected domain is initialized using SSL, and displayed.

## Starting a Product Console

Although there is no Product Console for Enterprise Domain Server, you can access the Product Console of any of the domains in the enterprise from the TADDM Enterprise Domain Manager.

### Procedure

To start a Product Console for a domain in your enterprise, complete the following steps:

1. Optional: If you are using Firefox version 3.0 or later, make sure the TLS 1.0 encryption protocol is enabled in your browser settings. To enable TLS 1.0 in Firefox 3.6:
   a. Click **Tools** → **Options**.
   b. In the Options window, click **Advanced**.
   c. Click the Encryption tab.
   d. Select **Use TLS 1.0**.
2. Click **Domain Management** → **Domain Summary** . The Domain Summary pane is displayed.
3. Ensure that the domain for which you want to access the Product Console has an open padlock icon alongside the domain name, specifying that it accepts unsecured connections.
4. Select the domain for which you want to start the Product Console.
5. Click **Start**. The Product Console for the selected domain is displayed.

## Domain Summary pane

The Domain Summary pane in the TADDM Enterprise Domain Manager contains information on the domains in your environment. From this pane, you can perform various operations on the domains.

The Distributed Domain Summary section contains the following buttons:

**New** Adds a domain to your enterprise.

**Edit** Edits the selected domain in your enterprise.

**Delete** Deletes the selected domain from your enterprise.

**Refresh**
Updates the Domain Summary table information for the selected domain.

**Start** Starts a Product Console for a domain in your enterprise.

**Start in Secure Mode**
Starts a Product Console for a domain in your enterprise using a secure SSL connection.

**SSL Connection Settings**
Displays the SSL connection settings.

The Domain Summary pane contains a table with the following fields:

**Domain**
Name of this domain.

**Host Name**
Name of the host for this domain.

**Last Synchronized**
The time of the last synchronization for this domain.

**Domain Status**
Status of the host.

## Add Domain and Edit Domain workspaces

You can use the Add Domain and Edit Domain workspaces in the TADDM Enterprise Domain Manager to work with or change the domains that make up your enterprise. You can use these workspaces to add a domain to your enterprise or change an existing domain.

The Add Domain and Edit Domain panes contain the following sections:

- Domain Details: Use this section to enter information describing the domain that you are adding or changing.
- Admin Details: Use this section to enter information about the contacts for this domain.

**Important:** To add a domain or change an existing domain, you must log in to the TADDM Enterprise Domain Manager as a user that has the Admin runtime permission.

The Domain Details section of the Add Domain and Edit Domain panes contains the following fields:

**Domain Name**
(Required) The name of the domain.

**Server Address**
(Required) The fully qualified host name or IP address of the TADDM server.

**Listening Port**
(Required) The listening port of the TADDM database. Use the unicast discovery port of the domain. To obtain the unicast discovery port, use the value of the com.collation.jini.unicastdiscoveryport property from the domain server in the dist/etc/collation.properties file. The default value is *4160*.

In the Edit Domain pane, the fields are completed with current values.

The Admin Details section of the Add Domain and Edit Domain panes contains the following fields:

**Name** The name of the domain administrator.

**Contact**
The contact for the domain.

**Escalation Contact**
The name of the escalation contact for the domain.

**Notes** User notes about the domain.

The Add Domain and the Edit Domain panes contain the following buttons:

**Add Domain**
(Add Domain pane only) Adds this domain.

**Save Changes**
(Edit Domain pane only) Saves the changed information.

**Apply** (Edit Domain pane only) Saves the changed information and returns to the Domain Summary pane.

**Cancel**
Returns to the Domain Summary pane without saving any information.

# Security problems

This information covers common problems that occur with security in the Tivoli Application Dependency Discovery Manager (TADDM).

## Errors when trying to log in with an SSL connection

**Problem**
When you log in to the Product Console with an SSL connection, you select the **Establish a secure (SSL) session** check box, but an SSL connection is not completed. An error message is displayed that states that the server is not running.

**Solution**
Ensure that you have downloaded the truststore and specified the location of the truststore file on the client system. To download the truststore, click **Show SSL Options** on the TADDM launch page and follow the displayed instructions. To use the truststore correctly, complete the following tasks:

- Make sure that the specified directory contains a valid truststore file, and that the truststore file has not been renamed.
- When specifying the location of the truststore file, do not include the file name.
- Ensure that the name of the directory that contains the truststore file does not have a trailing path separator at the end. For example, if you saved the truststore file as

  ```
  C:\domain_certs\Domain.cert
  ```

  enter the directory for the truststore as

  ```
  C:\domain_certs
  ```
- Ensure that the directory that you specify exists.

If the problem persists, delete the truststore file, and download it again.

# Configure different elevated privilege commands for different versions of Solaris

The following information has been added or updated for this feature.

## Commands that might require elevated privilege

These properties specify the operating system commands used by TADDM that might require elevated privilege, root or superuser, to run on the target system.

Typically, sudo is used on UNIX and Linux systems to provide privilege escalation. The following alternatives can be used instead of sudo:

- Enable the setuid access right on the target executable
- Add the discovery service account to the group associated with the target executable
- Use root for the discovery service account (not preferred)

For each property, sudo can be configured globally, meaning to run the command with sudo on every operating system target, or restricted to a specific IP address or scope set.

**Important:** On each target system for which privilege escalation is needed, sudo must be configured with the NOPASSWD option. Otherwise, your discovery hangs until sudo times out.

**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*

- These properties are needed to discover process information on Solaris systems.

  **Note:** You can specify a particular Solaris version by appending the SunOS version number to the property name. For example, the following property is specific to Solaris 10:

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

**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*

- These properties are needed to discover process/port information.

  **Note:** You can specify a particular Solaris version by appending the SunOS version number to the property name. For example, the following property is specific to Solaris 9:

  ```
  com.collation.discover.agent.command.lsof.SunOS5.9=sudo /usr/local/bin/lsof
  ```

# Externalization of the netstat command for Windows discoveries

The following information has been added or updated for this feature.

## Configuring the collation.properties file entries

This topic lists the collation.properties file entries that the sensor uses.

The sensor uses the following entries in the `collation.properties` file:

**com.collation.discover.agent.command.netstat.Windows**

You can use this property to specify a custom command to use instead of the `netstat -nao` command on a Windows target.

You must ensure that any alternative command you specify returns information in the same format as the `netstat -nao` command.

For example,

```
com.collation.discover.agent.command.netstat.Windows.ip_address=type c:\\\\folder\\\\mynetstat.txt
```

where `mynetstat.txt` contains the output of the `netstat -nao` command, and the `type` command is used to print the contents of the file.

# Launch in context for IBM Tivoli Monitoring using single sign-on

The following information has been added or updated for this feature.

## Specifying the URL to launch TADDM views

To launch TADDM views in context from other Tivoli applications, you must specify a URL.

The URL format for launching in context is:
`Protocol://TADDMHostname:TADDMPort/ContextRoot/?queryString`

The following list describes the valid values for each variable in the URL format:

*Protocol*
The Web protocol to use. Valid values are `http` or `https`.

*TADDMHostname*
The host name for the TADDM server to which you are launching.

*TADDMPort*
The port number for the TADDM server to which you are launching. The default value is `9430`.

*ContextRoot*
The following values are valid:

**`cdm/servlet/LICServlet`**
The relative path to the Java servlet that is deployed in the Apache Tomcat server.

**`cdm/queryHomePage.do`**
The relative path to the Query Home Page, when launched from IBM Tivoli Monitoring, using single sign-on, and specifying search text.

*queryString*
Contains name-value pair parameters that are delimited by separators. The format for a name-value pair is `name=value`. Use `=` to separate names and values, and use `&` to separate name-value pairs.

The following list describes the valid name-value pairs that can be used in the *queryString* variable:

**console**
Specifies whether to launch into the Product Console (a Java console) or the Domain Manager (a Web console).

If this parameter is not provided, the Product Console is launched.

The following string values are valid:

- `java`
- `web`

**target**
Specifies whether to launch a new or existing instance of the Product Console.

If this parameter is not provided, an existing Product Console is launched.

If `console=web` is specified, the target parameter is not applicable (does not have any effect).

The following string values are valid:

- `existing`
- `new`

**view**

Specifies that you want to display change history.

The only valid value is `changehistory`.

**days_previous**

Specifies the time period (the number of past days) for which to show the change history of a particular configuration item.

The valid value is a positive integer.

**guid**

Specifies the Globally Unique Identifier (GUID) for a configuration item.

If the graph parameter is specified with any of the following values, the guid parameter is optional:

- `businessapplications`
- `applicationinfrastructure`
- `physicalinfrastructure`

If the graph parameter is specified with any other type of topology graph, the guid parameter is required.

The valid value is a valid string representation of a GUID, as shown in the following example:

```
BA2842345F693855A3165A4B5F0D8BDE
```

You should specify only one GUID for each URL request for launch in context.

**graph**

Specifies the type of topology graph to be launched.

If you also specify a configuration item by providing its GUID on the guid parameter, the requested configuration item is then selected, if it is found in the topology graph that is specified on this graph parameter.

**Valid values regardless of whether the guid parameter is also specified:**

- `businessapplications`
- `applicationinfrastructure`, **except that this value is not valid for the Enterprise Domain Server**
- `physicalinfrastructure`

**Valid values only if the guid parameter is also specified:**

- For business application objects:
  - `app_software` for Business Application Software Topology
  - `app_physical` for Business Application Physical Topology
- For business service objects:
  - `bus_svc_software` for Business Service Software Topology
  - `bus_svc_physical` for Business Service Physical Topology

- For collection objects, except that these values are not valid for the Enterprise Domain Server:
  - `collection_relationship` for Collection Relationship Topology,
  - `collection_physical` for Collection Physical Topology

**username**

Specifies the user name used to log in to TADDM.

**password**

Specifies the password used to log in to TADDM.

**launchsource**

The only valid value is `ITM`. It is always used with the `searchtext=`*`search_term`* name-value pair.

The search is confined to the configuration items of type ComputerSystem and TMSAgent, listed in the `$COLLATION_HOME/etc/cdm/xml/itm_query_components.xml` configuration file.

From the Query Home Page results, for each configuration item listed, you can launch the following:

- Change History pane
- Details pane
- Product Console, displaying the Details pane

**searchtext**

Specifies the search term. It is always used with the `launchsource=ITM` name-value pair.

## Examples of how to specify the URL

The following examples show how to specify the URL to launch TADDM views:

**URL for launching the Product Console, specifying only a GUID**

```
http://home.taddm.com:9430/cdm/servlet/LICServlet?guid=BA2842345F693855A3165A4B5F0D8BDE
```

**URL for launching the Product Console, specifying only a graph name**

```
http://home.taddm.com:9430/cdm/servlet/LICServlet?graph=businessapplications
```

**URL for launching the Product Console, specifying a graph name with GUID**

```
http://home.taddm.com:9430/cdm/servlet/LICServlet?graph=app_software
  &guid=213l3jlk120bksdf
```

**URL for launching the Product Console to display the change history view, with the change history starting 20 days prior to the current date**

```
http://home.taddm.com:9430/cdm/servlet/LICServlet?guid=BA2842345F693855A3165A4B5F0D8BDE
 &view=changehistory&days_previous=20
```

**URL for launching the Domain Manager, specifying a graph name**

```
http://home.taddm.com:9430/cdm/servlet/LICServlet?console=web
  &graph=applicationinfrastructure
```

**URL for launching the Domain Manager, without entering authorization information separately**

```
http://home.taddm.com:9430/cdm/servlet/LICServlet?username=administrator
&password=adminpwd&console=web&guid=BA2842345F693855A3165A4B5F0D8BDE
```

**Important:** You must only use credentials as part of the URL for launching in context if you are using a trusted connection because the user name and password are not encrypted.

### URL for launching Query Home Page for IBM Tivoli Monitoring when using single sign-on and searching for a configuration item matching the search text

```
http://home.taddm.com:9430/cdm/queryHomePage.do?launchsource=itm&searchtext=127.0.0.1
```

# Mark for comparison

The following information has been added or updated for this feature.

## Application Summary pane

You can display applications summary information for the domains in your environment in the Domain Manager console.

The Application Summary pane contains the following information:

**Application Name**
The name of the business application.

**Groups**
Displays information associated with the functional groups in your environment.

**Changes**
Displays the change history for your business application.

**Details**
Displays details on your business application.

**Explore**
Displays the node centered topology for the application.

**Software Topology**
Displays the software topology for the application.

**Physical Topology**
Displays the physical topology for the application.

**Inventory**
Displays the inventory summary for the application.

**Mark For Comparison**
Adds the selected component to the list of components to be compared.

## Custom Query pane

You can manage custom query information in the Custom Query pane.

The Custom Query pane contains the **Saved Query** tab and depending on the tasks you are performing, it may contain some of the following additional tabs:

- New Query
- Edit Query
- Results

The **Saved Query** tab of the Custom Query pane contains the following information:

**New** Creates a custom query and adds it to the list of custom queries.

**Edit** Edits the attributes of the selected custom query.

**Copy** Creates a custom query based on the selected one.

**Delete** Deletes the selected custom query from the list of custom queries.

**Run Query**
Runs the selected custom query.

**Name** Displays the name of a custom query.

**Description**
Displays a description of a custom query.

The **Results** tab of the Custom Query pane contains the following information:

**Details**
Displays the details for the component.

**Explore**
Displays the node centered topology for the component.

**Changes**
Displays the change history for the component.

**Mark For Comparison**
Adds the selected component to the list of components to be compared.

**Save** Exports a report to a PDF, CSV or XML file.

## Services Summary pane

You can display a summary of the services for the domains in your enterprise in the Domain Manager console.

The Services Summary pane contains the following information:

**Service Name**
The name of the business service.

**Changes**
Displays the change history for your business service.

**Details**
Displays the details on your business service.

**Explore**
Displays the node centered topology for the business service.

**Software Topology**
Displays the software topology for the application.

**Physical Topology**
Displays the physical topology for the application.

**Inventory**
Displays the inventory summary for the application.

**Mark For Comparison**
Adds the selected component to the list of components to be compared.

## Displaying application summary information

You can display your application summary details information and history in the Domain Manager using the **Analytics** function.

## Procedure

To display a summary of your business applications, complete the following steps:

1. In the Functions pane, click **Analytics**.
2. Click the **Application Summary** item. The Application Summary pane opens.
3. In the Application Summary pane, select an entity in the **Application Name** column and do one of the following steps:
   - Click **Details** to display the Application Summary Details pane.
   - Click **Change** to display the Change History pane.
   - Click **Software Topology** to display the topology of software for the business application.
   - Click **Physical Topology** to display the topology of hardware used by the business application.
   - Click **Inventory** to display the Inventory Summary pane.
   - Click **Mark For Comparison** to add this component to the list of components to be compared.

     The Component Comparison window is displayed containing the selected component. To continue with the component comparison, complete the following steps:

     a. In the Application Summary pane, click the second component to be compared.

        **Important:** You do not need to close the Component Comparison window prior to clicking on another component in the Application Summary pane.
        The Component Comparison window is displayed containing the second selected component.

     b. Repeat this step for as many additional components that you want to compare.

     c. In the Component Comparison window, select the components that you want to compare and click **Compare**. The Component Comparison pane is displayed.

     For more information about component comparison, see the *Displaying component comparison information* and *Component Comparison pane* topics.

# Running a custom query

You can run an existing custom query, view the results in a table, and view additional information about each of the objects listed in the query results.

## Procedure

To run a custom query and view the results, complete the following steps in the Domain Manager:

1. In the Functions pane, click **Analytics** → **Custom Query**. The Custom Query pane is displayed in the workspace.
2. Do one of the following:
   - From the **Saved Queries** tab, select a query and click **Run Query**.
   - From the **Edit Query** tab, click **Run Query**.
   - In the **Edit Query** tab, ensure that from the **Component Type** list, a component has been selected. Click **Run Query**.

**Note:** You do not need to specify a name to run a custom query, but you must specify a name to save a custom query.
The **Results** tab is displayed. The columns displayed in the table correspond to the attributes selected in the query.

3. To view detailed information about an object listed in the query results, in the Results pane, select a query result and click **Details**. The Details notebook is displayed.
4. To view the relationships of an object listed in the query results, in the Results pane, select a query result and click **Explore**. A node centered topology graph of the object is displayed.
5. To add an object listed in the query results to the list of components to be compared, select a query result and click **Mark For Comparison**. The Component Comparison window is displayed containing the selected component. To continue with the component comparison, complete the following steps:
   a. In the Custom Query pane, click the second component to be compared.

      **Important:** You do not need to close the Component Comparison window prior to clicking on another component in the Custom Query pane.
      The Component Comparison window is displayed containing the second selected component.
   b. Repeat this step for as many additional components that you want to compare.
   c. In the Component Comparison window, select the components that you want to compare and click **Compare**. The Component Comparison pane is displayed.

   For more information about component comparison, see the *Displaying component comparison information* and *Component Comparison pane* topics.
6. To view the change history of an object listed in the query results, in the Results pane, select a query result and click **Changes**. The change history of the object is displayed.
7. To export the query results to a file (for example, Adobe® Portable Document Format, Comma Separated Values, or XML), complete the following steps:
   a. In the Results pane, select a query result and click **Save**. The Export pane, is displayed.
   b. From the **Format** list, select the format in which you want to save the query results.
   c. Click **Save**.

## Displaying services summary information

You can look at a summary of your business services in the TADDM Domain Manager using the Analytics function.

### Procedure

To display a summary of business services, complete the following steps:

1. In the Functions pane, click **Analytics**.
2. Click **Service Summary**. The Services Summary pane is displayed.
3. In the Services Summary pane, select an entity in the **Service Name** column and do one of the following steps:
   - Click **Details** to display the Service Summary Details pane.

- Click **Change** to display the Change History pane.
- Click **Software Topology** to display the topology of software for the business application.
- Click **Physical Topology** to display the topology of hardware used by the business application.
- Click **Inventory** to display the Inventory Summary pane.
- Click **Mark For Comparison** to add this component to the list of components to be compared.

  The Component Comparison window is displayed containing the selected component. To continue with the component comparison, complete the following steps:

  a. In the Services Summary pane, click the second component to be compared.

     **Important:** You do not need to close the Component Comparison window prior to clicking on another component in the Services Summary pane.
     The Component Comparison window is displayed containing the second selected component.

  b. Repeat this step for as many additional components that you want to compare.

  c. In the Component Comparison window, select the components that you want to compare and click **Compare**. The Component Comparison pane is displayed.

For more information about component comparison, see the *Displaying component comparison information* and *Component Comparison pane* topics.

# Storage array information in Domain Manager topology graphs

The following information has been added or updated for this feature.

## Topology graphs overview

The information shown in the topology graph depends on the type of component that is displayed.

The following describes the available topology graphs and the corresponding information displayed.

**Business Applications**

A business application is a collection of components that is typically deployed and assigned a version number as a unit. The topology graph displays the business applications in the environment. Business applications generally represent business functions supported by the organization, such as Order Entry or Billing.

The topology graph also displays dependencies between business applications. These dependencies represent a single derived relationship based on a grouping of the transactional dependencies that exist between the components of the respective business applications.

For example, if Business Application A has an ApplicationToApplication relationship to Business Application B, the following conditions exist:

- At least one transactional dependency exists from at least one component of Business Application A to a component of Business Application B.
- A Business Application component is one of the following types: appserver, compsys, cluster, or module.

Note however that TADDM does not discover module-to-module transactional dependencies, so these dependencies are not included in determining ApplicationToApplication relationships.

**Physical Infrastructure**

The physical infrastructure is the collection of network and hardware components in the IT environment that provides the platform on which the applications run. For example, IP subnets, routers, firewalls, load balancers, and any other IP device with routing functions are components in the Physical Infrastructure. The hardware in the IT environment must fit the criteria previously described and share subnet dependencies to be displayed in this topology graph.

The links in the topology graph represent connectivity between subnets and IP devices that route between them. These IP devices can include routers, firewalls, load balancers, and other IP devices with routing functions. For example, a router that routes between two subnets is displayed with a link to each of the subnets.

**Application Infrastructure**

The application infrastructure is the collection of server software and service components that support the business applications. For example, Web servers, J2EE application servers, Oracle databases, DNS services, and LDAP services are components in the Application Infrastructure. This graph represents the entire data center, which includes all the software servers in the data center and their relationships with each other.

This topology graph also displays the transactional and service dependencies between the software components. Components without dependencies are shown grouped by type with the label "No Dependency" displayed for the grouping.

**Software Topology for a Business Application**

The software topology of a business application displays the software relationships used by the specified or selected business application. A typical distributed application includes multiple functional groups, with each group combining components of similar type such as Apache Web servers, WebSphere application servers, DB2 databases, and so on. The software topology graph displays the components of the application.

The topology graph also displays the transactional and service dependencies between these software components. Note that if a cluster is in one of the functional groups, it is displayed along with the application servers in the cluster. If no dependencies are found, the software components are grouped by type with the label "No Dependency" displayed for the grouping.

**Physical Topology for a Business Application**

The physical topology of a business application displays the hardware used by the specified or selected business application. This graph displays the topology of the application showing how the tiers are dependent on each other. Examples of tiers include the software tier, computer tier, network (switch) tier, and storage tier.

A distributed application includes multiple functional groups, with each group combining components of similar type such as Apache Web servers, WebSphere application servers, DB2 databases, and so on. The physical topology graph displays the components of the business application.

The topology graph displays the host computer systems that are connected to the corresponding software module components. Similarly, if a computer system has, for example, a SwitchToDevice relationship with a switch, that relationship is also depicted in the graph. This is a derived relationship that traces Layer 2 connectivity from a switch to a device, host, or computer system. If a computer system uses a storage array, this derived relationship is also depicted in the graph.

Note that if a cluster is in one of the functional groups, it is displayed along with the contained application servers. When you view the physical topology of a business application, and the business application contains an application cluster that was created because of a discovered load balancer, the physical topology displays the load balancer connected to the application cluster. If the load balancer is connected to a discovered switch, that link and switch are also shown.

**Software Topology for a Business Service**

The software topology of a business service displays the components of the specified or selected business service. A business service can contain business applications, clusters, application servers, modules (such as Enterprise Archives, or EAR files), computer systems, and other business services.

A business service is typically a collection of components integrated across multiple business applications that deliver functionality for a specific task. The topology displays the transactional dependencies between the components along with the cluster relationships from clusters to their members.

**Physical Topology for a Business Service**

The physical topology of a business service displays the software components of the service and the physical resources, including the computer and network/switch tier, on which the components depend.

The topology graph also displays the constituent components linked to the specified business service. In addition, the topology graph displays the host computer systems connected to the corresponding software module components.

As with the physical topology for a business application, the physical topology for a business service shows the host computer systems connected to the corresponding software module components. If a computer system, for example, has a SwitchToDevice relationship with a switch, that relationship is also depicted in the graph. This is a derived relationship that traces Layer 2 connectivity from a switch to a device, host, or computer system.

Similarly, if a cluster is in one of the functional groups, it is displayed along with the contained application servers. When you display the physical topology of a business service, components within that business service that have their own physical topology, such as business applications or business services, are not expanded in the topology graph.

For example, if a business service contains a business application, the business application is displayed as a single component. From the topology graph, you can, however, expand the business application individually.

**Relationship Topology for a Collection**

The relationship topology of a collection displays both the explicit and logical relationships of the members of the specified or selected collection. The topology graph uses links to represent these relationships.

Note that a collection might contain a child collection as one of its components. You can manually expand this child collection within the same graph. In this case, relationships are displayed between members of the expanded collection and other objects within the topology.

In addition, logical relationships are displayed for members of a collection. Logical relationships are usually dependencies found during a discovery. There are two kinds of logical relationships: transactional and service, which you can display in the topology graph.

**Physical Topology for a Collection**

The physical topology of a collection displays all members of the collection along with the physical computer and network tiers on which members of the collection depend. The graph also displays how the tiers are dependent on each other.

The contained components of these collections are displayed in the graph linked to the specified collection. In addition, the topology graph displays the host computer systems that are connected to the corresponding software module components.

If a computer system has a SwitchToDevice relationship with a switch, that relationship is also depicted in this graph. This is a derived relationship that traces Layer 2 connectivity from a switch to a device, host, or computer system.

Note that if a cluster is in one of the functional groups, it is displayed along with the contained application servers. Also, when you display the physical topology of a business service, components within the service that have their own physical topology, such as business applications or business services, are not expanded in the topology graph.

For example, if a business service contains a business application, the business application is shown as a single component. You can expand the business application individually, if required. You cannot, however, expand child collection objects in the topology graph.

Similarly, when displaying the physical topology of a collection, only components that do not have a physical topology are expanded automatically. Components such as business services, business applications, and collections that can have a physical topology are not expanded automatically. You can, however, display the physical topology for these types of components.

**Ip Subnet Topology**

The topology of computer systems belonging to the specified Ip Subnet.

**Switch L2 IpDevice Topology**

The switch L2 IpDevice topology represents L2Devices connected to the specified switch. Specifically, the connections in the topology graph represent SwitchToDevice relationships. This is a derived relationship that traces Layer 2 connectivity from a switch to a device, host, or computer system.

A SwitchToDevice relationship implies that there is one or more L2 interfaces connected to one or more switch ports that have a segment in common with one or more L2 interfaces connected to the device. A segment is a connector between L2Interface model objects.

A SwitchToDevice relationship has the following constraints:

- A single switch port can be connected to a single or multiple ports on a host
- Multiple switch ports can be connected to a single or multiple ports on a host

For all multi-cardinality connections, the topology graph includes a link from the switch to the segment. In addition, a link is displayed from the segment to the device or devices. When there is a one-to-one connection between the switch and the device, there is a link between them. The connecting segment, however, is not displayed.

Note that only devices that are a target in the SwitchToDevice relationship, where the source is the switch for which the graph is being displayed are in the topology graph. All devices that are either a MacDevice or IpDevice are not displayed.

**Switch L2 Application Topology**

The switch L2 application topology represents business applications connected to the specified switch. Specifically, the connections in the topology graph represent the Layer 2 connectivity from a switch to a device, host, or computer system that is either a component of, or used by, a component of the business application.

**Router L2 IpDevice Topology**
The topology of L2Devices connected to the specified router.

**Router L2 Application Topology**
The topology of business applications connected to the specified router.

# Predefined BIRT reports

Because the System Utilization Hourly Peak Report is not a valid report in TADDM 7.2, the information about this report has been removed.

The following information has been added or updated for this feature.

## Fibre Channel Network Report

The Fibre Channel Network Report displays fibre channel connections between a selected Fibre Channel switch and other computer systems.

To run the report, you specify the fully qualified domain name of the Fibre Channel switch to view the fibre channel connections between this switch and other computer systems. In the Parameter window, type the fully qualified domain name or select it from the drop-down list of discovered Fibre Channel switches.

The following information is displayed in the report for each connected computer system:

- Computer System (Fully qualified domain name)
- Manufacturer
- Model
- Serial Number

You can click a fully qualified domain name in the report to open another Fibre Channel Network Report. This report shows the fibre channel connections between the selected computer system and other computer systems.

## Host Bus Adaptor Inventory Report

The Host Bus Adaptor Inventory Report displays a list of all discovered host bus adaptors and the computer systems on which they are installed.

For each host bus adaptor discovered, the following information is displayed in the report:

**Host Bus Adaptor Name**
The name of the host bus adaptor.

**Fully Qualified Domain Name**
The fully-qualified domain name of the computer systems on which the host bus adaptor is installed.

**Host uses storage arrays**
A Boolean value indicating whether the hosting computer system uses any storage volumes located in a storage array.

## Storage Array Consumers Report

The Storage Array Consumers Report displays a list of the computer systems and application servers that use a specified storage array.

When running the report, you are prompted to enter the name of a storage array. In the Parameter window, type the name of the storage array or select it from the drop-down list.

The report is displayed in the form of the following three tables:

**Computer systems using the storage array** *storage_array_name*
This table lists all discovered computer systems that use the specified storage array.

**Application servers using the storage array** *storage_array_name*
This table lists all discovered application servers that use the specified storage array.

**Business applications using the storage array** *storage_array_name*
This table lists all discovered business applications that use the specified storage array.

## Storage Arrays by Host Report

The Storage Arrays by Host Report displays a list of the storage volumes and storage arrays that are used by a specified computer system.

When running the report, you are prompted to enter the host name of the computer system for which you want to view storage information. In the Parameter window, type the host name or select it from the drop-down list.

The following information is displayed in the report:

- Storage Volume
- Storage Array
- Manufacturer
- Model
- Serial Number
- Available Capacity
- Allocated Capacity

# IBM Informix sensor

The following information has been added or updated for this feature.

## Sensor overview

This topic provides general information about the IBM Informix sensor. It includes information about supported versions and operating systems and if applicable, information about any prerequisites, security issues, and limitations.

### Sensor name that is used in the GUI and logs

Informix

### Supported versions

The sensor supports IBM Informix Dynamic Server Version 11.x and IBM Informix Java Database Connectivity (JDBC) Driver Version 3.50.

### Supported operating systems

The sensor supports the Linux operating system.

### Prerequisites

IBM Informix JDBC driver must be installed on the IBM Informix Dynamic Server.

### Limitations

To discover an Informix Dynamic Server it must be configured with the minimum requirement for discovery. Add the discovery service account to the group Informix on the Informix Dynamic Server.

## Model objects with associated attributes

The IBM Informix sensor creates model objects with associated attributes. The attributes indicate the type of information that the sensor collects about IBM Informix Dynamic Server resources in your IT environment.

The sensor creates the following model objects. The attributes that are associated with each model object are shown below the model object name.

**app.db.ids.IDSAlias**

- AliasName
- Parent
- Protocol
- ServiceName

**app.db.ids.IDSBufferPool**

- BufferPoolID
- NumBuffers
- Size

**app.db.ids.IDSChunk**

- ChunkNumber
- FreeSpace
- Offset
- Size
- MirrorOffset
- Parent

**app.db.ids.IDSConfigValue**

- ConfigID
- ConfigName
- DefaultValue
- EffectiveValue
- OriginalValue

**app.db.ids.IDSDatabase**

- DatabaseLocale
- LoggingType
- Name

**app.db.ids.IDSInstance**

- BitSize
- ConnectOption
- Home
- Host
- Name
- ProductName
- ProductVersion
- OnConfig
- Protocol
- SQLHostFile
- Status
- VersionString

**app.db.ids.IDSSegment**

- OS_SHM_ADDR
- OS_SHM_ID
- OS_SHM_KEY
- SegmentClass
- Size

**app.db.ids.IDSServerProcess**

- OSProcessName
- PID
- VpClass
- VpID

**app.db.ids.IDSSpace**

- Chunks
- ObjectType

- PageSize
- SpaceName
- SpaceNumber

**app.db.ids.IDSStartupEnvironmentVar**

- StartupEnvVarName
- StartupEnvVarValue

## Configuring the access list

To give the IBM Informix sensor access to the Informix Dynamic Server, you must configure the access list.

To configure the access list, complete the following steps:

1. From the Product Console, create a discovery scope set that contains the IP address of the Informix Dynamic Server.
2. To create an access list, click the **Access List** icon.
3. In the Access List window, click **Add**.
4. In the **Component Type** field of the Access Details window, click **ComputerSystem**.
5. Type the credentials to access the target Informix Dynamic Server. TADDM uses JDBC to connect to the Dynamic Server.

## Troubleshooting the sensor

This topic describes common problems that occur with the IBM Informix sensor and presents solutions for those problems.

### Sensor cannot retrieve server information

**Problem**

The sensor cannot retrieve information as the Informix Dynamic Server is not started.

**Solution**

Enter the oninit command to start the database server.

### Message states that nothing exists to be discovered

**Problem**

The sensor runs and completes successfully with the following message:

```
There was nothing to be discovered.
```

**Solution**

No active Informix instance is running on the target computer system.

# IBM Lotus Domino server sensor

The following information has been added or updated for this feature.

## Sensor overview

This topic provides general information about the IBM Lotus Domino server sensor. It includes information about supported versions and operating systems and if applicable, information about any prerequisites, security issues, and limitations.

### Sensor name that is used in the GUI and logs

DominoDomainSensor, DominoServerDetailSensor, and DominoSeverInitialSensor

### Supported versions

The sensor discovers the following versions of Lotus Domino:

- 6.0x
- 6.6
- 7.0.2
- 8.0
- 8.5

### Supported operating systems

The sensor supports the following operating systems:

- AIX 5.x
- Microsoft Windows 2000/2003 (32-bit and 64-bit systems)

### Prerequisites

Lotus Domino offers a user account configured with proper access to resources being discovered, such as files and databases. Note the following:

- The IIOP server must be running on at least one Domino server for each Domino domain.
- For TADDM to connect to a Domino IIOP server with SSL, you must set the `LotusDominoInitialServer.xml` file to true. Then you must copy the `TrustedCerts.class` file to `$COLLATION_HOME/etc/domino_trusted` on the TADDM directory. The `TrustedCerts.class` file is located in the `domino data folder/domino/java` folder.
- Issue the **show task** command in the Domino console to determine whether DIIOP is running.
- If DIIOP is not running, issue the **load diiop** command using the Domino console to load DIIOP.
- Issue the **tell diiop show config** command to check the configuration.
- In the `$COLLATION_HOME/etc/discover-sensors/LotusDominoInitialSensor.xml` configuration file, configure the XML section with the proper server name. The name contains the IP address or FQDN (Fully Qualified Domain Name) of the

server. Optionally it contains the port number of the Domino IIOP server. The port can be used in the case of an HTTP server where the default is 80.

For example:

```
<IIOPServers>
    <item>
        <name>example1-server.ibm.com[:Port_number]</name>
        <SSL>false</SSL>
    </item>
    <item>
        <name>example2-server.ibm.com[:Port_number]</name>
        <SSL>false</SSL>
    </item>
</IIOPServers>
```

where:

<*name*>

Contains the IP address or FQDN (Fully Qualified Domain Name) of the server.

Contains the Port number of the Domino IIOP server and is optional. The default port number for DIIOP task in the Lotus Domino configuration is 63148.

## Model objects created

The sensor creates the following model objects:

- app.lotus.AgentManager
- app.lotus.AdminProcess
- app.lotus.DirectoryAssistance
- app.lotus.DirectoryCataloger
- app.lotus.DomainCatalog
- app.lotus.DominoCluster
- app.lotus.DominoConnection
- app.lotus.DominoDatabase
- app.lotus.DominoDomain
- app.lotus.DominoNamingContext
- app.lotus.DominoReplicas
- app.lotus.DominoSecurity
- app.lotus.DominoServer
- app.lotus.DominoTransactionLogging
- app.lotus.HTTPFilterSpecialtyServer
- app.lotus.IIOPConfig
- app.lotus.IMAPConfig
- app.lotus.InternetClusterManager
- app.lotus.LDAPConfig
- app.lotus.OtherDatabase
- app.lotus.POPConfig
- app.lotus.RemoteDebugManager
- app.lotus.SMTPConfig
- app.lotus.SpecialityServer

- app.lotus.WebConfig
- app.lotus.WebRetriever

# Oracle sensor

The following information has been added or updated for this feature.

## Sensor overview

This topic provides general information about the Oracle sensor. It includes information about supported versions and operating systems and if applicable, information about any prerequisites, security issues, and limitations.

### Sensor name that is used in the GUI and logs

OracleSensor

### Supported versions

The sensor discovers the following versions of Oracle database servers:

- 8i
- 9i
- 10g
- 11g

The OracleSensor discovers clustered Oracle databases as separate instances. If the information is available, the sensor also discovers connections between the clustered Oracle databases.

### Supported operating systems

The sensor supports the following operating systems:

- AIX 5.x
- HP-UX 11.0, 11i
- Microsoft Windows 2000/2003
- Open VMS for FDA
- Red Hat AS/ES 3.0, 4.0
- Solaris 2.8-2.10
- SuSE 9,10

### Prerequisites

Note the following prerequisites:

- Discovery of the computer system must succeed.
- Network connectivity between the TADDM server and the Oracle Listener must be functioning.

### Security issues

The Oracle user credentials used to discover an Oracle database from TADDM, must have execute privileges. To ensure that the correct privileges are granted to the Oracle user, run the following command: `grant execute on` *`dbms_system`* `to` *`oracle_user`*`;`

The Oracle database account requires CONNECT privileges.

The Oracle access list user must have the following role: SELECT_CATALOG_ROLE.

The role is required to read/query the following tables: v$version, global_name, v$parameter, dba_data_files, v$log, v$logfile, sys.dba_tables, v$bgprocess, v$process, v$controlfile, v$sga, v$sys_optimizer_env, dba_db_links, dba_tables, dba_views, dba_indexes, dba_sequences, dba_constraints, dba_source, dba_clusters, dba_db_links, dba_tablespaces, dba_synonyms, dba_mviews, dba_rollback_segs, dba_profiles, dba_roles, dba_users, dba_dimensions, dba_sys_privs, dba_role_privs, dba_tab_privs, and dba_ts_quotas.

### Model objects created

The sensor creates the following model objects:

- app.db.oracle.OracleBackgroundProcess
- app.db.oracle.OracleControlFile
- app.db.oracle.OracleDatabase
- app.db.oracle.OracleDataFile
- app.db.oracle.OracleDBLink
- app.db.oracle.OracleDBLinkdat
- app.db.oracle.OracleInitValue
- app.db.oracle.OracleInstance
- app.db.oracle.OracleListener
- app.db.oracle.OracleModule
- app.db.oracle.OracleRedoLogFile
- app.db.oracle.OracleSchema
- app.db.oracle.OracleSchemaDetail
- app.db.oracle.OracleSegment
- app.db.oracle.OracleServer
- app.db.oracle.OracleServerProcess
- app.db.oracle.OracleSGAValue
- app.db.oracle.OracleTableSpace

## Configuring the collation.properties file entries

This topic lists the `collation.properties` file entries that the sensor uses.

The sensor uses the following entries in the `collation.properties` file:

**com.collation.discover.agent.OracleAgent.searchWindowsRegistry=true**
Searches the Windows Registry keys for Oracle SIDs. This property is scoped, which means that it can have an IP address appended to it to make the behavior dependent on the host being discovered.

**com.collation.discover.agent.OracleAgent.registrySearchRegexes=ORA_([^_]+)_AUTOSTART**

This regular expression property is a semicolon-separated list of regular expressions applied to Windows registry key names to extract SID candidates. Extracting the same SID multiple times is not an issue; it is only discovered once.

**com.collation.discover.agent.OracleAgent.useConnectorThreads=true**
**com.collation.discover.agent.OracleAgent.connectionThread Timeout=10000**

These properties control the use of Connector threads that prevent hung JDBC connections from hanging the sensor.

**com.collation.discover.agent.OracleAgent.suppressPorts=**

Stops the connection to ports on the listener (the ports are specified as a comma-separated list).

**com.collation.discover.agent.OracleAgent.suppressSIDs=**

Stops the discovery of SIDs (the SIDs are specified as a comma-separated list).

**com.collation.platform.os.ignoreLoopbackProcesses=false**

The default value of this property is false. This means that all processes with listening ports are considered for discovery (matched against internal and custom server templates).

In most cases, you can leave the property set to false. If you set the property to true, processes bound to 127.0.0.1 are excluded from the discovery and are not matched against internal or custom server templates.

**com.collation.discovery.oracle.extended=N**

This value specifies if additional configuration values about Oracle database links are stored. The default value of this property is N. If you set the property to Y then the sensor stores additional configuration values about Oracle database links

# WebLogic sensor

The following information has been added or updated for this feature.

## Editing the WeblogicVersionSensor.xml file

You must edit the `WeblogicVersionSensor.xml` file.

The configuration file is located in the following directories:

- On Linux, Solaris, AIX, and Linux on System z operating systems, the file is in the `$COLLATION_HOME/etc/discover-sensors/` directory.
- On Windows operating systems, the file is in the `%COLLATION_HOME%\etc\discover-sensors\` directory.

The code sample in this section shows how to configure the directories and JDK using XML tags. In this example, the following directories and JDK pairs are configured:

- The JAR files from the `lib/weblogic/10.0` directory are paired with JDK 1.5.0.
- The JAR files from the `lib/weblogic/9.0` directory are paired with JDK 1.5.0.
- The JAR files from the `lib/weblogic/8.0` directory are paired with JDK 1.4.2.

The `<entry>` tag configures the directory name used to store the WebLogic JAR files. The WebLogic JAR files must be located in the `lib/weblogic` directory.

Similarly, the `<jdk>` tag configures the version of the Java JDK in use. Valid versions are 1.5.0 and 1.4.2. If the WeblogicServerVersionSensor sensor does not recognize the BEA WebLogic server that is running, you can use the `<WeblogicClassPathDefault>` tag to force a configuration.

```
<SensorPlugin xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="http://www.ibm.com/xml/schemas/taddm/FixedSensorSchema.xsd">
    <name>WeblogicServerVersionSensor</name>
    <osgiId>com.ibm.cdb.discover.sensor.app.j2ee.weblogicserverversion_7.1.0</osgiId>

    <sensorClassName>com.collation.discover.agent.app.j2ee.WeblogicServerVersionAgent</sensorClassName>
    <seedClassName>com.collation.discover.seed.app.j2ee.WeblogicVersionSeed</seedClassName>
    <resultClassName>com.collation.discover.result.app.j2ee.WeblogicVersionResult</resultClassName>
    <convertorClassName>com.collation.discover.engine.seedfactory.WeblogicVersionConvertor</convertorClassName>

    <defaultProfiles>
        <profile>Level 3 Discovery</profile>
    </defaultProfiles>

    <configuration className="com.ibm.cdb.discover.sensor.configuration.WeblogicServerVersionAgentConfiguration">
        <weblogicClassPath>
            <item>
                <entry>10.0</entry>
                <jdk>1.5.0</jdk>
            </item>
            <item>
                <entry>9.0</entry>
                <jdk>1.5.0</jdk>
            </item>
            <item>
                <entry>8.0</entry>
                <jdk>1.4.2</jdk>
            </item>
        </weblogicClassPath>
        <!--<weblogicClassPathDefault>
            <entry>10.0</entry>
             <weblogicVersion>10</weblogicVersion>
            <jdk>1.5.0</jdk>
        </weblogicClassPathDefault>-->
    </configuration>
</SensorPlugin>
```

In the sample, the WeblogicServerVersionSensor sensor uses JAR files from the `lib/weblogic/10.0` directory with JDK 1.5.0 and assumes that WebLogic Server 10.x is running.

## Editing the WeblogicSensor2.xml file

You must edit the `WeblogicSensor2.xml` file.

The configuration file is located in the following directories:

- On Linux, Solaris, AIX, and Linux on System z operating systems, the file is in the `$`*`COLLATION_HOME`*`/etc/discover-sensors/` directory.
- On Windows operating systems, the file is in the *`%COLLATION_HOME%`*`\etc\discover-sensors\` directory.

Use the following tags to edit the `WeblogicSensor2.xml` file:

```
<SensorPlugin xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.ibm.com/xml/schemas/taddm/FixedSensorSchema.xsd">
    <name>WeblogicSensor2</name>
    <osgiId>com.ibm.cdb.discover.sensor.app.j2ee.weblogic2_7.1.0</osgiId>

    <sensorClassName>com.collation.discover.agent.app.j2ee.WeblogicAgent2</sensorClassName>
    <seedClassName>com.collation.discover.seed.app.j2ee.WeblogicSeed2</seedClassName>
    <resultClassName>com.collation.discover.result.app.j2ee.WeblogicServerResult2</resultClassName>
    <convertorClassName>com.collation.discover.engine.seedfactory.SoftwareConvertor</convertorClassName>

    <defaultProfiles>
        <profile>Level 3 Discovery</profile>
    </defaultProfiles>

    <configuration className="com.ibm.cdb.discover.sensor.configuration.WeblogicServerAgent2Configuration">
        <allowSensorToBePooledInJVM>true</allowSensorToBePooledInJVM>
        <domains>
            <item>
                <domainAddress>
                    <address>DOMAIN_IP</address>
                    <port>DOMAIN_PORT</port>
                </domainAddress>
                <addresses>
                    <item>
                        <address>IP_OF_FIRST_INTERFACE_ADMIN_SERVER_IS_USING</address>
                        <port>PORT_ ADMIN_SERVER_IS_USING </port>
                    </item>
                    <item>
                        <address>IP_OF_SECOND_INTERFACE_ADMIN_SERVER_IS_USING</address>
                        <port>PORT_ ADMIN_SERVER_IS_USING </port>
                    </item>
                </addresses>
            </item>
        </domains>
    </configuration>
</SensorPlugin>
```

You can use this configuration when the WebLogic server is using multiple interfaces on the Domain Admin Server.

In this case, the value of `DOMAIN_IP` and `DOMAIN_PORT` is used instead of `IP_OF_FIRST_INTERFACE_ADMIN_SERVER_IS_USING:PORT_ ADMIN_SERVER_IS_USING` and `IP_OF_SECOND_INTERFACE_ADMIN_SERVER_IS_USING:PORT_ ADMIN_SERVER_IS_USING`.

# Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

## Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Printed in USA